

MACHINE LEARNING, ENERGY AND INDUSTRIAL APPLICATIONS IN TECHNOLOGY AND ENGINEERING SCIENCES

EDITOR

Assist. Prof. Dr. Şule YÜCELBAŞ

AUTHORS

Prof. Dr. Hüseyin PEKER

Assoc. Prof. Dr. Serdar ÖZYÖN

Assist. Prof. Dr. Şule YÜCELBAŞ

Dr. Lec. Hatice ULUSOY

Dr. Duygu BAYRAM KARA

Abdallah ELCHAKIK

Ahmad SHAHIN

Adnan YASSINE

Hassan ALABBOUD

Mina Ghorban Zadeh Badeli

Roaa SOLOH



MACHINE LEARNING, ENERGY AND INDUSTRIAL APPLICATIONS IN TECHNOLOGY AND ENGINEERING SCIENCES

EDITOR

Assist. Prof. Dr. Şule YÜCELBAŞ

AUTHORS

Prof. Dr. Hüseyin PEKER

Assoc. Prof. Dr. Serdar ÖZYÖN

Assist. Prof. Dr. Şule YÜCELBAŞ

Dr. Lec. Hatice ULUSOY

Dr. Duygu BAYRAM KARA

Abdallah ELCHAKIK

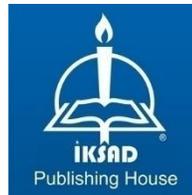
Ahmad SHAHIN

Adnan YASSINE

Hassan ALABBOUD

Mina Ghorban Zadeh Badeli

Roaa SOLOH



Copyright © 2021 by iksad publishing house
All rights reserved. No part of this publication may be reproduced,
distributed or transmitted in any form or by
any means, including photocopying, recording or other electronic or
mechanical methods, without the prior written permission of the publisher,
except in the case of
brief quotations embodied in critical reviews and certain other
noncommercial uses permitted by copyright law. Institution of Economic
Development and Social
Researches Publications®
(The Licence Number of Publicator: 2014/31220)
TURKEY TR: +90 342 606 06 75
USA: +1 631 685 0 853
E mail: iksadyayinevi@gmail.com
www.iksadyayinevi.com

It is responsibility of the author to abide by the publishing ethics rules.
Iksad Publications – 2021©

ISBN: 978-625-7562-47-8
Cover Design: İbrahim KAYA
August / 2021
Ankara / Turkey
Size = 16x24 cm

CONTENTS

PREFACE

Assist. Prof. Dr. Şule YÜCELBAŞ1

CHAPTER 1

ENVIRONMENTAL/ECONOMIC PUMPED- STORAGE HYDRAULIC UNIT SCHEDULING PROBLEM FOR DIFFERENT CYCLE EFFICIENCY VALUES

Assoc. Prof. Dr. Serdar ÖZYÖN3

CHAPTER 2

BACK PROPAGATION NEURAL NETWORK BASED CLASSIFICATION FOR ELECTRIC MOTORS

Mina Ghorban, Zadeh Badeli

Dr. Duygu BAYRAM KARA51

CHAPTER 3

IDENTIFICATION OF MYOCARDIAL INFARCTION DISEASE USING EFFECTIVE MACHINE LEARNING METHODS

Assist. Prof. Dr. Şule YÜCELBAŞ81

CHAPTER 4

EFFECT OF VARIOUS BINDERS ON HOLD / PHYSICAL PROPERTIES OF SPRUCE (PICEA ORIENTALIS (L.) LINK) WOOD

Dr. Lecturer Hatice ULUSOY

Prof. Dr. Hüseyin PEKER99

CHAPTER 5

**3D MESH MATCHING USING SURFACE DESCRIPTOR AND
INTEGER LINEAR PROGRAMMING**

Roa SOLOH, Abdallah ELCHAKIK

Hassan ALABBOUD, Ahmad SHAHIN

Adnan YASSINE111

CHAPTER 6

**PERFORMANCE ANALYSIS OF CHAOTIC
MAPS IN DIFFERENTIAL EVOLUTION
ALGORITHM FOR HIGH-DIMENSIONAL
MULTIMODAL BENCHMARK FUNCTIONS**

Assoc. Prof. Dr. Serdar ÖZYÖN129

PREFACE

This book presents the machine learning, energy and industrial applications in technology and engineering sciences. The matters discussed and presented in the chapters of this book cover a wide spectrum of topics and research methods in the field of technology and engineering sciences. The book contains six chapters: differential evolution algorithm for problem of environmental economic operation solution, classification for electric motors, effective machine learning methods for disease diagnosis, applications for wood industry, approach for 3D shape matching, and determining effects of chaotic mapping methods.

This book, which consists of researches covering the fields of machine learning, algorithms, engineering, and industrial sciences, will constitute an academic data source for academicians and researchers. I believe that this book, which consists of different scientific fields, will make significant contributions to the world of science and I would like to thank the authors who have contributed.

Sincerely Yours

Assist. Prof. Dr. Şule YÜCELBAŞ

Aug 2021

CHAPTER 1
**ENVIRONMENTAL/ECONOMIC PUMPED-STORAGE
HYDRAULIC UNIT SCHEDULING PROBLEM FOR
DIFFERENT CYCLE EFFICIENCY VALUES**

Assoc. Prof. Dr. Serdar ÖZYÖN¹

¹Department of Electrical and Electronics Engineering, Kütahya Dumlupınar University, KÜTAHYA, TURKEY. serdar.ozyon@dpu.edu.tr
ORCID ID 0000-0002-4469-3908

INTRODUCTION

One of the important problems of power system engineering is the problem of economic power dispatch. The gradual increase in the need for electrical power and the use of fossil-based fuels in the generation of most of this power requires the consideration of environmental effects in economic power dispatch. The problem which requires the decrease of the emission amount dispatched to the environment as well as the optimization of the financial cost is called environmental/economic power dispatch problem. The use of thermal and hydraulic generation units together in environmental/economic power dispatch problem requires providing much more constraints at the same time. There is neither emission nor fuel cost of the electrical power generated by the hydraulic generation units. One of the widely used hydraulic generation units is pumped- storage (*P/S*) hydraulic generation units. These units are one of the renewable power generation systems. The main purpose in *P/S* hydraulic units is to store the extra electrical power generated by the system as hydraulic potential power when its cost is low and to meet the power demand of the system (high power demand) with this stored potential power when the generation cost is high. The problem of the optimum operation of the systems including these kinds of units by considering the environmental effects as well, is a complex problem having many constraints (Jebaraj et al., 2017; Mahdi et al., 2018).

The problem of the environmental/economic operation of the generation units during a day (short term) is defined as meeting the

demanded load with the least fuel cost and emission amount by providing all the constraints of the generation units in the system. Since both the fuel cost and environmental pollution effect are evaluated together in the solution of environmental economic power dispatch problems, the problem turns into a multi-objective optimization problem. The solution of this problem determines active power generation values of the generation units in which the possible thermal and hydraulic constraints in the system are provided and which minimize the total fuel cost function (total thermal cost and total emission amount) scalarized with weighted sum method (*WSM*) during the foreseen operation time. Generally, there is not only one solution in multi-objective optimization problems, but an alternative set of solutions. When all objectives are thought for the elements of this alternative set of solutions, it cannot be said that any of the solutions is more “the best” solution than the others. These kinds of solutions are called pareto optimum solutions (Özyön et al., 2012).

Many studies have been done on *P/S* hydraulic generation systems in literature. The main purpose in these studies is to develop faster and more efficient solutions for the operation of the system. The mathematical modelling of *P/S* hydraulic generation units, the importance of their use, the benefit they provided for the system they are in and various statistical data have been given in (Deane et al., 2010; Ardizzon, Cavazzini, & Pavesi, 2014; Rehman, Al-Hadhrani, & Alam, 2015). The problem of economic operation of the systems with *P/S* hydraulic generation units has been solved with various

approaches and optimization methods under different constraints. These approaches that have been used can be listed as pseudo spot price algorithm (*PSPA*) (Fadıl & Urazel, 2010), modified subgradient algorithm based on feasible values (*F-MSG*) (Fadıl & Urazel, 2013), annealing neural network (*ANN*) (Liang, 2000), mixed-integer linear programming (*MILP*) (Cheng et al., 2018), mixed-integer non-linear programming (*MINLP*) (Simab, Javadi, & Nezhad, 2018; Nezhad, Javadi, & Rahimi, 2014), evolutionary particle swarm optimization (*EPSO*) (Chen, 2008), two stage stochastic programming (*TS-SP*) (Kocaman, 2019), and artificial sheep algorithm (*BASA*) (Wang, Li, & Liao, 2017). As for the environmental/economic operation of these systems, it has been solved with the genetic algorithm (*GA*) in (Demir, 2010; Fadıl, Demir, & Urazel, 2013) and with modified subgradient algorithm based on feasible values (*F-MSG*) in (Fadıl & Urazel, 2014).

The environmental/economic *P/S* hydraulic unit scheduling problem the solution of which has been done in this study is a complex problem in terms of its size and constraints and a difficult one to be solved with numeric methods. Many optimization algorithms have been and are still being developed for the faster and more decisive solution of these kinds of problems. In this study, the problem of environmental economic operation of a lossy power system consisting of thermal and *P/S* hydraulic generation units, has been solved with differential evolution algorithm (*DE*) for the first time in literature. Transmission line losses have been considered in the solution of the

sample system and Newton-Raphson AC load flow method has been used for the calculation of these losses. The sample system selected in the study has been solved for four cases. The economic and environmental benefit that *P/S* hydraulic generation units provided for the system they are in, has been proved with the data obtained from these solutions and the results have been discussed.

1. ENVIRONMENTAL/ECONOMIC *P/S* HYDRAULIC UNIT SCHEDULING PROBLEM

Today thermal generation units which meet a great part of electrical power generation cannot adapt to sudden load changes because of various constraints. Therefore, these kinds of generation units are generally operated at basic load. As for the hydraulic generation units, since they are able to reach full capacity power at a short time, they are used to meet the peak load. For the system to operate without problems it is important that the peak load demand is met. Another renewable power generation unit frequently used to meet the peak load demand is *P/S* hydraulic generation unit. The aim of these generation units is to store the water in the reservoir in a high place when the load is low, and to generate power with this stored water in cases of peak load demand (Fadıl & Yaşar, 2000; Wood & Wollenberg, 1996). In pumped storage hydraulic units there are two types of ways of operation as pumping and generation. The presentation belonging to the operation principle of these generation units has been given in Figure 1 (Ma et al., 2015).

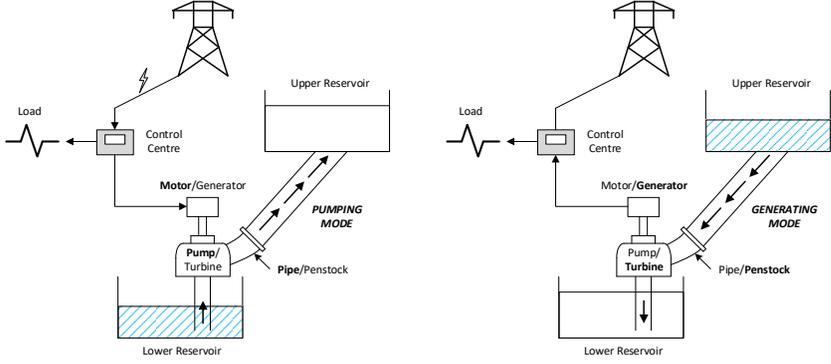


Figure 1: *P/S* Hydraulic Power Generation System

The fuel cost function for each thermal generation unit in the system is shown with a second-degree convex function. The fuel cost function of each of the thermal generation units has been taken as in equation (1) (Özyön et al., 2012 ;Wood & Wollenberg, 1996).

$$F_n(P_{GT,n}) = a_n + b_n \cdot P_{GT,n} + c_n \cdot P_{GT,n}^2 \text{ (\$/h)}, \quad n \in N_T \quad (1)$$

In the equation $F_n(P_{GT,n})$, shows the fuel cost function of n . generation unit, a_n , b_n ve c_n coefficients show the cost function coefficients of n . generation unit, as for $P_{GT,n}$, it shows the output power of n . thermal generation unit and its unit is taken as *MW*. The emission amount spread to the environment by each of the thermal units is defined in terms of the output power of the generation unit as in the following (Özyön et al., 2012 ;Wood & Wollenberg, 1996) .

$$E_n(P_{G,n}) = d_n + e_n P_{G,n} + f_n P_{G,n}^2 + g_n \exp(h_n P_{G,n}), \text{ (ton/h)} \quad (2)$$

Active power balance constraints for the pumping case in a lossy system consisting of thermal and *P/S* hydraulic generation units have

been shown in equation (3), and for the generation case in equation (4). As for the reactive power balance constraints in the system, they have been shown for the pumping case in equation (5) and for the generation case in equation (6) (Demir, 2010; Fadıl, Demir, & Urazel, 2013; Fadıl & Urazel, 2014).

$$\sum_{n \in N_T} P_{GT,nj} - P_{PPS,j} - P_{load,j} - P_{loss,j} = 0, \quad j \in J_{pump} \quad (3)$$

$$\sum_{n \in N_T} P_{GT,nj} + P_{GPS,j} - P_{load,j} - P_{loss,j} = 0, \quad j \in J_{gen} \quad (4)$$

$$\sum_{n \in N_T} Q_{GT,nj} - Q_{PPS,j} - Q_{load,j} - Q_{loss,j} = 0, \quad j \in J_{pump} \quad (5)$$

$$\sum_{n \in N_T} Q_{GT,nj} + Q_{GPS,j} - Q_{load,j} - Q_{loss,j} = 0, \quad j \in J_{gen} \quad (6)$$

Active and reactive power operation limit values of the thermal generation units in the system have been given in equation (7) and (8).

$$P_{GT,n}^{\min} \leq P_{GT,nj} \leq P_{GT,n}^{\max}, \quad n \in N_T, \quad j = 1, \dots, j_{\max} \quad (7)$$

$$Q_{GT,n}^{\min} \leq Q_{GT,nj} \leq Q_{GT,n}^{\max}, \quad n \in N_T, \quad j = 1, \dots, j_{\max} \quad (8)$$

Electrical constraints belonging to *P/S* hydraulic generation units have been given in equation (9)-(13), and the hydraulic constraints have been given in equation (14)-(22) (Demir, 2010; Fadıl, Demir, & Urazel, 2013; Fadıl & Urazel, 2014).

$$P_{GPS,m}^{\min} \leq |P_{GPS,mj}| \leq P_{GPS,m}^{\max}, \quad m \in N_{PS}, \quad j \in J_{gen} \quad (9)$$

$$Q_{GPS,m}^{\min} \leq |Q_{GPS,mj}| \leq Q_{GPS,m}^{\max}, \quad m \in N_{PS}, \quad j \in J_{gen} \quad (10)$$

$$q_m^{\min} \leq q_{mj}(|P_{GPS,mj}|) \leq q_m^{\max}, \quad m \in N_{PS}, \quad j \in J_{gen} \quad (11)$$

$$P_{PPS,m}^{\min} \leq |P_{PPS,mj}| \leq P_{PPS,m}^{\max}, \quad m \in N_{PS}, \quad j \in J_{pump} \quad (12)$$

$$Q_{PPS,m}^{\min} \leq |Q_{PPS,mj}| \leq Q_{PPS,m}^{\max}, \quad m \in N_{PS}, \quad j \in J_{pump} \quad (13)$$

$$q_m^{\min} \leq q_{mj}(|P_{PPS,mj}|) \leq q_m^{\max}, \quad m \in N_{PS}, \quad j \in J_{pump} \quad (14)$$

$$q_{GPS}(P_{GPS,mj}) = \begin{cases} d + eP_{GPS,mj} \text{ (acre-ft/h)} & \text{if } 0 < P_{GPS,mj} \leq P_{GPS}^{\max} \\ 0 & \text{if } P_{GPS,mj} = 0 \end{cases}, \quad m \in N_{PS}, \quad j \in J_{gen} \quad (15)$$

$$q_{PPS}(P_{PPS,mj}) = \begin{cases} f + g|P_{PPS,mj}| \text{ (acre-ft/h)} & \text{if } 0 < |P_{PPS,mj}| \leq P_{PPS}^{\max} \\ 0 & \text{if } |P_{PPS,mj}| = 0 \end{cases}, \quad m \in N_{PS}, \quad j \in J_{pump} \quad (16)$$

$$q_{GPS,total} = \sum_{j \in J_{gen}} q_{GPS}(P_{GPS,mj})t_j, \quad m \in N_{PS} \quad (17)$$

$$q_{PPS,total} = \sum_{j \in J_{pump}} q_{PPS}(P_{PPS,mj})t_j, \quad m \in N_{PS} \quad (18)$$

$$q_{GPS,total} - q_{PPS,total} = q_{total} = 0 \quad (19)$$

$$V_m^{\min} \leq V_{mj} \leq V_m^{\max}, \quad m \in N_{PS}, \quad j = 1, \dots, j_{\max} \quad (20)$$

$$V_{mj} = \begin{cases} V_{mj-1} - q_{GPS}(P_{GPS,mj})t_j & \text{if } j \in J_{gen} \\ V_{mj-1} + q_{PPS}(P_{PPS,mj})t_j & \text{if } j \in J_{pump} \end{cases}, \quad m \in N_{PS} \quad (21)$$

$$V_{m0} = V_{mj \max} = V_m^{start} = V_m^{end}, \quad m \in N_{PS} \quad (22)$$

Total fuel function that was scalarized with *WSM* and will be minimized by *DE* for the solution of environmental/economic *P/S* hydraulic unit scheduling problem has been given in equation (23) (Özyön et al., 2012).

$$TC = w.F_T(P_{GT,n}) + (1-w).\gamma.E_T(P_{GT,n}), \quad n \in N_T \quad (23)$$

In the equation, $F_T(P_{G,n})$ shows the total fuel cost as (\$) and $E_T(P_{GT,n})$ shows the total emission amount function as (ton). γ (\$/ton) shows the scaling factor, w shows the weight factor that changes as ($0 \leq w \leq 1$) and N_T shows the set of all thermal generation units in the system. Here $w=1.0$ value only corresponds to the minimization of total fuel cost and $w=0.0$ value only corresponds to the minimization of total emission amount. The total fuel cost taking place in the equation $F_T(P_{GT,n})$ and total emission amount $E_T(P_{GT,n})$ are calculated by using respectively equation (24) and (25) (Özyön et al., 2012 ;Wood & Wollenberg, 1996).

$$F_T(P_{GT,n}) = \sum_{j=1}^{j_{\max}} t_j \sum_{n \in N_T} F_n(P_{GT,nj}), \quad (\$) \quad (24)$$

$$E_T(P_{GT,n}) = \sum_{j=1}^{j_{\max}} t_j \sum_{n \in N_T} E_n(P_{GT,nj}), \quad (\text{ton}) \quad (25)$$

2. DIFFERENTIAL EVOLUTION ALGORITHM (DE)

DE, is a population-based algorithm consisting of mutation, crossing and selection operators. It uses the mutation operator as main search strategy and selection operator to direct the search that is done, to the potential solution area. Two sequences are formed in the algorithm as primary and secondary. Both series consist of potential solutions at NP number and each solution includes variables at D number. These solutions are vectors with real values. In short, in the beginning there is D dimensional vector at NP number. The sum of all vectors is called

population. The algorithm, as such was first suggested by Storn and Price in 1997 (Storn, 1997; Naila et al., 2018).

The basic algorithm parameters used in *DE* are the size of the population (NP_{max}), maximum iteration number (g_{max}), crossover rate (CR) and scaling factor (F). The process steps of the algorithms can be defined as variable assignment and formation of the initial population, calculation of the fitness of the individuals, selection, mutation, crossover and stopping of the algorithm. The flowchart belonging to the algorithm has been given in Figure 2 (Mandal, Chatterjee, & Bhattacharjee, 2013).

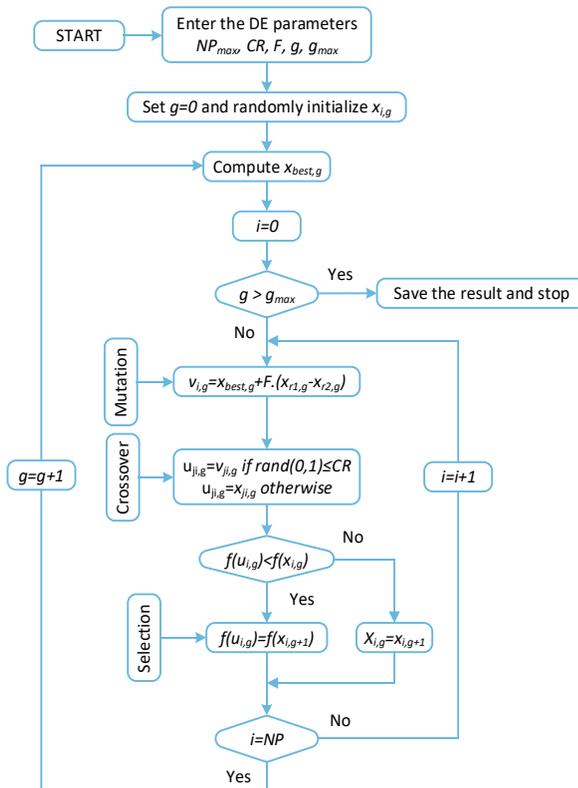


Figure 2: The flowchart of DE

3. NUMERICAL APPLICATIONS

Single line diagram selected as the example for the environmental/economic operation of power generation systems including P/S hydraulic unit has been given in Figure 3. There are 12 buses, 5 thermal generation units generating production with fossil-based fuel, one P/S hydraulic unit, 25 transmission lines and 7 load buses in this power system. In the study, all values used in the solution of the sample system and the obtained results have been taken as *pu*. As for the base values $S_{base}=100$ MVA, $V_{base}=230$ kV ve $Z_{base}=529$ Ohm. In the system G_1 generator connected to the bus number 1 has been selected as slack bus. Bus voltage value for the oscillation bus has been taken as $V_1=1,05\angle 0^\circ$ *pu*. The sample system has been solved with DE algorithm for four different cases by considering transmission line losses. In the first case, when there is no P/S hydraulic unit in the system, thermal fuel cost and the amount of emission have been minimized. As for the second, third and fourth cases, the system has been solved again by including P/S hydraulic unit respectively with 0.60, 0.67 and 0.75 cycle efficiency values in the system. The benefit that P/S hydraulic unit has for different cycle efficiency cases has been determined by calculating the thermal fuel costs and emission amounts of the system for four cases each, under the same load demand. Transmission line losses in the solution of the sample system have been found by doing AC load flow with Newton-Raphson method. In the solution of the system, one-day (short term) operation time, which consists of six equal four-hour time slices

($t_j=4h, j=1, \dots, j_{max}$), has been considered (Demir, 2010; Fadıl, Demir, & Urazel, 2013; Fadıl & Urazel, 2014; Özyön, 2020).

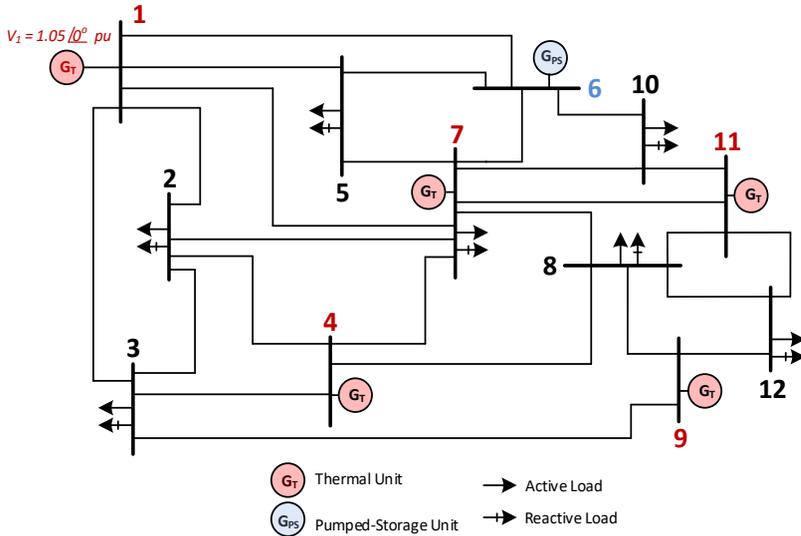


Figure 3: One-line diagram of the example power system

Line parameter values R (resistance), X (reactance) and B (susceptance) of nominal π equivalent circuits belonging to the transmission lines in the sample system, active and reactive load values in each period and initial reactive power generation values in each time slice belonging to the generation units have been taken from reference (Özyön, 2020).

Convex thermal fuel and emission cost function coefficients belonging to thermal generation units and active power generation limits of these units have been shown in Table 1 (Demir, 2010; Fadıl, Demir, & Urazel, 2013; Fadıl & Urazel, 2014; Özyön, 2020).

Table 1: Fuel and emission cost function coefficients belonging to the thermal generation units

<i>Bus no (n)</i>	<i>1</i>	<i>4</i>	<i>7</i>	<i>9</i>	<i>11</i>
a_n	10	20	20	70	70
b_n	0.5	0.85	0.85	2.15	2.05
c_n	0.0025	0.003	0.003	0.008	0.213
d_n	4.426	4.258	4.258	7.743	8.531
e_n	-5.55E-2	-6.09E-2	-6.09E-2	-2.15E-2	-2.36E-2
f_n	2.03E-4	3.89E-4	3.89E-4	8.44E-4	8.65E-4
g_n	2.00E-6	1.00E-6	1.00E-6	5.00E-5	5.00E-6
h_n	1.50E-2	2.35E-2	2.35E-2	8.83E-2	8.97E-2
$P_{GT,n}^{\min}$ (pu)	0.50	0.45	0.40	0.05	0.05
$P_{GT,n}^{\max}$ (pu)	3.50	1.80	1.70	1.00	1.00

DE parameter values used for all cases have been given in Table 2. For the solution of the problems, sub-programs developed in MATLAB R2015b have been operated in a work station with Intel Xeon E5-2637 v4 3.50 GHz processor and 128 GB RAM memory. The scaling factor taking place in equation (23) has been taken as 1000.

Table 2: DE parameters

<i>Iteration number (g_{max})</i>	<i>Individual number (NP_{max})</i>	<i>Function call (F_{Call})</i>	<i>Crossover rate (CR)</i>	<i>Scaling factor (F)</i>	<i>Run</i>
300	50	15000	0.8	0.4	30

3.1. Case-I: Non P/S Hydraulic Unit

In the first case, the sample system has been solved by assuming that the demanded load was met only by thermal generation units. The statistical values belonging to the total cost function of the problem which was solved 30 runs with the parameter values given in Table 2, have been given in Table 3. As for the power generation values

belonging to the best solution among these solutions, transmission line losses and solution times for different weight values, they have been given in Table 4-6 respectively.

Table 3: Values obtained from 30 runs (Case-I: Non P/S)

Non P/S	w		
	1.0	0.5	0.0
Worst	16414.499738 (Run: 13)	10147.829703 (Run: 10)	4084.708522 (Run: 17)
Mean	15551.990248	9832.602982	3783.839844
Best	15458.008141 (Run: 9)	9424.251183 (Run: 13)	3373.464713 (Run: 5)
Std	210.581316	190.860112	177.467403
Total time (s)	3762.168518	3595.311464	3757.381171
Mean time (s)	125.405617	119.843715	125.246039

Table 4: Values belonging to the best solution (Case-I: Non P/S) ($w=1.0$)

$w=1.0$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.162194	2.982822	3.200000	2.982809	1.547352	1.162366
	$Q_{GT,1}$	1.044028	2.897716	2.522277	2.897708	1.570472	1.044045
4	$P_{GT,4}$	0.450000	1.800000	1.800000	1.800000	0.768128	0.450000
7	$P_{GT,7}$	0.441258	1.700000	1.700000	1.700000	0.807127	0.441091
9	$P_{GT,9}$	0.000000	0.000000	0.801442	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.5458e+04					
E_T (ton)		3.3982					
$P_{total\ loss}$ (pu)		1.696606					
Time (s)		126.872289					

Table 5: Values belonging to the best solution (Case-I: Non P/S) ($w=0.5$)

$w=0.5$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.057801	2.982822	3.200000	2.982822	1.503546	1.058378
	$Q_{GT,1}$	1.035168	2.897716	2.522277	2.897716	1.565998	1.035215
4	$P_{GT,4}$	0.493967	1.800000	1.800000	1.800000	0.799463	0.493797
7	$P_{GT,7}$	0.499053	1.700000	1.700000	1.700000	0.818222	0.498660
9	$P_{GT,9}$	0.000000	0.000000	0.801442	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.5462e+04					
E_T (ton)		3.3861					
$P_{total\ loss}$ (pu)		1.689975					
Time (s)		117.772149					

Table 6: Values belonging to the best solution (Case-I: Non P/S) ($w=0.0$)

		$w=0.0$					
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	0.937014	3.062652	3.090112	3.062654	1.455649	0.937011
	$Q_{GT,1}$	1.027043	2.910232	2.399395	2.910233	1.561368	1.027043
4	$P_{GT,4}$	0.556796	1.724270	1.740207	1.724268	0.831592	0.556799
7	$P_{GT,7}$	0.554691	1.700000	1.700000	1.700000	0.832570	0.554691
9	$P_{GT,9}$	0.000000	0.000000	0.468820	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.462824	0.000000	0.000000	0.000000
F_T (\$)		1.7417e+04					
E_T (ton)		3.3735					
$P_{total\ loss}$ (pu)		1.652625					
Time (s)		135.532172					

In the first case, total fuel cost and total emission amounts belonging to the best solutions obtained from the solution of the sample system with DE algorithm and different weight values for 30 times have been given in Figure 4.

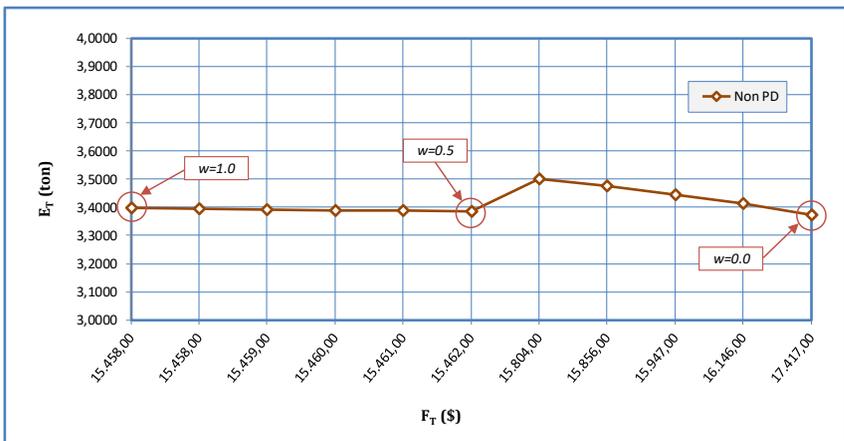


Figure 4: Total fuel cost and total emission amounts (Case-I: Non P/S)

In the first case, the change of total fuel cost and total emission amounts that belong to the best solution obtained for $w=1.0$, according

to the iterations have been given in Figure 5, the change of the transmission line losses according to the iterations and the active powers generated in each period have been given in Figure 6. Similar graphics have been obtained for all weight values but have not been added to the study due to the lack of space.

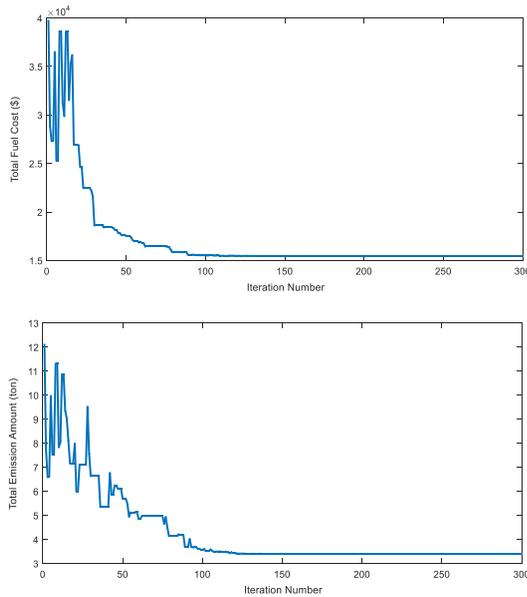


Figure 5: The change of *TFC* ve *TEA* according to the iterations (Case-I: Non *P/S*) ($w=1.0$)

As is seen in Figure 5, in the solution of the sample system with DE algorithm, total fuel cost value and total emission amount values for case-I have caught the optimum result approximately at 120th iteration.

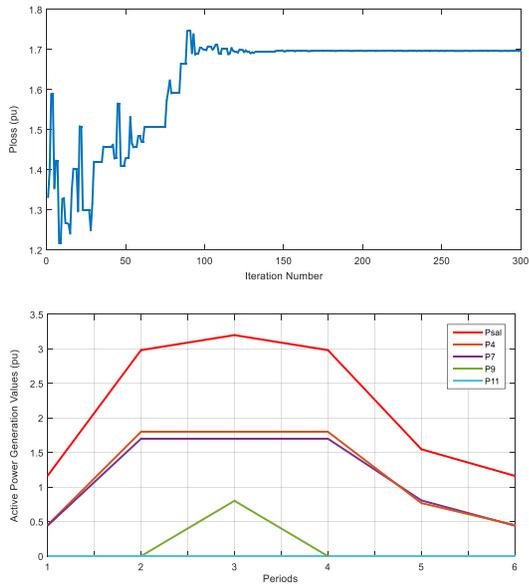


Figure 6: The change of P_{loss} according to iterations, The change of generated active powers according to the periods (Case-I: Non P/S) ($w=1.0$)

When the graphic of change of active powers according to the periods, which has been generated for this case, is looked at, it is seen that 11th thermal generation unit did not make any generation and as for the 9th thermal generation unit, it made generation only in the 3rd period. The reason is that the generation cost of these generation units is higher. In this case, the graphic where the best TFC 's obtained from 30 solutions done for $w=1.0$ take place, has been given in Figure 7 and the boxplot belonging to these solutions has been given in Figure 8.

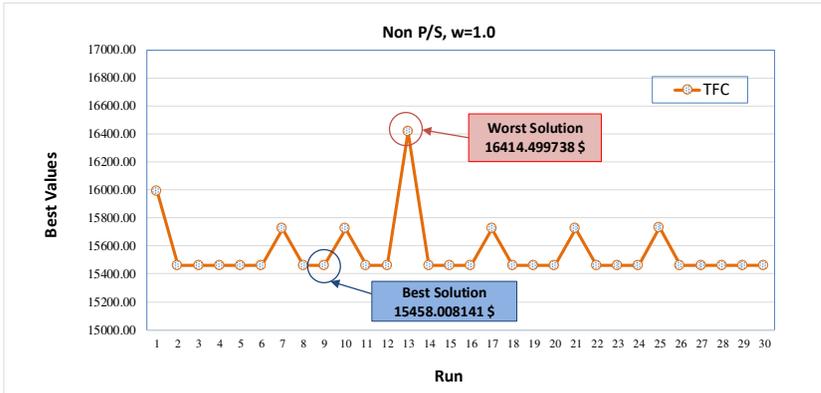


Figure 7: The best *TFC* values obtained from 30 runs (Case-I: Non *P/S*) ($w=1.0$)

When Figure 7 is examined, it is seen that in the solution of the sample system with *DE* for 30 times for case-I, the worst cost value has been obtained at 13th solution as 16414.499738 \$ and the best cost value has been obtained at 9th solution as 15458.008141 \$. *DE* algorithm has searched at a 956.491597 \$ interval which provides all the constraints at 30 solutions.

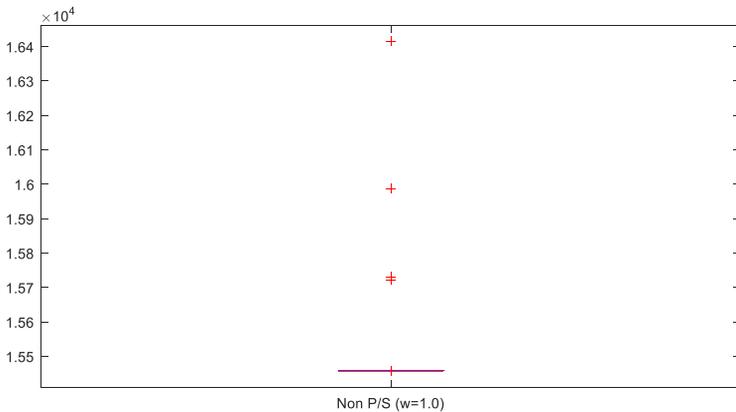


Figure 8: Boxplot (Case-I: Non *P/S*) ($w=1.0$)

When the boxplot for this case is examined, it has been seen that *DE* algorithm did a decisive search and there were only 4 divergent values in 30 runs.

In the second, third and fourth cases, solution has been done by connecting a P/S hydraulic unit, whose characteristics were given in Table 7, to the sixth bus in the sample system with different cycle efficiency values. In order to be able to make more accurate comparisons, reactive generation and reactive pumping load of P/S hydraulic unit have been taken as 0 MVAR in all cases.

Table 7: P/S hydraulic unit parameters

Bus no	P_{GPs}^{\min} (MW)	P_{GPs}^{\max} (MW)	d, f	e	g
	6	0	130	200	2.00
	V^{start} (acre-ft)	V^{end} (acre-ft)	V^{min} (acre-ft)	V^{max} (acre-ft)	Cycle efficiency ($\mu = g/e$)
	10000	10000	5000	15000	0.60, 0.67, 0.75

3.2. Case-II: With P/S Hydraulic Unit ($\mu=0.60$)

In the second case, the sample system has been solved by assuming that the demanded power was met with a P/S hydraulic unit having 0.60 cycle efficiency and together with thermal generation units. The statistical values belonging to the total cost function of the problem which was solved 30 runs, have been given in Table 8, power generation values belonging to the best solution, transmission line losses and solution times have been given in Tables 9-11.

