



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

A THESIS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Boundary estimation with gravitational
search algorithm based optimization in
electrical impedance tomography

GRADUATE SCHOOL
JEJU NATIONAL UNIVERSITY

Faculty of Applied Energy System
Major of Electronic Engineering

Sharma Sunam Kumar

2022.06

Boundary estimation with gravitational search algorithm based optimization in electrical impedance tomography

Sharma Sunam Kumar
(Supervised by Professor Kim Kyung Youn)

A thesis submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Electronic Engineering

2022.06.

This thesis has been examined and approved.

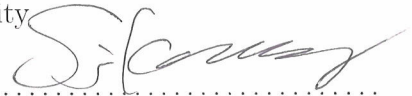
.....
Thesis director, Doh Yang Hoi, Professor, Department of Electronic Engineering,
Jeju National University



.....
Kim Kyung Youn, Professor, Department of Electronic Engineering,
Jeju National University

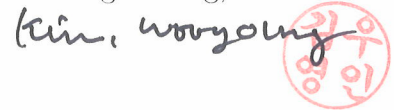

.....
Kang Soon Ju, Professor, Department of Electronic Engineering,
Kyungpook National University



.....
Ko Seok Jun, Professor, Department of Electronic Engineering,
Jeju National University

.....
Kim Woo Young, Professor, Department of Electronic Engineering,
Jeju National University

.....
Date

Faculty of Applied Energy System
Major of Electronic Engineering
GRADUATE SCHOOL
JEJU NATIONAL UNIVERSITY

To family

Acknowledgements

This dissertation is the result of four years of effort to study the boundary estimation by electrical impedance tomography. Writing this dissertation would not have been possible without the help and support of those around me. Now, I would like to thank everyone from the bottom of my heart.

First of all, I would like to thank Professor Kim Kyung-young, my academic advisor, for his continued support and guidance. During my doctoral course, he has helped me select a proper path for my research work. I owe him a lot of gratitude for showing me this way of research. Besides being an excellent supervisor, Prof. Kim is a very close person and a good friend to me.

I am very grateful to Dr. Khambampati Anil Kumar for mentoring me in all my work. His guidance, support, and in-depth knowledge of EIT have always been a great help. His support for personal problems has greatly helped me a lot to settle in this unknown country far from my hometown. I would like to thank the other members of my Ph.D. committee (Prof. Doh Yang Hoi, Prof. Kang Soon Jun, Prof. Kim Woo Youn, and Prof. Ko Seok Jun). They all monitored my research work and took an effort in reading and provide me with valuable comments on earlier versions of this dissertation.

I would like to thank Brain Korea 21+ program for the financial support for presenting the research work held in Korea and abroad. I would also like to acknowledge the financial support from the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) under Grant NRF- 2017R1A2B4002224, and NRF-2020R1A2C2006463 for the research work.

Living away from home is very difficult, but I'm happy to have friends. They made me forget about my loneliness. It helped me a lot to get my PhD. course. I am very thankful to Dr. Banjade Maniram, Khanal Amrita, Bhattarai Roshan Mangal, and Dr. Saud Shirjana for all the help and support. They made my stay in Jeju a memorable one.

My PhD. trip would not have been possible without the help of Prof. Dahal Ranjan Kumar. He believed in me and helped me to continue this journey. The help he gave me is indescribable. I am very grateful for everything he has done. I would also express

my sincere gratitude to Prof. Saptoka Achyut.

I will never have words to express my gratitude toward my family. Their support, love, and affection have always been a driving force in my academic journey. Missing to see my son grow every day, my mother and wife gave me much-needed support to go through ups and downs in this journey. I would like to thank all those whom I have not mentioned above but who helped me in many ways.

Abbreviations and notations

Adam	Adaptive moment estimation
BEM	Boundary element method
CEM	Complete electrode model
DNN	Deep neural network
ERT	Electrical resistance tomography
EIT	Electrical impedance tomography
FEM	Finite element method
FDM	Finite difference method
GA	Genetic algorithm
GPR	Ground penetrating radar
GSA	Gravitational search algorithm
IR	Infrared thermography
ML	Machine learning
MLP	Multi-layer perceptron
mNR	modified Newton-Raphson
NN	Neural network
PCC, r	Pearson correlation coefficient
PSO	Particle swarm optimization
PSOGSA	Particle swarm optimization gravitational search algorithm
RMSE, $RMSE$	Root mean square error
RELU	Rectified Linear Unit
RMSprop	Root mean square propagation
SGD	Stochastic gradient descent

TDS	Time-domain spectroscopy
Ω	Domain (object) under study
$\partial\Omega$	Boundary of the object
$\sigma, \sigma(x, y)$	Internal conductivity distribution of the object
E, \bar{E}	Electrical field
B	Magnetic induction, Component of stiffness matrix
\bar{B}	Magnetic induction
H	Magnetic field
J	Current density
D	Electric displacement, Component of stiffness matrix
ϵ	Permittivity, small constant
μ	Permeability
ω	Frequency
J^o	Ohmic current
J^s	Current source
$u, u(x, y)$	Potential distribution inside an object
A	Magnetic vector potential, stiffness matrix
z_l	Contact impedance of the l^{th} electrode
I_l	Current applied to the l^{th} electrode
α_i	Parameters used in FEM formulation
β_i	Parameters used in FEM formulation, Hyper-parameter
ϕ	Two-dimensional first-order basis function, Periodic basis function
$U, U(\Gamma_i, \sigma_i^b)$	Measured voltages from the electrodes in FEM formulation
C	Component of stiffness matrix

M	Spare matrix, Mass of the particle, Number of organ in pelvis domain
$\Phi, \varphi(y_t, y_p)$	Cost function
V	Voltage, Measured voltage recorded from the electrical impedance system
G	Gravitational force
g_c	Gravitational constant
a	Acceleration of particle
R	Distance between two particles
t	Iteration time, Time stamp
Γ_i	Particle, Shape coefficient
λ_i^d	Position of i^{th} particle (Γ_i) in the d^{th} dimension
N	Number of particles, Number of test data
M_{ai}	Active gravitational mass of particle i
M_{pi}	Passive gravitational mass of particle i
$rand$	Random number [0,1]
M_{ii}	Inertial gravitational mass of particle i
$v_i^d(t)$	velocity of particle i in the d^{th} dimension at t^{th} iteration
$iter$	current iteration
$iter_{max}$	maximum iteration
κ	Descending coefficient
$best(t)$	Best cost function
$worst(t)$	Worst cost function
N_{best}	First N particles with best fitness

c_j	Positive constant
w	Inertia weight, weighting function, neurons weight
$pbest_i$	Best previous position of the i^{th} particle
$gbest$	Best previous position among all the particles
$B_m(s)$	Boundary of bladder, Boundary of defect on graphene
F_ϕ	Order of the truncated Fourier series
Γ_i^t	True valued shape coefficient
σ_i^b	Background conductivity of the graphene sheet
λ_i^b	Dimension of the particle defining the conductivity
c_k, C'_j	Weighting factors
η	Number of Monte Carlo runs
$\bar{\lambda}$	Mean
\overline{MAE}	Mean absolute error
\overline{MES}	Mean error square
E_{sd}	Error standard deviation
\overline{RMSE}	Mean root mean square error
b	Bias
Θ	Activation function
Y, y_i, y_p	Node output, Predicted output
y_t, y^t	True output
α	Learning rate
w_{ik}	Weight value between output neuron k and input neuron i
j	Number of neurons
θ	Model parameter

m_t, \hat{m}_t	Moving average of gradient
v_t, \hat{v}_t	Squared of the gradient
B_γ	Discrete front points matrix
y_γ	Unknown parameter of Front point method
x_γ	Reference point of Front point method
H	Mapping function
W	Weight of the node
$\min V$	Minimum values of the voltage V
$\max V$	Maximum values of the voltage V
v_{norm}	Normalized voltage V
m	Total number of data samples
\bar{y}	Mean of predicted output
\bar{y}^t	Mean of true output

Abstract

Electrical impedance tomography (EIT) is a noninvasive image reconstruction method. It reconstructs the cross-sectional conductivity distribution of the domain. EIT is applied in various areas of applications such as medical, industrial, and geophysical. However, it suffers from poor spatial resolution due to the ill-posed and non-linear nature of the problem. Boundary or shape estimation is the alternative approach to solve this poor resolution problem. In this approach, the number of unknowns to be estimated is reduced which improves the spatial resolution. The conductivity of the closed disjoint region of the domain is assumed to be known as prior for the boundary estimation. In this thesis, for the closed boundary, the complex shape is defined by the Fourier series coefficients and the shape estimation is done with a heuristic algorithm. In this work, we have presented three scenarios for the boundary estimation in EIT. The first study is the estimation of the bladder boundary in the pelvic domain. In this, the boundary is estimated by a heuristic algorithm gravitational search algorithm (GSA). The estimation of the bladder using the noninvasive method is necessary for paraplegia patients. These patients are unable to discharge urine at the right time due to a weaker sensation for bladder volume. If the urine is not discharged in time, then the bladder size will increase and affect the neighboring organs and tissues. Size estimation of the bladder with EIT can clarify the bladder status. The bladder is a nonuniform structure with a complex shape; therefore, higher-order Fourier series is needed to represent the true shape. Estimating higher-order Fourier coefficients by a conventional modified Newton-Raphson (mNR) algorithm does not give the desired performance. GSA is proposed in this work to estimate the Fourier series coefficients as it is known for solving optimization problems in high-dimensional search space. Also, GSA has fast convergence and does not require the computation of Jacobian. Numerical experiments and phantom studies are performed to estimate the bladder size and it is compared with the estimated result by mNR.

The second case is the estimation of the defect on the single-layer graphene sheet by PSO-GSA. A PSO-GSA is a hybrid algorithm that is the combination of particle swarm optimization (PSO) and GSA. Recently, graphene has gained a lot of attention

in the electronic industry due to its unique properties and can overcome the limits of miniaturization making way for novel devices in the field of electronics. For the development of new device applications, it is necessary to grow large wafer-sized monolayer graphene. Among the methods to synthesize large graphene films, chemical vapour deposition (CVD) is one of the promising and common techniques but defects such as cracks, holes, or wrinkles are hard to avoid. Electrical impedance tomography (EIT) can be used to detect those defects on a graphene sheet. The conductivity is assumed to be known as prior and the geometry of the defect is estimated. These defect geometries are defined by truncated Fourier series coefficient which can represent the complex shapes. Numerical and experimental studies are done for graphene characterization and the results showed that the proposed PSOGSA has good performance in locating the defects present on a graphene surface.

The third is the open boundary case where the interlayer boundary of the subsurface is estimated. Subsurface topology estimation is important for the geophysical survey. The subsurface region can be approximated as piece-wise separate regions with constant conductivity in each region; therefore, the conductivity estimation problem is transformed to estimate the shape and location of the layer boundary interface. Each layer interface boundary is treated as an open boundary that is described using front points. A DNN model is used to estimate the front points describing the multi-layer interface boundaries. This DNN model is tuned for hidden layer nodes using PSOGSA. The PSOGSA tuned DNN model is trained for interlayer boundary reconstruction using training data that consists of pairs of voltage measurements of the subsurface domain. The tuned DNN model estimation result is compared with the 7-layer DNN model. The study on all three cases shows the proposed method has a better estimation result than the compared method.

Contents

List of Figures	xii
List of Tables	xviii
1 Introduction	1
1.1 Electrical impedance tomography	1
1.2 Boundary estimation studies	4
1.3 Motivation	5
1.4 Aims and content	8
2 Forward problem	11
2.1 Governing equation	11
2.2 Boundary conditions	14
2.3 Mathematical Electrode models	14
2.3.1 Complete electrode model	15
2.4 Numerical Solution	16
2.5 Current injection method	18
3 Inverse problem	20
3.1 Introduction	20
3.2 Heuristic algorithm	20
3.3 Neural network	21
4 Heuristic algorithm	23
4.1 Introduction	23
4.2 Gravitational search algorithm	24

4.3	Particle Swarm Optimization	29
4.4	Hybrid PSO-GSA algorithm	31
5	PSO-GSA tuned DNN	34
5.1	Deep neural network	34
5.2	Training of DNN	35
5.3	Hyper-parameter	37
5.4	Hyper-parameter tuning	38
5.5	Tuning with PSO-GSA	39
6	Bladder boundary reconstruction with GSA	43
6.1	Boundary representation of the bladder	43
6.2	Results	44
6.2.1	Numerical simulation	46
6.2.2	Phantom experiment	53
6.3	Conclusion	58
7	Defect detection in graphene with PSO-GSA	60
7.1	Results	61
7.1.1	Numerical study	64
7.1.2	Experimental study	77
7.2	Conclusion	81
8	Interlayer boundary estimation by PSO-GSA-DNN	82
8.1	Interlayer boundary representation	82
8.2	Results	83
8.3	Conclusion	96
9	Conclusion	98
	Summary	100
	List of publications	102
	Bibliography	104

List of Figures

1.1	A domain with 16 electrodes on the surface. A pair of electrodes used as current source and remaining for voltage measurement in an EIT system	2
1.2	Schematic diagram of an EIT system for estimating a defect on a graphene sheet sample.	3
2.1	Mesh used in this thesis. (a)Graphene sheet mesh, (b)Pelvic mesh, and (c)Subsurface mesh.	16
2.2	Adjacent and opposite current injection electrode selection for first current pattern.	19
4.1	Flowchart for the gravitational search algorithm.	28
4.2	PSO flowchart for estimating the coefficient of Fourier series-defining bladder boundary.	30
4.3	PSOGSA flowchart for EIT estimation problem.	31
5.1	Rectified Linear Unit (RELU) function.	35
5.2	Dropout concept illustration where nodes are randomly selected and their outputs are set to zero to present as a deactivate nodes.	36
5.3	Learning rate illustration.	37
5.4	A schematic diagram of the DNN model designed as funnel.	38
5.5	A schematic diagram of PSOGSA tuned DNN model.	40
5.6	Hyper-parameter tuning flow-chart.	41
5.7	Flowchart of DNN with PSOGSA tuning the hyper-parameter.	42

6.1	CT scan image of pelvis (Anjos et al., 2007). The bladder is the center top oval shape region with side muscles and bones included in two side lobes and circle represent the rectum.	44
6.2	Flowchart for the gravitational search algorithm for estimating the bladder boundary.	46
6.3	Mesh of the pelvic domain for estimating the bladder boundary. (a) Fine mesh used for the forward problem. (b) Coarse mesh used for inverse problem.	47
6.4	Convergence graph of GSA for estimating the bladder boundary in the pelvic domain	48
6.5	A simple numerical case for bladder estimated by GSA and PSO. The axis units are on a millimeter scale. (a) Big bladder case, and (b) small bladder case in the pelvic domain.	49
6.6	Numerical results for shape estimating of a big bladder case (case 1) using GSA for a multi-target scenario. The axis units are on a millimeter scale.	50
6.7	Numerical results for shape estimating of a small bladder case (case 2) using GSA for a multi-target scenario. The axis units are on a millimeter scale.	51
6.8	Numerical results for shape estimating of a small bladder case (case 3) at a different location using GSA for a multi-target scenario. The axis units are on a millimeter scale.	51
6.9	Comparison of estimated numerical results for GSA with a different number of particles assigned (5, 10,20, 75, 100). (a) True bladder boundary and the estimated bladder boundary by GSA, (b) Corresponding RMSE of the estimate result by GSA, and (c) The computational time of algorithm with a different number of particles.	52
6.10	Comparison between GSA estimation result and PSO estimation result with the different iteration number. GSA#75 is the GSA estimate result with 75 iteration steps, PSO#75 is PSO estimate result with 75 iteration steps, PSO#100 is PSO estimate result with 100 iteration steps, and PSO#300 is PSO estimate result with 300 iteration steps.	53

6.11	(a) Phantom experiment setup for big bladder (case 1). The red target is the bladder and yellow target regions are side lobes and rectum. (b) Phantom experiment estimation result for big bladder cases with GSA and PSO. The axis units are on a millimeter scale.	55
6.12	(a) Phantom experiment setup for medium bladder size (case 2). The red target is the bladder and yellow target regions are side lobes and rectum. (b) Phantom experiment estimation results for medium bladder size case with GSA and PSO. The axis units are on a millimeter scale.	56
6.13	(a) Phantom experiment setup for small bladder size (case 3). The red target is the bladder and yellow target regions are side lobes and rectum. (b) Phantom experiment estimation results for medium bladder size case with GSA and PSO. The axis units are on a millimeter scale.	57
7.1	Fine and coarse mesh of graphene used in the study. The boundary electrodes are represented in blue colour attached to the mesh boundary.	61
7.2	Flowchart of the PSOGSA algorithm for estimating defect geometry and the background conductivity of the graphene sheet.	62
7.3	Numerical case of single defect on graphene sheet for parameter optimization. (a) True defect location along with the estimated result from PSOGSA. (b) Weighting factors (C'_j) combination, (c) RMSE and cost function value for different cases of weighting factor combination, (d) cost function value for a different level of iteration, (e) RMSE and cost function with a different number of particles in PSOGSA.	65
7.4	Numerical case of two defects on the graphene sheet for parameter optimization. (a) True defect location along with the estimated result from PSOGSA. The black circle represents the true position of the defect. The estimated location of a defect by PSOGSA is represented by a blue circle. (b) Corresponding RMSE and cost function with a different number of particles in PSOGSA.	66