Table 8: Values obtained from 30 solutions/runs (Case-II: With $P/S, \mu=0.60$)

With $P/S, \mu=0.60$	w		
	1.0	0.5	0.0
Worst	15290.742974 (Run: 28)	8821.079736 (Run: 6)	2400.291739 (Run: 25)
Mean	15144.006011	8793.879524	2367.612348
Best	15076.984327 (Run: 7)	8779.554683 (Run: 17)	2358.440065 (Run: 7)
Std	52.785791	12.252794	8.542862
Total time (s)	3977.279036	3038.843043	4108.762891
Mean time (s)	132.575968	101.294768	136.958763

Table 9: Values belonging to the best solution (Case-II: With $P/S, \mu=0.60$) ($w=1.0$)

$w=1.0$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.487681	2.927614	3.128951	2.867508	1.600308	1.494114
	$Q_{GT,1}$	1.114867	2.883670	3.055393	2.868640	1.578680	1.112940
4	$P_{GT,4}$	0.717516	1.800000	1.800000	1.800000	0.837511	0.717923
6	$P_{GPS,6}$	-	0.054013	1.040925	0.106934	-	-
	$P_{PPS,6}$	-0.901489	-	-	-	-0.228316	-0.873320
7	$P_{GT,7}$	0.773348	1.696507	1.700000	1.698689	0.915926	0.737665
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.5077e+04					
E_T (ton)		2.5684					
$P_{total\ loss}$ (pu)		1.900012					
V_{end} (acre-ft)		10000.001868					
Time (s)		123.716002					

Table 10: Values belonging to the best solution (Case-II: With $P/S, \mu=0.60$) ($w=0.5$)

$w=0.5$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.598970	2.730209	2.858289	2.800431	1.765388	1.629742
	$Q_{GT,1}$	1.177192	2.830528	3.007194	2.846146	1.617677	1.183844
4	$P_{GT,4}$	0.869288	1.708196	1.800000	1.722821	0.942831	0.848013
6	$P_{GPS,6}$	-	0.322400	1.295941	0.242305	-	-
	$P_{PPS,6}$	-1.251718	-	-	-	-0.560775	-1.288585
7	$P_{GT,7}$	0.881159	1.699529	1.699554	1.700000	0.990801	0.910674
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.5148e+04					
E_T (ton)		2.4109					
$P_{total\ loss}$ (pu)		1.915469					
V_{end} (acre-ft)		10000.000327					
Time (s)		95.631006					

Table 11: Values belonging to the best solution (Case-II: With P/S , $\mu=0.60$) ($w=0.0$)

Bus no (n)	Generation (pu)	$w=0.0$					
		Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.580470	2.801500	2.953031	2.774792	2.075094	1.574821
	$Q_{GT,1}$	1.184755	2.852200	3.040017	2.851180	1.744646	1.183630
4	$P_{GT,4}$	0.896926	1.606857	1.719817	1.561455	1.131711	0.905123
6	$P_{GPS,6}$	-	0.443941	1.299959	0.539730	-	-
	$P_{PPS,6}$	-1.298338	-	-	-	-1.217894	-1.289850
7	$P_{GT,7}$	0.921222	1.614916	1.691746	1.590763	1.191111	0.909824
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.5297e+04					
E_T (ton)		2.3584					
$P_{total\ loss}$ (pu)		1.978732					
V_{end} (acre-ft)		10000.015648					
Time (s)		133.106572					

Total fuel cost and total emission amounts belonging to the best solutions obtained from the solution of the sample system with different weight values for 30 runs by adding a P/S unit with 0.60 cycle efficiency to the 6th bus, have been given in Figure 9.

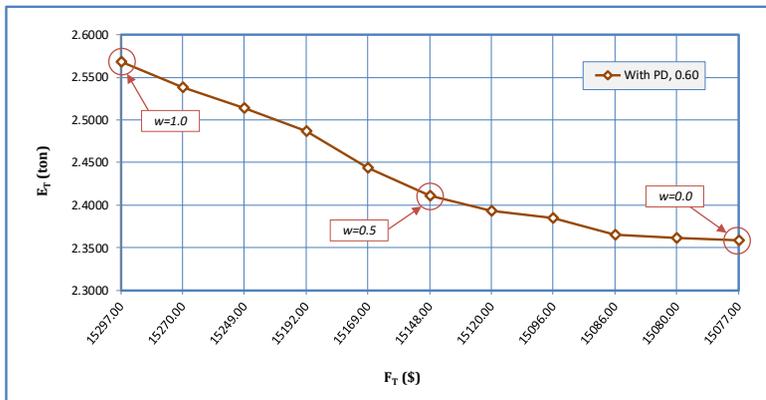


Figure 9: Total fuel cost and total emission amounts (Case-II: With P/S , $\mu=0.60$) ($w=1.0$)

The change of the total fuel cost and total emission amounts, belonging to the best solution obtained for $w=1.0$, according to the

iterations has been shown in Figure 10, the change of the transmission line losses according to the iterations and the active powers generated in each period have been given in Figure 11. Similar graphics have been obtained for all weight values, but have not been added to the study due to the lack of space.

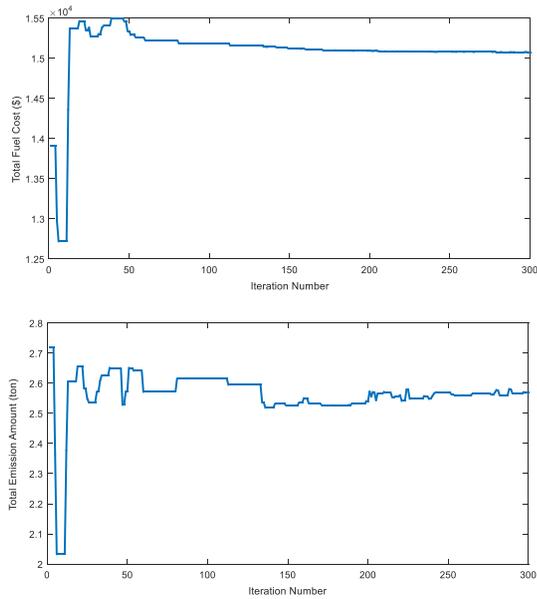


Figure 10: The change of *TFC* and *TEA* according to the iterations (Case-II: With $P/S, \mu=0.60$) ($w=1.0$)

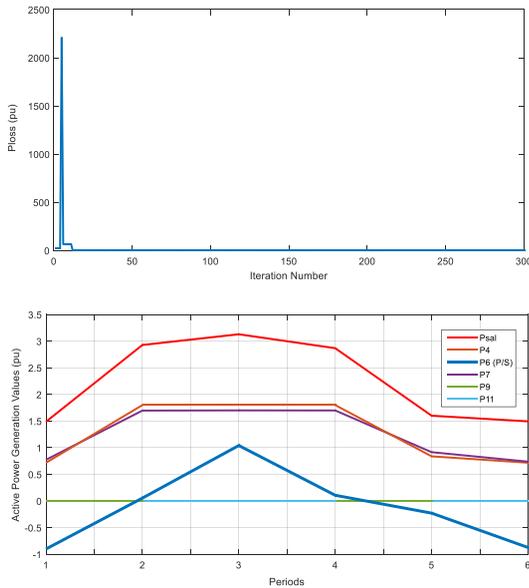


Figure 11: The change of P_{loss} according to the iterations, the change of generated active powers according to the periods (Case-II: With P/S , $\mu=0.60$) ($w=1.0$)

When Figure 10 is examined, it is seen that the total fuel cost value of the sample system with DE algorithm has caught the optimum solution for case-II at about 250th iteration. When the graphic of change of generated active powers according to the periods is looked at, it is seen that the 9th and 11th thermal generation units whose generation costs are high, have not done any production. While P/S hydraulic unit has pumped water to its upper reservoir in 1st, 5th and 6th periods where the load value is low, it has made generation as a generator in 2nd, 3rd and 4th periods where the load value is higher.

In the second case the amounts of water discharge-pumped in every period by the P/S hydraulic unit belonging to the best solution obtained from 30 solutions with DE algorithm for $w=1.0$ and the amount of water in the upper reservoir of this unit at the end of each

period during the operation time have been given in Figure 12, and the final amount of water in the upper reservoir of P/S hydraulic unit at the end of each iteration has been given in Figure 13.

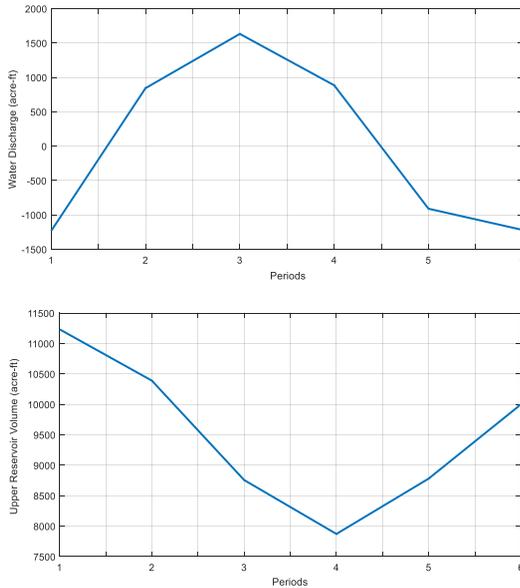


Figure 12: The amounts of water discharge-pumped in hydraulic unit in every period and the amount of water in the upper reservoir at the end of every period (Case-II: With $P/S, \mu=0.60$) ($w=1.0$)

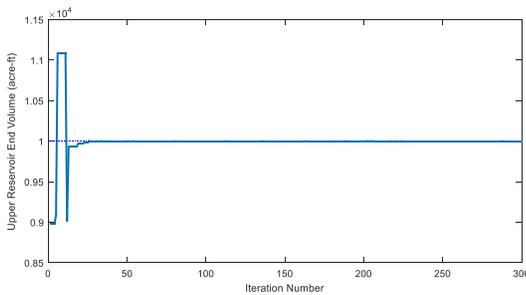


Figure 13: The final amount of water in the upper reservoir of P/S hydraulic unit at the end of every iteration (Case-II: With $P/S, \mu=0.60$) ($w=1.0$)

In this case, the graphic in which the best *TFC*'s obtained from 30 solutions done for $w=1.0$ take place, has been given in Figure 14 and the boxplot belonging to these solutions has been given in Figure 15.

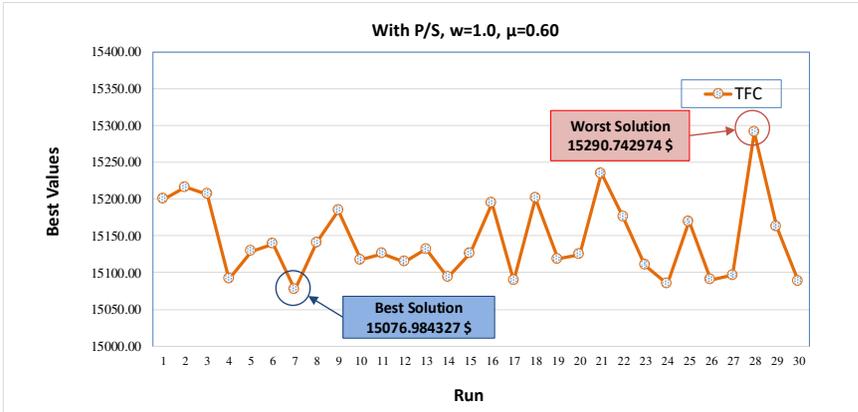


Figure 14: The best *TFC* values obtained from 30 runs (Case-II: With $P/S, \mu=0.60$) ($w=1.0$)

When Figure 14 is examined, it is seen that the best cost value has been obtained as 15076.984327 \$ and the worst cost value has been obtained as 15290.742974 \$ from the 30 solutions of the sample system with *DE* for case-II.

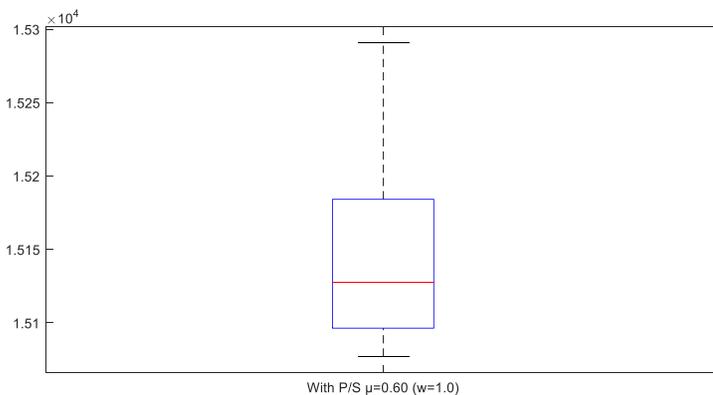


Figure 15: Boxplot (Case-II: With $P/S, \mu=0.60$) ($w=1.0$)

3.3. Case-III: With P/S Hydraulic Unit ($\mu=0.67$)

In the third case, the sample system has been solved by assuming that the demanded power was met with a P/S hydraulic unit having 0.67 cycle efficiency and together with thermal generation units. The statistical values belonging to the total cost function of the problem which was solved 30 runs with the parameter values given in Table 2 and 7, have been given in Table 12.

Table 12: Values obtained from 30 solutions (Case-III: With P/S, $\mu=0.67$)

With P/S, $\mu=0.67$	w		
	1.0	0.5	0.0
Worst	15077.390227 (Run: 15)	8696.077820 (Run: 2)	2332.380116 (Run: 15)
Mean	14982.642048	8681.422216	2308.324221
Best	14956.669569 (Run: 20)	8671.573880 (Run: 9)	2296.667967 (Run: 26)
Std	27.988238	7.990395	8.647647
Total time (s)	3682.686204	3490.484979	3964.108983
Mean time (s)	122.756207	116.349499	132.136966

Power generation values belonging to the best solution among these solutions, transmission line losses and solution times have been given respectively in Tables 13-15.

Table 13: Values belonging to the best solution (Case-III: With P/S, $\mu=0.67$) ($w=1.0$)

w=1.0							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.581844	2.814638	2.894096	2.813690	1.666924	1.559129
	$Q_{GT,1}$	1.145528	2.855755	3.013461	2.855329	1.587983	1.135197
4	$P_{GT,4}$	0.739065	1.797211	1.794454	1.794094	0.833766	0.738720
6	$P_{GPS,6}$	-	0.156992	1.267318	0.162183	-	-
	$P_{PPS,6}$	-1.093435	-	-	-	-0.263660	-1.022660
7	$P_{GT,7}$	0.859650	1.700000	1.700000	1.698726	0.891360	0.808520
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.4957e+04					
E_T (ton)		2.4674					
$P_{total\ loss}$ (pu)		1.892629					
V_{end} (acre-ft)		10000.008069					
Time (s)		125.222724					

Table 14: Values belonging to the best solution (Case-III: With P/S , $\mu=0.67$) ($w=0.5$)

$w=0.5$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.680378	2.650130	2.898605	2.666608	1.820541	1.613879
	$Q_{GT,1}$	1.190297	2.813406	3.014815	2.816325	1.630447	1.182266
4	$P_{GT,4}$	0.848918	1.648483	1.799281	1.663755	0.945445	0.874985
6	$P_{GPS,6}$	-	0.459346	1.262592	0.425182	-	-
	$P_{PPS,6}$	-1.298787	-	-	-	-0.648084	-1.273811
7	$P_{GT,7}$	0.871362	1.696583	1.695836	1.700000	1.024519	0.884304
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.4997e+04					
E_T (ton)		2.3465					
$P_{total\ loss}$ (pu)		1.910054					
V_{end} (acre-ft)		10000.001979					
Time (s)		113.696032					

Table 15: Values belonging to the best solution (Case-III: With P/S , $\mu=0.67$) ($w=0.0$)

$w=0.0$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.582883	2.737091	2.932709	2.676853	2.092758	1.607866
	$Q_{GT,1}$	1.184506	2.845294	3.033027	2.845036	1.758510	1.184099
4	$P_{GT,4}$	0.892012	1.571658	1.746683	1.539540	1.152004	0.870127
6	$P_{GPS,6}$	-	0.571272	1.295540	0.710429	-	-
	$P_{PPS,6}$	-1.297958	-	-	-	-1.275702	-1.292220
7	$P_{GT,7}$	0.923254	1.584750	1.687324	1.537669	1.215539	0.914214
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.5114e+04					
E_T (ton)		2.2967					
$P_{total\ loss}$ (pu)		1.976301					
V_{end} (acre-ft)		10000.008996					
Time (s)		131.911723					

Total fuel cost and total emission amounts belonging to the best solutions obtained from the solution of the sample system with different weight values for 30 runs by adding a P/S unit with 0.67 cycle efficiency to the 6th bus, have been given in Figure 16.

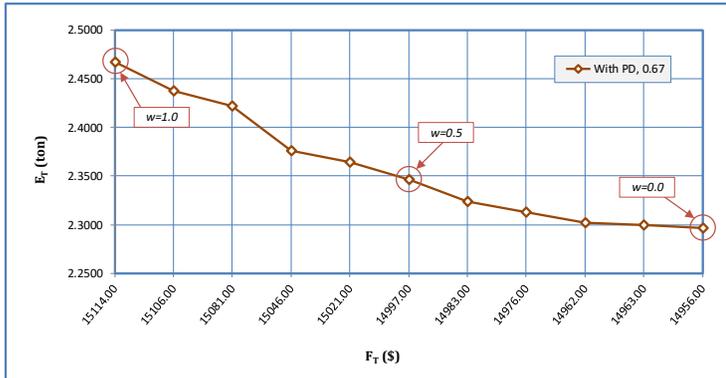


Figure 16: Total fuel cost and total emission amounts (Case-III: With P/S , $\mu=0.67$) ($w=1.0$)

The change of the total fuel cost and total emission amounts, belonging to the best solution obtained for $w=1.0$, according to the iterations has been shown in Figure 17, the change of the transmission line losses according to the iterations and the active powers generated in each period have been given in Figure 18.

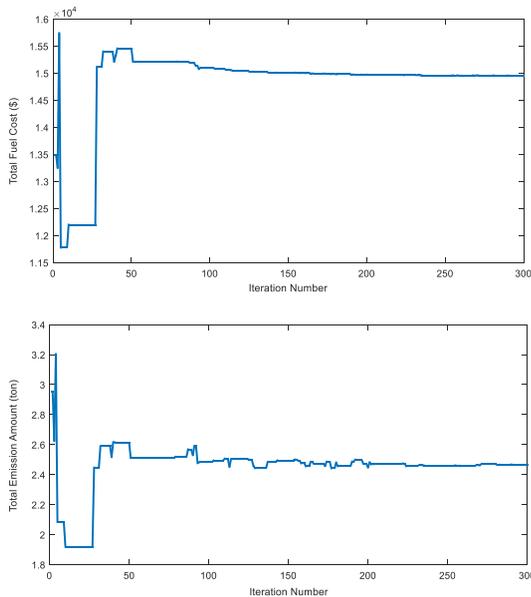


Figure 17: The change of TFC and TEA according to the iterations (Case-III: With P/S , $\mu=0.67$) ($w=1.0$)

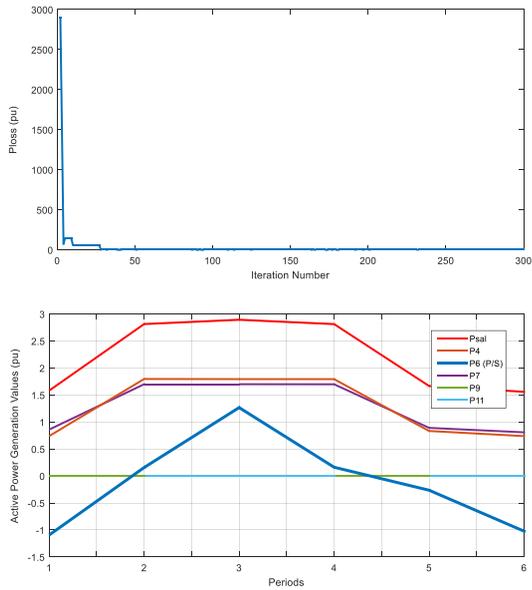


Figure 18: The change of P_{loss} according to the iterations, the change of the generated active powers according to the periods (Case-III: With P/S , $\mu=0.67$) ($w=1.0$)

In the third case the amounts of water discharge-pumped in every period by the P/S hydraulic unit belonging to the best solution obtained from 30 solutions with DE algorithm for $w=1.0$ and the amount of water in the upper reservoir of this unit at the end of each period during the operation time have been given in Figure 19 , and the final amount of water in the upper reservoir of P/S hydraulic unit at the end of each iteration has been given in Figure 20.

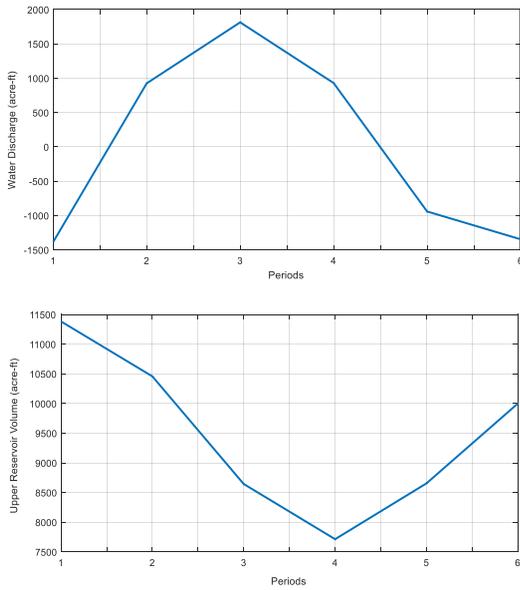


Figure 19: The amounts of water discharge-pumped in every period in hydraulic unit and the final amount of water in the upper reservoir at the end of each period (Case-III: With $P/S, \mu=0.67$) ($w=1.0$)

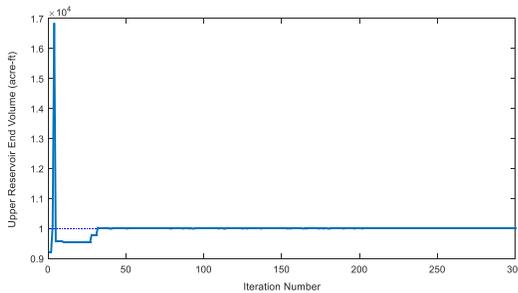


Figure 20: The final amount of water in the upper reservoir of P/S hydraulic unit at the end of every iteration (Case-III: With $P/S, \mu=0.67$) ($w=1.0$)

When Figure 20 is examined, it is seen that in the solution of the sample system with DE algorithm for case-III, the last amount of water in the upper reservoir of P/S hydraulic unit has caught its exact value at about 45th iteration. When the graphic of change of generated active powers according to the periods is looked at, it is seen that the

9th and 11th thermal generation units whose generation costs are high, have not done any production. While P/S hydraulic unit has pumped water to its upper reservoir in 1st, 5th and 6th periods where the load value is low, it has done water discharge in 2nd, 3rd and 4th periods where the load value is higher.

In this case, the graphic in which the best TFC 's obtained from 30 solutions done for $w=1.0$ take place, has been given in Figure 21 and the boxplot belonging to these solutions has been given in Figure 22.

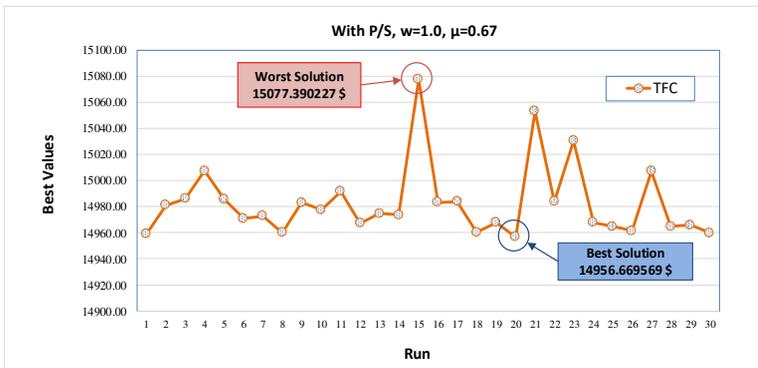


Figure 21: The best TFC values obtained from 30 runs (Case-III: With P/S , $\mu=0.67$) ($w=1.0$)

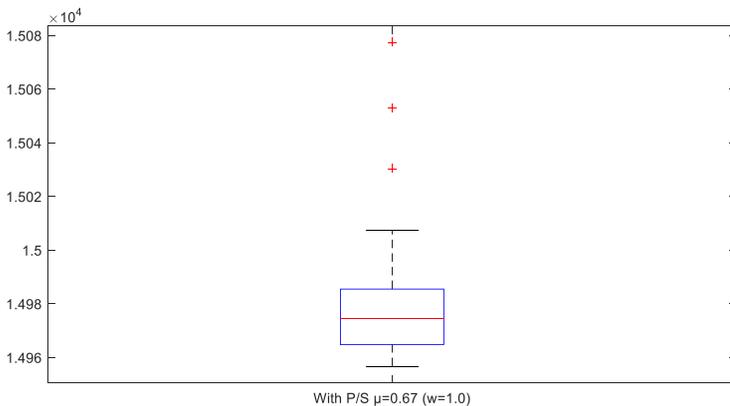


Figure 22: Boxplot (Case-III: With P/S , $\mu=0.67$) ($w=1.0$)

When Figure 21 is examined, it is seen that the worst cost value has been obtained as 15077.390227 \$ at the 15th solution and the best cost value has been obtained as 14956.669569 \$ at the 20th solution from the 30 solutions of the sample system with *DE* for case-III. *DE* algorithm searched in a small interval of 120.720066 \$.

3.4. Case-IV: With *P/S* Hydraulic Unit ($\mu=0.75$)

In the last case, the sample system has been solved 30 times with *DE* by assuming that the demanded power was met with a *P/S* hydraulic unit having 0.75 cycle efficiency and together with thermal generation units. The statistical values belonging to the total cost function obtained from these solutions have been given in Table 16, the data belonging the solution with the best total fuel cost among 30 solutions have been given respectively in Tables 17-19.

Table 16: Values obtained from 30 solutions (Case-IV: With *P/S*, $\mu=0.75$)

With <i>P/S</i> , $\mu=0.75$	<i>w</i>		
	<i>1.0</i>	<i>0.5</i>	<i>0.0</i>
<i>Worst</i>	14842.281806 (Run: 29)	8554.727550 (Run: 19)	2291.772890 (Run: 18)
<i>Mean</i>	14801.465410	8541.299428	2240.676481
<i>Best</i>	14783.249509 (Run: 21)	8532.421966 (Run: 3)	2226.728468 (Run: 9)
<i>Std</i>	17.802415	6.183135	14.258554
<i>Total time (s)</i>	3885.976407	4098.221020	3605.055298
<i>Mean time (s)</i>	129.532547	136.607367	120.168510

Table 17: Values belonging to the best solution (Case-IV: With P/S , $\mu=0.75$) ($w=1.0$)

$w=1.0$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.644966	2.599128	2.857502	2.627915	1.718534	1.644542
	$Q_{GT,1}$	1.175726	2.807173	3.006924	2.812035	1.599174	1.165783
4	$P_{GT,4}$	0.814873	1.740374	1.799975	1.733614	0.866790	0.790109
6	$P_{GPS,6}$	-	0.413995	1.296226	0.393003	-	-
	$P_{PPS,6}$	-1.239877	-	-	-	-0.384481	-1.179951
7	$P_{GT,7}$	0.877123	1.699099	1.699993	1.699675	0.931243	0.839053
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.4783e+04					
E_T (ton)		2.3474					
$P_{total\ loss}$ (pu)		1.883428					
V_{end} (acre-ft)		10000.006417					
Time (s)		126.528977					

Table 18: Values belonging to the best solution (Case-IV: With P/S , $\mu=0.75$) ($w=0.5$)

$w=0.5$							
Bus no (n)	Generation (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.645207	2.598073	2.859327	2.574878	1.885448	1.622598
	$Q_{GT,1}$	1.186215	2.809873	3.008309	2.811589	1.672331	1.186942
4	$P_{GT,4}$	0.864842	1.556280	1.797901	1.573493	1.084894	0.883614
6	$P_{GPS,6}$	-	0.633586	1.298986	0.674555	-	-
	$P_{PPS,6}$	-1.285972	-	-	-	-0.895272	-1.294949
7	$P_{GT,7}$	0.876519	1.665201	1.697933	1.630715	1.081230	0.889637
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.4812e+04					
E_T (ton)		2.2524					
$P_{total\ loss}$ (pu)		1.918726					
V_{end} (acre-ft)		10000.014570					
Time (s)		138.161685					

Table 19: Values belonging to the best solution (Case-IV: With P/S , $\mu=0.75$) ($w=0.0$)

$w=0.0$							
Bus no (n)	Generati on (pu)	Period (j)					
		1	2	3	4	5	6
1	$P_{GT,1}$	1.576437	2.673772	2.996025	2.583456	2.137614	1.653285
	$Q_{GT,1}$	1.184194	2.840465	3.049161	2.853150	1.771951	1.187736
4	$P_{GT,4}$	0.903219	1.525373	1.705896	1.483308	1.179666	0.860587
6	$P_{GPS,6}$	-	0.700384	1.272118	0.939844	-	-
	$P_{PPS,6}$	-1.293632	-	-	-	-1.296287	-1.293223
7	$P_{GT,7}$	0.914079	1.563486	1.693548	1.460232	1.167978	0.880435
9	$P_{GT,9}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	$P_{GT,11}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
F_T (\$)		1.4894e+04					
E_T (ton)		2.2267					
$P_{total\ loss}$ (pu)		1.987604					
V_{end} (acre-ft)		10000.008784					
Time (s)		123.667136					

Total fuel cost and total emission amounts belonging to the best solutions obtained from the solution of the sample system with different weight values for 30 runs by adding a P/S unit with 0.75 cycle efficiency to the 6th bus, have been given in Figure 23.

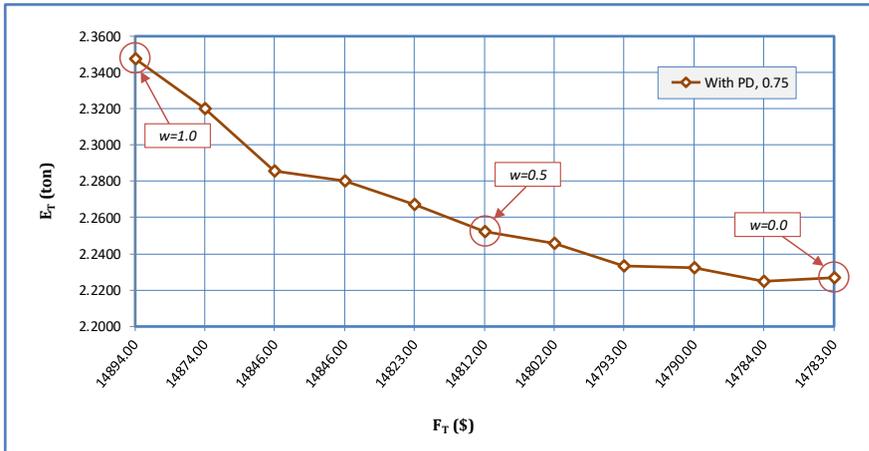


Figure 23: Total fuel cost and total emission amounts (Case-IV: With P/S , $\mu=0.75$) ($w=1.0$)

The change of the total fuel cost and total emission amounts, belonging to the best solution obtained for $w=1.0$, according to the iterations has been shown in Figure 24, the change of the transmission line losses according to the iterations and the active powers generated in each period have been given in Figure 25. Similar graphics have been obtained for all weight values but have not been added to the study due to the lack of space.

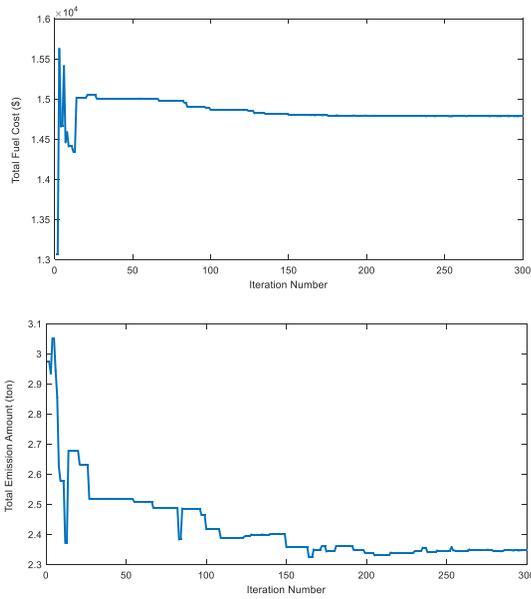


Figure 24: The change of *TFC* and *TEA* according to the iterations (Case-IV: With P/S , $\mu=0.75$) ($w=1.0$)

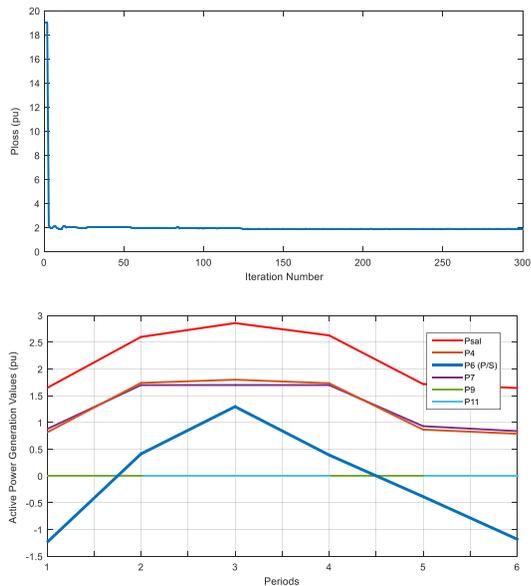


Figure 25: The change of *Ploss* according to the iterations, the change of the generated active powers according to the periods (Case-IV: With P/S , $\mu=0.75$) ($w=1.0$)

When Figure 24 is examined, it is seen that in the solution of the sample system with *DE* algorithm for case-III, the total fuel cost value has caught the optimum result at about 150th iteration. When the graphic of change of generated active powers according to the periods is looked at, it is seen that the 9th and 11th thermal generation units whose generation costs are high, have not done any production again. When the graphic of line losses is looked at, it is seen that after the 75th iteration the line losses have not changed. While *P/S* hydraulic unit has pumped water to its upper reservoir in 1st, 5th and 6th periods where the load value is lower, it has generated as a generator in 2nd, 3rd and 4th periods where the load value is higher.

In the last case the amounts of water discharge-pumped in every period by the *P/S* hydraulic unit belonging to the best solution obtained from 30 solutions with *DE* algorithm for $w=1.0$ and the amount of water in the upper reservoir of this unit at the end of each period during the operation time have been given in Figure 26, and the final amount of water in the upper reservoir of *P/S* hydraulic unit at the end of each iteration has been given in Figure 27.

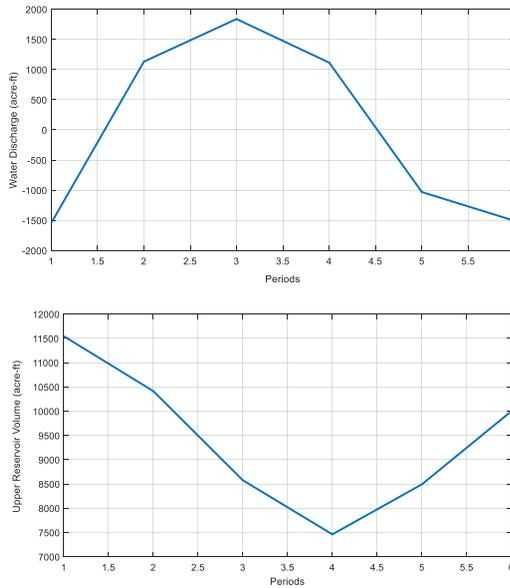


Figure 26: The amounts of water discharge-pumped in every period in hydraulic unit and the amount of water in the upper reservoir at the end of each period (Case-IV: With P/S , $\mu=0.75$) ($w=1.0$)

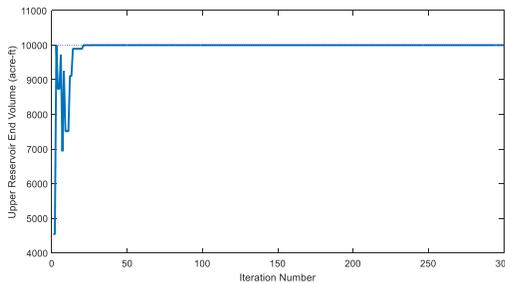


Figure 27: The final amount of water in the upper reservoir of P/S hydraulic unit at the end of each iteration (Case-IV: With P/S , $\mu=0.75$) ($w=1.0$)

In this case, the graphic in which the best TFC 's obtained from 30 solutions done for $w=1.0$ take place, has been given in Figure 28 and the boxplot belonging to these solutions has been given in Figure 29.

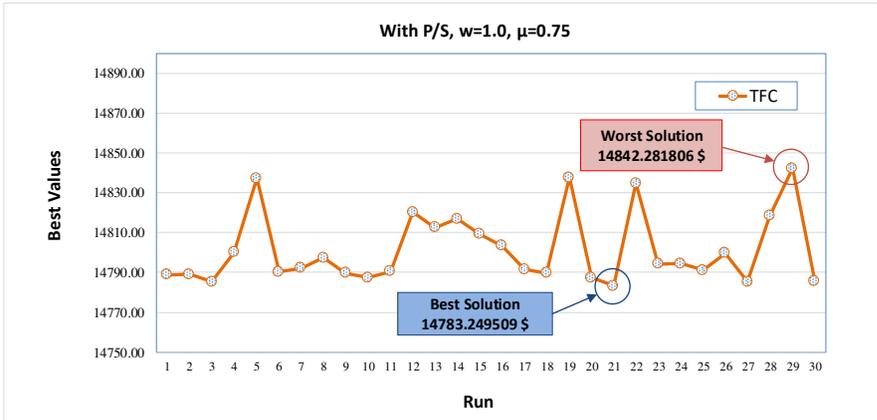


Figure 28: The best *TFC* values obtained from 30 runs (Case-IV: With *P/S*, $\mu=0.75$) ($w=1.0$)

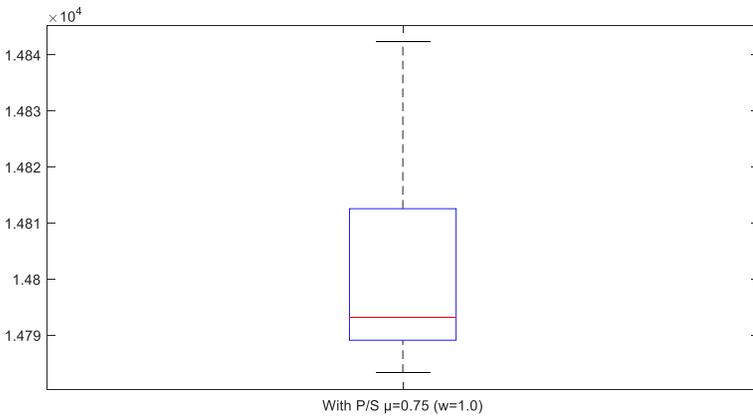


Figure 29: Boxplot (Case-IV: With *P/S*, $\mu=0.75$) ($w=1.0$)

When Figure 28 and 29 are examined, it is seen that the worst cost value has been obtained as 14842.281806 \$ at the 29th solution and the best cost value has been obtained as 14783.249509 \$ at the 21st solution from the 30 solutions of the sample system with *DE* for case-IV.