7.5	Numerical results for cases 1-6 with a single defect on graphene surface by PSO-GSA and GSA. The black circle represents the true position of the defect. The estimated location of the defect by PSO-GSA and GSA is represented by a blue circle and red circle, respectively.	67
7.6	Numerical results for cases 7-9 with two defects on graphene surface by PSO-GSA and GSA. The black circle represents the true position of the defect. The estimated location of a defect by PSO-GSA and GSA is represented by a blue circle and red circle, respectively.	68
7.7	Standard deviation error of the estimated Fourier series coefficients of the all 9 numerical cases.	76
7.8	Electrode coated graphene sample used for the experimental study. . . .	77
7.9	Experimental study for case 1 with defect identification on graphene of size 2.5×2.5 cm. (a) Graphene with a single defect. (b) Experimental result for defect location on the graphene sheet. The estimated location of the defect by PSO-GSA and GSA is represented by a blue circle and red circle, respectively.	78
7.10	Experimental study for case 2 with defects identification on graphene of size 2.5×2.5 cm. (a) Graphene with two defects. (b) Experimental result for defect location on the graphene sheet. The estimated location of a defect by PSO-GSA and GSA is represented by a blue circle and red circle, respectively.	80
8.1	Interlayer boundary parameterized with the discrete front points.	83
8.2	A sample of the input voltage reading from the dataset. (a) Normal voltage reading, (b) Normalized voltage reading.	85
8.3	Subsurface mesh used for the calculation of the voltage based on the boundaries of the interlayer.	86
8.4	Training scheme of the deep neural network model for estimating the front points of interlayer boundaries. The number represents the number of nodes for each layer.	86

8.5	Analysis of PSOGSA with different number of particle size to tune DNN model. (a) Different particle size. (b) Per iteration execution time. (c) Fitness of PSOGSA.	88
8.6	89
8.7	Optimum node size for hidden layer of the DNN model estimated by PSOGSA.	89
8.8	Training and validation loss and RMSE the PSOGSA tuned DNN model with different batch size	90
8.9	Training and validation loss and RMSE the model PSOGSA tuned DNN model with different learning rate.	90
8.10	(a) Different activation function configuration for the hidden layers. (b) Training and validation loss and RMSE the model for the same.	92
8.11	Numerical results with case 1 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSOGSA tuned DNN and DNN model.	93
8.12	Numerical results with case 2 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSOGSA tuned DNN and DNN model.	93
8.13	Numerical results with case 3 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSOGSA tuned DNN and DNN model.	94
8.14	Numerical results with case 4 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSOGSA tuned DNN and DNN model.	94
8.15	Numerical results with case 5 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSOGSA tuned DNN and DNN model.	94
8.16	Numerical results with case 6 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSOGSA tuned DNN and DNN model.	95

8.17 Numerical results with case 7 for interlayer boundary estimation of subsurface.	
(a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA	
tuned DNN and DNN model.	96
8.18 RMSE and PCC for the estimated front points by PSO-GSA-tuned DNN	
and DNN model.	96

List of Tables

6.1	Comparison of RMSE for the estimated Fourier coefficients by PSO and GSA in multi-target numerical scenarios.	50
6.2	Comparison of RMSE of the estimated result of PSO with different iteration steps with GSA estimated result with 75 iteration steps.	54
6.3	Comparison of RMSE for the estimated Fourier coefficients by PSO and GSA in phantom experiments.	58
7.1	Statistical parameter for numerical case 1 (figure 7.5a): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	69
7.2	Statistical parameter for numerical case 2 (figure 7.5b): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	69
7.3	Statistical parameter for numerical case 3 (figure 7.5c): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	70
7.4	Statistical parameter for numerical case 4 (figure 7.5d): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	71

7.5	Statistical parameter for numerical case 5 (figure 7.5e): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	72
7.6	Statistical parameter for numerical case 6 (figure 7.5f): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	72
7.7	RMSE of the estimated Fourier series coefficients by PSO GSA and GSA for single defect cases.	72
7.8	Statistical parameter for numerical case 7 (figure 7.6a): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	73
7.9	Statistical parameter for numerical case 8 (figure 7.6b): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	74
7.10	Statistical parameter for numerical case 9 (figure 7.6c): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA	75
7.11	RMSE of the estimated Fourier series coefficients by PSO GSA and GSA for two defects cases.	75
7.12	Estimated conductivity (mS/cm) of graphene sheet by PSO GSA and GSA	80
8.1	Comparison of RMSE and PCC for the estimated front points by PSO GSA tuned DNN and DNN model for interlayer boundaries in subsurface domain.	97

1 Introduction

1.1 Electrical impedance tomography

Electrical impedance tomography (EIT) is a non-destructive, non-invasive imaging technique (Cheney et al., 1999) which estimates the cross-sectional conductivity distribution of an object. In EIT, an array of electrodes are attached to the boundary of the object. A small amplitude alternating current is injected and the resulting voltages are measured on the surface electrodes. A simple diagram representing injected current and measured voltage are presented in figure 1.1. The figure also presents the conductivity distribution of the domain. The conductivity distribution can be estimated based on the injected current and measured voltages. The conductivity distribution of the object is computed according to Ohm's law. A schematic diagram presented in figure 1.2 shows the complete EIT system with the control and data flow. The EIT system is mostly divided into two parts, one contains the switch with a multiplexer, and another contains the analysis part. With the help of a switch and multiplexer current is supplied and resulting voltages are measured from the surface electrodes. The other part of the system is the data analysis which gives the estimated result. EIT is also known as soft field imaging techniques as the electrical quantities are dispersed inside the object and are affected by the inside objects. The conductivity estimation by EIT is a non-linear ill-posed problem and suffers from low spatial resolution. There are various advantages of EIT, such as being portable, non-invasive, non-destructive, radiation-free, and cheap. Also, EIT can be used for continuous monitoring as it can be used with a high-speed data acquisition system.

EIT has been studied and implemented in various area of medical, industry, and geophysical. In medical application, EIT has been implemented in detection and

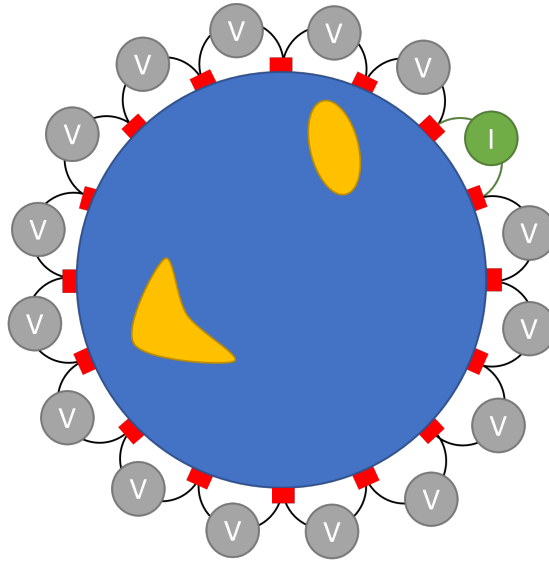


Figure 1.1: A domain with 16 electrodes on the surface. A pair of electrodes used as current source and remaining for voltage measurement in an EIT system

classification of tumors in breast tissue (Mueller et al., 2001; Osterman et al., 2000), and study of gastric function (Dijkstra et al., 1993). Also, for the continuous monitoring of a several physiological phenomena, i.e., pulmonary ventilation (Frerichs et al., 1998), hyperthermia (Moskowitz et al., 1995), perfusion (Frerichs et al., 2002), cardiac (Isaacson et al., 2006), and respiratory functions (Adler et al., 1997). Various clinical applications has been applied with EIT such as lung imaging (Brown, 2001; Frerichs, 2000), head imaging (Holder, 1992), and breast imaging (Cherepenin et al., 2001; Choi et al., 2007; Halter et al., 2008; Ybarra et al., 2007) . Monitoring of various industrial applications has been done with EIT such as monitoring the multi-phase flow in the process pipelines (Dickin and Wang, 1996; Jones et al., 1993; Khambampati et al., 2012), mixing in pipes (Pinheiro et al., 1997), flow process (Mann et al., 1997), sedimentation (Tossavainen et al., 2006), and also estimating air bubbles in pipe flow (Plaskowski et al., 1995). In geophysical applications, resistivity imaging is estimated rather than the conductivity for exploring aquifers (Sharma et al., 2020), minerals (Maillol et al., 1999), ground water (Barker and Moore, 1998), detection of fractures (Spies and Ellis, 1995), faults (Daily et al., 1992), underground pollutant (D'Antona et al., 2002), oil reservoir monitoring (Stacey et al., 2006), and geological mapping (Meads et al., 2003). In the field of robotics EIT is used for detecting the pressure on an artificial skin of the

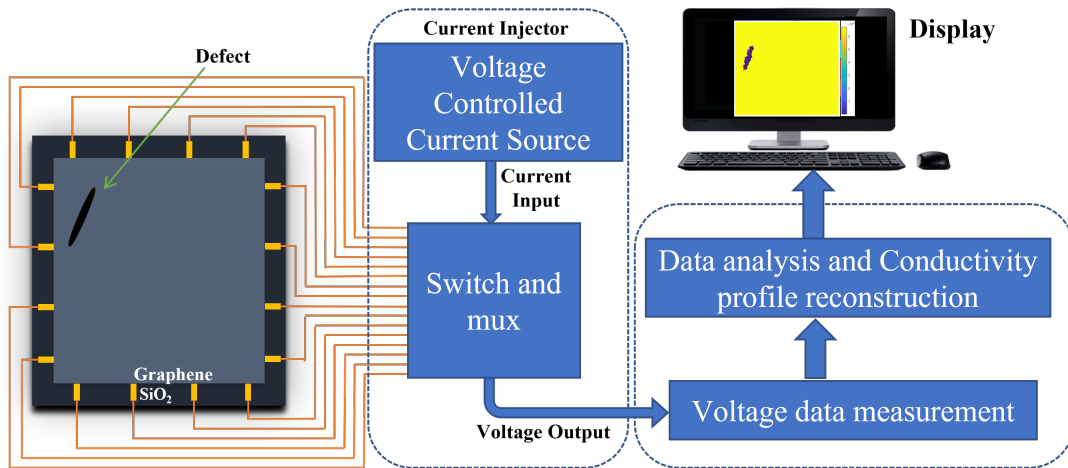


Figure 1.2: Schematic diagram of an EIT system for estimating a defect on a graphene sheet sample.

robots (Liu et al., 2020; Park et al., 2019).

The injected current and measured voltage physical relationship is governed by partial differential equations derived from Maxwell equations. A physical model is used to describe the boundary condition for those equations. There are many physical models used in EIT, namely continuum, gap, average-gap, shunt, and complete electrode model (CEM). Out of these models, CEM is preferred because it considers discreteness, shunt effect, and contact impedance between the electrode and outer boundary of the object. It can generate a mathematical model close to the true situation than the other models. EIT uses a forward and inverse problem for reconstructing an image. Calculation of the boundary voltages on the electrode surface is done in the forward problem of EIT. An analytic solution for the Laplace equation is only possible for the simple regular geometries and not for the complex geometry. However, the numerical method is used for the complex geometry and EIT utilizes the finite element method (FEM), and boundary element method (BEM) for solving the Laplace equation (Zienkiewicz and Taylor, 2000). Out of this, FEM is the preferred method for the numerical calculation in EIT as BEM involves additional computation of fluxes on the inclusion boundary and potential on the electrode surface (Khambampati et al., 2012).

In the inverse problem, the internal conductivity distribution is estimated from the measurement voltage reading and the injected current data. The inverse problem is non-linear and ill-posed because of the voltage weak relationship with the conductivity

distribution. This means that a small change in voltage measurement results a huge change in the conductivity distribution.

1.2 Boundary estimation studies

Due to the ill-posed nature of EIT, it suffers from poor spatial resolution while estimating the conductivity of the domain (Beretta et al., 2018; Lionheart, 2004). Boundary or shape estimation is the alternative approach that can eliminate the spatial resolution problem. In this approach the number of unknowns decreases which improves the spatial resolution. It incorporates the prior information on the conductivity of each closed disjoint region of the domain into the solution. The unknowns to be estimated is the boundary of the anomalies in the domain. The boundary can be used to define its shape, size, and location. There are various methods used to define the shape of the anomalies namely the Fourier series method, level set method, front-point method, B-spline based method, and super-shape method (Gu et al., 2021; Khambampati et al., 2018; Kim et al., 2007b; Liu et al., 2015, 2019; Soleimani et al., 2006).

There are few studies conducted for the boundary estimation in EIT. Liu et al. used parametric level set function to define the shape of the anomaly and the image was reconstructed with EIT where the geometry of the anomaly was estimated (Liu et al., 2017). Another method used to describe the boundary of the anomaly was supershape-based in the study conducted by Gu et al. for EIT in (Gu et al., 2021). They assumed conductivity distribution as piecewise constant distributed and shape reconstruction was done. Han and Prosperetti used BEM and the boundary voltages were used to estimate the Fourier series which defined each target in the domain (Han and Prosperetti, 1999). An organ elliptic boundaries were defined using Fourier series coefficients and were estimated by the Kalman filter. This method was introduced by Vauhkonen et al. in 1998 where the author assumed the boundaries of the lungs were known a priori and the boundary of the heart was estimated. A more general case was presented by Kolehmainen et al. where the heart and both lungs boundaries were represented by Fourier series coefficients and the conductivity was assumed to be known as a priori (Kolehmainen et al., 2001). In this study, the author estimated the

boundaries with the extended Kalman filter and also with the traditional Gauss-Newton method. However, the Gauss-Newton method exhibited convergence problems and the boundaries of the target were self-intersecting when the number of Fourier series coefficients was high. Even when the initial guess was assumed close to the true boundaries, the Gauss-Newton method sometimes showed deviation from the true boundaries. Also, the Gauss-Newton method heavily relies on the calculation of Jacobian to linearise the non-linear EIT problem.

1.3 Motivation

As stated earlier EIT suffers from poor spatial resolution which fails to estimate anomalies boundary. The boundary estimation of the region as an alternative method to overcome the resolution problem. If the conductivity of the domain is piecewise constant within the same region of the domain, then the shape estimation becomes the problem. In other words, the boundary of the target becomes the unknown to be estimated. There are many methods to define the geometry of a target as described in section 1.2.

The region boundary can be a complex shape which are estimated by an inverse algorithm. The most widely used inverse problem algorithm is the modified Newton-Raphson (mNR) algorithm. It is a very robust algorithm and its estimation result are accurate with fewer iterations. However, in the case of the complex boundary of the anomalies, mNR tends to produce the intersecting boundaries of the region (Konki et al., 2020; Li et al., 2005). Also, mNR heavily depends on the initial guess in the search space and contains the calculation of Jacobian matrix (Brandstatter, 2003) which is very complex.

An alternative algorithm that is capable to solve complex shapes without the Jacobian matrix in the EIT domain is required. Such an algorithm is the heuristic algorithm that can estimate the complex boundary in the EIT domain. Various studies have been conducted where heuristic algorithms are used in solving complex problems. Mostly particle swarm optimization (PSO) has been implemented for a non-linear complex problem, where it provided an optimum solution (Shi and Eberhart, 1999; Simon,

2013). Ijaz et al. in 2007 has introduced PSO in the EIT domain (Ijaz et al., 2007). In this PSO estimated the Fourier series coefficient of the target with good accuracy. However, PSO has a low convergence rate and a high number of iterations is required to obtain a good solution for a complex estimation problem. Rashedi et al. in 2009 introduced a new heuristic algorithm known as a Gravitational search algorithm (GSA) and evaluated it against another heuristic algorithm. The GSA outperformed most of the other algorithms in the standard benchmark functions and also the convergence of GSA is very fast than other heuristic algorithms (Rashedi et al., 2009).

The introduction of the GSA has inspired us to implement it as an estimation algorithm for the EIT domain. In this thesis, GSA and PSOGSA are used as estimating algorithm for estimating anomaly boundary. Three different studies are conducted in this work and the boundary estimation is done with an algorithm based on GSA. In the first case, GSA is used for estimating the urine bladder boundary. A second case presents the estimation of a defect boundary on the graphene sheet with the hybrid PSOGSA algorithm. The third case is the PSOGSA tuned deep neural network (DNN) which is used for estimating the subsurface interlayer boundary.

Bladder size estimation

The GSA is used for estimating the bladder boundary and the estimation result is compared with PSO. This is required for monitoring of the bladder for a paraplegia patient. A paraplegia patient which has low sensation in the lower part of the body has urine incontinence as a major problem. This problem is also present in the elderly person, which patient cannot determine the time to urinate. This problem arises due to the damage in the neural damage in the sensing area. As a result of this problem, a person is unable to determine the time to discharge the urine. This problem can create a very critical situation if the bladder pressure is not released in time. To avoid this situation there are two methods practiced in the clinic i.e. self-catherization and continuous monitoring. A self-catherization method involves minor surgery and prolonged use can lead to infection (Madersbacher, 1999). This problem does not occur in the continuous monitoring method which utilizes ultrasound technique (Koomen et al., 2002; Seif et al., 2004). Moreover, this method required a skilled person to

operate the equipment and continuous monitoring makes patients discomfort as they are unable to move.

Every organ and/or tissue has different conductivity values and urine has high conductivity when compared to other parts due to the high concentration of salt in it (Bayford, 2006; Najarian and Splinter, 2005). Due to different conductivity values of the organs in the human body, EIT can be used for continuous monitoring of the bladder. Also, it is a non-invasive method, fast, and portable (Chen et al., 2020; Xu et al., 2018) making it more suitable for continuous monitoring. When conductivity distribution is known a priori using data from alternative methods such as MRI or CT scan (Hu, 1999; Khoo et al., 1997), estimation of bladder boundary becomes the inverse problem. We can monitor the bladder condition by estimating the bladder size, i.e. bladder boundary.

Estimating defect geometry on graphene

GSA having a strong exploitation performance suffers from poor exploration performance during the search process. This performance is improved by combining GSA with another algorithm. We have used PSO-GSA which is the hybrid algorithm which is combination of GSA and PSO. The exploration performance of PSO is used in this algorithm which improved the performance of the algorithm. This hybrid algorithm is used for estimating the defect boundary on the graphene sheet. The background conductivity of graphene sheet is also estimating along with the defect boundary by PSO-GSA.

For the past few years, graphene has been the focus of the research work due to its remarkable mechanical, physical, electrical, chemical properties (Bolotin et al., 2008; Bunch et al., 2007; Schwierz, 2011). Graphene has shown promising results to overcome the limits of miniaturization for the electric channels in the nanoscale electronics applications (Fuchs and Goerbig, 2008). Graphene sheets can be fabricated with many methods such as exfoliation, colloidal suspension, epitaxial growth, and chemical vapour deposition (Jayasena and Subbiah, 2011; Jin et al., 2010; Pu et al., 2009; Reina et al., 2009). During fabrication of graphene inhomogeneity such as wrinkles, holes, and cracks develop (Li et al., 2009; Liang et al., 2011). During sample preparation of a specific size,

human errors can occur which can affect the electrical property of sample (Vasić et al., 2016; Zhu et al., 2012). Hence, identifying these defects is very useful in understanding the properties of the graphene sample. PSO-GSA is used as the estimating algorithm for estimating the defect boundary on the graphene sheet and the background conductivity.