The whole of the total fuel costs obtained with different weight values for all cases has been given in Figure 30.

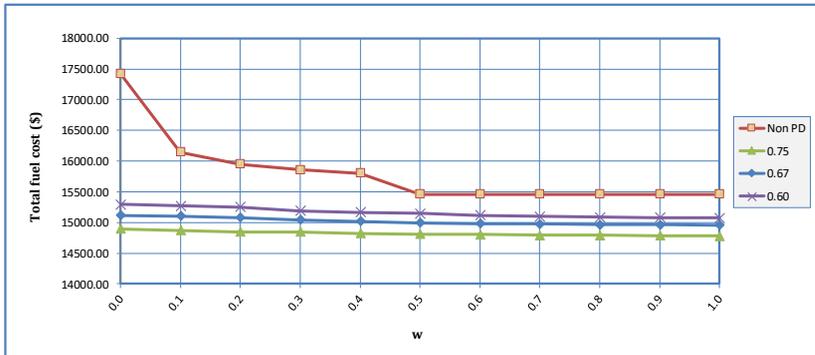


Figure 30: Total fuel cost values obtained for different cycle efficiencies and weight values

Those results can be inferred from Figure 30. More decisive results have been obtained in the cases where there is P/S unit, compared with the cases without P/S unit. P/S hydraulic unit added to the system has provided a visible economic benefit. As the cycle efficiency of P/S hydraulic unit increases, the economic benefit it provides increases as well. Therefore, the most economic benefit has been provided by the P/S hydraulic unit with 0.75 cycle efficiency.

The whole of the total emission amounts obtained with different weight values for all cases has been given in Figure 31.

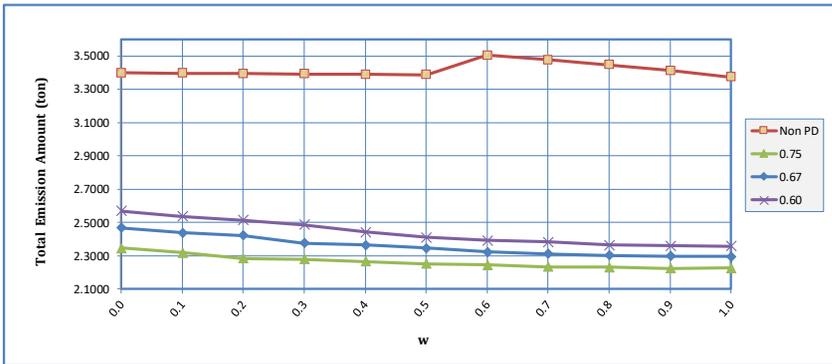


Figure 31: Total emission amount values obtained for different cycle efficiencies and weight values

When Figure 31 is examined, it is seen that as is expected in cases with *P/S* unit, total emission amounts decreasing with weight values have been obtained. The *P/S* hydraulic unit added to the system has provided a visible decrease in emission amount. The reason of this is that the emission amount of *P/S* hydraulic unit, which is a renewable generation system, is 0. With the generation done by *P/S* hydraulic unit, 9th and 11th units whose emission coefficients are high, were never used. As the cycle efficiency of *P/S* hydraulic unit increases, the emission amount benefit it provides increases. Therefore, the least amount of emission has been provided by the *P/S* hydraulic unit with 0.75 cycle efficiency.

The comparison of the best *TFC*, obtained from the solution of the sample power system by using *DE* for all cases, with the results in literature have been given in Table 20 and 21. When the table is examined, it has been seen that the *TFC* values of the problem, which was solved previously, obtained with *DE* for all cases in this study are close to the values obtained by *GA* and *F-MSG* approaches. The

sample system was solved with *F-MSG* method in the table by doing both *P* and *Q* optimizations. Hydrothermal scheduling problem including P/S hydraulic unit has been solved with different approaches (Basu, 2020).

Table 20: Comparison of the optimal solution values which are obtained by different methods ($w=1.0$)

	<i>GA (Demir, 2010)</i>	<i>GA (Demir, 2010)*</i>	<i>F-MSG (Fadil & Urazel, 2014)*</i>	<i>DE</i>
	F_T (\$) E_T (ton)	F_T (\$) E_T (ton)	F_T (\$) E_T (ton)	F_T (\$) E_T (ton)
<i>Case-I</i> <i>Non P/S</i>	15474.0000	15387.0000	15280.4746	15458.0081
	3.384400	3.357900	3.327797	3.398181
<i>Case-II</i> <i>With P/S, $\mu=0.60$</i>	15195.0000	15102.0000	14730.4413	15076.9843
	2.538100	2.512400	2.314850	2.568396
<i>Case-III</i> <i>With P/S, $\mu=0.67$</i>	14947.0000	14844.0000	14575.8040	14956.6695
	2.461900	2.432700	2.273622	2.467393
<i>Case-IV</i> <i>With P/S, $\mu=0.75$</i>	14731.0000	14629.0000	14394.4752	14783.2495
	2.456600	2.428300	2.233563	2.347402

*with active and reactive power optimization

Table 21: Comparison of the optimal solution values which are obtained by different methods ($w=0.0$)

	<i>GA (Demir, 2010)</i>	<i>GA (Demir, 2010)*</i>	<i>F-MSG (Fadil & Urazel, 2014)*</i>	<i>DE</i>
	F_T (\$) E_T (ton)	F_T (\$) E_T (ton)	F_T (\$) E_T (ton)	F_T (\$) E_T (ton)
<i>Case-I</i> <i>Non P/S</i>	15562.0000	15475.0000	15447.9144	17416.9875
	3.309200	3.281600	3.271275	3.373464
<i>Case-II</i> <i>With P/S, $\mu=0.60$</i>	15038.00	14916.0000	14767.559	15296.6314
	2.319600	2.285100	2.289196	2.358440
<i>Case-III</i> <i>With P/S, $\mu=0.67$</i>	14858.00	14742.0000	14632.1935	15113.8638
	2.289900	2.258300	2.250756	2.296667
<i>Case-IV</i> <i>With P/S, $\mu=0.75$</i>	14575.0000	14458.0000	14431.9547	14894.2880
	2.212000	2.180100	2.205826	2.226728

*with active and reactive power optimization

CONCLUSION

In the study the problem of the environmental/economic operation of *P/S* hydraulic generation units which has a great advantage in terms of storing the generated electrical power, has been handled. In order to turn the environmental/economic operation problem which is a multi-objective optimization problem, into a single-objective problem, Weight sum method (*WSM*) has been used, while differential evolution (*DE*) algorithm has been used for the optimization of the scalarized objective function. The use of only thermal fuel generation units in power generation causes an environmental cost as well as the financial cost. The economic and environmental benefit provided by the use of (*P/S*) hydraulic generation units together with these generation units has been calculated approximately. With these units, in the case of low load demand, water has been stored in the upper reservoirs with the extra power generated by the thermal generation units; and in the case of high-power demand, generation has been done and the continuity of the system has been contributed.

As an example, for the problem of environmental/economic operation of *P/S* hydraulic units a system taking place in literature has been selected. The sample system has been solved with *DE* with different weight values for four different cases. In the first case the sample system has been solved 30 runs under present constraints assuming that only the thermal generation units did generation. The best total fuel cost value obtained from those solutions has been *15458.0081* \$, and the best total emission amount has been *3.3981 tons*. In the second

case, the same system has been solved by adding a *P/S* hydraulic unit with 0.60 cycle efficiency to it and in this case the best total fuel cost has been found as 15296.6314 \$, and the best total emission amount as 2.3584 tons. In the third case, the same system has been solved by adding a *P/S* hydraulic unit with 0.67 cycle efficiency to it and in this case the best total fuel cost has been found as 15113.8638 \$, and the best total emission amount as 2.2966 tons. In the fourth case, the same system has been solved by adding a *P/S* hydraulic unit with 0.75 cycle efficiency to it and in this case the best total fuel cost has been found as 14894.2880 \$, and the best total emission amount as 2.2267 tons.

With *P/S* hydraulic unit in the same operation time/duration and with the same load demand values, for 0.60 efficiency 381.0238 \$ cost, 0.8298 tons of emission amount gain, for 0.67 efficiency 501.3386 \$ cost, 0.9308 tons of emission amount gain, for 0.75 efficiency 674.7586 \$ cost, 1.0507 tons of emission amount gain have been obtained in the sample system. It has been seen in these cases that as the gain value of the *P/S* hydraulic unit increases, there becomes a decrease in the total fuel cost and the total emission amount.

The aim of the study, is to show that environmental/economic operation of a system consisting of thermal and *P/S* hydraulic generation units which have a great importance in electrical engineering, and which is one of the complex and nonlinear optimization problems with many constraints and is difficult to solve with mathematical methods, can also be solved with *DE*. In addition

to this, another aim is to define the cost and emission amount benefit for the producer that will be provided by the use of *P/S* hydraulic generation units having different efficiencies together with the thermal generation units in an electrical power system. In the following studies it is thought to integrate different renewable power sources into the sample system together with *P/S* hydraulic units and to add the solutions of the sample system to the literature.

REFERENCES

- Jebaraj, L., Venkatesan, C., Soubache, I., & Christober, C. (2017). Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review. *Renewable Sustainable Energy Rev.*, 77: 1206-1220.
- Mahdi, F. P., Vasant, P., Kallimani, V., Watada, J., Fai, P. Y. S., & Abdullah-Al-Wadud, M. (2018). A holistic review on optimization strategies for combined economic emission dispatch problem. *Renewable Sustainable Energy Rev.*, 81: 3006-3020.
- Özyön, S., Temurtaş, H., Durmuş, B., & Kuvat, G. (2012). Charged system search algorithm for emission constrained economic power dispatch problem. *Energy*, 46(1): 420-430.
- Deane, J. P., Gallachoir. B.P.O., & McKeogh, E. J. (2010). Techno-economic review of existing and new pumped hydro energy storage plant. *Renewable Sustainable Energy Rev.*, 14: 1293-1302.
- Ardizzon, G., Cavazzini, G., & Pavesi, G. (2014). A new generation of small hydro and pumped-hydro power plants: Advances and future challenges. *Renewable Sustainable Energy Rev.*, 31: 746-761.
- Rehman, S., Al-Hadhrani, L. M., & Alam, MdM. (2015). Pumpes hydro energy storage system: A technological review. *Renewable Sustainable Energy Rev.*, 44: 586-598.
- Fadıl, S. & Yaşar, C. (2000). A pseudo spot price algorithm applied to the pumped-storage hydraulic unit scheduling problem. *Turk J. Elec. Eng.*, 8(2): 93-109.
- Fadıl, S. & Urazel, B. (2013). Solution to security-constrained non-convex pumped-storage hydraulic unit scheduling problem by modified subgradient algorithm based on feasible values and pseudo water price. *Electr. Power Compon. Syst.*, 41: 111-135.
- Liang, R. H. (2000). A noise annealing neural network for hydroelectric generation scheduling with pumped-storage units. *IEEE Trans. Power Syst.*, 15(3): 1008-1013.

- Cheng, C., Su, C., Wang, P., Shen, J., Lu, J., & Wu, X. (2018). An MILP-based model for short-term peak shaving operation of pumped-storage hydropower plants serving multiple power grids. *Energy*, 163: 722-733.
- Simab, M., Javadi, M. S., & Nezhad, A. E. (2018). Multi-objective programming of pumped-hydro-thermal scheduling problem using normal boundary intersection and VIKOR. *Energy*, 143: 854-866.
- Nezhad, A. E., Javadi, M. S., & Rahimi, E. (2014). Applying augmented ϵ -constraint approach and lexicographic optimization to solve multi-objective hydrothermal generation scheduling considering the impact of pumped-storage units. *Int. J. Electr. Power Energy Syst.*, 55: 195-204.
- Chen, P. H. (2008). Pumped-storage scheduling using evolutionary particle swarm optimization. *IEEE Trans. Energy Convers.*, 23(1): 294-301.
- Kocaman, A. S. (2019). Optimization of hybrid energy systems with pumped hydro storage-A case study for Turkey. *J. Faculty Eng. Archit. Gazi Univ.*, 34(1): 53-67.
- Wang, W., Li, C., & Liao, X. (2017). Qin H. Study on unit commitment problem considering pumped storage and renewable energy via a novel binary artificial sheep algorithm. *Appl. Energy*, 187: 612-626.
- Demir, C. (2010). Solution to environmental economic dispatch problem of electrical energy system included pumped storage hydraulic unit using genetic algorithm. Eskişehir Osmangazi University, Master of Science Thesis, (in Turkish)
- Fadıl, S., Demir, C., & Urazel, B. (2013). Solution to security constrained environmental pumped-storage hydraulic unit scheduling problem by genetic algorithm. *ISRN Power Eng.*, 1-12.
- Fadıl, S. & Urazel, B. (2014). Solution to security constrained environmental/economic pumped-storage hydraulic unit scheduling problem by modified subgradient algorithm based on feasible values and pseudo water price. *Int. J. Electr. Power Energy Syst.*, 60: 399-413.
- Wood, A. J. & Wollenberg, B. F. (1996). *Power Generation Operation and Control*. John Wiley & Sons, New York.

- Ma, T., Yang, H., Lu, L., & Peng, J. (2015). Pumped storage-based standalone photovoltaic power generation system: Modeling and techno-economic optimization. *Appl. Energy*, 137: 649-659.
- Storn, R. (1997). Price K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11: 341-359.
- Naila, Haroon, S. S., Hassan, S., Amin, S., Sajjad, I. A., Waqar, A., Aamir, M., Yaqoob, M., & Alam, I. (2018). Multiple fuel machines power economic dispatch using stud differential evolution. *Energies*, 11(1393): 1-20.
- Mandal, D., Chatterjee, A., & Bhattacharjee, A. K. (2013). Design of Fully digital controlled shaped beam synthesis using differential evolution algorithm. *Int. J. Antennas Propag.*, 1-9.
- Özyön, S. (2020). Optimal short-term operation of pumped-storage power plants with differential evolution algorithm. *Energy*, 194: 1-13, Article 116866.
- Basu, M. (2020). Optimal generation scheduling of hydrothermal system with demand side management considering uncertainty and outage of renewable energy sources. *Renewable Energy*, 146: 530-542.

CHAPTER 2

BACK PROPAGATION NEURAL NETWORK BASED CLASSIFICATION FOR ELECTRIC MOTORS

Mina Ghorban Zadeh Badeli¹

Dr. Duygu BAYRAM KARA²

¹Istanbul Technical University, Electrical and Electronics Fac., Electrical Eng.,
Istanbul, Turkey, badeli18@itu.edu.tr, ORCID ID 0000-0003-1855-9278

²Istanbul Technical University, Electrical and Electronics Fac., Electrical Eng.,
Istanbul, Turkey, bayramd@itu.edu.tr, ORCID ID 0000-0001-8184-8510

*A prior version of this study is presented in the 10th International Conference on
Engineering and Natural Sciences on May 16-18, 2021 / Siirt, Turkey.

INTRODUCTION

Electric motors play an essential role in modern industry. In recent years, the most common types of electric motors used in industrial applications were: DC motors, induction motors (IMs), permanent magnet synchronous motors (PMSMs), switched reluctance motors and brushless DC motors (Hashemnia & Asaei, 2008). Among different types of electric motors, IMs are the most widely used electro-mechanical devices in industrial applications due to their high reliability, simple structure, high efficiency and less maintenance need (Faiz, Ebrahimi, & Sharifian, 2006; Hashemnia & Asaei, 2008). Owing to these significant characteristics, condition monitoring (CM) and fault detection (FD) of IMs are vital issues that must be considered seriously (Seera, Lim, Nahavandi, & Loo, 2014). In order to reach this aim, some useful signal-based techniques are developed to detect failures and prevent sudden breakdowns and probable damages in IMs. The motor current signature analysis (MCSA) is one of the most common methods that uses the Fast Fourier Transform (FFT) and power spectral density (PSD) of the signals to diagnose the faults (Gu et al., 2015). The Short-Time Fourier Transform (STFT) and Wavelet Transform were also used to analyze the signals in both time and frequency domains for CM of IMs largely (Benbouzid, Vieira, & Theys, 1999; Gritli et al., 2017). It is important to notice that different types of signals may be used to monitor the condition of the IM such as voltages, currents, vibrations and temperatures. Nevertheless, vibration monitoring is the most appropriate condition

monitoring technique. Since vibration analysis results in comprehensive information associated with both mechanical and electrical faults. Fortunately, by the improvement of the automated tools, the conventional signal processing methods have been replaced with soft computing techniques over time. ANNs are one of the famous methods of soft computing techniques that have been applicable for FD of power systems in recent years. There are different types of vibration-based ANN applications for the condition monitoring and fault identification of the IMs. As an example, to detect the bearing faults, J. Zarei (Zarei, 2012) compared the efficiency of the extracted time-domain features and the extracted time-frequency-domain features as the input parameters of the ANN (Zarei, 2012). In another study paper, J. Zarei. et al. (Zarei, Tajeddini, & Karimi, 2014) employed the neural network as a filter for removing non-bearing fault components (RNFC) of the bearing vibration signals. Afterward, they have utilized multi-layered perceptron (MLP) neural networks to classify the bearing faults using pattern recognition techniques (Zarei et al., 2014). Other researchers such as M. Farajzadeh-Zanjani (Farajzadeh-Zanjani, Razavi-Far, Saif, & Rueda, 2016) performed useful research work on the best feature extraction method and the best feature selection method for the vibration signals to train the classifiers as the fault detectors of the bearings in the IMs. In the proposed study, the Wavelet Packet Transform (WPT) was availed to analyze the vibration signals by extracting the informative frequency-domain features. Besides, the Linear Discriminant Analysis (LDA) was selected for the dimension reduction of the extracted set of

the features. A fault classification module is also designed consisting of different classifiers such as MLP, Naive Bayes, support vector machine (SVM) with the radial basis function (RBF) kernel, K-nearest neighbor (KNN), random forest, and decision tree (Farajzadeh-Zanjani et al., 2016).

In this study, a feedforward neural network (FFNN) trained with a backpropagation algorithm (BPNN) is designed to classify the vibration data associated with different aging levels of an IM. The motivation of the study is tracking the condition of the motor, providing required precautions, reducing the consequential damages and increasing motor life. In order to ensure the accuracy rates of the proposed network, the created model is primarily applied to artificial vibration data, and then it is verified on experimental vibration data. In this study, after presenting the introduction section, the first section discusses the mathematical background of the ANNs with clarity. In the second section, associated information with the artificial and experimental vibration data is explained and the time-domain statistical features are discussed in detail. The third section investigates the condition estimations of the proposed BPNN model using three different performance assessment plots and three different quality measure parameters. Eventually, it was ascertained that the developed BPNN could efficiently classify various conditions of an IM.

1. ARTIFICIAL NEURAL NETWORK (ANN)

Artificial neural networks can be defined as very functional tools that act very similar to biological neurons. In fact, ANNs consist of straightforward, distributed, and strong processing units called artificial neurons which can be used to emulate the behavior and the functions of the biological neurons in the human brain. These simple elements of an ANN are able to replicate intelligent data evaluation techniques such as pattern recognition, classification, and generalization (Malik, 2005). Three significant concepts that must be defined before designing a simple ANN model are:

- Topology of the network
- Type of the training algorithm (learning procedure)
- Type of the activation functions

Among the different shallow network topologies and learning algorithms of neural networks, the multi-layered FFNN was the first and the simplest type of artificial neural network (Schmidhuber, 2015). There are three layers of nodes in the structure of a multi-layered perceptron network: an input layer, one or more hidden layers (HLs), and an output layer. Neurons belonging to a layer have to be connected to neurons of a different layer (Rosenblatt, 1961). Besides, the information in this network moves only from the input layer directly through any hidden layers to the output layer without any cycle or feedback (Schmidhuber, 2015). A three-layered FFNN model is demonstrated in Figure 1. In the MLP neural networks, each input

element, $x_i (i = 1, 2, \dots, n)$, is multiplied by an interconnection weight, w_{ij} , and then the weighted input is inserted into a sum function for each neuron. Afterward, by the sum of a bias value, b , and the weighted sum of the input data for each neuron, the resultant input can be calculated. The mathematical model of the resultant input, P_j , for neuron j is presented in Equation 1.

$$P_j = \sum_{i=1}^n w_{ij} x_i + b_j \quad (1)$$

Afterward, a non-linear activation function, f , is applied to the resultant input, P_j , in the hidden layer. The mathematical model for the outputs of the network, $Y_o (o = 1, 2, \dots, k)$, is represented in Equation 2.

$$Y_o = \sum_{j=1}^m V_{jo} f(P_j) \quad (2)$$

where V_{jo} is defined as the interconnection weight of the nodes between the hidden layer and the output layer.

Assigning the proper training algorithm is a significant issue that affects the success of prediction in neural networks. Many training methods have been established to optimize the learning procedure of the ANNs. The most prevalent technique is the backpropagation algorithm in which the training is implemented in both forward and backward directions. Indeed, in the BP algorithm, the difference between output and the actual value is computed as the error.

Afterward, the resultant error is fed back towards the input layer through the network. Using the produced information, the algorithm adjusts the weights of each connection repeatedly to minimize the error function (Rumelhart, Hinton, & Williams, 1986). Levenberg–Marquardt (LM), Bayesian Regulation and Scaled Conjugate Gradient (SCG) are some of the useful backpropagation training algorithms used for many ANN applications. After completing the training process, the neural network is usually tested with the new data sets. In this study, both Levenberg–Marquardt and Scaled Conjugate Gradient backpropagation algorithms are examined as the optimization algorithms, and then the optimum classifier is selected according to their optimum success. Further information regarding the developed ANN topology is presented in detail in section 3.

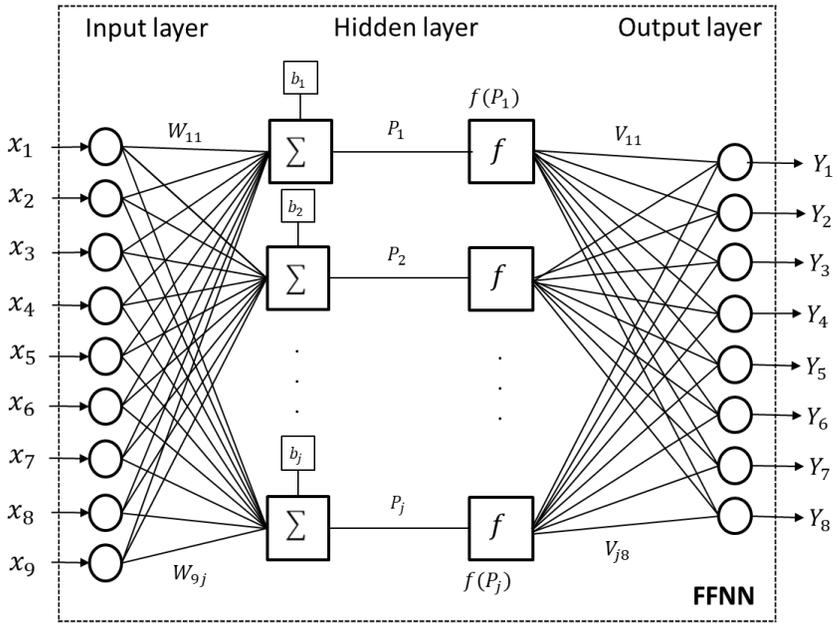


Figure 1: A Three-layered FFNN Architecture

2. VIBRATION DATA AND STATISTICAL FEATURES

2.1. Experimental Vibration Data

In order to collect the experimental vibration data, a 5 HP, three phases and four poles IM is subjected to Electrical Discharge Machining (EDM), thermal aging and chemical aging processes. As a result, seven aging cycles are obtained. In this study, the electrical and mechanical signals that have been collected besides vibration signals are not used. In this experiment, the accelerometers are located in six different locations on the motor and only the vibration data collected from one sensor are used eventually. The data acquisition system consists of a patch panel and a NI SCXI as the signal conditioning interface. NI SCXI-1142 8-channel elliptic low-pass filter module is

also used for the conditioning of the vibration signals. In order to prevent high-frequency noise interference, an anti-aliasing filter with a 4 kHz cut-off frequency is employed. Moreover, the vibration data is collected at the 12 kHz sampling frequency. Figure 2 represents the experimental vibration signals of the healthy and the aged IM in both time and frequency domains. Compared to healthy vibration signals, the frequency components for the aged case are amplified intensely between 2 kHz and 4 kHz. Also, regarding time-domain analysis, the vibration amplitude increases noticeably with aging.

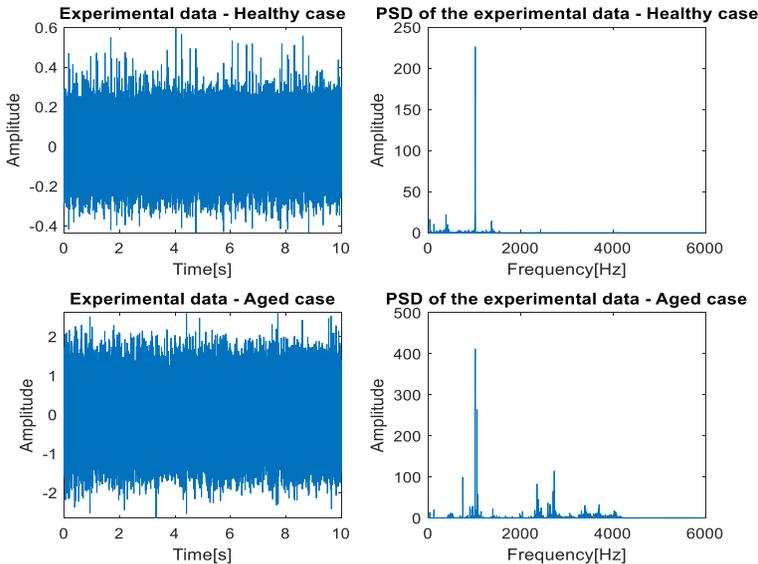


Figure 2: Representation of the Healthy and Aged Experimental Vibration Data in Time and Frequency Domains

2.2. Artificial Vibration Data

In order to develop the algorithm, an artificial vibration data set is needed for this study. To construct the artificial vibration data, a

structure similar to the experimental vibration data is employed to ensure the behavior of the created artificial vibration data. Since the speed of the motor in the experiment is 1742 rpm and the rotating frequency of the shaft is calculated as 29.03 Hz, the fundamental frequency of the artificial vibration data is also assumed to be 29 Hz. Indeed, six different cycles of vibration data associated with different aging levels of the IM have been constructed artificially. The number of aging cycles is intentionally chosen different from experimental data. The first cycle exhibits the vibration signal acquired from a healthy IM and then by weighting the proposed noises, five more signals are constructed according to the different aging levels of the motor. In summary, each artificial vibration cycle is constructed by the sum of a fundamental signal containing the rotating frequency with its harmonics and the weighted noises. The noises are considered to be the Gaussian (normal) distributed random signals (GDRS) filtered by four different 6th order Butterworth filters to obtain specific band noises. The cut-off frequency and the sampling frequency for the artificial vibration data are 4 kHz and 15 kHz, respectively. The mathematical definition of the constructed artificial vibration data for each cycle is depicted in Equation 3.

$$\textit{Artificial Vibration Data} = FC + \sum_{n=1}^4 W_n N_n \quad (3)$$

where

- FC: Fundamental component that consists of rotation frequency and its specific harmonics,

- W_n : Arbitrary values of the weight factors; the weight factors for the healthy case are the smallest values while they increase by the aging,
- N_1 : GDRS filtered between 0.02 kHz and 1 kHz,
- N_2 : GDRS filtered between 1 kHz and 2 kHz,
- N_3 : GDRS filtered between 2 kHz and 3 kHz,
- N_4 : GDRS filtered between 3 kHz and 4 kHz.

The amplitude of the first cycle of the artificial vibration data varies in a small interval. Besides, the amplitude of each aged artificial vibration cycle is greater than its prior cycle. Therefore, the artificial cycle 6 has the greatest amplitude as an aged IM. The healthy and the aged artificial vibration data in both time and frequency domains are illustrated in Figure 3. By comparing the power spectral densities (PSDs), it is determined that the amplitudes of frequency components for the aged case are amplified drastically between 2 kHz and 4 kHz.

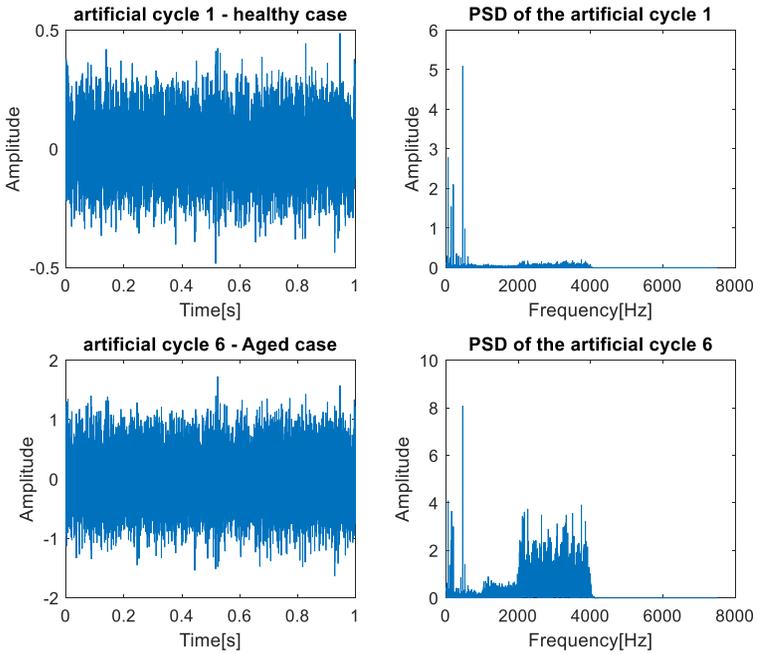


Figure 3: Representation of the Healthy and Aged Artificial Vibration Data in Time and Frequency Domains

2.3. Statistical Features

The major challenge for the training of the proposed ANN is the selection of suitable features as the input parameters of the network that may complicate the network structure. It is necessary to select the correct features to increase the performance of the network and reduce the network input dimensions and training time (Rad, Torabizadeh, & Noshadi, 2011). One of the most popular techniques for extracting the specific features is to calculate the statistical parameters of each data set. There are many research works in which the most efficient time-domain statistical parameters are determined as the inputs of an ANN

for CM and FD purposes in IMs (Saravanan, Siddabattuni, & Ramachandran, 2010; Zarei, 2012; Zarei et al., 2014). In this study, nine statistical features are calculated as the input parameters of the proposed BPNN. Mean value, maximum value, kurtosis, skewness, variance, standard deviation, root mean squared (RMS) value, peak-to-peak amplitude and crest factor are the nine statistical features that have been computed for each data set of both artificial and experimental data. Table 1 indicates the mathematical representation of the employed statistical features.

Table 1: Mathematical Representation of the Statistical Parameters

Statistical Parameters	Equations
Mean value	$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$
Maximum value	$x_{max} = \max \{x_i\}$
Kurtosis	$\bar{x}^4 = \frac{\sum_{i=1}^N (x_i - \bar{x})^4}{N} \times \frac{1}{(\delta^2)^2}$
Skewness	$\bar{x}^3 = \frac{\sum_{i=1}^N (x_i - \bar{x})^3}{N} \times \frac{1}{(\delta^2)^{\frac{3}{2}}}$
Variance	$\delta^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$
Standard deviation	$\delta = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$
Root mean squared (RMS) value	$x_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i ^2}$
Peak-to-peak amplitude	$x_{peak-to-peak} = \max \{x_i\} - \min \{x_i\}$
Crest factor	$C = \frac{ x_{max} }{x_{rms}}$

where x_i is defined as the data samples and N is the total number of samples for each data set.

3. AGING CLASSIFICATION BY BPNN

In this study, a multi-layered BPNN model is developed on MATLAB to classify the different aging levels of an IM. The proposed BPNN comprises one hidden layer. Levenberg-Marquardt and Scaled Conjugate Gradient optimization algorithms are applied to the proposed network. Besides, for both types of training methods, two different transfer functions are employed in the hidden layer. The logistic-sigmoid (log-sig) and the hyperbolic-tangent-sigmoid (tan-sig) are the two activation functions that are selected to quantify the outputs of the hidden layer. The amplitude of the logistic-sigmoid transfer function varies between 0 and 1 while the amplitude of the hyperbolic-tangent-sigmoid transfer function varies between -1 and 1. Besides, the derivative of the hyperbolic-tangent-sigmoid transfer function will be increased sharply compared to the derivative of the logistic-sigmoid function. The training, validation, testing and total accuracy rates of the proposed network for both artificial and experimental data are shown in Table 2 and Table 3, respectively. It is important to notice that the neural network models with different optimization algorithms and transfer functions are firstly examined on the created artificial vibration data and then they have been verified on the experimental data. By comparison of the estimated output results written in Table 2 and Table 3, the BPNN with 4 hidden neurons is chosen as the optimal hidden layer geometry for both artificial and experimental data. Indeed, the optimum classifier is trained with the Levenberg-Marquardt BP algorithm and contains one logistic-sigmoid

transfer function in the hidden layer. It is obvious that the proposed neural network with the logistic-sigmoid transfer function outperforms the network containing the hyperbolic-tangent-sigmoid transfer function. This is due to the fact that the logistic-sigmoid functions possess the unique ability to respond more sensitively for very small differences in the training set. Furthermore, for the classification studies, the logistic-sigmoid transfer function is the better choice. Since the output can be predicted as the probability values which must be between 0 and 1. The total accuracy rate of the optimum classifier for the artificial vibration data and the experimental vibration data are 100% and 98.3%, respectively.

Table 2: Table of the BPNN Accuracy Rates for the Artificial Vibration Data

Number of hidden neurons	Training function	Transfer function used in the HL	Artificial vibration data			
			Training accuracy	Validation accuracy	Testing accuracy	Total accuracy
10	LM	log-sig	90.3%	71.4%	57.1%	82.2%
9	LM	log-sig	74.2%	28.6%	28.6%	60%
8	LM	log-sig	100%	85.7%	85.7%	95.6%
7	LM	log-sig	83.9%	42.9%	14.3%	66.7%
6	LM	log-sig	100%	100%	100%	100%
5	LM	log-sig	100%	85.7%	57.1%	91.1%
4	LM	log-sig	100%	100%	100%	100%
3	LM	log-sig	77.4%	85.7%	71.4%	77.8%
10	LM	tan-sig	100%	71.4%	42.9%	86.7%
9	LM	tan-sig	100%	57.1%	71.4%	88.9%
8	LM	tan-sig	100%	71.4%	71.4%	91.1%
7	LM	tan-sig	96.8%	57.1%	28.6%	80%
6	LM	tan-sig	67.7%	71.4%	57.1%	66.7%
5	LM	tan-sig	100%	100%	42.9%	91.1%
4	LM	tan-sig	100%	85.7%	85.7%	95.6%
3	LM	tan-sig	100%	100%	100%	100%
10	SCG	log-sig	100%	100%	71.4%	95.6%
9	SCG	log-sig	100%	57.1%	57.1%	86.7%
8	SCG	log-sig	54.8%	28.6%	28.6%	46.7%
7	SCG	log-sig	93.5%	71.4%	28.6%	80%
6	SCG	log-sig	100%	85.7%	85.7%	95.6%
5	SCG	log-sig	90.3%	57.1%	71.4%	82.2%
4	SCG	log-sig	87.1%	100%	71.4%	86.7%
3	SCG	log-sig	67.7%	71.4%	57.1%	66.7%
10	SCG	tan-sig	100%	85.7%	57.1%	91.1%
9	SCG	tan-sig	96.8%	42.9%	57.1%	82.2%
8	SCG	tan-sig	100%	42.9%	28.6%	80%
7	SCG	tan-sig	93.5%	85.7%	42.9%	84.4%
6	SCG	tan-sig	100%	100%	85.7%	97.8%
5	SCG	tan-sig	93.5%	71.4%	71.4%	86.7%
4	SCG	tan-sig	100%	100%	71.4%	95.6%
3	SCG	tan-sig	90.3%	100%	42.9%	84.4%

Table 3: Table of the BPNN Accuracy Rates for the Experimental Vibration Data

Number of hidden neurons	Training function	Transfer function used in the HL	Experimental vibration data			
			Training accuracy	Validation accuracy	Testing accuracy	Total accuracy
10	LM	log-sig	98.2%	94.4%	97.2%	97.5%
9	LM	log-sig	98.8%	94.4%	100%	98.3%
8	LM	log-sig	99.4%	100%	86.1%	97.5%
7	LM	log-sig	97.6%	80.6%	94.4%	94.6%
6	LM	log-sig	99.4%	91.7%	97.2%	97.9%
5	LM	log-sig	100%	94.4%	94.4%	98.3%
4	LM	log-sig	100%	94.4%	94.4%	98.3%
3	LM	log-sig	93.5%	86.1%	91.7%	92.1%
10	LM	tan-sig	97.6%	97.2%	91.7%	96.7%
9	LM	tan-sig	100%	97.2%	97.2%	99.2%
8	LM	tan-sig	100%	100%	88.9%	98.3%
7	LM	tan-sig	100%	83.3%	91.7%	96.3%
6	LM	tan-sig	97.6%	97.2%	94.4%	97.1%
5	LM	tan-sig	100%	91.7%	97.2%	98.3%
4	LM	tan-sig	97.6%	94.4%	91.7%	96.3%
3	LM	tan-sig	98.8%	88.9%	97.2%	97.1%
10	SCG	log-sig	94%	91.7%	94.4%	93.8%
9	SCG	log-sig	98.8%	94.4%	100%	98.3%
8	SCG	log-sig	98.8%	100%	91.7%	97.9%
7	SCG	log-sig	97.6%	91.7%	94.4%	96.3%
6	SCG	log-sig	96.4%	94.4%	97.2%	96.3%
5	SCG	log-sig	84.5%	72.2%	75%	81.3%
4	SCG	log-sig	96.4%	91.7%	91.7%	95%
3	SCG	log-sig	69%	72.2%	80.16%	71.3%
10	SCG	tan-sig	97.6%	94.4%	94.4%	96.7%
9	SCG	tan-sig	97.6%	97.2%	97.2%	97.5%
8	SCG	tan-sig	97%	100%	88.9%	96.3%
7	SCG	tan-sig	97%	100%	97.2%	97.5%
6	SCG	tan-sig	97%	94.4%	94.4%	96.3%
5	SCG	tan-sig	82.7%	75%	77.8%	80.8%
4	SCG	tan-sig	96.4%	94.4%	91.7%	95.4%
3	SCG	tan-sig	96.4%	91.7%	97.2%	95.8%

3.1. Confusion Matrix Plots

A confusion matrix is a significant performance measurement for classification applications. In this study, the confusion matrix of the optimum classifier for training, validation, testing, and all samples of the artificial and experimental vibration data is calculated separately to ensure the value of the accuracy rates. The confusion matrix plots of the optimum classifier for both data are illustrated in Figure 4 and Figure 5, respectively. According to these figures, the confusion matrices of the network indicate the approval of the accuracy of the optimum network for the aging classification problem. For the artificial data, the predicted values in the four confusion matrices are the same as their target values. However, for the experimental data, there are two misclassified samples in the validation and testing confusion matrices. Therefore, the confusion matrix for all samples of the experimental data determines four incorrect estimated samples which are resulted in a 98.3% accuracy rate. For the optimum classifier, three different quality measures have also been evaluated according to the confusion matrices acquired from the classification of the samples of both artificial and experimental vibration data. The overall misclassification rate, correctness and completeness are the evaluated quality measures considered for this study. The misclassification rate is defined as the ratio of the number of wrongly classified samples for different aging levels to the total number of samples classified by the prediction model (Vinodhini & Chandrasekaran, 2016). According to the misclassification rates

computed in Table 4, the overall misclassification rates for the artificial and experimental vibration data are 0% and 1.7%, respectively. The correctness is defined as the number of samples correctly classified as class I to the total number of samples classified as class I (Vinodhini & Chandrasekaran, 2016). Table 5 represents the correctness of each class assigned in the optimum classifier for both artificial and experimental data. According to Table 5, the optimum classifier operates perfectly well in terms of the classification of the artificial vibration data. However, the values of the correctness of the proposed classifier for class 4, class 5 and class 6 in the experimental vibration data are calculated as 93.3%, 97.1% and 96.7%, respectively. It means that the less correctness of the proposed classifier for the classification of the experimental data is relevant to the four incorrect predicted samples located in class 4, class 5 and class 6. Completeness is also defined as the ratio of the number of samples of class I classified as class I to the total number of actual samples in class I (Vinodhini & Chandrasekaran, 2016). For both artificial and experimental vibration data, the completeness of the optimum classifier for each class is represented in Table 6. By analysis of the completeness of the optimum classifier for the artificial vibration data, it is determined that the proposed classifier predicts all the samples in all classes correctly while for the classification of the experimental data, the computed completeness of class 3, class 5 and class 6 has been reduced to 96.2%, 94.3% and 96.7%, respectively. Indeed, the optimum classifier predicts the samples of class 5 with the minimum value of completeness which means that the ratio of the

correctly predicted samples in class 5 to the total number of actual samples in class 5 is the smallest value compared to the same ratio in other classes.