Subsurface interlayer boundary estimation

The DNN model is an emerging algorithm for estimation. The model is trained based on dataset but is not able to tune with that dataset. We have used PSO-GSA to tune the DNN model in this work. This tuned DNN model is then used to estimate the interlayer boundaries of the subsurface. In geology the study of the subsurface is very important as it helps us to understand the properties and exploration of minerals in a particular location (Hoover et al., 1995; Kearey et al., 2002; Philp and Crisp, 1982). The subsurface is composed of soil, rocks, water, etc. in different shapes, and sizes. The electrical current flow in the soils suggests a relationship between soil strength and electrical resistivity (Sudha et al., 2009). The content of clay in soil determines the soil strength and the soil moisture affects the electrical conductivity (Katsube et al., 2003). Different layers in the subsurface have resistivity contrast making it possible to detect those layers using ERT or EIT. If the conductivity of the layers is known a priori then interlayer boundary estimating can be performed. A borehole method can be used to obtain material conductivity information. It can be used as a priori in estimating the interlayer subsurface boundaries. A PSO-GSA tuned deep neural network (DNN) is used to estimate the interlayer boundaries of the subsurface. In this thesis, we have optimized the node's size of the hidden layers of the DNN model using PSO-GSA.

1.4 Aims and content

This proposed thesis help to develop a novel reconstruction technique with EIT for estimating the boundaries of the anomalies in a domain. The boundary estimation is done for two cases, i.e. closed boundary and open boundary. GSA is used as an inverse problem solver for estimating the bladder boundary. The bladder boundary is described with Fourier series coefficients which is a closed boundary case. This is the

first case study of the thesis work. The second case is the estimation of defect location on graphene sheet using the PSOGSA hybrid heuristic algorithm. This algorithm combines the characteristic of PSO and GSA algorithms which can estimate the defect geometry in the highly conductive domain. In both these studies, the boundary is a closed boundary scenario and the conductivity inside the boundary is assumed to be uniformly distributed. The open boundary is the interfacial boundary between the layers of the subsurface. This interlayer boundary is estimated by the DNN model which is optimized by PSOGSA. A PSOGSA algorithm is used to optimize the number of nodes in the hidden layers of the model. Statistical parameters are analyzed for all the cases for the proposed algorithm.

This thesis is divided into 9 chapters with the first chapter giving the introduction to EIT. It also introduces the application and methodology of the EIT. It introduces the need for the boundary estimation of the anomalies using prior information. The related boundary estimation work is presented and the overview of the thesis is stated. Chapter 2 introduces the governing equation with the mathematical models used in solving the EIT problem. A complete electrode model is used in this thesis as the physical model. Finite element formulation based on the mathematical model is briefly explained. Various data collection methods used in EIT are discussed briefly.

Chapter 3 explains the cost function used in the inverse problem. This chapter enhances the need for a fast convergence algorithm and is free of Jacobian matrix calculation. The studies were done on a heuristic algorithm as an inverse problem algorithm is presented. Also, the development of a neural network with EIT is briefly discussed. Chapter 4 introduces the heuristic algorithm i.e. gravitational search algorithm and particle swarm optimization. The formulation of both algorithms is illustrated in the chapter along with the introduction of a hybrid heuristic algorithm. A hybrid PSOGSA algorithm is explained with the mathematical formulation. Chapter 5 presents need to tune the DNN model hybrid-parameter. Along with the proposed method of tuning. From chapter 6 to 8 presents the study result of the boundary estimation done in this study. All 3 chapters present a study conducted on the closed and open boundary estimation problem. Chapter 6 introduces the bladder boundary formulation along with the numerical and experimental study. The bladder boundary estimation is done

with the GSA in a numerical and phantom study. The hybrid algorithm PSOGSA is used as the inverse problem solver to estimate the defect geometry on the graphene sheet as presented in chapter 7. The numerical and experimental study for defect detection with PSOGSA is presented in this chapter. Chapter 8 presents the result of analysis of the hyper-parameter tuning. The tuned DNN model is used to estimate the interlayer boundary of the subsurface and the estimation results are presented along with the proposed method of tuning the hyper-parameter of the DNN model. Finally, chapter 9 presents the conclusion of the thesis and future work is envisaged.

2 Forward problem

EIT is a non-invasive image reconstruction technique that is composed of two steps, i.e., forward problem and inverse problem. Generally, a conductivity distribution is reconstructed based on the injected current and the responding voltage measurement readings from the surface-mounted electrodes of a domain. In the forward problem, a constant current is injected into the domain, and voltages are computed inside the domain (Ω) and on the electrode surface ($\partial\Omega$) with the current and known conductivity distribution. In an inverse problem, the reconstruction of an image of internal properties is done using voltage measurements and the injected current. A mathematical (physical) model is needed to describe the problem in EIT to derive the forward problem solution. The physical model is derived from Maxwell's equation of electromagnetism for EIT (Cheney et al., 1999). The potential distribution of the domain and the boundary conditions are discussed as it is necessary to solve the governing equation. In this study for solving the boundary estimation problem, a complete electrode model (CEM) is used as a physical model as it is efficient and accurate when compared to other models (Cheng et al., 1989). The forward problem is formulated with the help of the finite element method (FEM). The data collection methods are also discussed later in this chapter.

2.1 Governing equation

Maxwell equations of electromagnetism is used to derive the mathematical model for EIT. These equations provide a relation between current, voltage measurement, and conductivity distribution. A partial differential equation derived from Maxwell's equation gives the relationship between the internal conductivity $\sigma(x, y)$ and the

electrical potential $u(x, y)$ on the domain ($\Omega \in R^2$) (Somersalo et al., 1992). Let us consider E as an electrical field, B as magnetic induction, H as magnetic field, J as current density and D as electric displacement, then the electromagnetic field in the domain $\Omega \in R^2$ can be described as

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad \text{Faraday's Law} \quad (2.1)$$

$$\nabla \times H = J + \frac{\partial D}{\partial t} \quad \text{Coulomb's Law} \quad (2.2)$$

Assuming the domain Ω consists of linear and isotropic medium, following holds true.

$$D = \epsilon E \quad (2.3)$$

$$J = \sigma E \quad (2.4)$$

$$B = \mu H \quad (2.5)$$

where ϵ , σ , and μ is permittivity, conductivity, and permeability of the medium. If the injected currents are time harmonic with frequency ω , then we get following

$$E = \bar{E}e^{i\omega t} \quad (2.6)$$

$$B = \bar{B}e^{i\omega t} \quad (2.7)$$

Using equations (2.3) to (2.7) for solving equations (2.1) and (2.2), we get

$$\begin{aligned} \nabla \times E &= -\frac{\partial B}{\partial t} = -\frac{\partial(\bar{B}e^{i\omega t})}{\partial t} \\ &= -i\omega\bar{B}e^{i\omega t} - e^{i\omega t}\frac{\partial\bar{B}}{\partial t} = -i\omega\mu H - e^{i\omega t}\frac{\partial\bar{B}}{\partial t} \end{aligned} \quad (2.8)$$

$$\begin{aligned} \nabla \times H &= J + \frac{\partial D}{\partial t} = J + \frac{\partial(\epsilon E)}{\partial t} = J + \frac{\epsilon\partial(\bar{E}e^{i\omega t})}{\partial t} \\ &= J + i\omega\epsilon\bar{E}e^{i\omega t} + \frac{\epsilon e^{i\omega t}\partial\bar{E}}{\partial t} = J + i\omega\epsilon E + \frac{\epsilon e^{i\omega t}\partial\bar{E}}{\partial t} \end{aligned} \quad (2.9)$$

A time harmonic current is assumed as an injected current having frequency (ω) and the medium is conductive, then the current density (J) can be separated into two components, i.e., ohmic current ($J^o = \sigma E$) and current source (J^s). A simplified

Maxwell's equation can be obtained after the oscillatory exponential terms are canceled and the medium is conductive (Doerstling, 1995; Ola et al., 1993; Somersalo et al., 1992).

We get

$$\nabla \times E = -\omega\mu H \quad (2.10)$$

$$\nabla \times H = (\sigma + i\omega\epsilon)E + J^s \quad (2.11)$$

The electrical field (E) can be expressed as

$$E = -\nabla u - \frac{\partial A}{\partial t} \quad (2.12)$$

where u is electric potential and A is magnetic vector potential.

The effect of magnetic induction which produces an induced electrical field is neglected after assuming a static conditions for EIT. Also assuming, the capacitive effects ($i\omega\epsilon E$) in equation (2.11) is neglected (Baker, 1989; Barber, 1984). The above equations can be simplified with these assumptions as

$$E = -\nabla u \quad (2.13)$$

$$\nabla \times H = \sigma E + J^s \quad (2.14)$$

Also the source current (J^s) is zero at a given frequency (ω) in EIT and taking the divergence on the both side of equation (2.14) and putting the value of equation (2.13) into (2.14), we get

$$\nabla \cdot (\nabla \sigma u) = 0 \quad (2.15)$$

where $u = u(x, y)$ for $x, y \in \Omega$. The above equation (2.15) represents the governing equation for EIT. To solve this equation we need to define a boundary conditions. Boundary conditions are defined with the help of physical models in EIT. The following section describe the boundary conditions and after that physical model is explained.