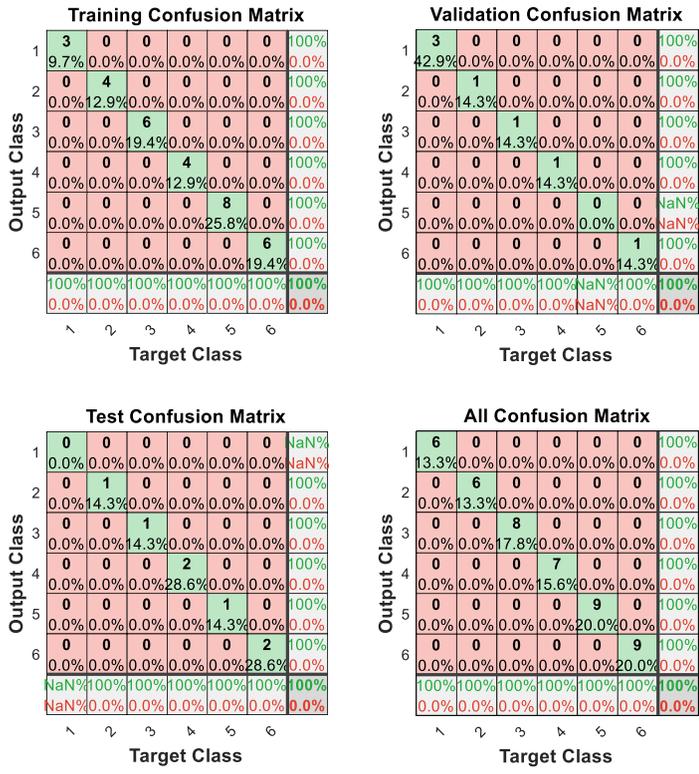


Figure 4: Confusion Matrices of the Optimum Classifier for the Artificial Data

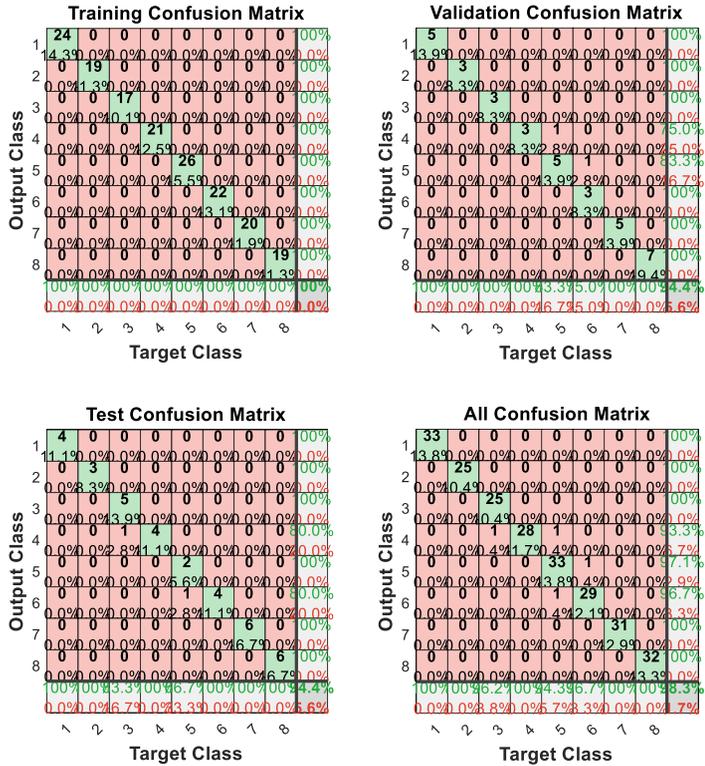


Figure 5: Confusion Matrices of the Optimum Classifier for the Experimental Data

Table 4: Overall Misclassification Rate of the Optimum Classifier

Type of the Data Sets	Overall misclassification rate
Artificial Vibration Data	0%
Experimental Vibration Data	1.7%

Table 5: Correctness of the Optimum Classifier for Aging Classes

Type of the Data Sets	Correctness							
	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8
Artificial Vibration Data	100%	100%	100%	100%	100%	100%	-	-
Experimental Vibration Data	100%	100%	100%	93.3%	97.1%	96.7%	100%	100%

Table 6: Completeness of the Optimum Classifier for Aging Classes

Type of the Data Sets	Completeness							
	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8
Artificial Vibration Data	100%	100%	100%	100%	100%	100%	-	-
Experimental Vibration Data	100%	100%	96.2%	100%	94.3%	96.7%	100%	100%

3.2. ROC Plots

The receiver operating characteristic (ROC) curves are the other performance assessment tools for classification purposes. The ROC curves calculate the true positive rate (TPR) and the false positive rate (FPR) with different values of thresholds for the model. A perfect classifier is expected to have a TPR = 1 and an FPR = 0. Figure 6 and Figure 7 represent the ROC curves of the optimum classifier for the artificial and the experimental vibration data, respectively. For the artificial data, since the ROC curves hug the left and top edges, the proposed classifier is enumerated as an efficient model. Furthermore, it is necessary to mention that since there is not any sample as the class 5 in the validation data set and also there is not any sample as the class 1 in the testing data set, the ROC curves for class 5 and class 1 in

the validation and testing processes hug the right and bottom edges of the plots. For the experimental data, the training ROC curves for all classes show the perfect values of TPR and FPR. However, for the validation and testing processes, the ROC curves for class 5 and class 6 are pulled down slightly. Nevertheless, for the classification of all samples of the experimental data, acceptable values of TPR and FPR are acquired which proves that the classifier is still operating very well.

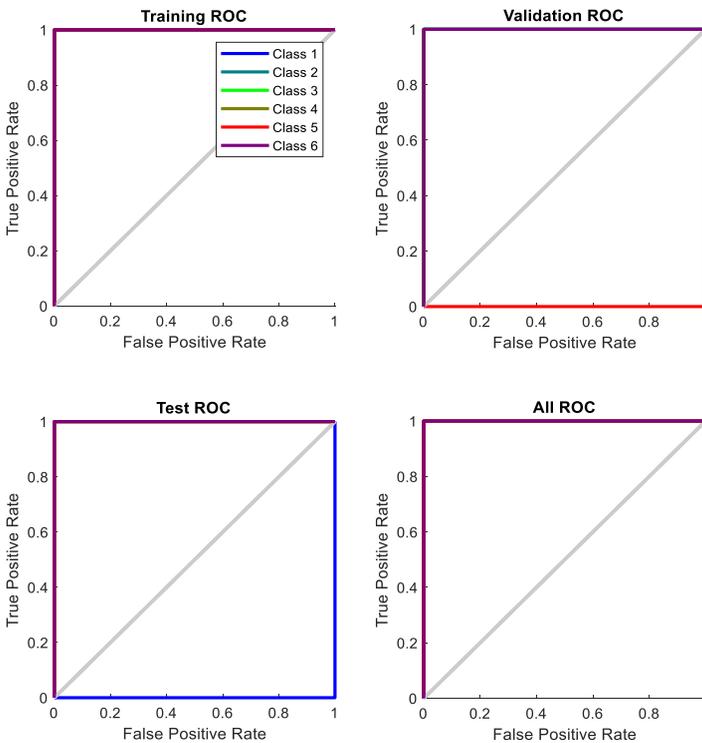


Figure 6: ROC Plots of the Optimum Classifier for the Artificial Data

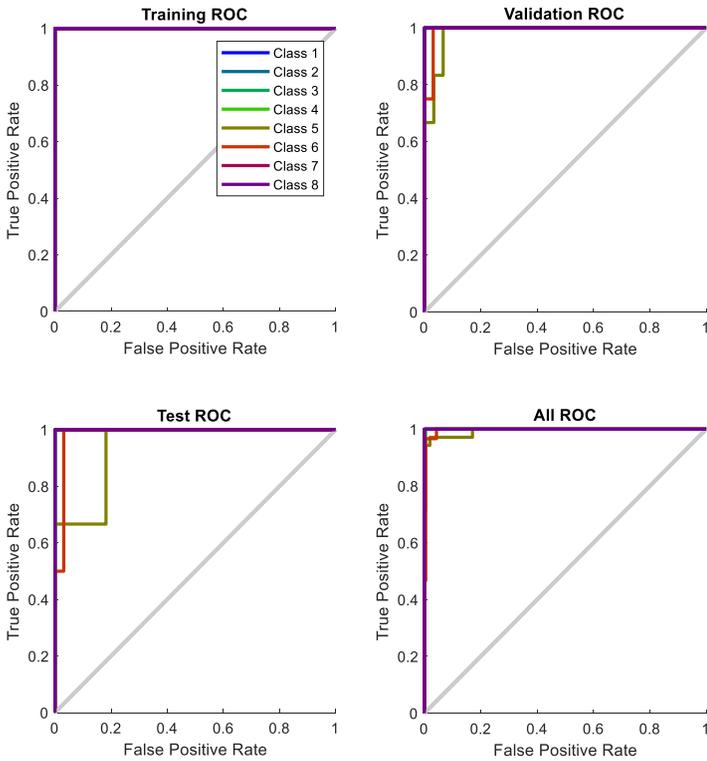


Figure 7: ROC Plots of the Optimum Classifier for the Experimental Data

3.3 Error Histogram Plots

The error histograms are also suitable indicators for performance in classification applications. Indeed, the error values of the training, validation and testing processes are determined clearly by using the error histogram plots. The error histograms of the optimum classifier for both artificial and experimental data are also given in Figure 8 and Figure 9, respectively. According to the error histograms, the error value that corresponds to most samples of the data is -0.00619 and -

0.04778 for the artificial and experimental data, respectively. Besides, the total calculated error values of the optimum network for both data are very small values indicating the high performance of the networks.

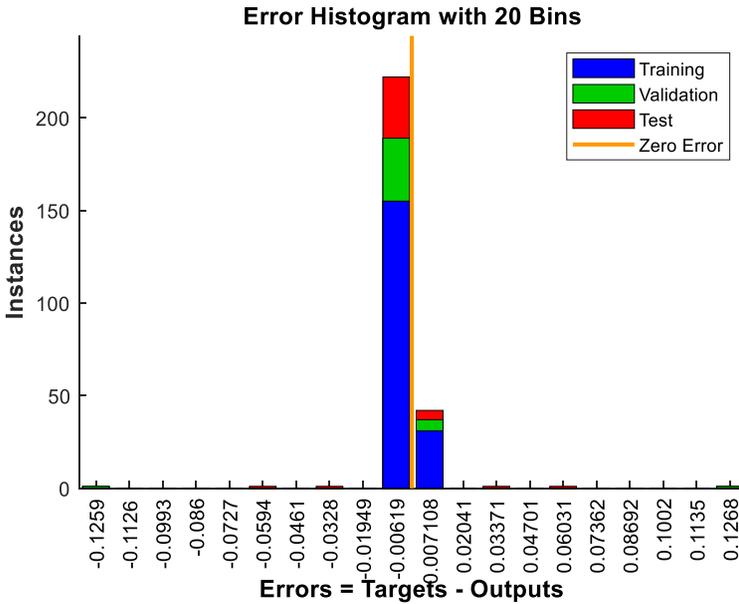


Figure 8: Error Histograms of the Optimum Classifier for the Artificial Data

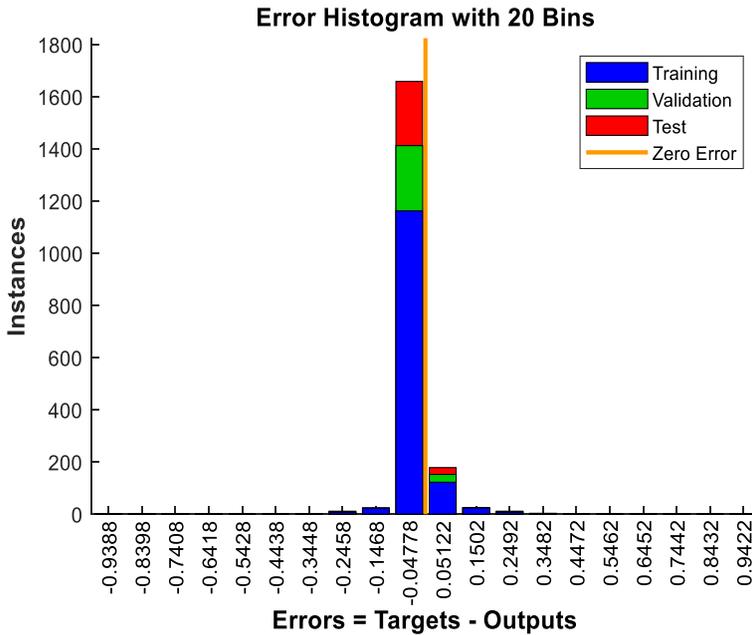


Figure 9: Error Histograms of the Optimum Classifier for the Experimental Data

CONCLUSION

In this research work, we present a FFNN model trained with BP as an aging classifier of IM. Since the vibration signals are the most informative signals acquired from an IM, the developed BPNN model is assessed by using artificial and experimental vibration signals. In this study, by constructing the artificial vibration signals, the behavior of the designed BPNN is analyzed precisely and also it is verified on the experimental vibration signals to ensure the accurate performance of the proposed model.

In order to increase the performance of the proposed model, nine time-domain statistical features are specified as the input parameters of the

BPNN model. Besides, the performance of the proposed model is examined using different number of neurons in the hidden layer, different optimization algorithms and different transfer functions. By comparing the accuracy rates of these models, it is assigned that the optimal topology is trained with the Levenberg-Marquardt algorithm consisting of 4 neurons and a log-sigmoid transfer function in the hidden layer. For the artificial and experimental vibration signals, the total accuracy rates of the optimum classifier are calculated as 100% and 98.3%, respectively.

Eventually, the quality of the performance of the optimum classifier is investigated appropriately by using the confusion matrix, receiver operating characteristic (ROC) and the error histogram plots. Besides, using the information acquired from the confusion matrices, three different quality measures are also discussed to approve the success of the optimum classifier in detail. Therefore, the fact that the proposed BPNN model is suggested as a robust and reliable model for monitoring and detecting the aging levels of an IM can be asserted certainly.

REFERENCES

- Benbouzid, M. E. H., Vieira, M., & Theys, C. (1999). Induction motors' faults detection and localization using stator current advanced signal processing techniques. *IEEE Transactions on power electronics*, *14*(1), 14-22.
- Faiz, J., Ebrahimi, B., & Sharifian, M. (2006). Different faults and their diagnosis techniques in three-phase squirrel-cage induction motors—A review. *Electromagnetics*, *26*(7), 543-569.
- Farajzadeh-Zanjani, M., Razavi-Far, R., Saif, M., & Rueda, L. (2016). *Efficient feature extraction of vibration signals for diagnosing bearing defects in induction motors*. Paper presented at the 2016 International Joint Conference on Neural Networks (IJCNN).
- Gritli, Y., Bellini, A., Rossi, C., Casadei, D., Filippetti, F., & Capolino, G. (2017). *Condition monitoring of mechanical faults in induction machines from electrical signatures: Review of different techniques*. Paper presented at the 2017 IEEE 11th international symposium on diagnostics for electrical machines, power electronics and drives (SDEMPED).
- Gu, F., Wang, T., Alwodai, A., Tian, X., Shao, Y., & Ball, A. (2015). A new method of accurate broken rotor bar diagnosis based on modulation signal bispectrum analysis of motor current signals. *Mechanical Systems and Signal Processing*, *50*, 400-413.
- Hashemnia, N., & Asaei, B. (2008). *Comparative study of using different electric motors in the electric vehicles*. Paper presented at the 2008 18th International Conference on Electrical Machines.
- Malik, N. (2005). Artificial neural networks and their applications. *arXiv preprint cs/0505019*.
- Rad, M. K., Torabizadeh, M., & Noshadi, A. (2011). *Artificial Neural Network-based fault diagnostics of an electric motor using vibration monitoring*. Paper presented at the Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE).

- Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Retrieved from
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533-536.
- Saravanan, N., Siddabattuni, V. K., & Ramachandran, K. (2010). Fault diagnosis of spur bevel gear box using artificial neural network (ANN), and proximal support vector machine (PSVM). *Applied soft computing*, 10(1), 344-360.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Seera, M., Lim, C. P., Nahavandi, S., & Loo, C. K. (2014). Condition monitoring of induction motors: A review and an application of an ensemble of hybrid intelligent models. *Expert Systems with Applications*, 41(10), 4891-4903.
- Vinodhini, G., & Chandrasekaran, R. (2016). A comparative performance evaluation of neural network based approach for sentiment classification of online reviews. *Journal of King Saud University-Computer and Information Sciences*, 28(1), 2-12.
- Zarei, J. (2012). Induction motors bearing fault detection using pattern recognition techniques. *Expert Systems with Applications*, 39(1), 68-73.
- Zarei, J., Tajeddini, M. A., & Karimi, H. R. (2014). Vibration analysis for bearing fault detection and classification using an intelligent filter. *Mechatronics*, 24(2), 151-157.

CHAPTER 3

IDENTIFICATION OF MYOCARDIAL INFARCTION DISEASE USING EFFECTIVE MACHINE LEARNING METHODS

Assist. Prof. Dr. Şule YÜCELBAŞ ¹

¹ Hakkari University, Electrical and Electronics Engineering Department, Hakkari,
TURKEY suleyucelbas@hakkari.edu.tr ,ORCID ID 0000-0002-6758-8502

INTRODUCTION

The heart, which is located in the thorax a little to the left of the midline of the chest and is of vital importance, is a muscular organ. The weight of this organ, which pumps almost 8000 liters of blood into the circulation by contracting an average of 100 thousand times a day, is 340 grams in men, while it is about 300-320 grams in women. The filling of an organ with blood leakage is called an infarction (White & Chew, 2008; Thygesen et al., 2007; Boersma et al., 2003; Reed, Rossi, & Cannon, 2017; Frangogiannis, Smith, & Entman, 2002). This event begins when the feeding artery of an area is blocked by a clot. In the next step, this clot causes a slowdown in blood flow and then a lack of blood supply in that area, resulting in infarction. The final stage is tissue death. There are many types of infarct disease, and one of them is myocardial infarction. Myocardial infarction is also known as coronary occlusion. This condition occurs as a result of the blockage of one of the vessels called the coronary (White & Chew, 2008; Thygesen et al., 2007; Boersma et al., 2003; Reed, Rossi, & Cannon, 2017; Frangogiannis, Smith, & Entman, 2002). Myocardial infarction is a very common disorder and its causes include embolism, occlusion of the heart artery, and lack of blood supply to the heart muscle. This disease, which is more common in women than men, has many symptoms. Severe pain in the chest area on the stomach, cold sweats, bruises, shock, circulatory disorders are among the symptoms of heart infarction. This disease can sometimes come on suddenly and be fatal (White & Chew, 2008; Thygesen et al., 2007; Boersma et al., 2003; Reed, Rossi, & Cannon, 2017; Frangogiannis, Smith, &

Entman, 2002). However, most of the patients can be treated with early diagnosis. Exercise tests performed at regular intervals or scintigraphy examination applied to patients who cannot run can measure the susceptibility of the person to this disease (Margonato et al., 1992; Hung et al., 1984; Staniloff et al., 1986). In addition, patients can have a preliminary diagnosis about themselves with medical tests based on machine learning methods, which are applied online today. This makes a great contribution to the transfer of information to every corner of the world (White & Chew, 2008; Thygesen et al., 2007; Boersma et al., 2003; Reed, Rossi, & Cannon, 2017; Frangogiannis, Smith, & Entman, 2002; Margonato et al., 1992; Hung et al., 1984; Staniloff et al., 1986).

The death that occurs unexpectedly within an hour of the onset of symptoms is defined as sudden death. Half of the sudden deaths occur due to cardiovascular diseases. Among these rates, especially the share of myocardial infarction is quite high. Unfortunately, half of the patients who suddenly experience myocardial infarction symptoms die before reaching the hospital. Except for sudden death, coronary heart diseases take first place among the causes of death in all developed countries. Actually, we can minimize the risk of developing cardiovascular diseases. A healthy diet, regular exercise, and check-ups can reduce the risks that patients may face in the future. In addition to being one-on-one with a doctor or a specialist, these controls are made possible online thanks to the developing medical machine learning systems (Alonzo, Simon, & Feinleib, 1975; Thompson et al., 1995; Vismara, Amsterdam, & Mason, 1975).

Thanks to artificial intelligence-based medical expert systems, people can make their own preliminary checks. This situation both increases the opportunity of people to reach medical support and reduces the intensity that may occur in hospitals (Fan et al., 2020; He et al., 2019; Benjamens, Dhunnoo, & Meskó, 2020; Beckers, Kwade, & Zanca, 2021). For this reason, all kinds of equipment and software that can detect the presence of deadly diseases such as myocardial infarction are important. In this study, an idea that can contribute to artificial intelligence-based medical studies in the literature has been put forward, and therefore, effective features have been tried to be determined by machine learning methods in order to detect myocardial infarction with high accuracy. First of all, the data of patients with non-informative features were excluded from the data set to be used. In addition, features with very little information content were also eliminated from the data set. On the data set consisting of the remaining data, firstly, classification was carried out with various classifiers. Then, based on the most successful classifier, various feature selection methods were tried on the dataset. Thus, the features with the highest accuracy rate were determined. The process flow of the study is shown in Figure 1.

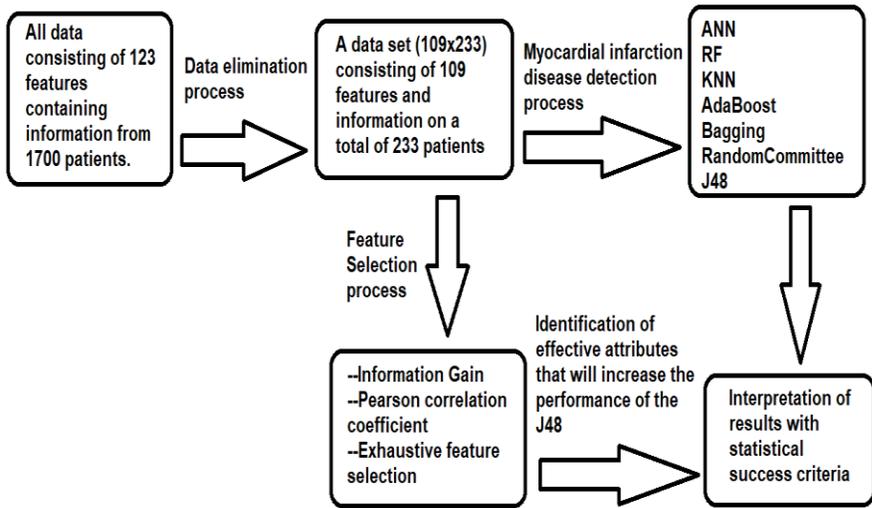


Figure 1: The Process Flow of the Study

METODOLOGIES

The dataset used in this study consists of information collected at the Krasnoyarsk Interregional Clinical Hospital (Russia) from 1992 to 1995 and recently made available to the public. The dataset used in the study by extracting missing data includes information from 1700 patients characterized by 111 features that define clinical phenotypes and 12 features representing possible complications of myocardial infarction disease (123 features in total) (Golovenkin et al., 2020). Details of the descriptive statistics and variable names of the data are provided in the dataset description available online (Golovenkin et al., 2020). The data of patients with non-informative features were excluded from the data set to be used. In addition, features with very little information content were also eliminated from the data set. As a

result, a total of 233 patient data with 109 features remained in the data set (109x233). Classifications and feature selection processes were performed on the final data set. Conditions with or without myocardial infarction in patients were labeled with '1' (cardiogenic shock, pulmonary edema, myocardial rupture, progress of congestive heart failure, thromboembolism, asystole, and ventricular fibrillation) and '0' (unknown) classes, respectively. Finally, a 10-fold cross-validation method was applied to the data before it was given to the classifiers.

In this study, the training and test data created from the eliminated dataset were presented to the various classifiers such as artificial neural networks (ANN), random forest (RF), k-nearest neighbors (KNN), AdaBoost (ABST), Bagging (BGG), random committee (RC), and J48 for the myocardial infarction detection process with a high accuracy rate. After reaching the highest classification accuracy, in the feature selection stage of this study, various selection algorithms such as information gain (IG), Pearson correlation coefficient (PCC), and exhaustive feature selection (EFS) were used to determine the effective attributes. The reduction in entropy from the transformation of a dataset is calculated by information gain. Correlation, on the other hand, gives a measure of the linear relationship of two or more variables. Finally, exhaustive feature selection method tries every possible combination of the variables and returns the most accurate subset (Chandrashekar & Sahin, 2017).

Confusion matrixes were used in the evaluation of the results obtained for this study. Statistical calculations were made according to these matrixes and the results were examined. The validity of the experimental results was statistically analyzed using Cohen's Kappa coefficient (K), true-positive rate (TP rate), false-positive rate (FP rate), precision (P), F-measure (F), Matthews correlation coefficient (MCC), receiver operating characteristic area (ROC), precision-recall area (PRC), cost curve, and accuracy (ACC) rates criteria. Detailed formulations of the parameters used were given in (Yücelbaş et al., 2017; Drummond & Holte, 2006).

EXPERIMENTAL RESULTS

In this study, the effective features have been tried to define with machine learning methods for the detection of myocardial infarction with high accuracy. First of all, the data of subjects with non-informative features were excluded from the dataset to be used. Then, attributes with very little information content were also eliminated from the dataset. On the dataset consisting of the remaining data, firstly, classification was carried out with the previously mentioned classifiers. Using the eliminated dataset, a total of 109 features were presented as input data to the mentioned classifiers. The performance results obtained according to these classifications are shown in Table 1.

Table 1: Performance Results Obtained by Classification of Dataset with the Previously Mentioned Classifiers (TNI: Total Number of Instances, CCI: Correctly Classified Instances)

Classifiers	TNI	CCI	K	TP rate	FP rate	P	F	MCC	ROC	PRC	ACC %
ANN		202	0.41	0.86	0.50	0.85	0.85	0.42	0.71	0.84	86.69
RF		201	0.17	0.86	0.75	0.88	0.81	0.30	0.85	0.92	86.26
KNN		199	0.23	0.85	0.68	0.82	0.82	0.27	0.59	0.77	85.40
ABST	233	208	0.47	0.89	0.51	0.88	0.87	0.51	0.74	0.86	89.27
BGG		209	0.47	0.89	0.54	0.90	0.87	0.54	0.77	0.87	89.69
RC		202	0.24	0.86	0.70	0.86	0.82	0.34	0.77	0.85	86.69
J48		209	0.51	0.89	0.47	0.89	0.88	0.54	0.73	0.85	89.69

As can be seen from Table 1, J48 was the classifier that reaches the highest performance values by correctly classifying 209 of 233 data among the seven classifiers. Although J48 achieves the same ACC% ratio as BGG, when examined carefully, it was understood that it has much better ratios than this classifier in many parameters such as the K criterion. In addition, J48 and BGG classifiers were followed by ABST with an ACC rate of 89.27%. Among these classifiers, the worst results were obtained with KNN. This classifier was able to correctly classify only 199 of 233 data and achieved an ACC rate of 85.40%.

According to the information in Table 1, the classification errors of the J48 classifier were shown in Figure 2. The blue ones represent the ‘0’ label and the red ones represent the ‘1’ label. In addition, crosses indicate correctly classified samples, while squares indicate incorrect ones.

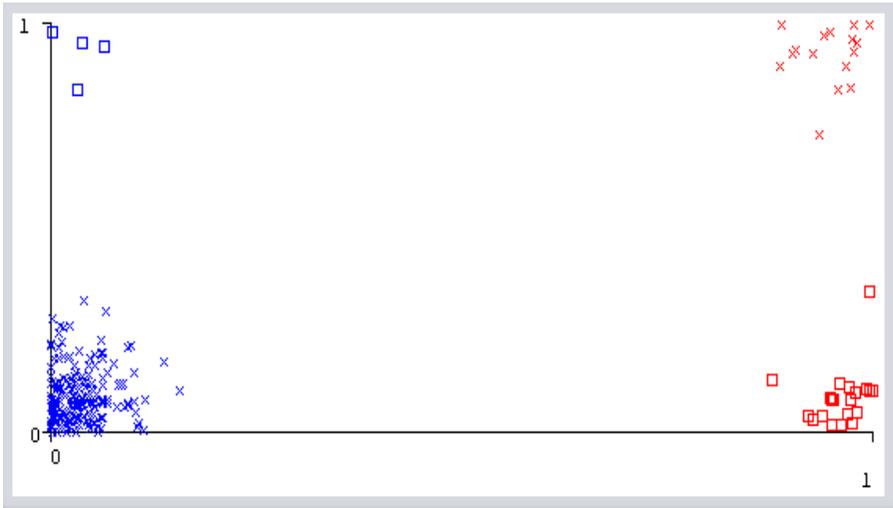


Figure 2: Classification Errors of J48 Classifier According to Information of the Table 1 (Blue: “0” and Red: “1”)

According to the features used in the dataset, the decision tree structure that J48 reaches as a result of the classification was shown in Figure 3. The explanations of the features in the decision tree structure, which were reached based on the statistical results of the J48 classifier in Table 1, and seen in Figure 3 are as follows (Golovenkin et al., 2020):

RAZRIV: Myocardial rupture

*S_AD_KBRIG: Systolic blood pressure according to
Emergency Cardiology Team (mmHg)*

*INF_ANAM: Quantity of myocardial infarctions in the
anamnesis*

*O_L_POST: Pulmonary edema at the time of admission to
intensive care unit*

SEX: Gender

post_im: Presence of a posterior myocardial infarction (left ventricular)

GEPAR_S_n: Use of a anticoagulants (heparin) in the intensive care unit

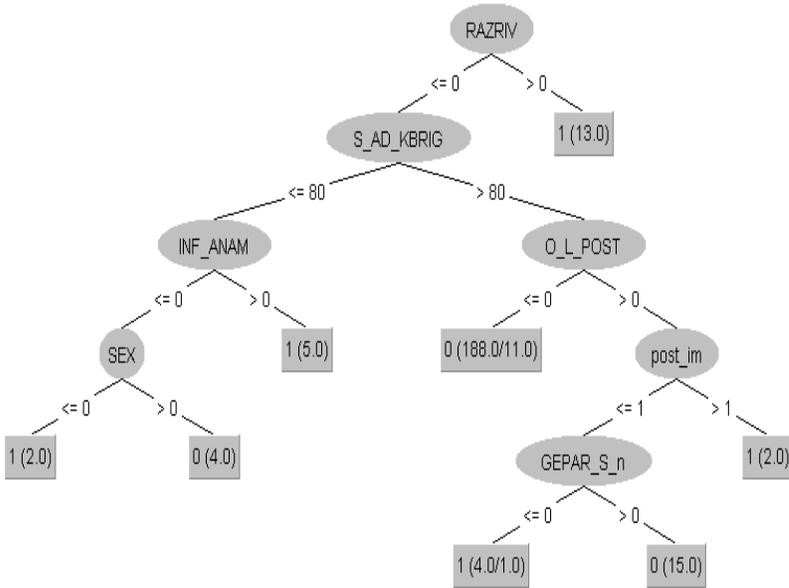


Figure 3: According to the Features Used in the Dataset, the Decision Tree Structure That J48 Reaches as a Result of the Classification (Based on the Classification Result in Table 1)

At this stage, the study was continued with the J48 classifier which had the best classification results than the others, and the performance outputs of this classifier were tried to be improved. For this reason, three different feature selection processes mentioned in Section 2 were carried out and the classification results obtained with J48 were shown in Table 2.

Table 2: The Classification Results Obtained with the J48 Classifier According to Feature Selection Methods (TNI: Total Number of Instances, CCI: Correctly Classified Instances)

Feature selection method	TNI	CCI	K	TP rate	FP rate	P	F	MCC	ROC	PRC	ACC %
IG		213	0.58	0.91	0.44	0.91	0.90	0.63	0.67	0.84	91.41
PCC	233	209	0.51	0.89	0.47	0.89	0.88	0.54	0.73	0.85	89.69
EFS		212	0.58	0.91	0.42	0.90	0.90	0.61	0.72	0.84	90.98

As a result of the classification of 10 effective attributes with J48, which were determined after the IG process, the outputs in Table 2 were reached. When Table 2 is examined, it is seen that the ACC rate of 91.41% was obtained with the features determined after the IG feature selection process. The gain values of these attributes were given in Table 3. It is clearly seen that the IG method is better when compared to other methods according to Table 2. However, using the PCC method, the number of features was reduced to 88 and the same values were achieved as in Table 1 (obtained using all features). This showed that the PCC method was also successful. As can be seen in Table 2, 5 of the 109 features with the higher effect were determined thanks to the EFS method, and an ACC rate of 90.98% was obtained. These features are age, systolic blood pressure, pulmonary edema at the time of admission to intensive care unit, white blood cell count (billions per liter), and myocardial rupture.

Table 3: The Gain Values of the Attributes Selected with the IG Algorithm

ATTRIBUTE NAME AND DESCRIPTION (Golovenkin et al., 2020)	GAIN VALUE
RAZRIV: Myocardial rupture	0.1647
S_AD_KBRIG: Systolic blood pressure according to Emergency Cardiology Team (mmHg)	0.0439
S_AD_ORIT: Systolic blood pressure according to intensive care unit (mmHg)	0.0394
ant_im: Presence of an anterior myocardial infarction (left ventricular)	0.0372
NA_R_1_n: Use of opioid drugs in the intensive care unit in the first hours of the hospital period	0.0365
D_AD_ORIT: Diastolic blood pressure according to intensive care unit (mmHg)	0.0354
ASP_S_n: Use of acetylsalicylic acid in the intensive care unit	0.0283
K_SH_POST: Cardiogenic shock at the time of admission to intensive care unit	0.0234
n_p_ecg_p_12: Complete right bundle branch block on electrocardiography at the time of admission to hospital	0.0214
P_IM_STEN: Post-infarction angina	0.0184

According to the effective features mentioned in Table 3, the decision tree structure that J48 reaches as a result of the classification was shown in Figure 4. The descriptions of the features in the decision tree structure, which is reached based on the statistical results of the J48 classifier in Table 2 and seen in Figure 4, were already given in Table 3.

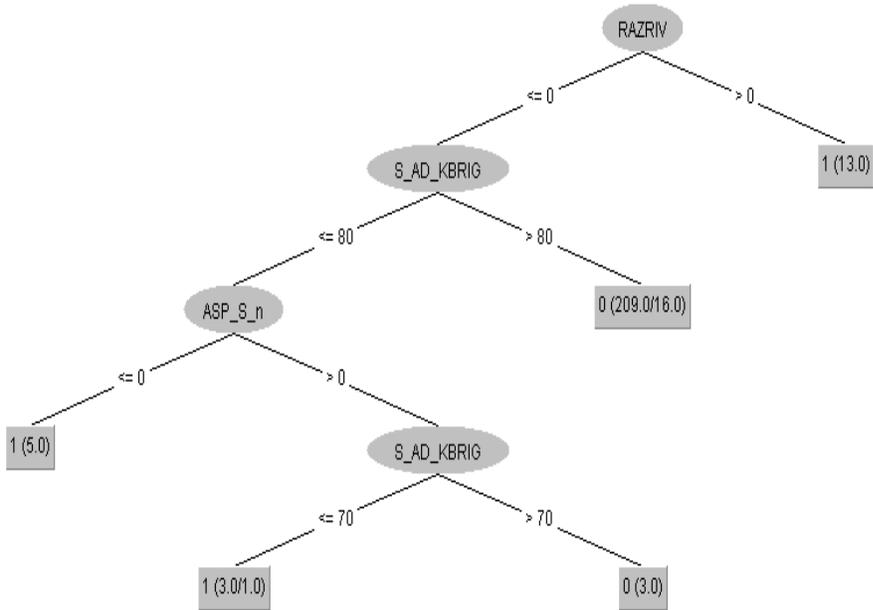


Figure 4: According to the Effective Features Mentioned in Table 3, the Decision Tree Structure that J48 Reaches as a Result of the Classification (Based on the Classification Result in Table 2)

DISCUSSION AND CONCLUSION

Myocardial infarction (colloquially known as a heart attack) is a very serious health problem that takes years to form however appears suddenly. Heart attack, which is the leading cause of death in the world, actually has warning symptoms. While patients who have knowledge about the subject can take into account the symptoms and be treated for this ailment, individuals who do not have knowledge may die. We can protect our heart health by exercising regularly, avoiding harmful habits such as smoking, preferring more vegetables and fruits, and keeping body weight within normal limits, in short, by

increasing our quality of life. Thus, we can minimize the risk of developing cardiovascular diseases. Moreover, regular check-ups can reduce the risks that patients may face in the future. In addition to being one-on-one with a doctor or a specialist, these controls are made possible online thanks to the developing medical machine learning systems. Thanks to artificial intelligence-based medical expert systems, people can make their own preliminary checks. This situation both increases the opportunity of people to reach medical support and reduces the intensity that may occur in hospitals. Therefore, all kinds of equipment and software that can facilitate the diagnosis of deadly diseases such as myocardial infarction are of great importance.

In this study, the effective attributes have been tried to determine with machine learning methods for the detection of myocardial infarction with high accuracy. The eliminated dataset consisting of the information of the remaining 233 patients was classified with ANN, KNN, RF, ABST, BGG, RC, and J48 classifiers. As a result of these processes, J48 among the seven classifiers correctly classified 209 of 233 data and reached 89.69% ACC rate. This classifier was followed by BGG and ABST in terms of performance, respectively. Among these classifiers, the worst results were obtained with KNN. This classifier was able to correctly classify only 199 of 233 data and achieved an ACC rate of 85.40%. Then, J48, the classifier with the most successful results, was taken as a basis and IG, PCC, and EF feature selection methods were tried on the dataset. Thus, the effective attributes that can increase the success of J48 were determined according to each feature selection method. The results showed that a

2% increase in the ACC rate was achieved with 10 effective features determined after the IG feature selection process. Thanks to the EFS method, 5 of the 109 features with the higher effect were determined and an ACC ratio of 90.98% was obtained. In addition, 89.96% ACC rate was reached when 88 features selected using the PCC method were given to J48 as input data. In other words, the success of J48 has not increased with the PCC method, but the number of features that tire the system has been reduced. This showed that all three feature selection methods used contributed to the performance.

The chief factor among the limitations of this study was, of course, the number of data. Because only 36 of 233 patients could be labeled as myocardial infarction and the remaining 197 patients were defined as unknown. This situation caused an imbalance in the data distribution and reduced the success. In addition, the fact that some of the features in the data set were binary information (that is, they consist of 0 or 1) reduced the quality of the numerical content used in the classification. This situation also indirectly affected the success. In order to obtain more successful results in future studies, the quality and number of data should be increased. In addition, researchers could contribute to artificial intelligence-based medical systems by presenting various studies with other feature selection methods and classifiers or clustering methods.

REFERENCES

- White, H. D., & Chew, D. P. (2008). Acute myocardial infarction. *The Lancet*, 372(9638), 570-584.
- Thygesen, K., Alpert, J. S., White, H. D., & Joint ESC/ACCF/AHA/WHF Task Force for the Redefinition of Myocardial Infarction. (2007). Universal definition of myocardial infarction. *Journal of the American College of Cardiology*, 50(22), 2173-2195.
- Boersma, E., Mercado, N., Poldermans, D., Gardien, M., Vos, J., & Simoons, M. L. (2003). Acute myocardial infarction. *The Lancet*, 361(9360), 847-858.
- Reed, G. W., Rossi, J. E., & Cannon, C. P. (2017). Acute myocardial infarction. *The Lancet*, 389(10065), 197-210.
- Frangogiannis, N. G., Smith, C. W., & Entman, M. L. (2002). The inflammatory response in myocardial infarction. *Cardiovascular research*, 53(1), 31-47.
- Margonato, A., Ballarotto, C., Bonetti, F., Cappelletti, A., Sciammarella, M., Cianflone, D., & Chierchia, S. L. (1992). Assessment of residual tissue viability by exercise testing in recent myocardial infarction: comparison of the electrocardiogram and myocardial perfusion scintigraphy. *Journal of the American College of Cardiology*, 19(5), 948-952.
- Hung, J., Goris, M. L., Nash, E., Kraemer, H. C., & DeBusk, R. F. (1984). Comparative value of maximal treadmill testing, exercise thallium myocardial perfusion scintigraphy and exercise radionuclide ventriculography for distinguishing high-and low-risk patients soon after acute myocardial infarction. *The American journal of cardiology*, 53(9), 1221-1227.
- Staniloff, H. M., Forrester, J. S., Berman, D. S., & Swan, H. J. C. (1986). Prediction of death, myocardial infarction, and worsening chest pain using thallium scintigraphy and exercise electrocardiography. *Journal of nuclear medicine*, 27(12), 1842-1848.
- Alonzo, A. A., Simon, A. B., & Feinleib, M. A. N. N. I. N. G. (1975). Prodromata of myocardial infarction and sudden death. *Circulation*, 52(6), 1056-1062.

- Thompson, S. G., Kienast, J., Pyke, S. D., Haverkate, F., & van de Loo, J. C. (1995). Hemostatic factors and the risk of myocardial infarction or sudden death in patients with angina pectoris. *New England Journal of Medicine*, 332(10), 635-641.
- Vismara, L. A., Amsterdam, E. A., & Mason, D. T. (1975). Relation of ventricular arrhythmias in the late hospital phase of acute myocardial infarction to sudden death after hospital discharge. *The American journal of medicine*, 59(1), 6-12.
- Fan, W., Liu, J., Zhu, S., & Pardalos, P. M. (2020). Investigating the impacting factors for the healthcare professionals to adopt artificial intelligence-based medical diagnosis support system (AIMDSS). *Annals of Operations Research*, 294(1), 567-592.
- He, J., Baxter, S. L., Xu, J., Xu, J., Zhou, X., & Zhang, K. (2019). The practical implementation of artificial intelligence technologies in medicine. *Nature medicine*, 25(1), 30-36.
- Benjamens, S., Dhunoo, P., & Meskó, B. (2020). The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database. *NPJ digital medicine*, 3(1), 1-8.
- Beckers, R., Kwade, Z., & Zanca, F. (2021). The EU medical device regulation: Implications for artificial intelligence-based medical device software in medical physics. *Physica Medica*, 83, 1-8.
- Golovenkin SE, Gorban AN, Mirkes EM, et al. Myocardial infarction complications Database. 2020. <https://doi.org/10.25392/leicester.data.12045261.v3>.
- Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16-28.
- Yücelbaş, Ş., Yücelbaş, C., Tezel, G., Özşen, S., Küçüktürk, S., & Yosunkaya, Ş. (2017). Pre-determination of OSA degree using morphological features of the ECG signal. *Expert Systems with Applications*, 81, 79-87.
- Drummond, C., & Holte, R. C. (2006). Cost curves: An improved method for visualizing classifier performance. *Machine learning*, 65(1), 95-130.

CHAPTER 4

EFFECT OF VARIOUS BINDERS ON HOLD / PHYSICAL PROPERTIES OF SPRUCE (*Picea orientalis* (L.) Link) WOOD

Dr. Lecturer Hatice ULUSOY¹
Prof. Dr. Hüseyin PEKER²

¹ Muğla Sıtkı Koçman University, Köyceğiz Vocational School, Forest Department, Muğla, Turkey. haticulusoy@mu.edu.tr ,ORCID ID 0000-0003-0960-3388

² Artvin Çoruh University, Forest Faculty, Forest Industrial Engineering Department, Artvin, Turkey. peker100@artvin.edu.tr ,ORCID ID 0000-0002-7771-6993

INTRODUCTION

Owing to its natural quality, organic composition, and superior properties, wooden material has been favored in many areas (building, yacht, cruise, home furniture, etc.) from the past to the present. Aside from being an environmentally friendly, renewable, and sustainable natural resource, wood is also superior in terms of health, aesthetics, durability, and fire resistance. However, wood material is adversely affected by external weather conditions and seasonal weather changes, and its value and service life are decreasing (Üstün, 2019).

Scotch pine (*Pinus sylvestris L.*) wood was impregnated with a solution made up of boron compounds and mixture of natural preservatives. It was examined that the amount of retention in quechua yellow pine, which is one of the natural preservatives, is lower than that of kebraco, and as the solution concentration increases, the total amount of retention also increases. In samples impregnated with 1 percent concentration solutions, the maximum retention levels were observed. It has been stated that the retention rate varies due to reasons such as the properties of the solutions and the anatomical structure (Alkan, 2016). Uludağ fir and eastern beech wood are impregnated with Imersol Aqua, Tanalith-E. The air-dry density and retention rate of the impregnated test samples were tried to be determined. According to the results of the experiment, the highest air-dry density value was obtained in the beech wood impregnated with Imersol Aqua (0.672 g/cm^3) and the highest retention rate

(1.53%) in the fir wood impregnated with Imersol Aqua (Kaçamer, 2010).

Boron is a material that is applied in a variety of applications, ranging from cleaning agents to space exploration. Furthermore, it is of great importance in terms of having a large part of the boron reserves in Turkey. It is stated that the use of boron in industrial areas requiring advanced technology is increasing and its value is further increased by being used as a raw material (Acarkan, 2002). In duncan test application results, It was determined that the highest value in retention is 37.3 kg/m^3 'from samples treated with pressure-vacuum method in scotch pine and the lowest value was 3.71 kg/m^3 from the samples treated with beech immersion method. The samples treated with the pressure-vacuum process in yellow pine had the maximum retention value of 6.42 %, while samples treated with immersion in beech had the lowest retention value of as 0.30 % (Özçifçi et al., 2009).

While magnesium sulphate is applied in a variety of applications, it is considered to be an excellent preservative material due to its mordant structure and constructive human-environmental interaction. In this study; It's been tried to figure out what kind of physical changes it creates in spruce wood. As a result, it aims to shed light on the fields that can be used in the wood industry to some extent.

MATERIAL and METHOD

Wood Material

In the framework of the research, among the wood species of our region, spruce wood (*Picea orientalis* (L.) Link) was chosen for the research, and sample cuts were obtained from sapwood sections by cutting in the radial direction in accordance with the principles of TS 53 and TS 2470. As an impregnation and binder, borax and magnesium sulphate ($MgSO_4$) were preferred (mordan). 1% concentrate was applied.

METHOD

Sample Preparation for Experiment

Drying the cuts was made according to the standards laid out in the experiment sample preparation. All samples were examined to ensure that the fiber structure was free of cracks, knots, and color defects (fungus, rot etc.). TS 2471 and TS 2472 standards (30x20x20 mm) were applied to determine the specific gravity change while preparing 100x50x30 mm test specimens to determine the adhesion level.

Impregnation Process

The impregnation process was applied in accordance with the conditions in “ASTM-D 1413-76”. Experimental samples were prepared in 13x5x3 cm dimensions and the test samples were made completely dry so that the impregnation agent was not affected by wood moisture (ASTM 1413, 1976).

% Retention (Net Dry Matter Amount)

After impregnation, the amount of substance remained (tkoao-% retention) compared to dry wood was calculated from the specified formula.

Moes-Moeö

$$R (\%) = \frac{\text{Moes} - \text{Moeö}}{\text{Moeö}} \times 100 \quad (1)$$

Moes = Sample full dry weight after impregnation (g)

Moeö = Sample full dry weight before impregnation (g)

Air / Full Dry Specific Gravity

Moisture amounts of the test samples were carried out according to TS 2471 and specific weights were carried out according to TS 2472. At the temperature of 20 ± 2 °C and relative humidity as 65 ± 5 %, all the weighing and measurements were performed. Calculations were made with below formula.

$$D_{12} = M_{12} / V_{12}$$

$$(D_{12}): \text{Air Dry specific gravity (g/cm}^3\text{)} \quad (2)$$

(M₁₂): Air dry weight of test sample (g)

(V₁₂): Air dry volume (cm³)

Test samples at 12 % relative humidity as normal were used to determine the full dry density. Here, TS 2472 was applied and 0 % value was obtained at 103 ± 2 °C. In order to avoid moisture intake, it

was cooled in CaCl₂ desiccator and necessary weighing/measurements were made. According to the specified formula (TS 2471, 1976; TS 2472, 1976).

$D_o = W_o / V_o \text{ g/cm}^3$ In Formula;

D_o: Full dry density (g/cm³)

W_o: Full dry weight (g)

V_o: Full dry volume (cm³) (3)

RESULTS AND DISCUSSION

Properties of Solution

The solution properties are given in Table 1.

Table 1: Solution properties

Impregnation Material	Solvent	Temperature (°C)	pH		Density (g/ml)	
			BI	AI	BI	AI
Borax (%1)	DW	22°C	9.35	9.36	1,025	1.025
Magnesium sulphate (%1)			6.37	6.37	1,320	1.320

*BI: Before Impregnate *AI: After Impregnate

There was no change in solution pH and density before and after impregnation. It has been reported in the literature that acidic and basic structure affects the technological and other properties of wood.

% Retention

% Retention values are given in Table 2.

Table 2: % Retention test results

Impregnation Material	Vacuum Time (Minute)	Diffusion Time (Minute)	Retention (%)		
			Mean	Std dev	HG
Magnesium Sulphate	30	20	2.29	1.87	C
		30	2.48	3.71	A
		40	2.34	5.22	B
Borax		20	1.72	1.45	E
		30	1.86	6.20	D
		40	1.07	2.93	F

The highest percent preservation was obtained in magnesium sulfate (2.48%) for 30 minutes vacuum and 30 minutes diffusion; the lowest value in borax for 30 minutes vacuum and 40 minutes diffusion (1.07%). The anatomical structure of the wood, as well as the preservative, can cause this situation.

In a study, the effects of barite on wood impregnation and density were investigated. The total retention value in beech wood was determined as maximum 50% barite solution (20.30 kg/m³) as a result of these experiment. In the contrary, the highest full dry density value was found in beech wood, barite in a 50 percent solution (0.60g /cm³) (Tan et al., 2015). In a study, total retention and % retention were determined in wood material impregnated with tea plants. In their results, the lowest % retention rate was found in iroko wood (1.58%), while the highest % retention rate was observed in beech wood (6.75 %). It is reported that the lowest total retention is in iroko (31.27 kg/m³) and the highest total retention value is in beech wood (100.65 kg/m³). According to the retention results; it was stated that organics

obtained from tea plant extract can be applied as preservative in wood material (Atilgan et al., 2013).

Air Dry / Full Dry Density Amount

The change in air dry and fully dry specific gravity is given in Table 3.

Table 3: The change in air dry and fully dry specific density

Impregnation Material	Vacuum Time (Minute)	Diffusion Time (Minute)	DRY-Density (g/cm ³)-AIR			
			Mean	Std dev	Mean	Std dev
Control	-	-	0.35	1.11	0.37	2.34
Borax 1 %	30	20	0.36	2.45	0.38	6.23
		30	0.38	2.01	0.42	3.23
		40	0.37	4.15	0.41	2.51
Magnesium Sulphate 1%		20	0.37	2.85	0.39	2.56
		30	0.39	3.66	0.41	1.93
		40	0.36	2.89	0.38	2.09

In comparison to the control group, there was no noticeable rise in preservative materials. The highest air dry specific gravity was determined as 30 minutes in vacuum and 30 minutes in borax (0.42 g/cm³), the highest full dry specific gravity in 30 minutes in vacuum and 30 minutes diffusion (0.39 g/cm³). We can infer that this is due to the wood type, anatomical structure, and preservative material.

In a study, spruce (*Picea orientalis* (L.) Link) wood was impregnated according to ASTM 1413-76 standard with three different solution concentrations (1, 3, 5%) prepared from boron compounds and barite and the rates of influence on some physical-mechanical properties

were investigated. As a result of his research, it was reported that the air-dry specific gravity value was the most Barite+Ba (0.80 g/cm³), while the mechanical properties had the highest bending resistance value Barite+Ba (100 N/mm²) (Çiçek, 2015). It has been reported that in experimental samples treated with aqueous solutions of boric acid, borax and sodium perborate, which are boron compounds, with various concentration levels, the full dry density values are higher than the non-impregnated (control) sample (Toker, 2007).

CONCLUSION

According to the experimental results; the highest % retention was in magnesium sulfate (2.48%) for 30 minutes vacuum and 30 minutes diffusion. The lowest retention was in borax (1.07%) for 30 minutes vacuum and 40 minutes diffusion. Highest air dry specific gravity for 30 minutes vacuum and 30 minutes diffusion was in borax (0.42 g/cm³), lowest in control sample (0.37 g/cm³), highest full dry specific gravity for 30 minutes vacuum and 30 minutes diffusion in magnesium sulphate (0.39 g/cm³), the lowest was found again in the control sample. Within the scope of the study, paints and varnishes, which are indispensable for their use/protection in the wood/furniture industry, have been tested for stronger adhesion on the surface and for human/environmental health. It will make significant contributions to science and the wood/furniture industry, as well as a significant contribution to the country's economy.

REFERENCES

- Acarkan, N. (2002). Bor ürün Çeşitleri ve Kullanım Alanları, I. Uluslararası Bor Sempozyumu, Kütahya, 1-5.
- Alkan, E. (2016). Doğal emprenye maddeleri ve borlu bileşikler ile emprenye edilen sarıçam (*Pinus sylvestris* L.) odununun fiziksel ve mekanik özelliklerinin incelenmesi, Yüksek Lisans Tezi, Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü, 59 s., Gümüşhane.
- ASTM D 1413–76 (1976). Standartd Methods of Testing Wood Preservatives by Laboratory Soilblock Cultures, Annual Book of ASTM Standarts, USA.
- Atilgan, A., Ersen, N., ve Peker, H. (2013). Different Types Of Wood Treated With Tea Plant Extract Retention Values, Kastamonu University Journal Of Forestry Faculty, 13(2), 278-286
- Çiçek, S. (2015). Barit Ve Bor Bileşiklerinin Ladin Odununda Tutunabilme Özelliği ve Dış Mekan Mobilyalarında Kullanılabilme Olanakları, Artvin Çoruh Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 59 s, Artvin.
- Kaçamer, S. (2010). İmersol Aqua ve Tanalith-E İle Emprenye Edilmiş Isıl İşlemler Ağaç Malzemelerin Yapışma ve Yanma Dirençlerinin Belirlenmesi. Yüksek Lisans Tezi, Karabük Üniversitesi Fen Bilimleri Enstitüsü, 72 s., Karabük
- Özçiğçi, A., Batan, F. (2009). Bor Yağının Ağaç Malzemenin Bazı Mekanik Özelliklerine Etkisi” Politeknik Dergisi, Cilt 12, Sayı 4.
- TS 53 (1981). Odunun Fiziksel Özelliklerini Tayin İçin Numune Alma, Muayene ve Deney Metotları, Ankara.
- TS 2470 (1976). Odunda Fiziksel Ve Mekaniksel Deneyler İçin Numune Alma Metotları ve Genel Özellikler, Ankara
- TS 2471 (1976). Odunda, Fiziksel ve Mekaniksel Deneyler İçin Rutubet Miktarı Tayini, Ankara.
- TS 2472 (1976). Odunda, Fiziksel Ve Mekaniksel Deneyler İçin Birim Hacim Ağırlığı Tayini, Ankara

- Tan, H., Peker, H. (2015). Barit Maddesinin Ahşapta Emprenye Edilebilme Özelliği ve Yoğunluk Üzerine Etkisi, Fırat Üniv. Mühendislik Bilimleri Dergisi, Elazığ s.27(1), 29-33. ThermoWood processing”, Journal of Environmental Biology, 31: 1007-1011.
- Toker, H. (2007). Borlu Bileşiklerin Ağaç Malzemenin Bazı Fiziksel Mekanik ve Biyolojik Özelliklerine Etkilerinin Belirlenmesi, Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Üstün, S. (2019). Bakır Esaslı Emprenye Maddeleri ile Muamele Edilen Odunun Doğal Koşullarda Fiziksel Performans Özellikleri Ve Vernikleme Öncesi Bakır Esaslı Emprenye Maddeleri ile Emprenye İşleminin Fiziksel Performans Özelliklerine Etkisi, Muğla Sıtkı Koçman Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 92 s. Muğla.

NOTE: IMCOFE-2020 / 07-09 July 2020 Russia/ St.Petersburg.
220-226 pp Presented at the symposium.

CHAPTER 5

3D MESH MATCHING USING SURFACE DESCRIPTOR AND INTEGER LINEAR PROGRAMMING

Roaa SOLOH ¹
Abdallah ELCHAKIK ²
Hassan ALABBOUD ³
Ahmad SHAHIN ⁴
Adnan YASSINE ⁵

¹ Lebanese University, LIA Laboratory, Doctoral School of Sciences & Technology, AZM Center - Mitein Street, Tripoli - Lebanon

Normandie Univ., UNIHAVRE, LMAH, FR CNRS 3335, ISCN, 76600 Le Havre, France roaa.soloh@etu.univ-lehavre.fr ORCID ID 0000-0003-0043-9892

² Beirut Arab University, Department of Computer Science, Tripoli- Lebanon a.alshakik@bau.edu.lb ORCID ID 0000-0002-6859-033X

³ Lebanese University, Faculty of Economics and Business administration Head of Common Core Department (including Department of Applied Mathematics), Tripoli - Lebanon hassan.alabboud@ul.edu.lb ORCID ID /0000-0002-8566-488X

⁴ Lebanese University, LIA Laboratory, Doctoral School of Sciences & Technology, Tripoli - Lebanon

Lebanese University, Department of Computer Information Systems, Faculty of Business – 3, Kobbah Campus, Tripoli - Lebanon

ashahin@ul.edu.lb ,ORCID ID 0000-0003-4704-3035

⁵ Normandie Univ., UNIHAVRE, LMAH, FR CNRS 3335, ISCN, 76600 Le Havre, France

Normandie Univ., UNIHAVRE, ISEL, 76600 Le Havre, France

adnan.yassine@univ-lehavre.fr ORCID ID 0000-0001-6875-072X

INTRODUCTION

Matching and similarity between 3D shapes is a fundamental task whether for shape segmentation, classification, retrieval, and all of these with many others are important procedure in different areas like computer vision, biomedical modeling, mechanical engineering. At the same time, the widespread of 3D data brings the need to develop algorithms in order to match, classify as well recognizing it. Unlike images and range scans, 3D models do not depend on the configuration of cameras, light sources, or surrounding objects (e.g., mirrors). As a result, they do not contain reflections, shadows, occlusions, projections, or partial objects, which greatly simplify finding matches between objects of the same type.

Descriptors whether local, or global ones, should capture most significant features of the shape in order to represent it in a well manner. Many approaches were proposed to capture the details of shapes, some lack the capacity in terms of huge memory required, others may be variant towards affine transformations, etc. And since this step is a crucial one for the dependence of other steps on it, in our work we proposed an efficient way to describe the 3D model, by modeling its surface as a weighted graph using the Gaussian function, followed by combining precise curvatures formulating compact feature vector. To follow on, we use Louvain clustering method implemented by (Bonald et al., 2020) in order to segment our mesh into regions. These regions are set to be our subgraphs, and they are

matched through linear sum assignment problem (Virtanen et al., 2020)

The rest of the paper is organized as follows. Section 2, shows some literature reviews on the recent work. Section 3, presents the full framework starting by the surface modeling, followed by regions segmentation ending this section by the 3D matching problem. In section 4, we show our experimental results and the robustness of our proposition. The work is concluded in section 5.

1. RELATED WORKS

Shape descriptors obtained locally, on each vertex becomes an important step behind matching, retrieval, classification, segmentation of 3D models. Local shape descriptors speed up the process by lowering the objects into a little feature that are responsible for matching. In (Gatzke & Grimm, 2006) an overview shows how computing curvature plays a vital role in representing a shape. Similarly, many other papers deal with the importance of curvatures, authors in (Soltanpour, Boufama, & Wu, 2017) works with curvature maps as a feature extraction for face recognition, as well showing the priority of using local feature on global ones. H'roura et. al in (H'roura et al., 2017) present in their paper an overview of 3D object recognition, as well the representative states of the art methods. According to (Körtgen et al., 2003) shape descriptors must obey some requirements in order to classify this descriptor as a good descriptor, so researchers tend to ensure the quality of their descriptor depending on, not limited to, discriminating power regarding similarity

measures, invariant under transformations, compact in order to minimize the storage though optimality in search dimensionality, as well insensitive to noise and topological deformations.

New contributions on level of multi-scale descriptors are trending nowadays, where in this paper (Cipriano, Phillips, & Gleicher, 2009) they show how extracting features in multi-scale summarizes the shapes of differently sized neighborhoods. The descriptor considers a local neighborhood around a central point with a roughly circular area specified by radial distance. While authors in (Chakik et al., 2020) define multi-scale descriptor map by merging single scale descriptor maps calculated with different Delaunay neighboring level. Our work which is nearly similar to this work, without a multi-scale level, where our descriptor maps at single level shows nearly familiar to the ground truth results.

The work in (Chakik et al., 2020) tends to project height maps with patches in order to obtain Zernike moments on the corresponding images obtained, while in our approach, being concise towards the speed of getting our descriptor maps.

In the recent years, scientific researchers model the matching problem as a combinatorial optimization problem, and since graphs are the most appropriate data representation, the problems are treated as graph matching problem. Several approaches arise to solve the problem, algorithm to image segmentation using two different kinds of local neighborhoods in constructing the graph is done in (Felzenszwalb & Huttenlocher, 2004), the algorithm uses a greedy approach and gives a

well notable results. Establishing a meaningful correspondence between shapes is often difficult, in (Van Kaick et al., 2011) authors shows a review on the available methods and algorithms detailed for the appropriate correspondence depending on applications. Sahillioğlu, Y. et. Al in their paper (Sahillioğlu & Yemez, 2011) propose a dense shape correspondence method which is computationally efficient. They minimize the isometric distortion directly in the 3D Euclidean space. A drawback for their work is the symmetry problem, which is inherent to all isometry-based correspondence algorithms. 3D shapes matching methods as Riemannian (Wang et al., 2005) or conformal (Kurtek et al., 2010) geometries have proposed to use a local optimization technique to minimize the non-convex energy. One way to solve the matching problem is to match maximum relational sub graph matching among the set of feature regions that represent an object as done in (El Chakik et al., 2020). To overcome several issues arise from the previous we tend to propose a framework that extract simple, yet powerful, well representative shape descriptors, that are concise to store. Followed by segmenting the vertices into regions having nearly same geometrical descriptors, in order to perform subgraph matching.

2. PROPOSED FRAMEWORK

In our framework, we tend to find matching between 3D models. We start by modeling the surface of mesh as a weighted graph, where our main target is to match subgraphs obtained from the feature regions extracted depending on the descriptor map. Our descriptor map is

defined by extracting several curvatures formulating a compact feature vector.

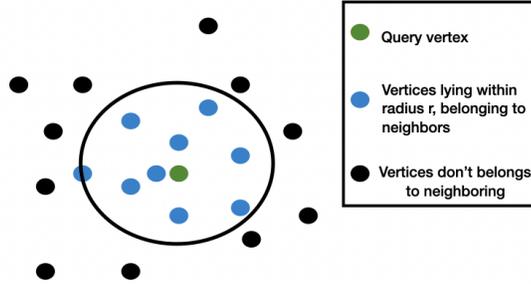


Figure 1: Neighboring Construction

2.1. Descriptor Map Estimation

We start by representing the 3D surface as a weighted graph $G=(\mathcal{V}, \mathcal{E}, \mathbf{w})$ where \mathcal{V} is finite set of vertices, \mathcal{E} is finite set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, and \mathbf{w} represents the weight function, it's a similarity measure between vertices. At the same time, we tend to obtain the set of neighbors (\mathbf{v}_i), for a given vertex \mathbf{v}_i where we find the neighbors within a given radius, even the vertices lying on the boundary are included. Figure 1 shows the neighborhood construction of a single vertex.

The weight function between two connected vertices is calculated as:

$$(\mathbf{v}_i, \mathbf{v}_j) = e^{-\frac{(d_{v_i} - d_{v_j})^2}{2\theta^2}} \quad (1)$$

Where d of any vertex represents its descriptor vector, while θ is the standard deviation.

Each vertex has a compact feature vector that called as a descriptor, and computed by obtaining four curvatures as below:

$$\mathcal{S}\mathcal{J} = \frac{2}{\pi} \tan^{-1} \left(\frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1} \right) \quad (2)$$

$$\mathcal{C}\mathcal{J} = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \quad (3)$$

$$\mathcal{G} = \kappa_1 \kappa_2 \quad (4)$$

$$\mathcal{M} = \frac{1}{2} (\kappa_1 + \kappa_2) \quad (5)$$

Ending up, with a matrix \mathcal{D} [$n * 4$] where n is the number of vertices we have in our model, and four is the components of the descriptor.

We show in Figure 2 the representation of our descriptor compared to the ground truth, and proposed approach from (El Chakik et al., 2020), it shows how the feature vector extracted by our approach is doing well in discriminating.

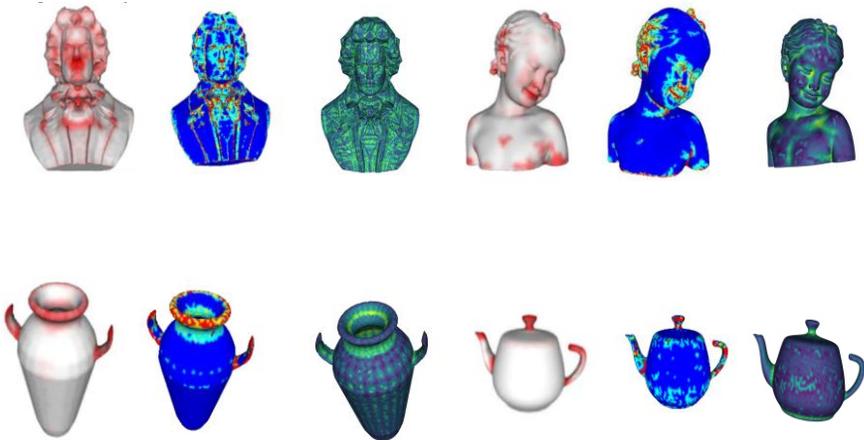


Figure 2: Descriptor Maps. Left to Right: Ground Truth, proposed approach

2.2. Feature Regions Detection

Clustering the 3D objects is an important step to do before matching these objects, where we end up with subgraphs which reduces the complexity of the matching. To do so, we cluster our vertices using the feature vectors and the weighted graph. Integer linear programming is the modeling for our problem, where we try to maximize the modularity of our function.

At first, each vertex has its own label, vertices similar to each other are merged and labels are decreased. The modeling is as follows:

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j), \quad (6)$$

where

- A_{ij} represents the edge weight between nodes i and j ;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j , respectively;
- m is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is Kronecker delta function $\delta(x, y) = 1$ if $x = y$, 0 otherwise).

To maximize Q , first, each node in the network is assigned to its own community. Then for each node i , the change in modularity is calculated for removing i from its own community and moving it into

the community of each neighbor j of i . This value is easily calculated by two steps: (1) removing i from its original community, and (2) inserting i to the community of j .

At the end, vertices that are similar in their geometric properties will have same label. This stage will assess us to find correspondence between the regions of two models.

2.3. 3D Surface Matching

The matching is done through finding a relation between maximum number of correspondence from the feature regions extracted. This problem is modeled as a linear sum assignment known as minimum weight matching, let X be a boolean matrix where $X[i, j] = 1$ iff row i is assigned to column j . Then the optimal assignment has cost,

$$\min \sum C_{i,j} X_{i,j} \quad (7)$$

This problem is a modified Jonker-Volgenant algorithm with no initialization, described in (Crouse, 2016).

3. EXPERIMENTAL DATA ANALYSIS AND RESULTS

In order to show the effectiveness of our proposed approach, we perform several experimentations on downloaded datasets from “MeshsegBenchmark “ and “CAPOD” (McLaren, 2014; Papadakis, 2014) the models are more than 500 object, with different classes, each contains different poses.

Our proposition gives well matching despite the movement changes in human, teddy, armadillo and the other models. Figure 3 shows the matchings between different poses of person, this obviously shows the invariance of our descriptor regarding movements.

The diversity between models within a class and getting matches is seen in Figure 4, each subfigure, shows a two retrieved models matched. The results ensure that our approach is invariant to different affine transformations, whether rotation, scaling, different poses.

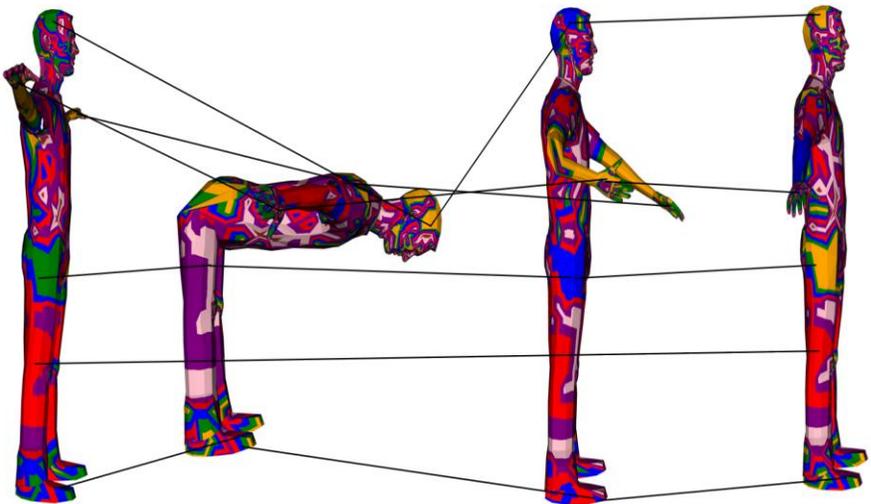


Figure 3: Matched Correspondences

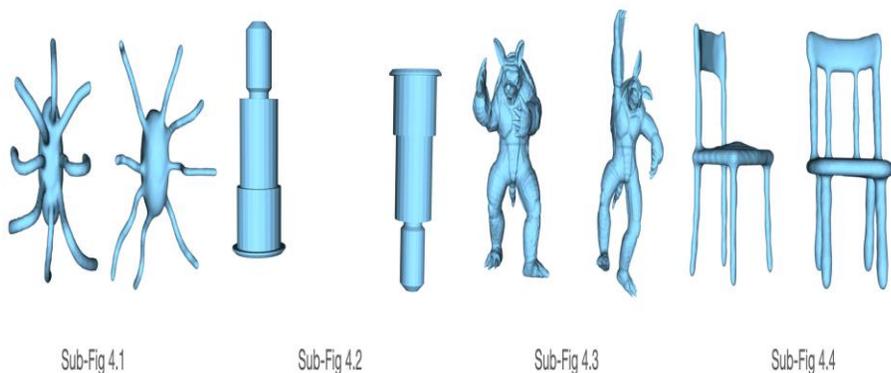


Figure 4: Articulated 3D shape matching

3D Objects of the same class share a fixed pose in terms of object center, scale and rotation while undergoing diverse shape deformations (Papadakis, 2014), the matching between two of the same class with their feature regions visualization is presented in the following figure.



Figure 5: 3D matching within same classes

Getting optimal matches, encourage us to test our framework in order to classify the models, with a query model we try to retrieve the similar models with cosine similarity more than 80%, using the cosine similarity refers to the concept of, if the two similar models are far apart by the Euclidean distance (due to the size of the model), the chances are they may still be oriented closer together through the angle between the models. The classification of two query models as instances, where for each one, the retrieved models that refer to the same class, as well the missed objects from the same class are shown in Table 1.

Accuracy is the ratio of correct predictions to the total number of predictions. It is one of the simplest measures of a model. We must aim for high accuracy for our model. If a model has high accuracy, we can infer that the model makes correct predictions most of the time. And this what we obtain through our framework, where almost each query gives more than 85% of accuracy. At the same time, we were able to get a high recall as a second measure.

Table 1: Retrievals for Bust Models

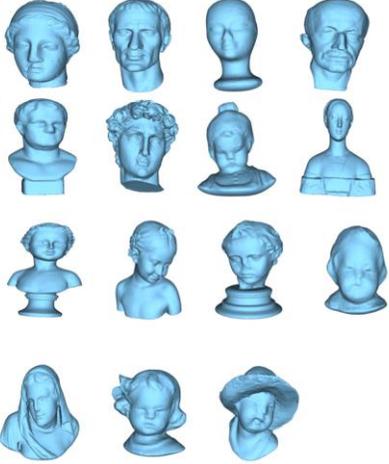
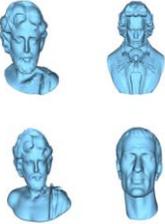
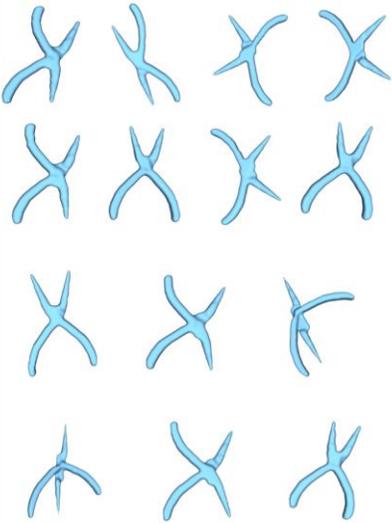
Query model	Retrieved models	Missed models
		

Table 2: Retrievals for Plier models

		
--	--	--

CONCLUSION

A robust approach for 3D shape matching was presented in this paper. The 3D shapes that were initially represented as meshes, their surfaces were modeled as weighted graphs. A new combination of curvatures was extracted formulating a compact feature vector in order to cluster nodes, depending on the similarity measure. Combinatorial optimization is used for the graph matching problem, this is done through integer linear programming approach. Following this, we show the robustness and efficiency of the proposed approach, with experimental results applied on two datasets, with different affine transformations. Finally, classification results are presented, ensuring the wide range of applications to be tested on.