2.2 Boundary conditions

On the domain boundary ($\partial\Omega$) the current source (J^s) is not zero, then the boundary can be represented as

$$\nabla \cdot \sigma E = -\nabla \cdot J^s \quad (2.16)$$

Integrating both sides of the above equation over the volume v ,

$$\int_v \nabla \cdot \sigma E dv = - \int_v \nabla \cdot J^s dv \quad (2.17)$$

Implementing the divergence theorem on above equation, we get

$$\int_s \sigma E \cdot \vec{n} dS = - \int_s J^s \cdot \vec{n} dS \quad (2.18)$$

where S is the surface of v , and \vec{n} is the unit-normal vector. Since $J^s = 0$ inside the object whereas $E = 0$ outside of the object, the above equation (2.18) change the form to

$$-\sigma E \cdot \vec{n} \Big|_{inside} = -J^s \cdot \vec{n} \Big|_{outside} \quad (2.19)$$

is valid. Using the equation (2.13) and (2.19) a Neumann-type boundary condition is obtained and defined as

$$\sigma \frac{\partial u}{\partial \vec{n}} = -J^s \cdot \vec{n} \equiv j^n \quad (2.20)$$

where j^n is the negative normal component of the injected current density J^s .

2.3 Mathematical Electrode models

There are different physical models used in EIT such as the continuum model, gap model, average-gap model, shunt model, and complete electrode model (CEM). A continuum model assumes the entire surface of the domain as a conductor which overestimates the conductivity because the effect of electrodes is not considered. A gap model considers the effect of electrodes in the model overcoming the problem of the continuum model. The average-gap model is based on the gap model for the boundary conditions with only one difference. The voltage measured at each electrode

is considered as the average value of potential on that electrode whereas the gap model considers voltage values measured at the center of each electrode. Since the continuum model, gap model, and average-gap model ignore the shunting effect and the contact impedance, they all overestimate the resistivity distribution inside the domain (Somersalo et al., 1992). The shunt model takes the potential on the electrode as a constant for the account of the shunting effect of the electrodes but ignores the contact impedance that arises due to the electrochemical effect at the electrode surface, it underestimates the resistivity. The shunting effect of electrodes and contact impedance between electrode and domain surface is considered in CEM.

2.3.1 Complete electrode model

The CEM consist of a governing equation as defined in equation (2.15) and the boundary conditions which is defined as

$$u + z_l \sigma \frac{\partial u}{\partial v} = U_l, \quad x \in e_l, l = 1, 2, \dots, L \quad (2.21)$$

$$\int_{e_l} \sigma \frac{\partial u}{\partial v} dS = I_l, \quad x \in e_l, l = 1, 2, \dots, L \quad (2.22)$$

$$\sigma \frac{\partial u}{\partial v} = 0, \quad x \in \partial\Omega \setminus \bigcup_{l=1}^L e_l \quad (2.23)$$

where z_l is the contact impedance between l^{th} electrode and domain surface and I_l is the current applied to the l^{th} electrode. To ensure existence and uniqueness of the result, two conditions defined by equations (2.24), (2.25) for the injected current and measured voltages are needed.

$$\sum_{l=1}^L I_l = 0 \quad (2.24)$$

$$\sum_{l=1}^L U_l = 0 \quad (2.25)$$

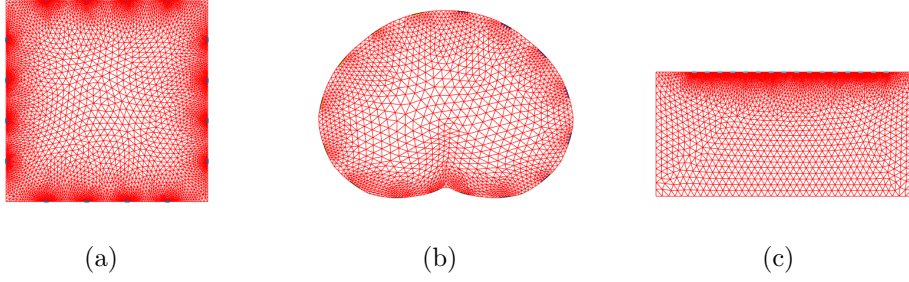


Figure 2.1: Mesh used in this thesis. (a)Graphene sheet mesh, (b)Pelvic mesh, and (c)Subsurface mesh.

2.4 Numerical Solution

The forward solution for the physical model is needed for solving the conductivity distribution. As the analytical solution is not feasible for a complex domain, a numerical method is used to find the solution for the Laplace equation. Numerical methods used for solving the partial differential equation or Laplace equation are finite difference method (FDM), finite element method (FEM), and boundary element method (BEM). FDM is easy to implement but in complicated boundary conditions and geometries, it is very difficult to implement and also suffers from slow convergence. In BEM only inhomogeneities boundaries are discretized but it is unfeasible in highly heterogeneous domains (de Munck et al., 2000; Khambampati et al., 2016). For complex geometries solving the partial differential equation, FEM is a feasible method (Bagshaw et al., 2003; Xu et al., 2005).

FEM is used in this work in solving the problem stated above as forward problem solution provider. The domain (Ω) is discretized into a small elements which are triangular as shown in figure 2.1. The triangle vertices are called nodes and the conductivity inside each triangular element are assumed to be constant. The potential distribution (u) within the domain can be approximate in finite dimension space can be written as

$$u = \sum_{i=1}^N \alpha_i \phi_i \quad (2.26)$$

where $\phi_i = \phi_i(x, y)$ is the two-dimensional first-order basis function and potential on

the electrodes are approximated as

$$U = \sum_{i=1}^{L-1} \beta_i n_i \quad (2.27)$$

where $n_1 = [1, -1, 0, \dots, 0]^T$, $n_2 = [1, 0, -1, \dots, 0]^T \in R^L$, etc. which ensures that the equation (2.25) is valid and α_i and β_i are the coefficients to be determined. Using finite element formulation a matrix equation is constructed as

$$Ab = \vec{I} \quad (2.28)$$

where $b = (\alpha, \beta)^T \in R^{N+L-1}$, $A \in R^{(N+L-1) \times (N+L-1)}$ is a sparse block matrix and \vec{I} is the data vector which are defined as

$$A = \begin{pmatrix} B & C \\ C^T & D \end{pmatrix} \quad (2.29)$$

$$\vec{I} = \begin{pmatrix} \vec{0} \\ \vec{I} \end{pmatrix} \quad (2.30)$$

where $\vec{0} = (0, \dots, 0)^T \in R^N$ and $\vec{I} = (I_1 - I_2, I_1 - I_3, \dots, I_1 - I_L)^T \in R^{N+L-1}$. Using the variational form with FEM basis functions ϕ_i and n_j the elements of the space block matrix A becomes

$$B(i, j) = \int_{\Omega} \sigma \nabla \phi_i \cdot \nabla \phi_j d\Omega + \sum_{l=1}^L \frac{1}{z_l} \int_{e_l} \phi_i \phi_j dS, \quad i = 1, 2, \dots, N \quad (2.31)$$

$$C(i, j) = -\frac{1}{z_l} \int_{e_l} \phi_i dS + \frac{1}{z_{j+1}} \int_{e_{j+1}} \phi_i dS, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, L-1 \quad (2.32)$$

$$D(i, j) = \begin{cases} \frac{|e_l|}{z_l} & i \neq j \\ \frac{|e_l|}{z_l} + \frac{|e_{j+1}|}{z_{j+1}} & i = j \end{cases}, \quad i, j = 1, 2, \dots, L-1 \quad (2.33)$$

where e_l is the area of the l^{th} electrode. Solving the above equation (2.28) as $b = A^{-1}\vec{I}$, the approximative solution is obtained for forward problem. The potentials on

electrodes are calculated with the assist of equation (2.27) as

$$\begin{aligned}
 U_1 &= \sum_{l=1}^{L-1} \beta_l \\
 U_2 &= -\beta_1 \\
 U_3 &= -\beta_2 \\
 &\vdots \\
 U_L &= -\beta_{L-1}
 \end{aligned} \tag{2.34}$$

The matrix form of above equation can be written as

$$U = M\beta \tag{2.35}$$

where $M \in R^{L \times (L-1)}$ is a spare matrix defined as

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & & & \\ & -1 & & \\ & & \ddots & \\ & & & -1 \end{pmatrix} \tag{2.36}$$

and $\beta = (\beta_1, \beta_2, \dots, \beta_{L-1})^T$.