We believe that more experiments should be performed to study the gap in retrieving false models, as well for future work, our assumption is to apply this proposition into real world application.

REFERENCES

- Bonald, T., de Lara, N., Lutz, Q., & Charpentier, B. (2020). Scikit-network: Graph Analysis in Python. *J. Mach. Learn. Res.*, 21, 185-1.
- Cipriano, G., Phillips Jr, G. N., & Gleicher, M. (2009). Multi-scale surface descriptors. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1201-1208.
- Crouse, D.F. (2016). On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679-1696, August 2016, DOI:10.1109/TAES.2016.140952
- El Chakik, A., El Sayed, A. R., Alabboud, H., & Bakkach, A. (2020). An invariant descriptor map for 3D objects matching. *International Journal of Engineering & Technology*, 9(1), 59-68.
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2), 167-181.
- Gatzke, T. D., & Grimm, C. M. (2006). Estimating curvature on triangular meshes. *International journal of shape modeling*, 12(01), 1-28.
- H'roura, J., Bekkari, A., Mammass, D., Bouzit, A., Mansouri, A., Roy, M., & Le Goïc, G. (2017). 3D objects descriptors methods: overview and trends. In *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)* (pp. 1-9). IEEE.
- McLaren D. A. (2014). <https://github.com/forsythrosin/cs348aHW4/tree/master/models/MeshsegBenchmark-1.0>.
- Kurtek, S., Klassen, E., Ding, Z., & Srivastava, A. (2010). A novel Riemannian framework for shape analysis of 3D objects. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 1625-1632). IEEE.
- Körtgen, M., Park, G. J., Novotni, M., & Klein, R. (2003). 3D shape matching with 3D shape contexts. In *The 7th central European seminar on computer graphics* (Vol. 3, pp. 5-17). Budmerice.

- M. Musy et al. (2021). vedo, a python module for scientific analysis and visualization of 3D objects and point clouds. Zenodo, 2021, doi.org/10.5281/zenodo.4287635
- Papadakis, P. (2014). The canonically posed 3D objects dataset. In Eurographics Workshop on 3D Object Retrieval (pp. 33-36).
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272.
- Sahillioğlu, Y., & Yemez, Y. (2011). Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer Graphics Forum* (Vol. 30, No. 5, pp. 1461-1470). Oxford, UK: Blackwell Publishing Ltd.
- Soltanpour, S., Boufama, B., & Wu, Q. J. (2017). A survey of local feature methods for 3D face recognition. *Pattern Recognition*, 72, 391-406.
- Van Kaick, O., Zhang, H., Hamarneh, G., & Cohen-Or, D. (2011). A survey on shape correspondence. In *Computer graphics forum* (Vol. 30, No. 6, pp. 1681-1707). Oxford, UK: Blackwell Publishing Ltd.
- Wang, Y., Gu, X., Hayashi, K. M., Chan, T. F., Thompson, P. M., & Yau, S. T. (2005). Brain surface parameterization using Riemann surface structure. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 657-665). Springer, Berlin, Heidelberg.

CHAPTER 6

PERFORMANCE ANALYSIS OF CHAOTIC MAPS IN DIFFERENTIAL EVOLUTION ALGORITHM FOR HIGH- DIMENSIONAL MULTIMODAL BENCHMARK FUNCTIONS

Assoc. Prof. Dr. Serdar ÖZYÖN¹

¹Department of Electrical and Electronics Engineering, Kütahya Dumlupınar University, KÜTAHYA, TURKEY. serdar.ozyon@dpu.edu.tr ,
ORCID ID 0000-0002-4469-3908

INTRODUCTION

In many studies done in recent years, complex problems difficult to solve with numeric methods are solved by the help of heuristic algorithms. The reason for this is that the solution of multi-variable problems with big solution spaces takes a long time with classic methods. Many algorithms have been developed to solve these kinds of problems in a faster and decisive way. Researches have shown that some points overlooked by the researchers while developing these algorithms can be realized by another researcher later and the performance of the heuristic algorithms can be improved. One of these improvement methods is to use the chaotic number generator in random number generation in the present algorithm (Peitgen, Jurgens, & Saupe, 1992).

As a result of the literature review, it has been seen that many algorithms were used together with chaotic mapping methods with different aims and approaches. These are chaotic biogeography-based optimisation-CBBO (Saremi, Mirjalili, & Lewis, 2014), chaotic charged system search-CCSS (Nouhi, Talatahari, Kheiri, & Cattani, 2013), chaotic fruit fly optimization-CFOA (Mitic, Vukovic, Petrovic, & Miljkovic, 2015), chaotic bat-CBA (Gandomi, & Yang, 2014), chaotic krill herd-CKHA (Wang, Guo, Gandomi, Hao, & Wang, 2014), chaotic harmony search-CHS (Alataş, 2016), chaotic particle swarm optimization-CPSO (Alataş, Akın, & Bedri, 2009; Alataş, & Akın, 2009), chaotic gravitational search-CGSA (Durmuş, Özyön, & Temurtaş, 2016; Chaoshun, Jianzhong, Jian, & Han, 2012),

chaotically initialized differential evolution-CIDE (Ozer, 2010), chaotic bat swarm optimisation-CBSO (Jordehi, 2015), chaotic artificial bee colony-CABC (Alataş, 2010) and chaotic improved imperialist competitive-CICA (Talatahari, Farahmand Azar, Sheikholesman, & Gandomi, 2012) algorithms. It has been seen that in these studies, selected algorithms have been applied to the chaotic mapping method once or several times and the obtained values have been examined. As far as is known 25 chaotic mapping methods defined or used in different studies in literature have been together in this study for the first time and the performance evaluation has been done. The aim is to present an extensive source and data for the future studies.

In order to do a more effective and decisive search together with chaotic number generators in different structures, DE algorithm has been selected for use due to the positive feedbacks obtained from literature. In the study 25 different chaotic mapping methods have been used together with DE algorithm. The algorithms formed by the use of DE algorithm together with chaotic mapping methods have been called Chaotic-based Differential Evolution Algorithm (CbDE-X). In the developed 25 different CbDE-X algorithms while the individuals in the first population are formed, they have been located in the first individual search space randomly. As for the other individuals, they have been located depending on the location of the first individual by using 25 different chaotic mapping methods which have been defined in the subsections. In other words, individuals in CbDE-X algorithms have not spread in the search space randomly, but

according to a certain chaotic order. 25 different chaotic DE structures (CbDE-1,...,CbDE-25) that have been defined differently, have been applied to 5 different high dimensional, multimodal test functions successfully and the obtained results have been evaluated. Two different aims come into prominence in this study, the first of these, is to improve the performance of DE algorithm. As for the second aim, it is to determine the positive or negative effects of the 25 different chaotic mapping methods in literature on heuristic algorithms. In the following studies that will be done by the researchers, the approaches proposed in this study can be integrated to other heuristic algorithms and their performances can be evaluated.

1. DIFFERENTIAL EVOLUTION ALGORITHM (DE)

The DE algorithm is a simple but heuristic optimization technique based on the genetic algorithm in terms of operation. It was developed by Price and Storn in 1995 for solution of the problems with continuous parameters, in particular. This algorithm does not operate on the basis of population and chromosomes are separately subjected to the operators in order to create a new individual. Mutation and crossover operators are used during this process. If the fitness of the new individual is better than that of the old one, the new individual is transferred to the next generation; otherwise, the old one is transferred to the next generation (Ozer, 2010; Storn, & Price, 1997; Mandal, Chatterjee, & Bhattacharjee, 2013; Li, & Yin, 2012).

As for the parameters used in the DE algorithm; population size is shown as NP , number of variables (number of genes - dimension) as

D , generation $(1,2,3,\dots,g_{max})$ as g , ratio of crossover as CR and the factor of scaling as F . Steps of the process in relation to the DE algorithm could be categorized as coding, mutation, crossover and selection. The flowchart of the DE algorithm is given in Figure 1.

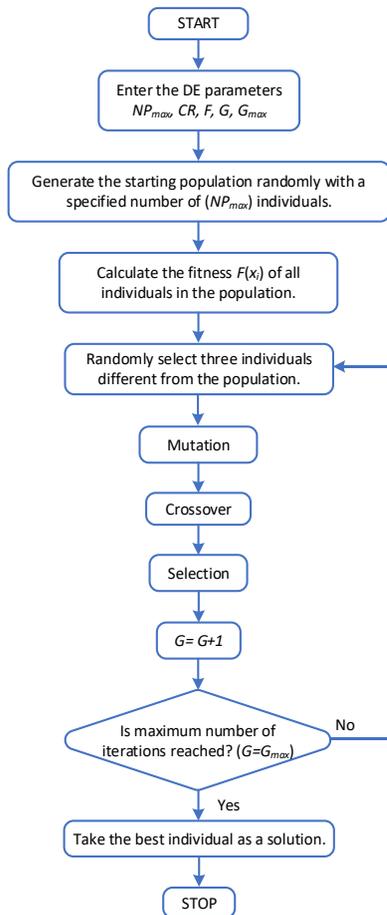


Figure 1: The DE algorithm flowchart

- **Initial population and coding**

In addition to the current chromosome, three chromosomes are required for the creation of new chromosomes in the DE algorithm. Therefore, population size must be higher than three $NP > 3$. The algorithm begins with a randomly initiated population which utilize NP D -dimension parameter vector within constrained by the minimum and maximum bounds (Ozer, 2010; Storn, & Price, 1997; Mandal, Chatterjee, & Bhattacharjee, 2013; Li, & Yin, 2012).

$$\bar{x}_{min} = \{x_{1,min}, x_{2,min}, x_{3,min}, \dots, x_{D,min}\} \quad (1)$$

$$\bar{x}_{max} = \{x_{1,max}, x_{2,max}, x_{3,max}, \dots, x_{D,max}\} \quad (2)$$

Creation of the initial population composed of chromosomes with NP number and D size is calculated by using the equation (3).

$$x_{j,i,g=0} = x_{j,min} + rand_{i,j} [0,1] \cdot (x_{j,max} - x_{j,min}), \quad i = 1, \dots, NP \quad j = 1, \dots, D \quad (3)$$

In this equation, $x_{j,i,g}$ represents j parameter of i chromosome in g generation while $(x_{j,min}, x_{j,max})$ represents the minimum and maximum values for the variables. Where $rand_{i,j} [0,1]$ is a uniformly distribution random number between 0 and 1.

- **Mutation**

Mutation means making random changes on the chromosome genes. Except for the chromosome to be subjected to the DE algorithm, three separate chromosomes are selected (r_1, r_2, r_3). Difference of the first two chromosomes is calculated and multiplied with F . F is generally

used as a value, which varies between 0 and 2. This weighted difference chromosome is summed with the third chromosome (Ozer, 2010; Storn, & Price, 1997; Mandal, Chatterjee, & Bhattacharjee, 2013; Li, & Yin, 2012).

$$n_{j,i,g+1} = x_{j,r_3,g} + F.(x_{j,r_1,g} - x_{j,r_2,g}) \quad (4)$$

In this equation, $n_{j,i,g+1}$ represents the trial chromosome subjected to $g+1$ mutation and crossover, r_1, r_2, r_3 represent the random chromosomes to be used for creation of new chromosomes $\{r_{1,2,3} \in \{1,2,3,\dots, NP\}, r_1 \neq r_2 \neq r_3 \neq i\}$.

- **Crossover**

A new chromosome ($u_{i,g+1}$) is created by using the different chromosome obtained as a result of the mutation and $x_{i,g}$ chromosome. As for the trial chromosome, genes are selected from the difference chromosome with a probability of CR while they are selected from the current chromosome with a probability of $(1-CR)$. $j=j_{rand}$ condition is used for ensuring that at least one gene is obtained from the newly created chromosome. Gene, which is located in random $j=j_{rand}$ is selected from $n_{j,i,g+1}$ irrespective of CR value (Ozer, 2010; Storn, & Price, 1997; Mandal, Chatterjee, & Bhattacharjee, 2013; Li, & Yin, 2012).

$$x_{j,u,g+1} = \begin{cases} x_{j,n,g+1} & \text{if } rand[0,1] \leq CR \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (5)$$

- **Fitness function**

Additional to the target chromosome, three different chromosomes are used to obtain a new chromosome (trial chromosome) by the help of mutation and crossover operators. The chromosome to be transferred to the new generation ($g=g+1$) is determined by considering the fitness value. Fitness value of the target chromosome is already known. Objective function value of the problem is calculated as the fitness function (Ozer, 2010; Storn, & Price, 1997; Mandal, Chatterjee, & Bhattacharjee, 2013; Li, & Yin, 2012).

- **Selection**

The chromosome with the highest feasibility is transferred to the new generation. The cycle continues until it becomes $g=g_{max}$, when it becomes g_{max} , the best individual is taken as solution (Ozer, 2010; Storn, & Price, 1997; Mandal, Chatterjee, & Bhattacharjee, 2013; Li, & Yin, 2012).

$$x_{i,g+1} = \begin{cases} x_{u,g+1} & \text{if } f(x_{u,g+1}) \leq f(x_{i,g+1}) \\ x_{i,g} & \text{otherwise} \end{cases}$$

(6)

- **Stopping criteria**

Aim of this equation is to obtain chromosomes with better fitness values in a continuous manner and to achieve or come close to the optimum value. This cycle continues until it becomes $g=g_{max}$. The algorithm is stopped depending on the number of iterations.

1.1. Case 0: No Chaos (DE)

In this case where chaotic mapping methods have not been used, the individuals in the population have been located in the search space randomly according to equation (3). For 200 numbers generated randomly, their locations in the search space mapped at (0-1) interval and the code block have been given in Figure 2. Figure and code block for nonchaotic, random case have been given in order to be able to make comparison with chaotic structures that will be given in other sections.

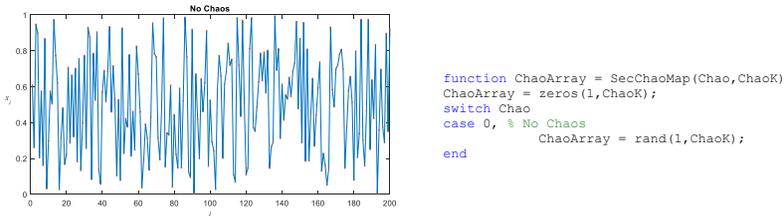


Figure 2: Search Area and for Matlab Code for No Chaos (DE)

2. CHAOS-BASED DIFFERENTIAL EVOLUTION ALGORITHM (CBDE)

Chaos, which is sensitive to the initial conditions but limited, is an indecisive and dynamic behavior. The idea of using chaotic systems instead of random selection process has become common recently. Especially in random selection-based optimization algorithms (with random selection basis) chaotic dynamics take the role of random selection. It has been shown in many studies that the use of chaotic number generators instead of random number generators will provide advantage. Notably in numeric calculation, decision making and

heuristic algorithm, random selection equations require a long period and unbalanced uniform distributions. Whereas chaos is a deterministic dynamic system in which there are random similar operations. By its very nature, it seems to be random and cannot be predicted. However, it behaves like an element of order and operates in an order. In this study 25 different chaotic mapping structures following a certain order have been used as the source of random number generation. In the formation of starting population in the initialization level of search algorithms the locations of the individuals are formed randomly. In the study, instead of this random localization, the individuals will be placed into the starting population according to the number values generated from chaotic maps. In this way, the starting population will not be formed randomly, but with a more systematic distribution (Peitgen, Jurgens, & Saupe, 1992).

In Chaotic-based Differential Evolution Algorithm (CbDE), when the individuals in the initial population are formed, the first individual has been located in the search space randomly while the other individuals have been located depending on the location of the randomly formed first individual by using 25 different chaotic mapping methods which have been defined in the subsections. 25 chaotic mapping structures that take place in literature have been defined in the following sections together with their mathematical equations and search spaces the search spaces of all defined chaotic structures have been mapped at the search interval of [0-1].

2.1. Case 1: Arnold's Cat Map (CbDE-1)

The mathematical definition of this mapping method has been given in equation (7), and its two-dimensional search space and code block have been given in Figure 3 (Arnold, & Avez, 1968).

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} \pmod{1}, \quad X_k, Y_k \in [0,1] \quad (7)$$

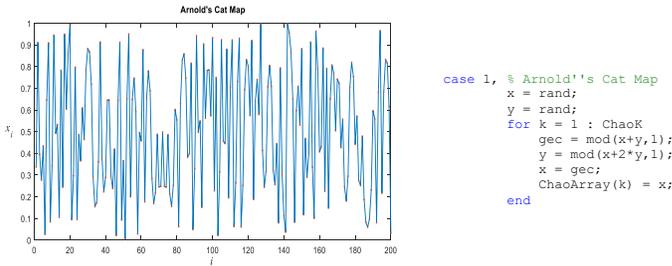


Figure 3: 2-D search area and for Matlab Code for Arnold's Cat Map (CbDE-1)

2.2. Case 2: Bernoulli Shift Map (CbDE-2)

The mathematical definition of this mapping method has been given in equation (8), and a special occasion of it in equation (9) (Ge, Tan, & Wang, 2009). As for its two-dimensional search space and code block, they have been given in Figure 4.

$$X_{k+1} = \begin{cases} \frac{X_k}{1-\lambda} & 0 < X_k \leq 1-\lambda \\ \frac{X_k - (1-\lambda)}{\lambda} & 1-\lambda \leq X_k < 1 \end{cases} \quad (8)$$

$$X_{k+1} = 2X_k \pmod{1} \quad (9)$$

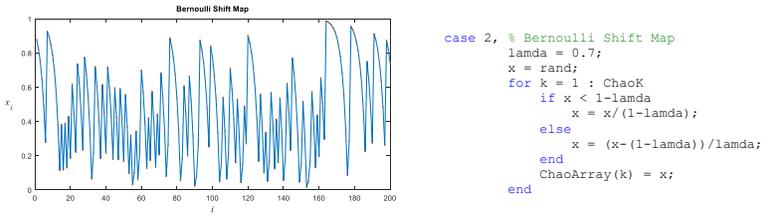


Figure 4: Search Area and for Matlab Code for Bernoulli Shift Map (CbDE-2)

2.3. Case 3: Chebyshev Map (CbDE-3)

The mathematical definition belonging to this mapping method has been given in equation (10), and its search space and code block have been given in Figure 5. f taking place in the equation has been taken in this study as 50 (Stoyanov, 2014; Stoyanov, & Kordov, 2014).

$$X_{k+1} = \cos(f \cos^{-1} X_k), \quad f \in [2, \infty) \quad (10)$$

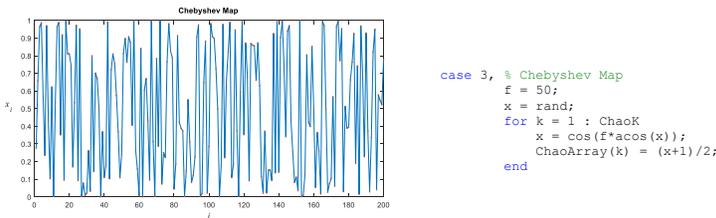


Figure 5: Search Area and for Matlab Code for Chebyshev Map (CbDE-3)

2.4. Case 4: Circle Map (CbDE-4)

The definition of The Circle Map, which is a chaotic mapping method generating chaotic cycles at $(0,1)$ interval, is as in equation (11) (Zheng, 1994). Two-dimensional search space and code block belonging to the method have been given in Figure 6.

$$X_{k+1} = \left\{ X_k + b - \left(\frac{a}{2\pi} \right) \sin(2\pi X_k) \right\} \bmod(1), \quad a = 0.5, b = 0.2 \quad (11)$$

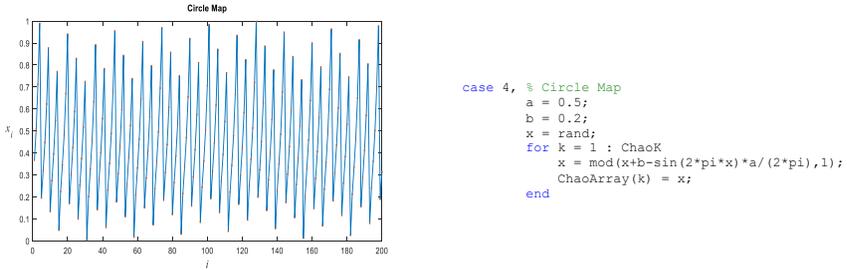


Figure 6: Search Area and for Matlab Code for Circle Map (CbDE-4)

2.5. Case 5: Cubic Map (CbDE-5)

The mathematical definition of The Cubic Map, which is a chaotic mapping method generating chaotic cycles at $(0,1)$ interval, has been given in equation (12) and its search space and code block have been given in Figure 7 (Google, 2019a).

$$X_{k+1} = pX_k(1 - X_k^2), \quad X_k \in (0,1), \quad p = 2.59 \quad (12)$$

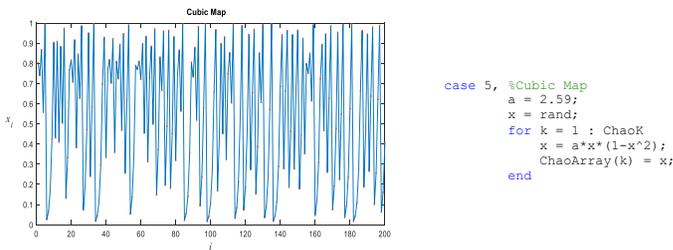


Figure 7: Search Area and for Matlab Code for Cubic Map (CbDE-5)

2.6. Case 6: Duffing Map (CbDE-6)

This mapping method depends on two invariables as a and b . These invariables have been generally taken as $a=2.75$ and $b=0.2$ in

literature. This method is a discrete version of the Duffing equation in mathematics (Google, 2019b; Mahdi, Jawad, & Hreshee, 2016). The mathematical definition belonging to this mapping method has been given in equation (13), and its search space and code block have been given in Figure 8.

$$\begin{aligned} X_{k+1} &= Y_k \\ Y_{k+1} &= -bX_k + aY_k - Y_k^3, \quad a = 2.75, \quad b = 0.2 \end{aligned} \quad (13)$$

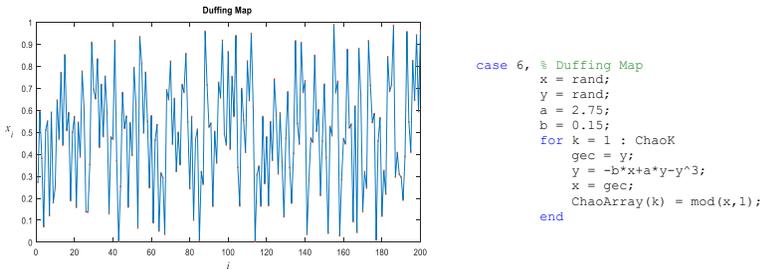
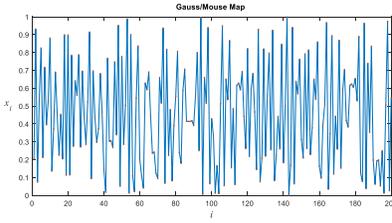


Figure 8: Search Area and for Matlab Code for Duffing Map (CbDE-6)

2.7. Case 7: Gauss/Mouse Map (CbDE-7)

One of the well-known mapping methods which is commonly used in the generation of chaotic series is Gauss/Mouse Map method (Bucolo, Caponetto, Fortuna, Frasca, & Rizzo, 2002; Hilborn, 2004). The mathematical definition belonging to this mapping method has been given in equation (14), and its search space and code block have been given in Figure 9.

$$\begin{aligned} X_{k+1} &= \begin{cases} X_k / 0.7 & X_k = 0 \\ 1 / X_k \bmod(1) & X_k \in (0,1) \end{cases} \\ 1 / X_k \bmod(1) &= \frac{1}{X_k} - \left\lfloor \frac{1}{X_k} \right\rfloor \end{aligned} \quad (14)$$



```

case 7, % Gauss/Mouse Map
x = rand;
for k = 1 : ChaoK
    if x ~ = 0
        x = mod(1/x,1);
    end
    ChaoArray(k) = x;
end

```

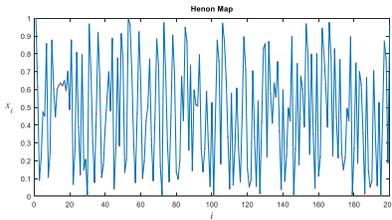
Figure 9: Search Area and for Matlab Code for Gauss/Mouse Map (CbDE-7)

2.8. Case 8: Henon Map (CbDE-8)

Two different mathematical definitions belonging to this mapping method have been given in equation (15) and (16) (Henon, 1976). Two-dimensional search space and code block of the method have been given in Figure 10.

$$\begin{aligned} X_{k+1} &= 1 - a(X_k)^2 + Y_k \\ Y_{k+1} &= bX_k, \quad a = 1.4, \quad b = 0.3 \end{aligned} \quad (15)$$

$$X_{k+1} = 1 - aX_k^2 + bX_{k-1}, \quad a = 1.4, \quad b = 0.3 \quad (16)$$



```

case 8, % Henon Map
a = 1.4;
b = 0.3;
x1 = 0;
x = rand;
for k = 1 : ChaoK
    x0 = x;
    x = 1 - a*x0^2 + b*x1;
    x1 = x0;
    ChaoArray(k) = mod(x,1);
end

```

Figure 10: Search Area and for Matlab Code for Henon Map (CbDE-8)

2.9. Case 9: ICMIC - Iterative chaotic map with infinite collapses (CbDE-9)

The mathematical definition belonging to this mapping method has been given in equation (17) (Caponetto, Fortuna, Fazzino, & Xibilia,

2003; Lu, Wang, Fei, & Qiu, 2014). Two-dimensional search space and code block of the method have been shown in Figure 11. In this study a has been taken as $a=2$.

$$X_{k+1} = \sin\left(\frac{a}{X_k}\right) \text{ for } a > 0, X_k \in [-1,0) \cup (0,1] \quad (17)$$

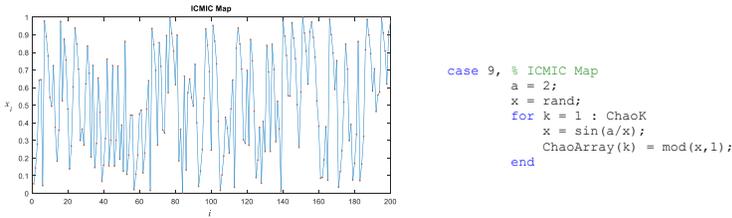


Figure 11: Search Area and for Matlab Code for ICMIC Map (CbDE-9)

2.10. Case 10: Ikeda Map (CbDE-10)

The mathematical definition belonging to this mapping method has been given in equation (18) (Ikeda, 1979; Ikeda, Daido, & Akimoto, 1980; Abottes, Liceaga, Kiryanov, & Wilson, 2016). Two-dimensional search space and code block of the method have been shown in Figure 12.

$$X_{k+1} = 1 + u(X_k \cos t_n - Y_k \sin t_n)$$

$$Y_{k+1} = u(X_k \sin t_n + Y_k \cos t_n) \quad t_n = 0, 4 - \frac{6}{1 + X_k^2 + Y_k^2}, \quad u \geq 0.6 \quad (18)$$

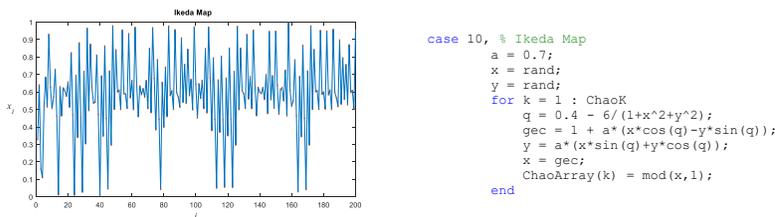


Figure 12: Search Area and for Matlab Code for Ikeda Map (CbDE-10)

2.11. Case 11: Intermittency Map (CbDE-11)

The mathematical definition belonging to this mapping method has been given in equation (19), and its two-dimensional search space and code block have been given in Figure 13. In this study c , n and P have been taken as $c=0.7$, $n=0.3$, $P=0.5$ (Tavazoei, & Haeri, 2007).

$$X_{k+1} = \begin{cases} \varepsilon + X_k + cX_k^n & 0 < X_k \leq P \\ \frac{X_k - P}{1 - P} & P < X_k < 1 \end{cases} \quad (19)$$

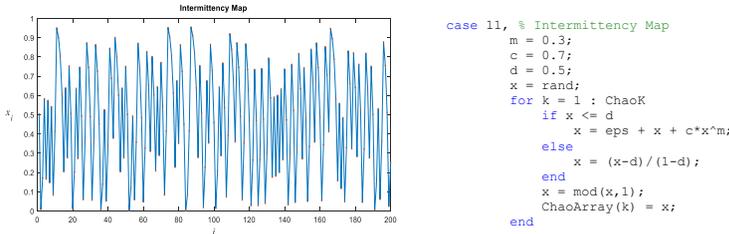
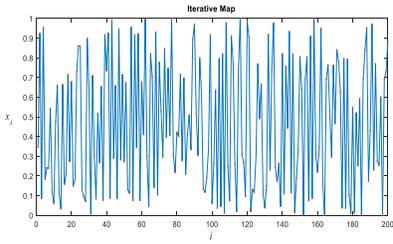


Figure 13: Search Area and for Matlab Code for Intermittency Map (CbDE-11)

2.12. Case 12: Iterative Map (CbDE-12)

The mathematical definition belonging to this mapping method has been given in equation (20), and its two-dimensional search space and code block have been given in Figure 14. For this study a has been taken as $a=0.9$ (May, 1976).

$$X_{k+1} = \sin\left(\frac{a\pi}{X_k}\right), \quad a \in (0,1) \quad (20)$$



```

case 12, % Iterative Map
a = 0.9;
x = rand;
for k = 1 : ChaoK
    x = sin(a*pi/x);
    ChaoArray(k) = mod(x,1);
end

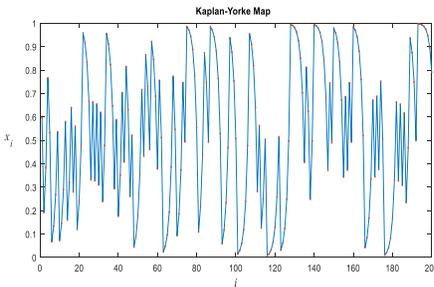
```

Figure 14: Search Area and for Matlab Code for Iterative Map (CbDE-12)

2.13. Case 13: Kaplan-Yorke Map (CbDE-13)

The mathematical definition belonging to this mapping method has been given in equation (21), and its two-dimensional search space and code block have been given in Figure 15 (Kaplan, & Yorke, 1979; Google, 2019c).

$$\begin{aligned}
 X_{k+1} &= aX_k \bmod(1) \\
 Y_{k+1} &= bY_k + \cos(4\pi X_k), \quad 0 \leq a \leq 2, \quad 0 \leq b \leq 1
 \end{aligned}
 \tag{21}$$



```

case 13, % Kaplan-Yorke Map
x = rand;
y = rand;
a = rand;
b = 1.2;
alfa = 0.2;
for k = 1 : ChaoK
    a = mod(2*a,b);
    y = alfa*y + cos(4*pi*x);
    x = a/b;
    ChaoArray(k) = x;
end

```

Figure 15: Search Area and for Matlab Code for Kaplan-Yorke Map (CbDE-13)

2.14. Case 14: Liebovtech Map (CbDE-14)

The mathematical definition belonging to this mapping method has been given in equation (22), and its two-dimensional search space and

code block have been given in Figure 16 (Liebovitch, & Thot, 1991).

In this study d_1 and d_2 have been used as $d_1=0.5, d_2=0.7$.

$$X_{k+1} = \begin{cases} \alpha_1 X_k & 0 < X_k \leq d_1 \\ \frac{d_2 - X_k}{d_2 - d_1} & d_1 < X_k \leq d_2 \\ 1 - \alpha_2 (1 - X_k) & d_2 < X_k \leq 1 \end{cases}, \quad d_1, d_2 \in (0,1), \quad d_1 < d_2 \quad (22)$$

$$\alpha_1 = \frac{d_2}{d_1} [1 - (d_2 - d_1)], \quad \alpha_2 = \frac{1}{d_2 - 1} [(d_2 - 1) - d_1 (d_2 - d_1)]$$

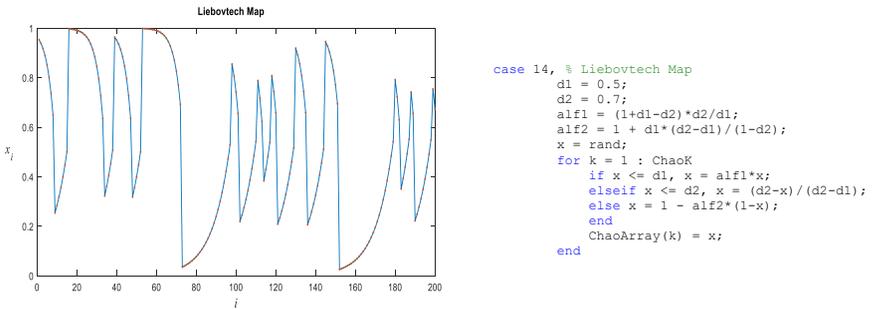


Figure 16: Search Area and for Matlab Code for Liebovitch Map (CbDE-14)

2.15. Case 15: Logistic Map (CbDE-15)

The mathematical definition belonging to this mapping method has been given in equation (23), and its two-dimensional search space and code block have been given in Figure 17 (Saremi, Mirjalili, & Lewis, 2014; Nouhi, Talatahari, Kheiri, & Cattani, 2013).

$$X_{k+1} = aX_k(1 - X_k), \quad 0 < a \leq 4 \quad (23)$$

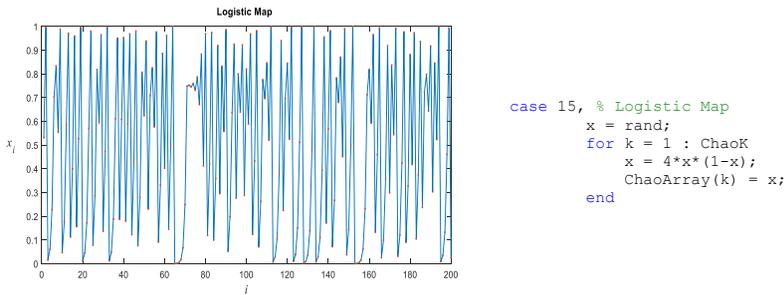


Figure 17: Search Area and for Matlab Code for Logistic Map (CbDE-15)

2.16. Case 16: Lozi Map (CbDE-16)

The mathematical definition belonging to this mapping method has been given in equation (24), and its two-dimensional search space and code block have been given in Figure 18 (Botella-Soler, Castelo, Oteo, & Ros, 2011). In this study a and b have been taken as $a=1.7$ and $b=0.5$.

$$\begin{aligned} X_{k+1} &= 1 - a|X_k| + Y_k \\ Y_{k+1} &= bX_k \end{aligned} \quad (24)$$

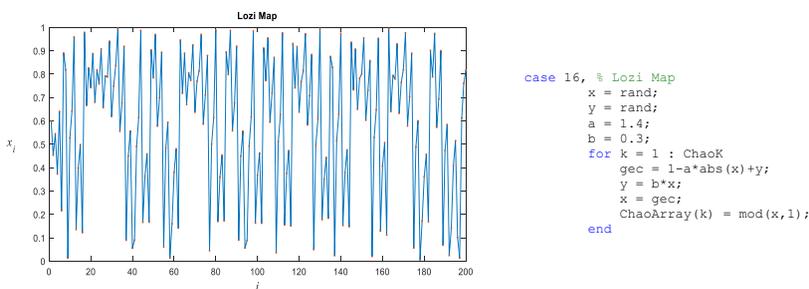


Figure 18: Search Area and for Matlab Code for Lozi Map (CbDE-16)

2.17. Case 17: Piecewise Map (CbDE-17)

The mathematical definition belonging to this mapping method has been given in equation (25), and its two-dimensional search space and code block have been given in Figure 19 (Saremi, Mirjalili, & Lewis, 2014; Mitic, Vukovic, Petrovic, & Miljkovic, 2015). For this study P , has been taken as $P=0.3$.

$$X_{k+1} = \begin{cases} \frac{X_k}{P} & 0 \leq X_k < P \\ \frac{X_k - P}{0.5 - P} & P \leq X_k < \frac{1}{2} \\ \frac{1 - P - X_k}{0.5 - P} & \frac{1}{2} \leq X_k < 1 - P \\ \frac{1 - X_k}{P} & 1 - P \leq X_k < 1 \end{cases} \quad 0 < P < 0.5 \quad (25)$$

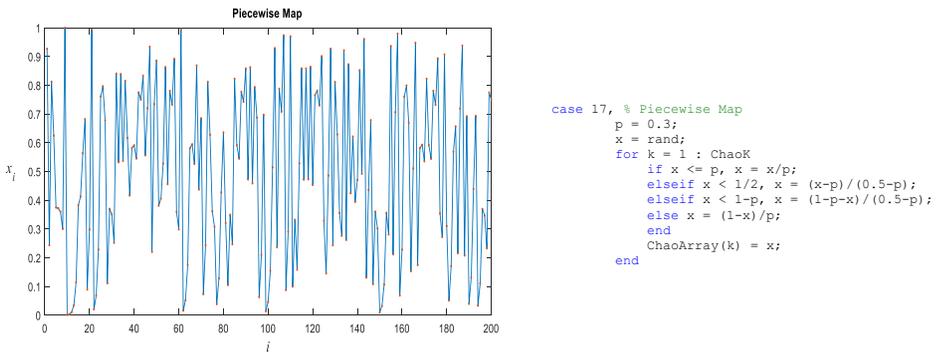


Figure 19: Search Area and for Matlab Code for Piecewise Map (CbDE-17)