2.5 Current injection method

For shape estimation inside a domain in EIT, the currents are injected through the electrodes, and voltages are measured. Since the voltmeter has a large input impedance, the contact impedance that exists between electrode and domain has a negligible effect. The current injection method can affect the performance of estimation as the sensitivity is different for each case. There are many current injection methods (Cheng et al., 1988; Webster, 1990) and the most commonly used in EIT are an adjacent method, opposite method, and cross method. In EIT, an image is reconstructed with the help of all the independent injected current patterns. The number of electrodes and current

patterns used helps to determine the total number of independent current patterns. In an adjacent method, a current is injected through two adjacent electrodes, and voltage is measured from all other electrodes as shown in figure 2.2a. This process is repeated for all the electrodes and the total number of voltage readings makes one frame of data. For example, a domain with 16 electrodes will have $16 \times 16 = 256$ voltage measurement, here even current-carrying electrodes are used to measure voltages.

In the opposite method, the current is injected through a pair of electrodes which are physically located opposite of each other in the domain presented in figure 2.2b. For example, a domain with 16 electrodes will have $16 \times 8 = 128$ number of voltage readings. In this, the number of voltage measurements will be half of the adjacent injection method because the current injection will repeat itself after the 8th injection pattern which is useless.

The Cross method is the combination of adjacent and opposite methods in which one electrode is the current reference sink and the current source is applied successively to the other electrodes. For each current pair, voltages are measured from all the electrodes.

All the current injection methods discussed in this section are suitable for the system of a single current source. Also, the selection of the current injection method depends on the problem for which it is applied. In this thesis opposite method is used for the current injection and voltages are measured on all electrodes for all the injected sources.

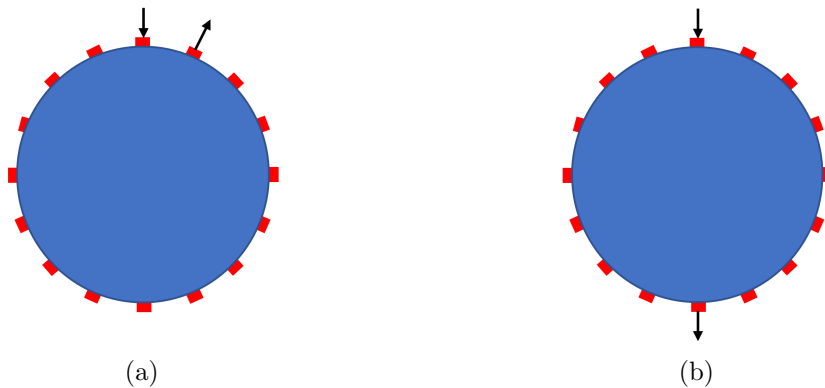


Figure 2.2: Adjacent and opposite current injection electrode selection for first current pattern.

3 Inverse problem

3.1 Introduction

As EIT is composed of a forward problem and inverse problem for the reconstruction of the image. The forward problem involves the calculation of the boundary voltages based on the given conductivity distribution and injected current. However, the inverse problem involves the reconstruction of the image presenting the conductivity distribution of the domain based on the cost function. The cost function is the error between the measured voltage and the calculated voltage. It is mathematically defined as

$$\Phi = \frac{1}{2}[V - U]^T[V - U] \quad (3.1)$$

where V is the measured voltage on the boundary electrodes and U is the calculated voltage obtained through FEM formulation. In EIT the inverse problem solver is the iterative algorithm that updates the estimation result based on the cost function. The main aim of the algorithm is to achieve a lower-valued cost function per iteration.

3.2 Heuristic algorithm

The conventional inverse problem algorithms such as mNR often show low performance when solving EIT inverse problems due to the ill-posed nature of the problem. Also, due to the highly nonlinear nature of the inverse problems, it is very difficult to solve the inverse problem in realistic conditions. The ill-posedness of the EIT can be reduced by transforming the conductivity distribution estimation problem into the boundary estimation problem. But the analytical solution to the EIT boundary estimation problem is very difficult to achieve. Also, mNR heavily depends on the initial guesses,

i.e. if the initial guess needs to be close to the true distribution for the very complex scenario. But mNR tends to produce intersecting boundaries for the complex shape and can get stuck at local minima.

An algorithm that can bypass or filter out the local minima to achieve an optimum solution with fast convergence is desired. Heuristic algorithms are algorithms that have a fast convergence rate. When compared to the traditional inverse algorithm applied to EIT, these algorithms do not require computation of Jacobian matrix and also outperform gradient-based method. A heuristic algorithm PSO has been studied as an inverse problem solver in EIT. In (Kumar et al., 2010) PSO was used to improve the reconstruction of brain EIT images and in (Ijaz et al., 2007) it was used to estimate the elliptic region boundary in EIT. PSO demonstrated good estimation for the EIT experiments however the convergence rate is very low. To overcome the limitation of PSO, GSA is used in this work as an inverse problem solver. GSA was first introduced by Rashedi et al. which showed a better performance than PSO in (Rashedi et al., 2009).

A hybrid heuristic algorithm is also known as the advanced heuristic algorithm where two heuristic algorithms are combined to make a new algorithm. In EIT very few hybrid heuristic algorithms are used as the inverse problem solver. In Mendoza and Lope (2012) a hybrid genetic algorithm was used to reconstruct images in EIT. Inspired by this, we have used the PSOGSA hybrid algorithm for estimating the boundary of the defects in the graphene sheet. The mathematical formulation of GSA, PSO, and PSOGSA are presented in chapter 4 and the details of the estimation by GSA and PSOGSA are illustrated in chapters 5 and 6.

3.3 Neural network

Nowadays due to big data availability, study on the neural network (NN) has increased and research works are being done to implement it in a variety of fields. A deep neural network (DNN) is the most researched algorithm in NN. DNN is the advanced version of NN where the number of hidden layers between the input and output layer is more than three. Neural Network has been used with EIT as an inverse solver for

estimating the boundary of the targets (Konki et al., 2020; Park et al., 2021). The NN is illustrated in chapter 7 along with the training and validating concept of the model. The hyperparameters of the model are also explained in the chapter. NN requires tuning of the hyper-parameters before training of the model is conducted. There are many hyperparameters present in the NN model and tuning is a very time taking task. The number of nodes in the hidden layer is one of the important hyperparameters of the NN model and finding the optimum value is very hard. Normally grid search algorithm is used to estimate the optimum value but it is a very time taking process. To solve this problem we have used the PSOGSA algorithm in this work to tune the number of nodes of the hidden layer. The process flow of the PSGOSA for tuning this parameter of the DNN model is explained in chapter 8 along with the numerical study.

4 Heuristic algorithm

4.1 Introduction

In high-dimensional search space, a classical optimization algorithm fails to provide a suitable solution because search space increases the problem size exponentially. Due to this solving these problems using exact techniques i.e. exhaustive search is not practical. To overcome this problem, heuristic techniques are studied as an alternative solution. The “heuristic” is a Greek word which means “to know”, “to find”, “to discover”, or “to guide a investigation” (Lazar, 2002). In other words, “Heuristics are techniques which seek good (near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particularly feasible solution is” (Russell and Norvig, 2002). Heuristic techniques use knowledge of previously tried solutions to guide the search for an optimal solution.

Heuristic algorithms are inspired by physical or biological processes. Genetic algorithm, simulated annealing, artificial immune system, ant colony optimization, and bacterial foraging algorithm are a few of the popular heuristic algorithms. Different heuristic algorithms are inspired by different physical or biological processes. Genetic algorithm is inspired by Darwinian evolution theory (Tang et al., 1996), simulated annealing is based on the thermodynamic effects (Van Laarhoven and Aarts, 1987), artificial immune system mimic biological immune systems (Farmer et al., 1986), ant colony optimization is based on the behavior of ants gathering for food (Dorigo et al., 2006), and bacterial foraging algorithm is based on search and optimal foraging of bacteria (Kim et al., 2007a).

Normally, the heuristic algorithms have a stochastic behavior where a search starts from

a single point and sequentially continues. However, most of the heuristic algorithms search in a parallel manner where multiple initial points are used to search. These points are referred to by many names such as particles, agents, or objects, and are the solution candidates for the problem at hand. For example, swarm-based algorithms use a collection of particles similar to a natural flock of birds where each particle executes a series of a particular operations and shares the information with others. This sharing of information between particles is known as swarm intelligence (Tarasewich and McMullen, 2002) and this local interaction between particles provides a solution to a problem without any central controller.

Exploration and exploitation are the two common aspects of the heuristic algorithms where exploration is the ability to find the optimum solution in the search space and exploitation is the ability to find optima around a good solution. The algorithm uses exploration initially to avoid trapping in local minima and exploitation is used at a later time. For a high-performance search of the algorithm, a suitable tuning of the trade-off between exploration and exploitation is needed. For realizing the concepts of exploration and exploitation, the algorithm in each iteration passes through three steps self-adaptation, cooperation, and competition. The performance of each particle is improved in each iteration under the self-adaptation step. The information updated is shared between every particle in a cooperation step and final competition step particles compete to survive in each iteration. All these three steps are inspired by nature and help algorithms achieve an optimum solution. In the following sections, we describe a mathematical formulation of heuristic algorithms (GSA, PSO, PSOGSA) in detail. The adaptation of the algorithms for estimating in EIT as the inverse problem solver is also explained.