2.18. Case 18: Sawtooth Map (CbDE-18)

The mathematical definition belonging to this mapping method has been given in equation (26), and its two-dimensional search space and

code block have been given in Figure 20 (Jessa, 2006). For this study a has been taken as $a = \sqrt{3}$.

$$X_{k+1} = aX_k \bmod(1) \quad 0 < a \leq 2 \quad (26)$$

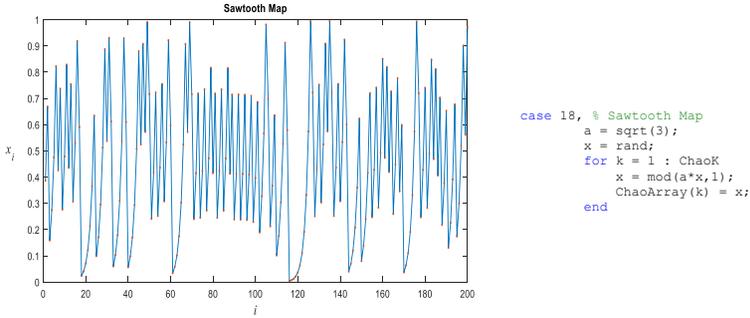


Figure 20: Search Area and for Matlab Code for Sawtooth Map (CbDE-18)

2.19. Case 19: Sine Map (CbDE-19)

The mathematical definition belonging to this mapping method has been given in equation (27), and its two-dimensional search space and code block have been given in Figure 21 (Pareek, Patidar, & Sud, 2005). In this study a has been taken as $a = \sqrt{15} = 3.873$.

$$X_{k+1} = \frac{a}{4} \sin(\pi X_k) \quad 0 < a \leq 4 \quad (27)$$

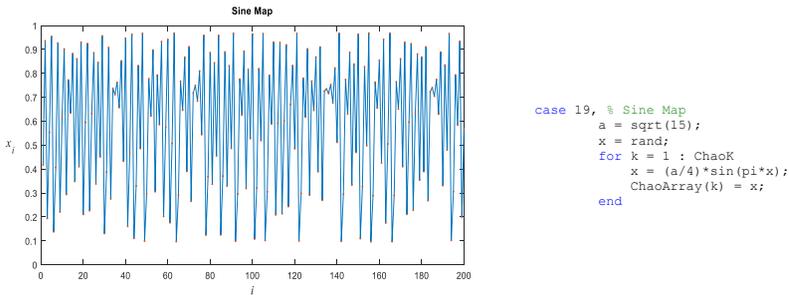


Figure 21: Search Area and for Matlab Code for Sine Map (CbDE-19)

2.20. Case 20: Singer Map (CbDE-20)

The mathematical definition belonging to this mapping method has been given in equation (28), and its two-dimensional search space and code block have been given in Figure 22 (Saremi, Mirjalili, & Lewis, 2014; Jordehi, 2015). For this study μ has been taken as $\mu=1.07$.

$$X_{k+1} = \mu(7.86X_k - 23.31X_k^2 + 28.75X_k^3 - 13.3X_k^4) \quad 0.9 \leq \mu \leq 1.08 \quad (28)$$

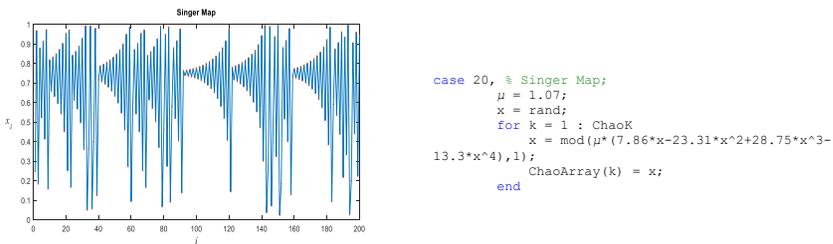


Figure 22: Search Area and for Matlab Code for Singer Map (CbDE-20)

2.21. Case 21: Sinus Map (CbDE-21)

The mathematical definition belonging to this mapping method has been given in equation (29), and its two-dimensional search space and code block have been given in Figure 23 (Nouhi, Talatahari, Kheiri, & Cattani, 2013). For this study a has been taken as $a=2.3$.

$$X_{k+1} = aX_k^{2\sin(\pi X_k)} \quad (29)$$

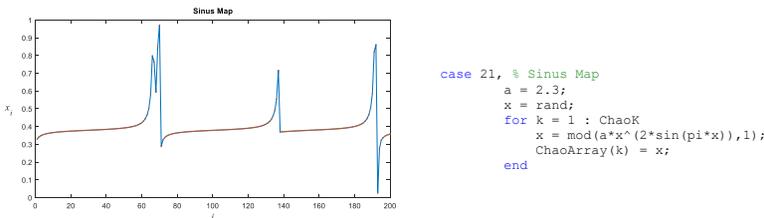


Figure 23: Search Area and for Matlab Code for Sinus Map (CbDE-21)

2.22. Case 22: Sinusoidal Iterator Map (CbDE-22)

The mathematical definition belonging to this mapping method has been given in equation (30), and its two-dimensional search space and code block have been given in Figure 24 (Saremi, Mirjalili, & Lewis, 2014; Nouhi, Talatahari, Kheiri, & Cattani, 2013). In this study a and X_0 have been taken as $a=2.3$ and $X_0=0.7$.

$$X_{k+1} = aX_k^2 \sin(\pi X_k) \quad (30)$$

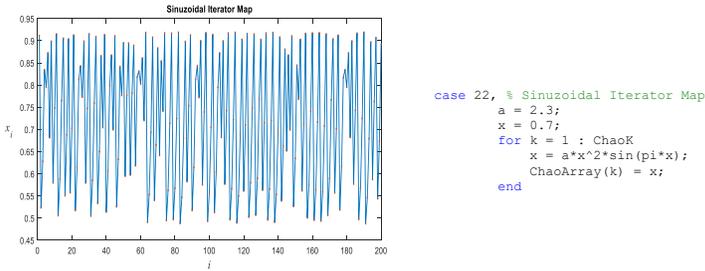
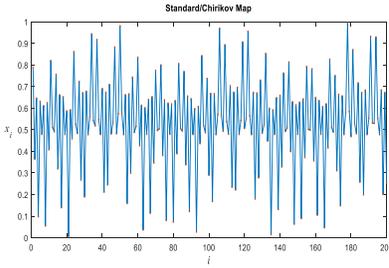


Figure 24: Search Area and for Matlab Code for Sinuzoidal Iterator Map (CbDE-22)

2.23. Case 23: Standard/Chirikov Map (CbDE-23)

The mathematical definition belonging to this mapping method has been given in equation (31), and its two-dimensional search space and code block have been given in Figure 25 (Chirikov, 1979; Zhang, & Xiao, 2013; Patidar, Pareek, & Sud, 2009). For this study a , X_0 and Y_0 have been taken as $a=0.6$, $X_0=0.5$ ve $Y_0=0.5$.

$$\begin{aligned} X_{k+1} &= X_k + a \sin(Y_k) \\ Y_{k+1} &= Y_k + X_{k+1} \end{aligned} \quad (31)$$



```

case 23, % Standard/Chirikov Map
x = 0.5;
y = 0.5;
a = 0.6;
for k = 1 : ChaoK
    x = x + a*sin(y);
    y = y + x;
    ChaoArray(k) = mod(x,1);
end

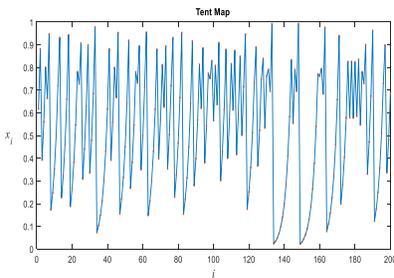
```

Figure 25: Search Area and for Matlab Code for Standard/Chirikov Map (CbDE-23)

2.24. Case 24: Tent Map (CbDE-24)

The mathematical definition belonging to this mapping method has been given in equation (32), and its two-dimensional search space and code block have been given in Figure 26 (Al-shameri, & Mahiub, 2013; Campos-Cantons, Murguia, & Rosu, 2009). For this study a has been taken as $a=0.7$.

$$X_{k+1} = \begin{cases} X_k / a & X_k < a \\ \frac{10}{3} X_k (1 - X_k) & else \end{cases} \quad (32)$$



```

case 24, % Tent Map
a = 0.7;
x = rand;
for k = 1 : ChaoK
    if x < a, x = x / a;
    else x = (1-x)/(1-a);
    end
    ChaoArray(k) = x;
end

```

Figure 26: Search Area and for Matlab Code for Tent Map (CbDE-24)

2.25. Case 25: Zaslavskii Map (CbDE-25)

The mathematical definition belonging to this mapping method has been given in equation (33), and its two-dimensional search space and code block have been given in Figure 27 (Zaslavsky, 1978). For this study v, r, a , have been taken as $v=400, r=3$ ve $a=12.6695$.

$$\begin{aligned} X_{k+1} &= (X_k + v + aY_{k+1}) \bmod(1) \\ Y_{k+1} &= \cos(2\pi X_k) + e^{-r} Y_k \end{aligned} \quad (33)$$

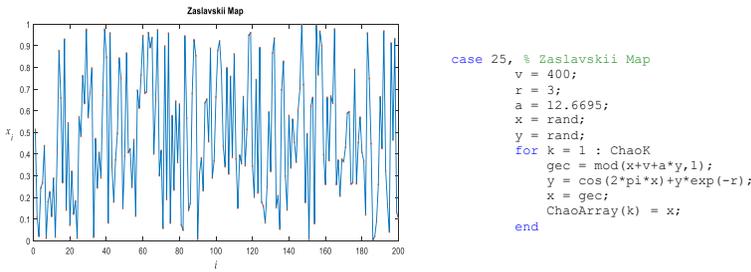


Figure 27: Search Area and for Matlab Code for Zaslavskii Map (CbDE-25)

In the standard DE, positions of genes are determined with random values. The derived CbDE-X algorithms make this definition in a chaotic way. The initial positions of the agents are formed by using the chaotic maps defined above. As it has been shown in (3), in standard DE, in the calculation of the weighted sum of the forces, which acts on each agent, random selection is needed. The derived CbDE-X algorithms define this random selection by chaotic maps. If (3) is modified again, it is defined as in the following:

$$x_{j,i,g=0} = x_{j,\min} + \text{Case}X_{i,j}[0,1].(x_{j,\max} - x_{j,\min}), \quad i = 1, \dots, NP \quad j = 1, \dots, D \quad (34)$$

where $CaseX_{i,j}[0,1]$ is a chaotic variable based on selected map. The flowchart created for the Chaotic based Differential Evolution Algorithm defined in this study has been given in Figure 28. The parts different from classic DE have been defined with red color in the figure.

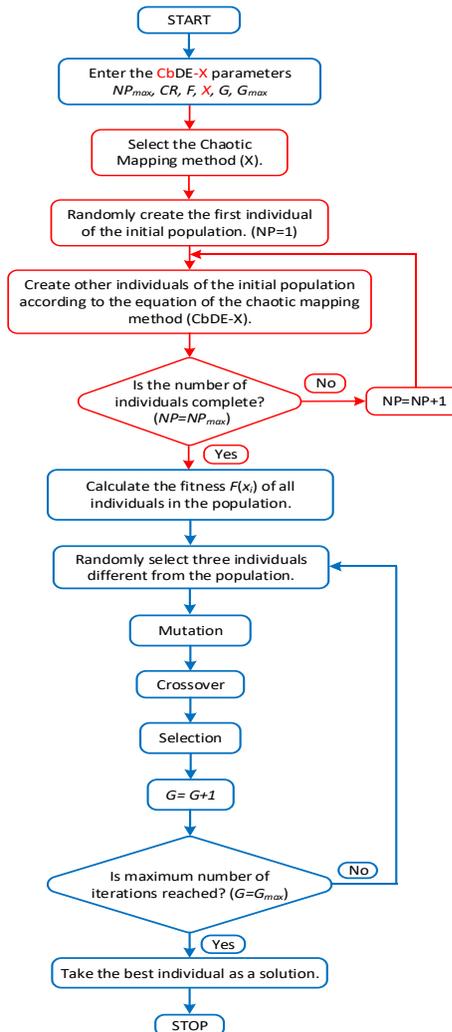


Figure 28: The CbDE-X algorithm flowchart

In the study 25 different approaches have been developed by integrating 25 different chaotic mapping methods to DE algorithm. These approaches have been called as CbDE-1, CbDE-2, CbDE-3, ... , CbDE-25 (*CaseX*). The developed approaches have been applied to 50-dimensional test functions 50 times each and the results have been evaluated.

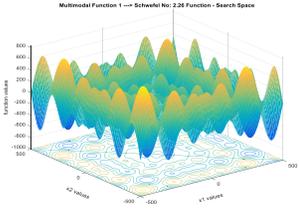
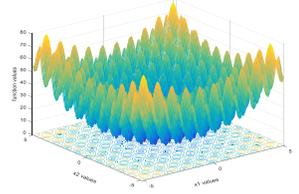
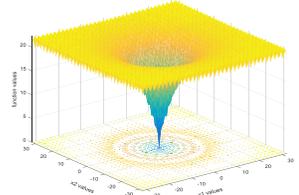
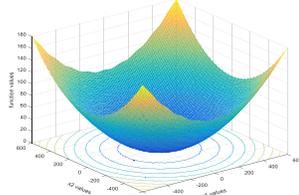
3. TEST FUNCTIONS

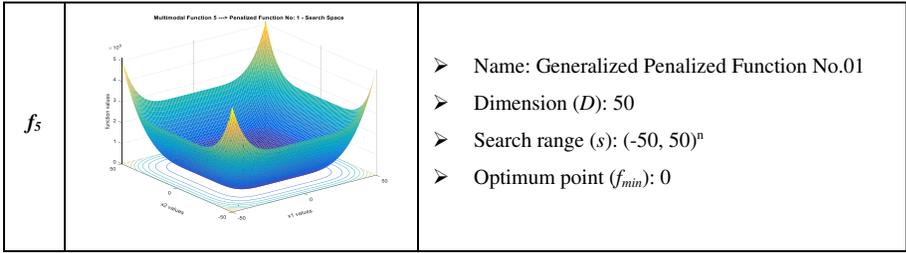
The proposed chaotic structures have been applied to 5 multimodal test functions in order to evaluate their performances. The formulations, definitions and 2-D drawings of these functions have been given in Table 1 and 2. The functions are high dimensional, multimodal functions having many local minimum points and wide search space (Özyön, Yaşar, & Temurtaş, 2018).

Table 1: Multimodal test functions formulation

	Formulation
f_1	$f_1(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x })$
f_2	$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
f_3	$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$
f_4	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
f_5	$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \right\} + \sum_{i=1}^{n-1} u(x_i, a, k, m)$ $y_i = 1 + \frac{x_i + 1}{4}, \quad a = 10, \quad k = 100, \quad m = 4, \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$

Table 2: Multimodal test functions definition

f_1	 <p>Multimodal Function 1 → Schwefel No. 2.26 Function - Search Space</p>	<ul style="list-style-type: none"> ➤ Name: Schwefel's No. 2.26 ➤ Dimension (D): 50 ➤ Search range (s): $(-500, 500)^n$ ➤ Optimum point (f_{min}): $(-418.9829) \times n$
f_2	 <p>Multimodal Function 2 → Rastrigin Function - Search Space</p>	<ul style="list-style-type: none"> ➤ Name: Rastrigin ➤ Dimension (D): 50 ➤ Search range (s): $(-5.12, 5.12)^n$ ➤ Optimum point (f_{min}): 0
f_3	 <p>Multimodal Function 3 → Ackley Function - Search Space</p>	<ul style="list-style-type: none"> ➤ Name: Ackley ➤ Dimension (D): 50 ➤ Search range (s): $(-32, 32)^n$ ➤ Optimum point (f_{min}): 0
f_4	 <p>Multimodal Function 4 → Griewank Function - Search Space</p>	<ul style="list-style-type: none"> ➤ Name: Griewank ➤ Dimension (D): 50 ➤ Search range (s): $(-600, 600)^n$ ➤ Optimum point (f_{min}): 0



4. NUMERICAL RESULTS

The test functions taking place in the study have been solved with DE algorithm and all chaotic mapping methods with the same parameter values in order to be able to make a one-to-one comparison. These parameter values have been given in Table 3. For the solution of the test functions the program coded in MATLAB R2015b has been operated at the workstation with Intel Xeon E5-2637 v4 3.50 GHz processor and 128 GB RAM memory.

Table 3: Parameter values

<i>DE - CbDE-X</i>						
<i>IteN - g_{max}</i>	<i>NP</i>	<i>f_{Call}</i>	<i>Run</i>	<i>D</i>	<i>CR</i>	<i>F</i>
1000	50	50000	50	50	0.15	0.25

In Table 3 *IteN - g_{max}* shows the iteration number which is the stopping criterion of the algorithm (maximum generation number), *NP* shows the number of the individuals in each population, *f_{Call}* shows how many purpose function calls have been made during the solution, *Run* shows how many times functions have been solved with the algorithm, *D* shows the number of the unknowns in the function

(dimension), CR crossbreeding rate and F scale factor. The values obtained by the use of DE together with 25 different chaotic mapping methods have been given in Table 4 for f_j function. The values in the Ranks column of the table respectively show the place of the all values taking place in the table (Worst, Average, Best, StdDev and Time), in 50 operations in 25 different algorithms together with DE. The values in the last line ie the order of the search intervals. Each algorithm has been scored according to its place in this ranking and the best method has been tried to be determined for each function.

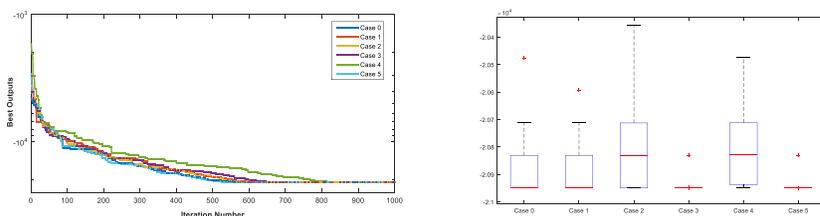
Table 4: The obtained data for f_j function (50-D, 50 run, 1000 iteration) (CbDE)

	$f_j - f_{min} = -2,094914e+04$					
	<i>Worst</i>	<i>Average</i>	<i>Best</i>	<i>StdDev</i>	<i>Time (sec)</i>	<i>Ranks</i>
Case-0	-2,047539e+04	-2,090403e+04	-2,094914e+04	8,823213e+01	0,824397	9-15-26-11-14-9 (84)
Case-1	-2,059383e+04	-2,089978e+04	-2,094914e+04	8,150544e+01	0,892212	13-14-25-12-3-13 (80)
Case-2	-2,035695e+04	-2,076911e+04	-2,094914e+04	1,539530e+02	0,856821	5-4-24-4-7-5 (49)
Case-3	-2,083068e+04	-2,092777e+04	-2,094914e+04	4,547824e+01	0,841379	22-19-23-20-10-22 (116)
Case-4	-2,047482e+04	-2,080115e+04	-2,094877e+04	1,349955e+02	0,789234	6-7-3-6-22-6 (50)
Case-5	-2,083070e+04	-2,094440e+04	-2,094914e+04	2,320877e+01	0,750286	23-24-22-24-26-23 (142)
Case-6	-2,047539e+04	-2,083997e+04	-2,094914e+04	1,028383e+02	0,775412	8-8-21-9-24-8 (78)
Case-7	-2,056004e+04	-2,079522e+04	-2,094905e+04	1,128138e+02	0,811732	10-6-4-8-19-10 (57)
Case-8	-2,071210e+04	-2,087303e+04	-2,094914e+04	8,115358e+01	0,807233	16-10-20-13-20-16 (95)
Case-9	-2,071192e+04	-2,089937e+04	-2,094914e+04	7,143284e+01	0,830424	15-13-19-14-12-15 (88)
Case-10	-2,073200e+04	-2,094006e+04	-2,094914e+04	3,770472e+01	0,850206	20-23-18-23-20-16 (112)
Case-11	-2,059383e+04	-2,091361e+04	-2,094914e+04	6,803705e+01	0,875837	12-18-17-15-4-12 (78)
Case-12	-2,082990e+04	-2,093246e+04	-2,094914e+04	4,111543e+01	0,900290	21-21-16-21-1-21 (101)
Case-13	-2,047539e+04	-2,084726e+04	-2,094914e+04	1,207938e+02	0,819528	7-9-15-7-15-7 (60)
Case-14	-1,976476e+04	-2,041880e+04	-2,094912e+04	2,844442e+02	0,796555	3-2-5-2-21-3 (36)
Case-15	-2,071227e+04	-2,091346e+04	-2,094914e+04	5,914250e+01	0,814375	19-17-14-19-17-19 (105)
Case-16	-2,083071e+04	-2,093493e+04	-2,094914e+04	3,848790e+01	0,813096	24-22-13-22-18-24 (123)

Case-17	-2,071226e+04	-2,089937e+04	-2,094914e+04	6,308867e+01	0,892342	18-12-12-16-2-18 (78)
Case-18	-1,954641e+04	-2,060338e+04	-2,094535e+04	2,237539e+02	0,759831	2-3-2-3-25-2 (37)
Case-19	-2,094912e+04	-2,094914e+04	-2,094914e+04	3,676058e-03	0,835922	26-26-11-26-11-26 (126)
Case-20	-2,070432e+04	-2,093122e+04	-2,094914e+04	6,088092e+01	0,864914	14-20-10-18-6-14 (82)
Case-21	-1,094082e+04	-1,254758e+04	-1,397933e+04	7,317948e+02	0,846612	1-1-1-1-9-1 (14)
Case-22	-2,094896e+04	-2,094914e+04	-2,094914e+04	2,648271e-02	0,778198	25-25-9-25-23-25 (132)
Case-23	-2,027799e+04	-2,078964e+04	-2,094914e+04	1,525902e+02	0,816655	4-5-8-5-16-4 (42)
Case-24	-2,071226e+04	-2,090645e+04	-2,094914e+04	6,155929e+01	0,829990	17-16-7-17-13-17 (87)
Case-25	-2,059383e+04	-2,088753e+04	-2,094914e+04	8,937995e+01	0,870619	11-11-6-10-5-11 (54)

When Table 4 is examined, it is seen that there are 8 methods being successful over 100 points for 50 operations. These are respectively Case-5, 22, 19, 16, 3, 10, 15 and 12 methods. It is predicted that the preference of the use of heuristic algorithms together with these methods in the solutions of functions having similar structures with f_1 , will better the decisiveness, performance and the speed of the algorithms.

The graphics and boxplots showing the convergence according to the iteration number belonging to the best solution values obtained for f_1 function for 50 operations in Table 4 have been given in Figure 29.



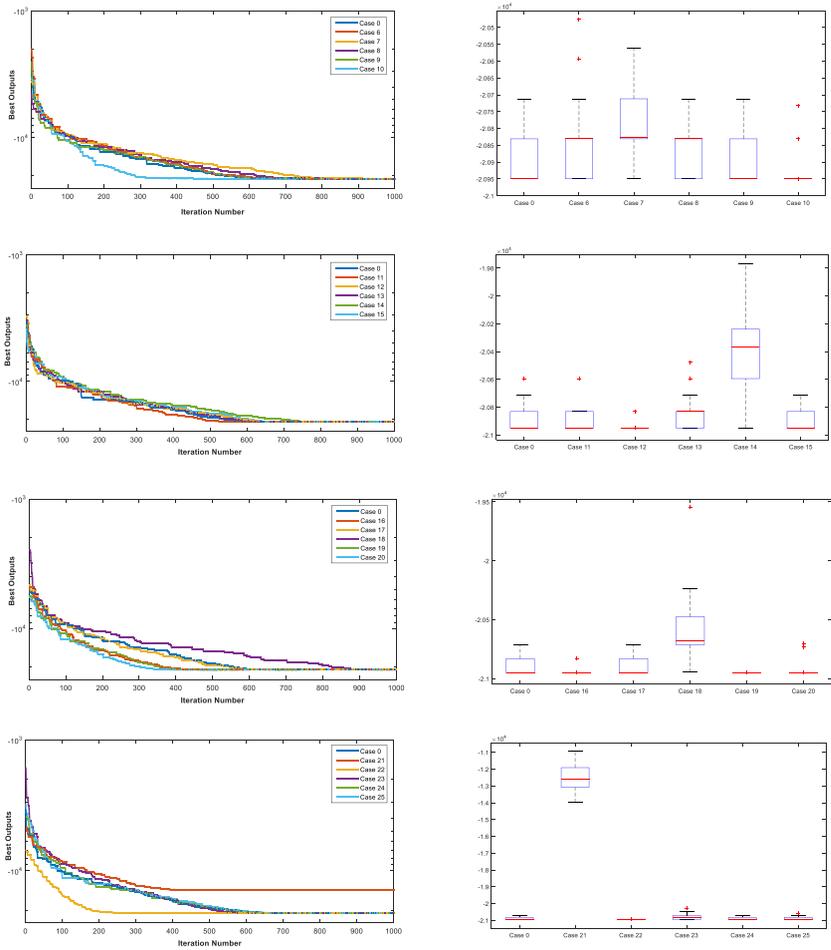


Figure 29: Boxplots and convergence curves for the best solutions obtained for f_1 (50-D)

When the graphics in Figure 29 are examined, it has been seen that for f_1 function 13 approaches out of 25 have bettered classic DE algorithm in terms of convergence and decisiveness.

The search intervals in 50 operations for f_1 function, which has been solved with 25 different chaotic mapping methods together with DE, have been given in Figure 30.

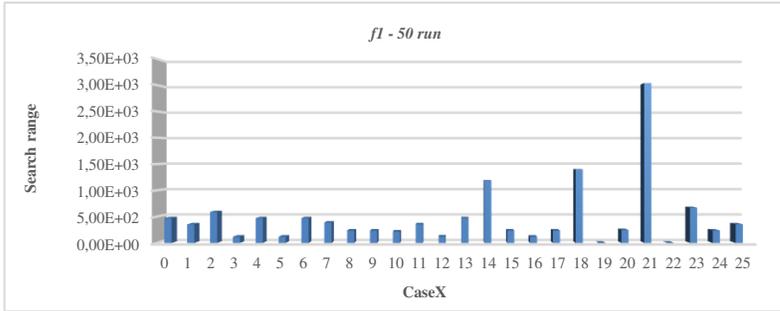


Figure 30: Search intervals for f_1 in 50 operations

The values obtained by the use of DE together with 25 different chaotic mapping methods have been given in Table 5 for f_2 function.

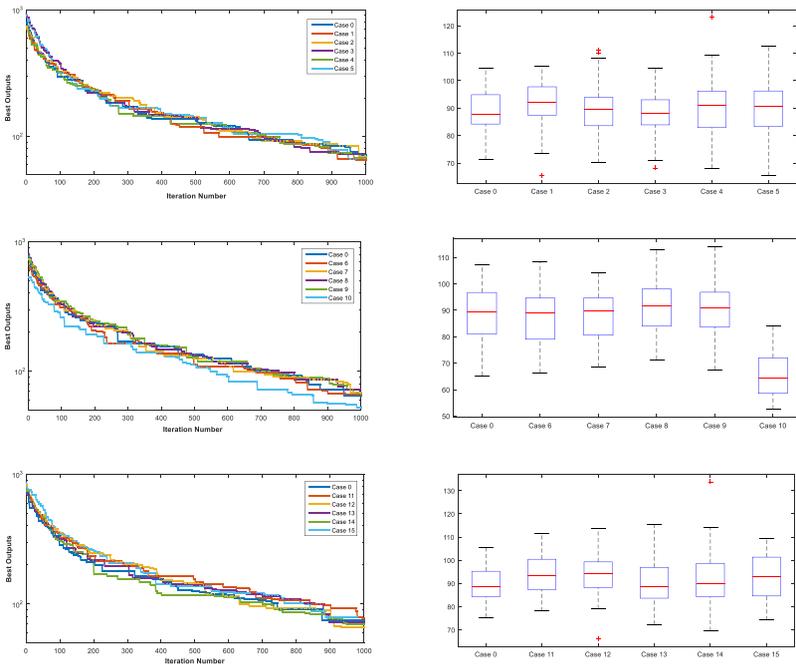
Table 5: The data obtained for f_2 function (50-D, 50 run, 1000 iteration) (CbDE)

	$f_2 - f_{min}=0,000000e+00$					
	<i>Worst</i>	<i>Average</i>	<i>Best</i>	<i>StdDev</i>	<i>Time (sec)</i>	<i>Ranks</i>
Case-0	1,046105e+02	8,849805e+01	7,130538e+01	7,506685e+00	0,86414	20-16-6-25-3-24 (94)
Case-1	1,054597e+02	9,167155e+01	6,557469e+01	8,709619e+00	0,798071	17-5-16-18-14-17 (87)
Case-2	1,112121e+02	8,914645e+01	7,007758e+01	9,285923e+00	0,821374	10-14-7-12-8-16 (67)
Case-3	1,045055e+02	8,774642e+01	6,818505e+01	7,950621e+00	0,781344	21-20-10-23-21-19 (114)
Case-4	1,230287e+02	9,002053e+01	6,809905e+01	1,118040e+01	0,815771	2-12-11-2-11-2 (40)
Case-5	1,126963e+02	9,024674e+01	6,535921e+01	9,621158e+00	0,781422	8-10-17-7-20-8 (70)
Case-6	1,085142e+02	8,800510e+01	6,620539e+01	9,875586e+00	0,741815	14-18-13-5-25-14 (89)
Case-7	1,041599e+02	8,824374e+01	6,850582e+01	8,810363e+00	0,888262	22-17-9-16-1-20 (85)
Case-8	1,129402e+02	9,146530e+01	7,138321e+01	8,690211e+00	0,735012	7-7-5-19-26-15 (79)
Case-9	1,141838e+02	9,009767e+01	6,742605e+01	9,707779e+00	0,847263	4-11-12-6-6-9 (48)
Case-10	8,425323e+01	6,537017e+01	5,248881e+01	8,140923e+00	0,817882	24-24-23-22-9-25 (127)
Case-11	1,114649e+02	9,326647e+01	7,806608e+01	8,532063e+00	0,824025	9-2-1-21-7-23 (63)
Case-12	1,136269e+02	9,376729e+01	6,612642e+01	9,127218e+00	0,880076	5-1-15-15-2-7 (45)
Case-13	1,154652e+02	9,064896e+01	7,236487e+01	1,019680e+01	0,760033	3-9-4-3-23-13 (55)
Case-14	1,338090e+02	9,123208e+01	6,959362e+01	1,229041e+01	0,786496	1-8-8-1-19-1 (38)
Case-15	1,095159e+02	9,251295e+01	7,419990e+01	9,546641e+00	0,790318	12-3-3-8-16-21 (63)
Case-16	1,070045e+02	8,426263e+01	5,763838e+01	9,359999e+00	0,743736	16-22-21-11-21-5 (99)
Case-17	1,095881e+02	8,932776e+01	6,518855e+01	9,510218e+00	0,787217	11-13-18-10-18-12 (82)

Case-18	1,091166e+02	9,150695e+01	7,526248e+01	7,885758e+00	0,817301	13-6-2-24-10-22 (77)
Case-19	1,081757e+02	9,194753e+01	6,324324e+01	9,196819e+00	0,851471	15-4-20-14-5-11 (69)
Case-20	1,135603e+02	8,851684e+01	6,397449e+01	9,214390e+00	0,76457	6-15-19-13-22-4 (79)
Case-21	3,087574e+01	2,578507e+01	1,907214e+01	3,085763e+00	0,78787	26-26-26-26-17-26 (147)
Case-22	7,760718e+01	5,710193e+01	3,167561e+01	1,007055e+01	0,799214	25-25-25-4-13-10 (102)
Case-23	9,059803e+01	6,672405e+01	3,851704e+01	9,516328e+00	0,790761	23-23-24-9-15-3 (97)
Case-24	1,053134e+02	8,582185e+01	5,703032e+01	8,790628e+00	0,86273	18-21-22-17-4-6 (88)
Case-25	1,051424e+02	8,776532e+01	6,617781e+01	8,560226e+00	0,799248	19-19-14-20-12-18 (102)

When the values in Table 5 are examined, there are 4 methods which achieved a success over 100 points for 50 runs. These are respectively Case-10, 3, 22 and 25 methods.

The graphics and boxplots showing the convergence according to the iteration number belonging to the best solution values obtained for f_2 function for 50 operations in Table 5 have been given in Figure 31.



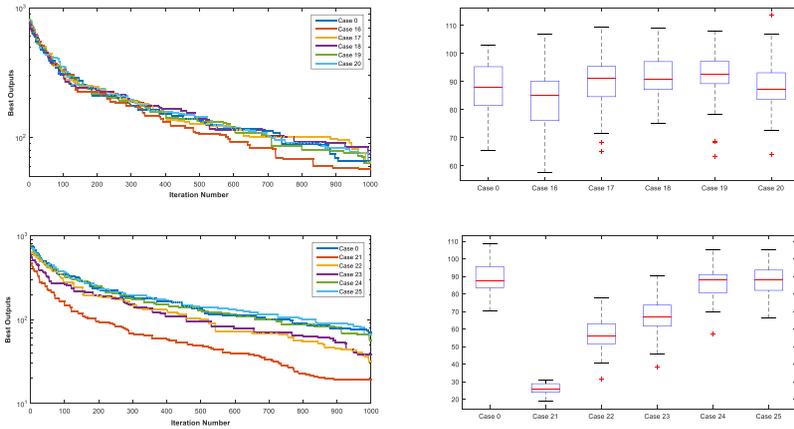


Figure 31: Boxplots and convergence curves for the best solutions obtained for f_2 (50-D)

When the graphics in Figure 31 are examined, it has been seen that many approaches have the same convergence and decisiveness data with DE algorithm for f_2 function. It can be said that Case 10, 16, 21 and 23 methods provide convergence a little faster.

The search intervals in 50 operations for f_2 function, which has been solved with 25 different chaotic mapping methods together with DE, have been given in Figure 32.

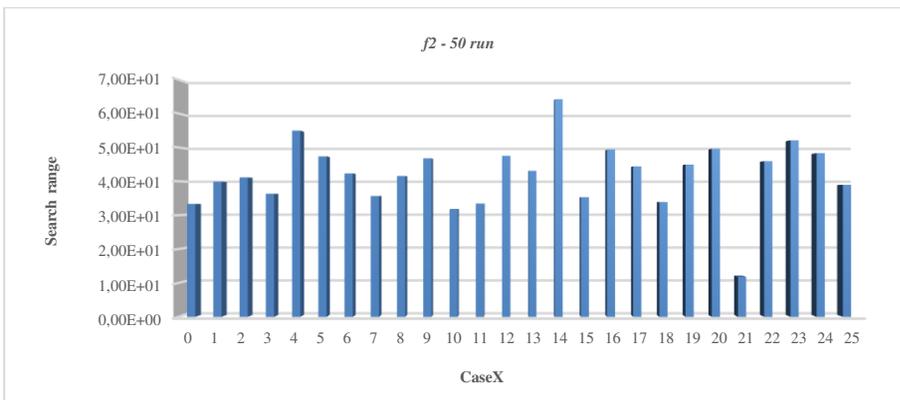


Figure 32: Search intervals for f_2 in 50 operations/runs

The values obtained by the use of DE together with 25 different chaotic mapping methods have been given in Table 6 for f_3 function.

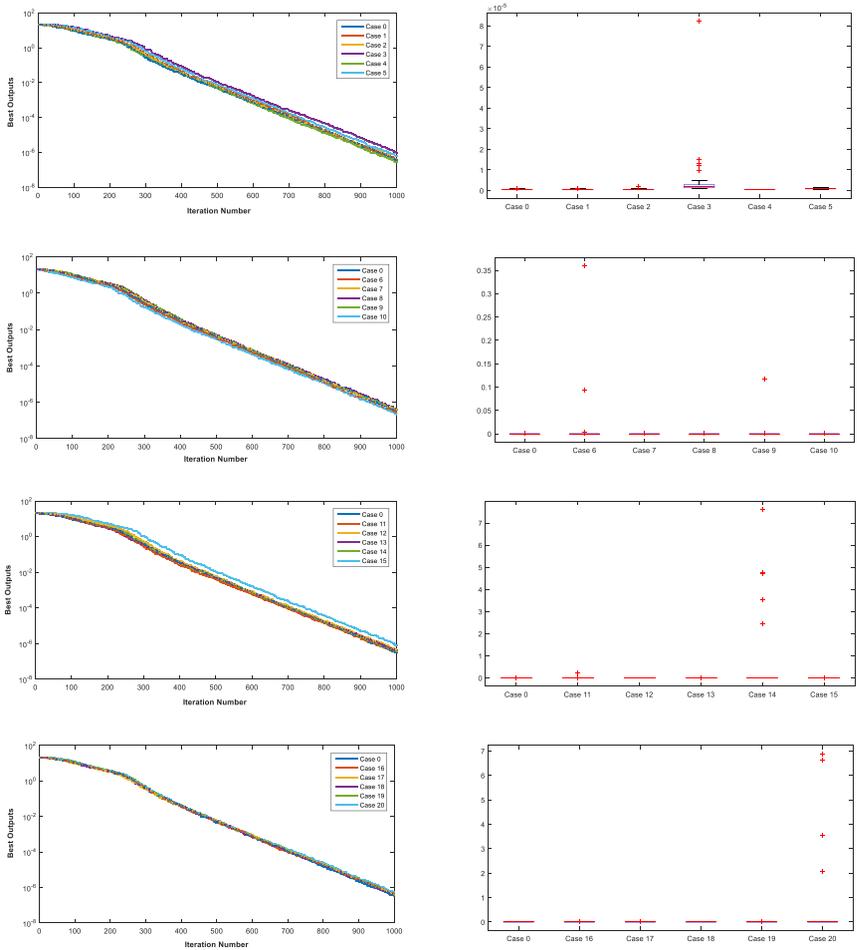
Table 6: The data obtained for f_3 function (50-D, 50 run, 1000 iteration) (CbDE)

	$f_3 - f_{min}=0,000000e+00$					
	<i>Worst</i>	<i>Average</i>	<i>Best</i>	<i>StdDev</i>	<i>Time (sec)</i>	<i>Ranks</i>
Case-0	9,810398e-07	5,562393e-07	3,612261e-07	1,182002e-07	0,933618	22-21-16-22-16-22 (119)
Case-1	8,920020e-07	5,559920e-07	3,918666e-07	1,102110e-07	0,927393	23-22-9-23-18-23 (118)
Case-2	1,807944e-06	5,843559e-07	3,915176e-07	1,996851e-07	0,900927	16-19-10-17-21-16 (99)
Case-3	8,229542e-05	4,404963e-06	9,259227e-07	1,153600e-05	1,000670	12-11-2-12-11-13 (61)
Case-4	6,593357e-07	4,497772e-07	2,685979e-07	9,102288e-08	1,135830	26-25-22-26-4-26 (129)
Case-5	1,378321e-06	9,430139e-07	6,213864e-07	1,682653e-07	1,142280	19-15-4-18-3-19 (78)
Case-6	3,599374e-01	9,128487e-03	2,615778e-07	5,180164e-02	0,937726	4-4-24-4-15-4 (55)
Case-7	8,224209e-05	2,197604e-06	3,763704e-07	1,143550e-05	1,077600	13-13-14-13-6-12 (71)
Case-8	7,905301e-06	7,323987e-07	4,046082e-07	1,029627e-06	0,893053	15-17-7-15-24-15 (93)
Case-9	1,168444e-01	2,337460e-03	3,857343e-07	1,635814e-02	0,974208	6-6-12-6-13-6 (49)
Case-10	1,398376e-06	3,352279e-07	2,103437e-07	1,632564e-07	0,901545	18-26-25-19-20-17 (125)
Case-11	2,432376e-01	4,890803e-03	3,162234e-07	3,405001e-02	1,168960	5-5-21-5-2-5 (43)
Case-12	1,584532e-06	9,305525e-07	4,630671e-07	2,348983e-07	1,017650	17-16-6-16-9-18 (82)
Case-13	8,870334e-05	2,305636e-06	3,945551e-07	1,234293e-05	1,083900	11-12-8-11-5-11 (58)
Case-14	7,600402e+00	4,607137e-01	3,480475e-07	1,485410e+00	0,874004	2-2-18-1-25-1 (49)
Case-15	2,264272e-03	4,742843e-05	7,834454e-07	3,166939e-04	1,010400	8-8-3-8-10-8 (45)
Case-16	3,087264e-05	1,198600e-06	3,575508e-07	4,244943e-06	0,896578	14-14-17-14-23-14 (96)
Case-17	1,025341e-06	5,685757e-07	3,914016e-07	1,274241e-07	1,253320	20-20-11-21-1-21 (94)
Case-18	7,814135e-07	5,398190e-07	3,203070e-07	1,080282e-07	0,940851	24-23-20-24-14-24 (129)
Case-19	1,023968e-06	6,321919e-07	3,773755e-07	1,314528e-07	0,927572	21-18-13-20-17-20 (109)
Case-20	6,887188e+00	3,820636e-01	5,173518e-07	1,418960e+00	0,899101	3-3-5-2-22-2 (37)
Case-21	8,182616e+00	6,200140e+00	4,373391e+00	9,315462e-01	1,048280	1-1-1-3-7-3 (16)
Case-22	6,302751e-02	1,264939e-03	2,662502e-07	8,823263e-03	0,913236	7-7-23-7-19-7 (70)
Case-23	2,221129e-04	4,739602e-06	1,853627e-07	3,105340e-05	1,020770	10-10-26-10-8-10 (74)
Case-24	7,414639e-07	5,319493e-07	3,474004e-07	9,451327e-08	0,860323	25-24-19-25-26-25 (144)
Case-25	9,254590e-04	1,906288e-05	3,645896e-07	1,294852e-04	0,983024	9-9-15-9-12-9 (63)

When Table 6 which has been created according to the best solution values obtained for f_3 function in 50 runs, is examined, it has been seen that there are 7 approaches that achieve success over 100 points.

These approaches are respectively Case 24, 4, 18, 0, 1 and 19 approaches. DE algorithm has also achieved a success over 100 points for this function.

The graphics and boxplots showing the convergence according to the iteration number belonging to the best solution values obtained for f_3 function for 50 operations in Table 6 have been given in Figure 33.



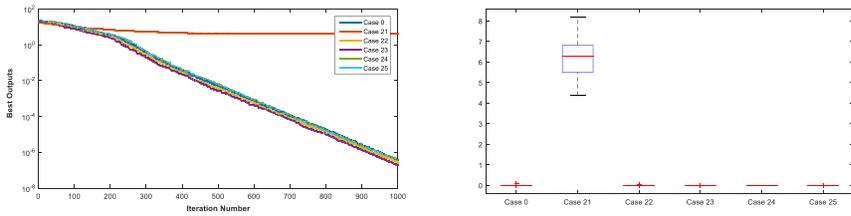


Figure 33: Boxplots and convergence curves for the best solutions obtained for f_3 (50-D)

When the graphics in Figure 33 are examined, it is seen that most of the approaches have been successful in terms of convergence and decisiveness. Only in Case 3, 6, 9, 14, 20 and 21 methods are encountered some diverging values. It can be commented that the decisiveness of these methods is a little bad for this function.

The search intervals in 50 operations for f_3 function, which has been solved with 25 different chaotic mapping methods together with DE, have been given in Figure 34.

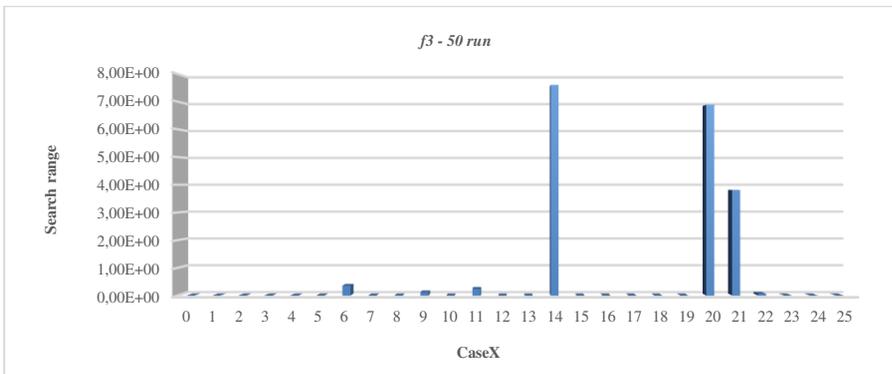


Figure 34: Search intervals for f_3 in 50 operations/runs

The values obtained by the use of DE together with 25 different chaotic mapping methods have been given in Table 7 for f_4 function.

Table 7: The data obtained for f_4 function (50-D, 50 run, 1000 iteration) (CbDE)

	$f_4 - f_{min}=0,000000e+00$					
	<i>Worst</i>	<i>Average</i>	<i>Best</i>	<i>StdDev</i>	<i>Time (sec)</i>	<i>Ranks</i>
Case-0	1,309977e-01	2,646913e-03	7,476908e-12	1,833680e-02	1,470410	6-6-12-6-16-6 (52)
Case-1	4,912803e-05	9,825873e-07	3,362977e-12	6,877920e-06	1,630640	9-9-24-9-4-9 (64)
Case-2	9,513867e-11	2,421124e-11	7,707612e-12	1,694552e-11	1,337540	26-25-11-26-26-26 (140)
Case-3	2,154351e-10	4,461181e-11	1,402001e-11	3,265902e-11	1,600640	23-22-2-23-5-23 (98)
Case-4	6,832222e-08	1,414611e-09	6,857404e-12	9,560062e-09	1,593990	17-17-18-17-6-17 (92)
Case-5	1,689736e-07	3,473977e-09	1,027811e-11	2,364519e-08	1,452920	15-15-4-15-18-15 (82)
Case-6	1,049520e-07	2,117416e-09	3,705591e-12	1,469066e-08	1,469560	16-16-23-16-17-16 (104)
Case-7	4,212455e-06	1,129201e-07	9,341972e-12	6,189780e-07	1,438160	11-11-7-11-19-11 (70)
Case-8	9,225016e-07	3,107653e-08	8,827161e-12	1,545546e-07	1,819070	13-13-9-13-1-13 (62)
Case-9	6,122365e-06	1,224798e-07	9,861223e-12	8,571264e-07	1,395110	10-10-6-10-23-10 (69)
Case-10	1,513580e-10	1,158892e-11	2,163492e-12	2,090792e-11	1,630680	24-26-26-25-3-24 (128)
Case-11	4,223005e-08	8,721149e-10	6,487255e-12	5,908379e-09	1,406940	18-18-19-18-22-18 (113)
Case-12	9,977682e-10	6,574534e-11	1,176137e-11	1,444340e-10	1,532700	19-19-3-19-11-19 (90)
Case-13	4,006289e-07	8,047301e-09	4,594880e-12	5,608312e-08	1,472790	14-14-21-14-15-14 (92)
Case-14	8,348084e+01	3,066448e+00	7,031820e-12	1,425759e+01	1,530000	1-2-17-1-12-1 (34)
Case-15	6,006287e-10	6,420108e-11	1,010492e-11	9,874025e-11	1,585750	21-20-5-20-7-21 (94)
Case-16	2,274661e+00	4,874399e-02	7,439938e-12	3,183971e-01	1,423260	5-5-13-5-21-5 (54)
Case-17	1,079932e-01	2,159865e-03	7,213230e-12	1,511905e-02	1,574360	7-7-14-7-9-7 (51)
Case-18	1,648270e-06	3,526949e-08	7,118639e-12	2,309610e-07	1,548360	12-12-16-12-0-12 (74)
Case-19	2,626747e-10	3,814226e-11	8,665180e-12	4,849315e-11	1,489050	22-23-10-22-14-22 (113)
Case-20	1,598232e+01	1,126848e+00	9,103385e-12	3,663686e+00	1,510060	3-3-8-2-13-2 (31)
Case-21	1,681683e+01	1,088380e+01	3,076413e+00	3,103759e+00	1,367320	2-1-1-3-25-3 (35)
Case-22	2,315748e+00	4,989260e-02	4,408141e-12	3,246524e-01	1,652970	4-4-22-4-2-4 (40)
Case-23	1,722629e-02	3,445259e-04	2,206346e-12	2,411681e-03	1,425680	8-8-25-8-20-8 (77)
Case-24	1,463665e-10	2,942576e-11	4,991563e-12	2,771935e-11	1,393830	25-24-20-24-24-25 (142)
Case-25	6,872746e-10	4,834058e-11	7,170375e-12	9,661373e-11	1,578790	20-21-15-21-8-20 (105)

When Table 7 is examined, it has been seen that there are 7 approaches that achieve success over 100 points. DE algorithm has gathered 52 points while only 5 approaches out of 25 have gathered less points than that. It has been seen that the rest 25 approaches have contributed to DE algorithm in a positive way.

The graphics and boxplots showing the convergence according to the iteration number belonging to the best solution values obtained for f_4 function for 50 operations in Table 7 have been given in Figure 35.

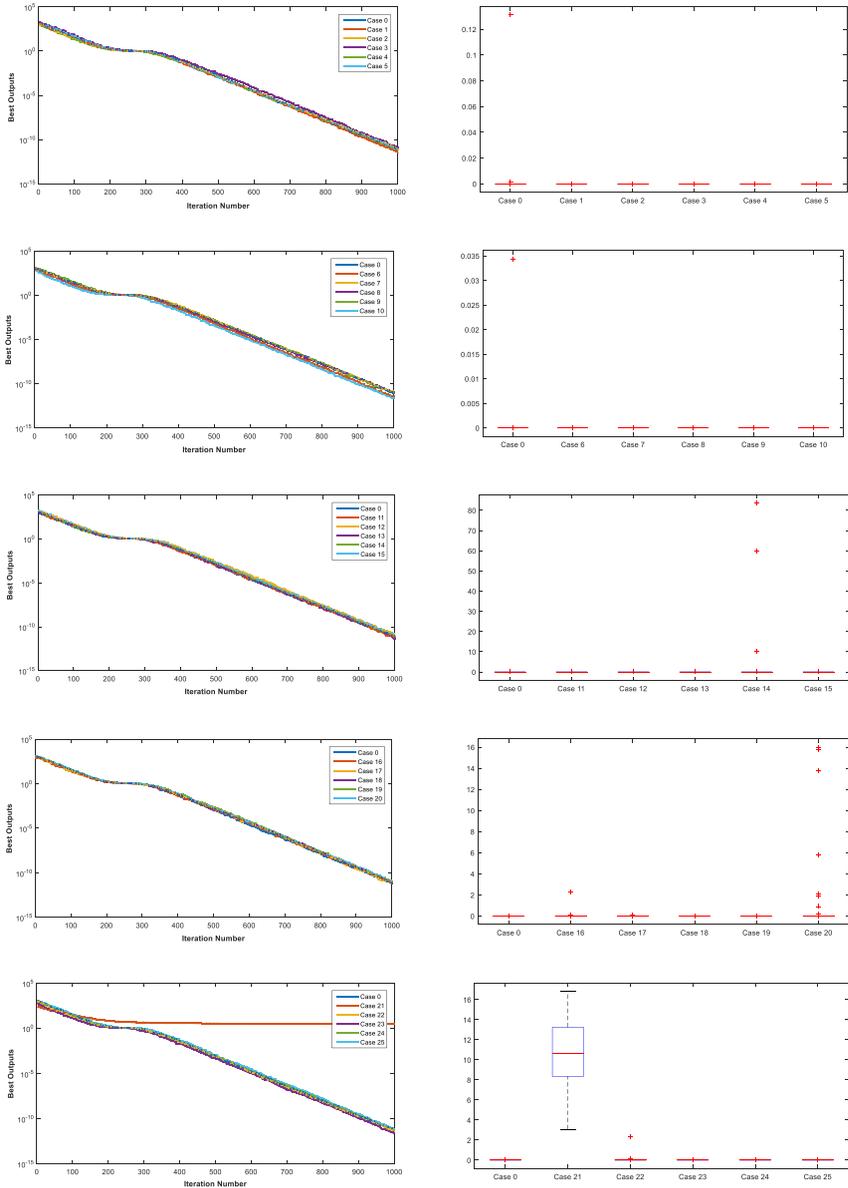


Figure 35: Boxplots and convergence curves for the best solutions obtained for f_4 (50-D)

When Figure 35 is examined, diverging values have been encountered in Case 0, 14 and 20 approaches. As for Case 21 approach, it has not been able to converge for the function with this structure.

The search intervals in 50 operations for f_4 function, which has been solved with 25 different chaotic mapping methods together with DE, have been given in Figure 36.

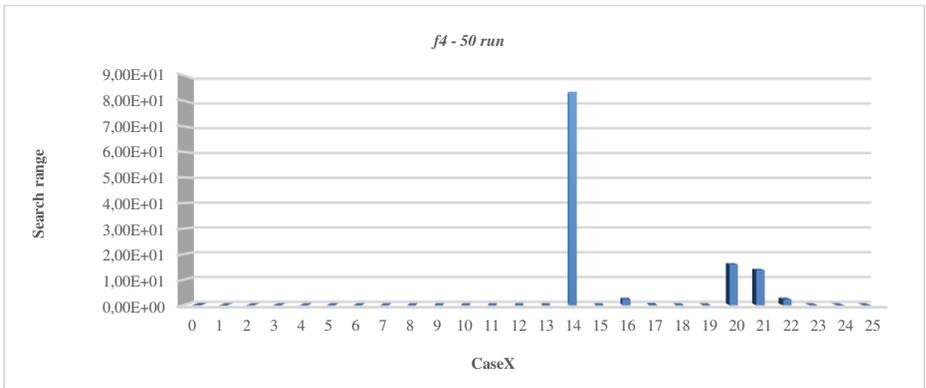


Figure 36: Search intervals for f_4 in 50 operations/runs

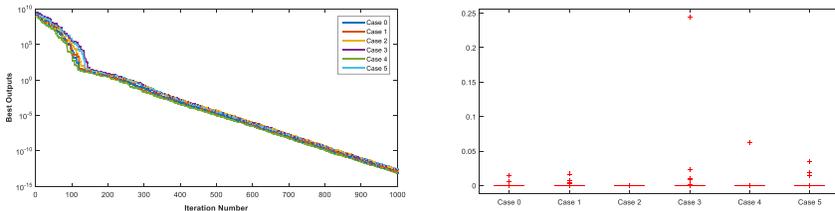
The values obtained by the use of DE together with 25 different chaotic mapping methods have been given in Table 8 for f_5 function.

Table 8: The data obtained for f_5 function (50-D, 50 run, 1000 iteration) (CbDE)

	$f_5 - f_{min}=0,000000e+00$					
	<i>Worst</i>	<i>Average</i>	<i>Best</i>	<i>StdDev</i>	<i>Time (sec)</i>	<i>Ranks</i>
Case-0	1,411975e-02	3,899035e-04	7,992672e-14	2,089053e-03	2,02988	20-21-22-21-1-20 (105)
Case-1	1,663889e-02	6,406688e-04	8,583680e-14	2,597067e-03	1,34930	19-18-20-19-16-19 (111)
Case-2	2,625902e-04	5,252218e-06	1,483199e-13	3,676257e-05	1,64310	24-24-12-24-2-24 (110)
Case-3	2,440827e-01	5,748431e-03	2,001754e-13	3,424332e-02	1,46605	6-8-7-6-6-6 (39)
Case-4	6,220141e-02	1,246512e-03	8,171266e-14	8,707860e-03	1,61849	12-13-21-12-3-11 (72)
Case-5	3,428255e-02	1,353507e-03	2,551863e-13	5,731778e-03	1,49758	13-12-4-13-5-13 (60)
Case-6	1,518494e-04	3,036988e-06	6,507838e-14	2,125892e-05	1,22701	25-25-23-25-22-25 (145)

Case-7	6,220141e-02	1,889497e-03	1,328261e-13	9,029336e-03	1,36773	11-11-14-11-14-12 (73)
Case-8	1,328706e-02	5,005344e-04	1,497552e-13	2,412769e-03	1,22680	21-19-11-20-23-21 (115)
Case-9	3,119660e-02	1,212997e-03	1,607098e-13	4,803303e-03	1,21163	14-14-9-15-25-14 (91)
Case-10	1,780034e-01	8,222054e-03	3,625083e-14	3,269172e-02	1,40031	8-6-25-7-10-8 (64)
Case-11	1,429263e-01	3,003772e-03	1,217306e-13	2,000526e-02	1,38621	9-9-16-9-11-9 (63)
Case-12	2,015177e-01	7,659925e-03	2,833534e-13	3,083587e-02	1,29720	7-7-3-8-19-7 (51)
Case-13	2,877329e-02	1,190811e-03	1,034880e-13	4,981774e-03	1,38341	15-15-18-14-12-15 (89)
Case-14	9,091757e+07	2,534953e+06	1,581040e-13	1,333314e+07	1,31078	1-1-10-1-17-1 (31)
Case-15	2,540074e-02	9,285151e-04	2,454552e-13	3,991084e-03	1,41303	16-16-5-16-8-16 (77)
Case-16	5,421064e+03	1,084759e+02	2,057004e-13	7,589413e+02	1,30910	3-3-6-3-18-3 (36)
Case-17	1,160362e-03	2,330925e-05	9,131824e-14	1,624376e-04	1,25045	23-23-19-23-20-23 (131)
Case-18	1,676256e-07	3,352824e-09	5,492238e-14	2,346754e-08	1,40608	26-26-24-26-9-26 (137)
Case-19	1,180766e-02	2,361608e-04	1,410968e-13	1,653072e-03	1,35390	22-22-13-22-15-22 (116)
Case-20	1,277374e+06	5,591710e+04	3,158748e-13	2,406205e+05	1,58668	2-2-2-2-4-2 (14)
Case-21	3,421366e+01	6,580415e+00	1,868253e+00	7,872075e+00	1,37389	4-4-1-4-13-4 (30)
Case-22	5,624480e-01	6,344862e-02	1,144759e-13	1,238278e-01	1,16911	5-5-17-5-26-5 (63)
Case-23	1,956572e-02	3,913144e-04	2,449494e-14	2,739201e-03	1,44482	18-20-26-18-7-18 (107)
Case-24	2,444936e-02	6,689852e-04	1,321672e-13	3,623237e-03	1,23162	17-17-15-17-21-17 (104)
Case-25	6,850046e-02	2,741248e-03	1,891332e-13	1,282658e-02	1,21275	10-10-8-10-24-10 (72)

When Table 8 is examined, it has been seen that there are 9 approaches including DE that achieve success over 100 points. While DE algorithm has gathered 105 points, out of other 25 approaches Case-18 has gathered the most point with 137 points. The graphics and box charts showing the convergence according to the iteration number belonging to the best solution values obtained for f_4 function for 50 operations in Table 8 have been given in Figure 37.



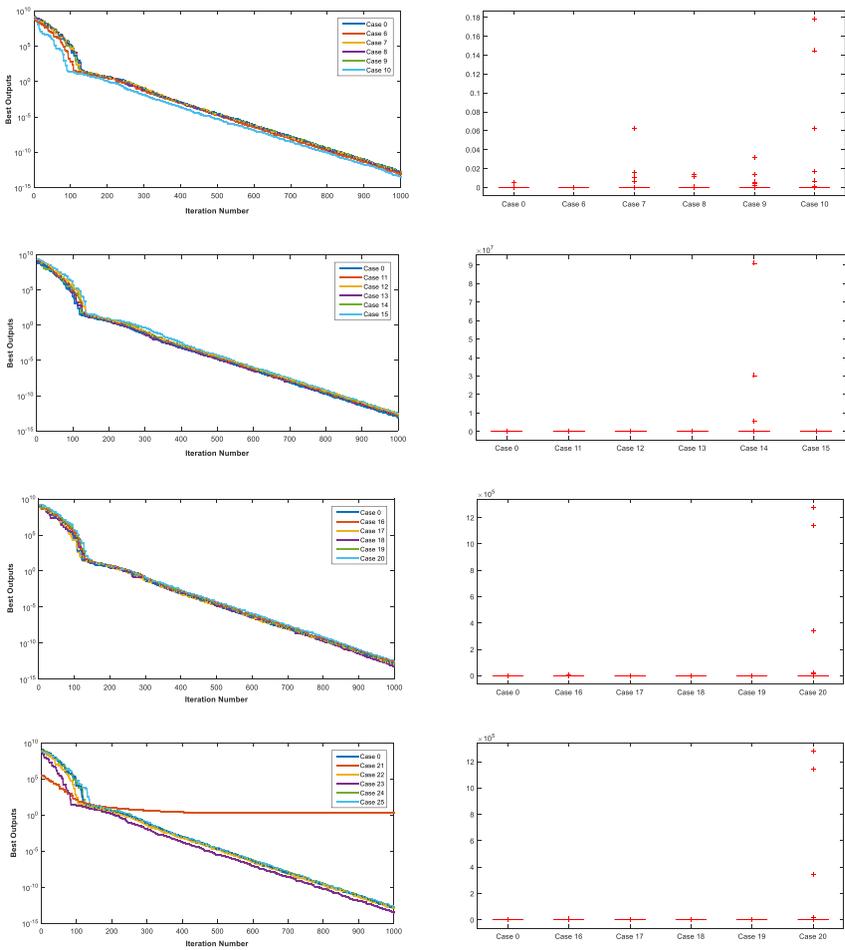


Figure 37: Box charts and convergence curves for the best solutions obtained for f_5 (50-D)

The search intervals in 50 operations for f_5 function, which has been solved with 25 different chaotic mapping methods together with DE, have been given in Figure 38.

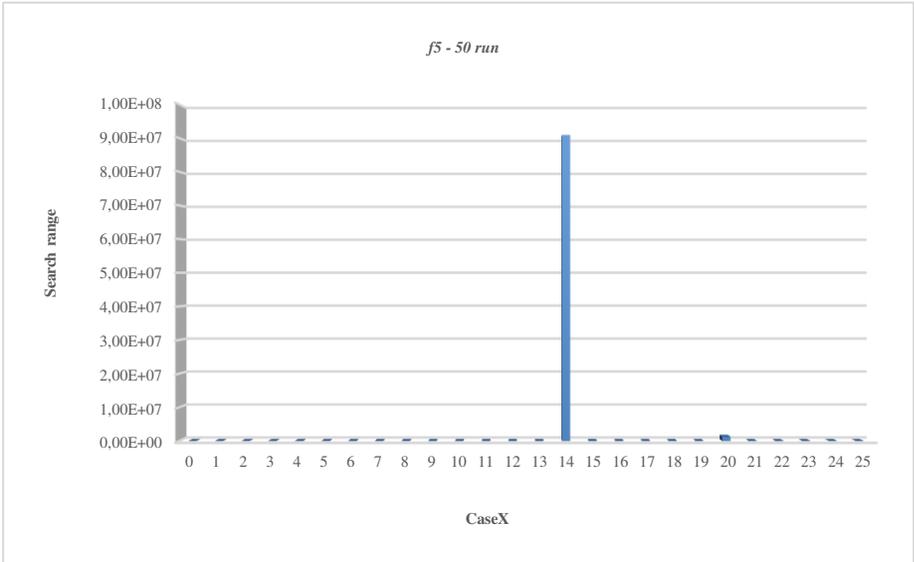


Figure 38: Search intervals for f_5 in 50 operations/runs

After all these evaluations, the most successful first ten methods defined for each function and the locations of classic DE algorithm have been given in Table 9.

Table 9: The order of the best methods

	f_1	f_2	f_3	f_4	f_5	f_1-f_5
1	Case-05	Case-21	Case-24	Case-24	Case-06	Case-24
2	Case-22	Case-10	Case-04	Case-02	Case-18	Case-10
3	Case-19	Case-03	Case-18	Case-10	Case-17	Case-19
4	Case-16	Case-22	Case-10	Case-11	Case-19	Case-06
5	Case-03	Case-25	Case-00	Case-19	Case-08	Case-02
6	Case-10	Case-16	Case-01	Case-25	Case-01	Case-01
7	Case-15	Case-23	Case-19	Case-06	Case-02	Case-00
8	Case-12	Case-00	Case-02	Case-03	Case-23	Case-18
9	Case-08	Case-06	Case-16	Case-15	Case-00	Case-08
10	Case-09	Case-24	Case-17	Case-04	Case-24	Case-17

5. STABILITY ANALYSIS

In literature Wilcoxon test is applied in order to compare these kinds of studies statistically (García, Molina, Lozano, & Herrera, 2009; Wilcoxon, Katti, & Wilcox, 1970). When the number of the data is a few, more sensitive and accurate results are obtained by applying non-parametric Wilcoxon test. Therefore, DE belonging to 5 test functions and the best values belonging to CbDE-1, CbDE-2, CbDE-3, ... , CbDE-25 algorithms in 50 runs have been subjected to Wilcoxon statistical evaluation tests defined in literature and the obtained results have been given in Table 10. Significance level in the data analysis has been taken as $p < 0.05$ as in literature.

Table 10: Wilcoxon test results of the best values belonging to f_1 - f_5 functions obtained for 50 runs and 50-D

	Wilcoxon (<i>p-values</i>)				
	f_1	f_2	f_3	f_4	f_5
DE - CbDE-1	0.9795	0.0236	0.7758	0.1284	0.5988
DE - CbDE-2	3.5838e-05	0.6123	0.4901	0.0349	0.2994
DE - CbDE-3	0.6912	0.4258	7.5569e-10	0.1602	0.0025
DE - CbDE-4	8.7266e-07	0.3667	6.0478e-05	0.0618	0.0015
DE - CbDE-5	1.0599e-04	0.4258	1.0235e-09	0.1358	0.0054
DE - CbDE-6	1.7088e-04	0.7537	3.6125e-04	0.0042	2.1162e-05
DE - CbDE-7	2.3493e-07	0.9730	0.6055	0.1358	0.7102
DE - CbDE-8	0.0056	0.0688	0.0781	0.8734	0.5592
DE - CbDE-9	0.6760	0.3876	0.1038	0.4258	0.7611
DE - CbDE-10	5.4422e-05	8.0311e-10	1.9841e-08	1.0163e-07	0.0093
DE - CbDE-11	0.1073	0.0122	0.8056	0.0978	0.7246
DE - CbDE-12	0.2887	0.0017	1.3831e-09	0.5399	0.0012
DE - CbDE-13	0.0224	0.3320	0.8887	0.1781	0.3131
DE - CbDE-14	2.2824e-09	0.2184	0.1630	0.4901	0.0393
DE - CbDE-15	0.6594	0.0393	7.5569e-10	0.1101	2.7856e-04

DE - CbDE-16	0.0011	0.0341	0.1781	0.1059	0.0014
DE - CbDE-17	0.7110	0.6191	0.4371	0.2567	0.2370
DE - CbDE-18	6.3812e-09	0.0441	0.5988	0.2370	0.0025
DE - CbDE-19	5.3749e-06	0.0261	0.0054	0.6887	0.7246
DE - CbDE-20	0.0072	0.7611	1.9094e-07	0.0032	4.5085e-09
DE - CbDE-21	7.5518e-10	7.5569e-10	7.5569e-10	7.5569e-10	7.5569e-10
DE - CbDE-22	1.1919e-05	7.5569e-10	8.7755e-06	0.2487	4.2959e-08
DE - CbDE-23	2.0040e-04	9.6344e-10	1.9841e-08	2.6489e-06	1.5860e-08
DE - CbDE-24	0.7264	0.1750	0.4840	0.1545	0.7907
DE - CbDE-25	0.0947	0.8963	0.3466	0.7684	0.2567

According to the p values given in Table 10, it has been seen that CbDE-X algorithms developed in the study have significant differences from DE algorithm for different test functions.

6. ASSESSMENT

In this study CbDE-X algorithms have been developed by integrating 25 different chaotic mapping methods into classic DE algorithm. In these approaches in the formation of populations while the first individual is assigned randomly, the other individuals are assigned depending on this first individual according to a certain rule. In all approaches 5 multidimensional multimodal test functions have been solved. As a result of these solutions, various improvements have been provided in the classic DE algorithm with CbDE-X algorithms. While doing this study, two aims have been considered. The first of these aims is to improve a selected optimization algorithm with chaotic structures and to obtain new faster and more decisive algorithms. As for the second aim, it is to determine the best method for chaotic mapping methods which are used frequently in many studies in

literature. After this study; researchers, who will study with chaotic structures, will be able to decide easily which method is more suitable according to the structure of the problem that they will solve. Some other optimization algorithms in literature can be bettered by using together with these approaches improved here. In our following studies, we are thinking of applying the algorithm that we have improved to various engineering problems.

REFERENCES

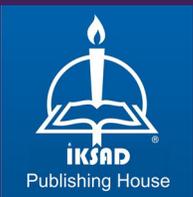
- Abottes, V., Liceaga, D., Kiryanov, A., & Wilson, M., (2016). Ikeda Map and phase conjugated ring resonator chaotic dynamics, *Applied Mathematics & Information Sciences*, 10 (6), 2071-76.
- Alataş, B., & Akın, E., (2009). Chaotically encoded particle swarm optimization algorithm and its applications. *Chaos, Solitons and Fractals*, 41 (2), 939-50.
- Alataş, B., (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, 37 (8), 5682-87.
- Alataş, B., (2016). Chaotic harmony search algorithms. *Applied Mathematics and Computation*, 216 (9), 2687-99.
- Alataş, B., Akın, E., & Bedri, O., (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons and Fractals*, 40 (4), 1715-34.
- Al-shameri, W.F.A., & Mahiub, M.A., (2013). Some dynamical properties of the family of Tent Maps. *International Journal of Mathematical Analysis*, 7 (29), 1433-49.
- Arnold, V.I., & Avez, A., (1968). Ergodic problems in classical mechanics, 1st edition, *Benjamin*, New York.
- Botella-Soler, V., Castelo, J.M., Oteo, J.A., & Ros, J., (2011). Bifurcations in the Lozi map. *Journal of Physics A: Mathematical and Theoretical*, 44 (30), 1-14.
- Bucolo, M., Caponetto, R., Fortuna, L., Frasca, M., & Rizzo, A., (2002). Does chaos work better than noise?. *IEEE Circuits and Systems Magazine*, 2 (3), 4-19.
- Campos-Canton, I., Campos-Canton, E., Murguia, J.S., & Rosu, H.C., (2009). A simple electronic circuit realization of the tent map. *Chaos, Solitons and Fractals*, 42 (1), 12-6.
- Caponetto, R., Fortuna, L., Fazzino, S., & Xibilia, M.G., (2003). Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7 (3), 289-304.

- Chaoshun, L., Jianzhong, Z., Jian, X., & Han, X., (2012). Parameters identification of chaotic system by chaotic gravitational search algorithm. *Chaos, Solitons and Fractals*, 45 (4), 539-47.
- Chirikov, B.V., (1979). A universal instability of many-dimensional oscillator systems. *Physics Reports*, 52 (5), 263-379.
- Durmuş, B., Özyön, S., & Temurtaş, H., (2016). Gravitational search algorithm with chaotic map (GSA-CM) dor solving optimization problems. *International Journal of Research in Engineering and Technology*, 5 (2), 204-12.
- Gandomi, A.H., & Yang, X.S., (2014). Chaotic bat algorithm. *Journal of Computational Science*, 5 (2), 224-32.
- García, S., Molina, D., Lozano, M., & Herrera, F., (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15, 617-44.
- Ge, F., Tan, L., & Wang, Y., (2009). Traffic modeling with Bernoulli Shift Map, *Fifth International Joint Conference on INC, IMS and IDC*, Seoul, 449-52.
- Google, (2019a). http://www.cmp.caltech.edu/~mcc/Chaos_Co/Lesson3/Oscillators.pdf
- Google, (2019b). https://en.wikipedia.org/wiki/Duffing_map
- Google, (2019c). https://en.wikipedia.org/wiki/Kaplan-Yorke_map
- Henon, M., (1976). A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, 50, 69-77.
- Hilborn, R.C., (2004). Chaos and nonlinear dynamics: An introduction for scientists and engineers. 2nd Edition, *Oxford Univ. Press*, New York.
- Ikeda, K., (1979). Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Optics Communications*, 30 (2), 257-61.
- Ikeda, K., Daido, H., & Akimoto, O., (1980), Optical turbulence: Chaotic behavior of transmitted light from a ring cavity. *Physical Review Letters*, 45 (9), 709-12.

- Jessa, M., (2006). Designing security for number sequences generated by means of the Sawtooth Chaotic Map. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53 (5), 1140-50.
- Jordehi, A.R., (2015). Chaotic bat swarm optimisation (CBSO). *Applied Soft Computing*, 26, 523-30.
- Kaplan, J.L., & Yorke, J.A., (1979). Chaotic behavior of multidimensional difference equations. *Lecture Notes in Mathematics*, 730, 204-27.
- Li, X., & Yin, M., (2012). Application of differential evolution algorithm on self-potential data. *PLoS ONE*, 7 (12), 1-11.
- Lievbovitch, S.L., & Thot, T.I., (1991). A model of ion channel kinetics using deterministic chaotic rather than stochastic processes. *Journal of Theoretical Biology*, 148, 243-67.
- Lu, H., Wang, X., Fei, Z., & Qiu, M., (2014). The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms. *Mathematical Problems in Engineering*, 1-16.
- Mahdi, A., Jawad, A.K., & Hreshee, S.S., (2016). Digital chaotic scrambling of voice based on Duffing Map. *International Journal of Information and Communication Sciences*, 1 (2), 16-21.
- Mandal, D., Chatterjee, A., & Bhattacharjee, A.K., (2013). Design of fully digital controlled shaped beam synthesis using differential evolution algorithm. *International Journal of Antennas and Propagation*, 4, 1-9.
- May, R.M., (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261, 459-67.
- Mitic, M., Vukovic, N., Petrovic, M., & Miljkovic, Z., (2015). Chaotic fruit fly optimization algorithm. *Knowledge-Based Systems*, 89, 446-58.
- Nouhi, B., Talatahari, S., Kheiri, H., & Cattani, C., (2013). Chaotic charged system search with a feasible-based method for constraint optimization problems. *Mathematical Problems in Engineering*, 1-8.
- Ozer, A.B., (2010). CIDE: Chaotically initialized differential evolution. *Expert Systems with Applications*, 37 (6), 4632-41.

- Özyön, S., Yaşar, C., & Temurtaş, H., (2018). Incremental gravitational search algorithm for high-dimensional benchmark functions. *Neural Computing and Applications*, 1-25.
- Pareek, K., Patidar, V., & Sud, K.K., (2005). Cryptography using multiple one-dimensional chaotic maps. *Communications in Nonlinear Science and Numerical Simulation*, 10 (7), 715-23.
- Patidar, V., Pareek, N.K., & Sud, K.K., (2009). A new substitution–diffusion based image cipher using chaotic standard and logistic maps. *Communications in Nonlinear Science and Numerical Simulation*, 14 (7), 3056-75.
- Peitgen, H., Jurgens. H., & Saupe D., (1992). *Chaos and Fractals: New frontiers of science*. Springer-Verlag, Berlin.
- Saremi, S., Mirjalili, S., & Lewis, A., (2014). Biogeography-based optimisation with chaos. *Neural Computing and Applications*, 25 (5), 1077-97.
- Storn, R., & Price, K., (1997). Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341-59.
- Stoyanov, B., & Kordov, K., (2014). Novel image encryption scheme based on Chebyshev polynomial and Duffing Map. *The Scientific World Journal*, 1-11.
- Stoyanov, B., (2014). Pseudo-random bit generation algorithm based on Chebyshev Polynomial and Tinkerbell Map, *Applied Mathematical Sciences*, 8 (125), 6205-10.
- Talatahari, S., Farahmand Azar, B., Sheikholesman, R., & Gandomi, A.H., (2012). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation*, 17 (3), 1312-19.
- Tavazoei, M.S., & Haeri, M., (2007). Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Applied Mathematics and Computation*, 187 (2), 1076-85.
- Wang, G.G., Guo, L., Gandomi, A.H., Hao, G.S., & Wang, H., (2014). Chaotic krill herd algorithm. *Information Sciences*, 274, 17-34.

- Wilcoxon, F., Katti, S., & Wilcox, R.A., (1970). Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected Tables in Mathematical Statistics*, 1, 171-259.
- Zaslavsky, G.M., (1978). The simplest case of a strange attractor. *Physics Letters A*, 69 (3), 145-47.
- Zhang, Y., & Xiao, D., (2013). Double optical image encryption using discrete Chirikov standard map and chaos-based fractional random transform. *Optics and Lasers in Engineering*, 51 (4), 472-80.
- Zheng, W.M., (1994). Kneading plane of the circle map. *Chaos, Solitons and Fractals*, 4 (7), 1221-33.



ISBN: 978-625-7562-47-8