4.2 Gravitational search algorithm

The gravitational search algorithm (GSA) is an iterative algorithm based on the Newton law of gravity and motion (Rashedi et al., 2009). The particles in GSA are the solution candidate and are considered objects whose performance is measured by their masses. As per the law of gravity, gravitation is the tendency of masses to accelerate toward

each other with gravitational force. The gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the square of the distance between them (Halliday et al., 2013).

$$G = g_c \frac{M_1 \times M_2}{R^2} \quad (4.1)$$

where G is the gravitational force, g_c is the gravitational constant, M_1 and M_2 are the masses of the two particles, and R is the distance between the two particles. A particle accelerate towards the heavier mass particle with an acceleration a depending only on the gravitational force G and it's own mass M as defined in the law of Newton's second law (Halliday et al., 2013).

$$a = \frac{G}{M} \quad (4.2)$$

The gravitational force from each particle causes the global movement of all the particles towards heavier masses. The gravitational constant depends on the actual age of the universe (Mansouri et al., 1999), i.e. in the algorithm after every iteration, the gravitational constant value should be decreased and mathematically it is defined as

$$g_c(t) = (g_{c0}, t) \quad (4.3)$$

where g_{c0} is gravitational constant during initialization, and t is the iteration time. In theoretical physics related to gravitational force, masses are differentiated into three types (active, passive, and inertial mass). Active gravitational mass is a measure of the strength of the gravitational field due to a particular object. Passive gravitational mass is a measure of the strength of an object's interaction with the gravitational field. Inertial mass is a measure of an object's resistance to acceleration by a force. Conceptually these masses are different but the theory of general relativity assumes that inertial and passive gravitational mass are equivalent also known as the equivalence principle. Also, a strong equivalent principle assumes that inertial mass and active gravitational mass are equivalent (Kenyon, 1990).

Each particle is attracted to the other and particle with heavier masses are considered good solutions and move slower than particles having light mass. As GSA is based

on Newtonian laws of gravitation and motion, each particle has four specifications (position, inertial mass, active gravitational mass, and passive gravitational mass). The position of the particle represents a solution to the problem and its masses are determined by a fitness function. That is by properly adjusting the inertia and gravitational masses, the algorithm progresses towards the optimum solution.

Let's consider a system with N number of particles and define the position of the i^{th} particle by

$$\Gamma_i = (\lambda_i^1, \lambda_i^2, \dots, \lambda_i^d), \quad i = (1, 2, \dots, N) \quad (4.4)$$

where λ_i^d is the position of i^{th} particle Γ_i in the d^{th} dimension. The position are randomly assigned within the problem search space. Based on the law of gravity, the gravitational force acting on mass i from mass j at a specific time t is defined as

$$G_{ij}^d(t) = g_c(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij} + \epsilon} (\lambda_j^d(t) - \lambda_i^d(t)) \quad (4.5)$$

where M_{pi} is a passive gravitational mass of particle i , M_{aj} is an active gravitational mass of particle j , $g_c(t)$ is gravitational constant at t , ϵ is a small constant, and R_{ij} is the distance between two particles i and j . The distance between two particle's position in d^{th} dimension is expressed as the Euclidean distance, see equation 4.6.

$$R_{ij}(t) = \|\lambda_i(t), \lambda_j(t)\|_2 \quad (4.6)$$

Let the total force on an particle i by the other particles in a dimension d be randomly weighted sum of d^{th} dimension of force.

$$G_i^d(t) = \sum_{j=1, j \neq i}^N \text{rand}_j G_{ij}^d(t) \quad (4.7)$$

where rand_j is a random number $[0,1]$. Let $a_i^d(t)$ be the acceleration of particle i at time t , defined by the law of motion as

$$a_i^d(t) = \frac{G_i^d(t)}{M_{ii}(t)} \quad (4.8)$$

where M_{ii} is the inertial mass of the i^{th} particle. The new state velocity of a particle is defined as a sum of the fraction of its current state velocity and current state acceleration. Hence, the position and velocity of the particle is updated as

$$v_i^d(t) = rand_i \times v_i^d(t-1) + a_i^d(t-1) \quad (4.9)$$

$$\lambda_i^d(t) = \lambda_i^d(t-1) + v_i^d(t) \quad (4.10)$$

where $rand_i$ is a random number $[0,1]$. The gravitational constant $g_c(t)$ is initialized and reduced with time to control search accuracy.

$$g_c(t) = g_c(t_0) \times e^{(\kappa \times iter / iter_{max})} \quad (4.11)$$

where κ is descending coefficient, $iter$ is current iteration, and $iter_{max}$ is maximum number of iterations. A particle with heavier mass means more efficient particle which moves slow compared to other particles. Assuming the gravitational and inertia masses are equal, i.e.

$$M_{ai} = M_{pi} = M_{ii} = M_i, \quad i = (1, 2, \dots, N) \quad (4.12)$$

The mass is updated using equation (4.14) in which fitness function is used. The fitness function is the cost function which is defined based on the problem.

$$m_i(t) = \frac{\Phi_i - worst(t)}{best(t) - worst(t)} \quad (4.13)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (4.14)$$

where Φ_i is the fitness value of the particle i or the cost function of that particle, $best(t)$ and $worst(t)$ functions are cost function with best and worst values respectively and defined as

$$best(t) = \min_{i \in (1, \dots, N)} \Phi_i \quad (4.15)$$

$$worst(t) = \max_{i \in (1, \dots, N)} \Phi_i \quad (4.16)$$

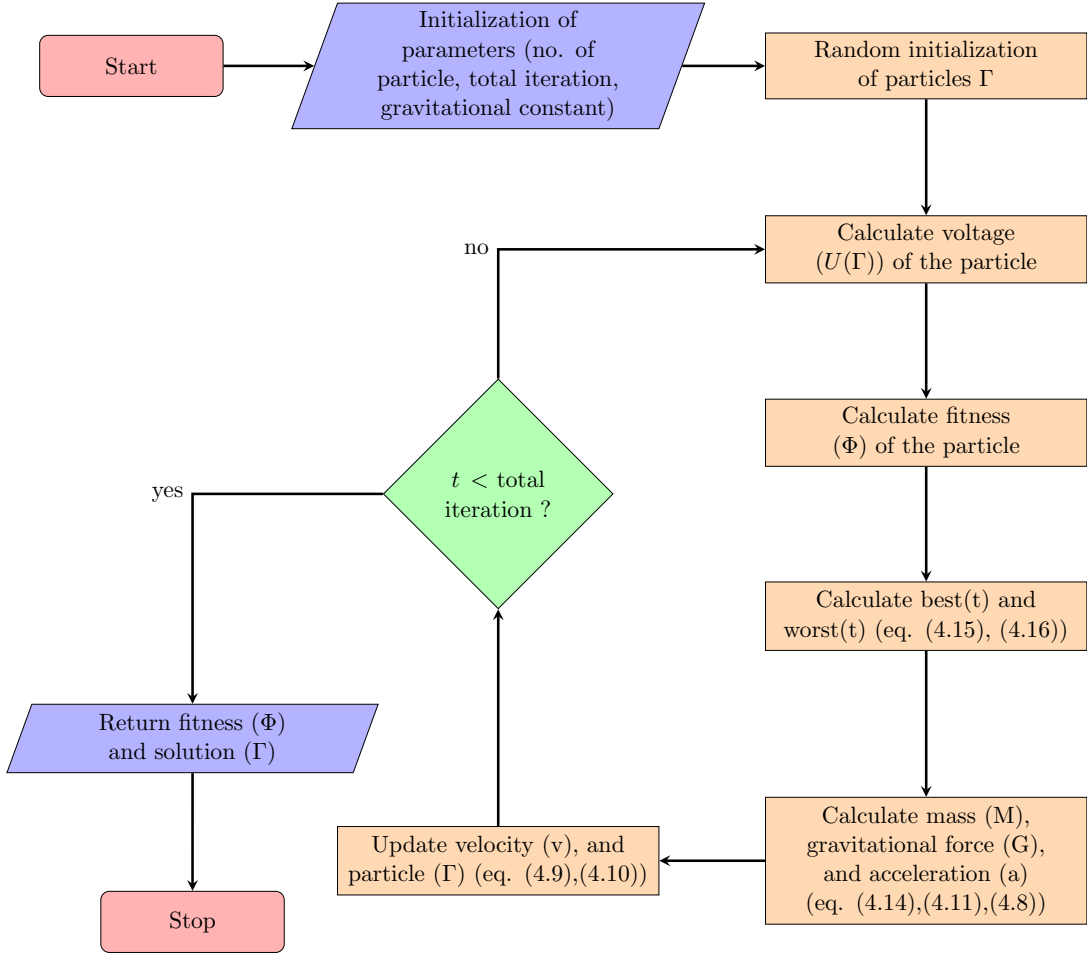


Figure 4.1: Flowchart for the gravitational search algorithm.

In the EIT problem, a cost function whose value is minimum is used as $best(t)$ and a function with maximum value is used as $worst(t)$.

Exploration of search space by GSA is required at the beginning stage to avoid local optimum. Exploration must fade out and exploitation must fade in with later iterations to find the best solution. To achieve optimum performance of GSA by controlling exploration and exploitation only $Nbest$ particles will attract the others. $Nbest$ particles will apply force and later decreased linearly with time. In end, only one particle will apply force to the others. To adjust $Nbest$, equation (4.7) need to be modified as

$$G_i^d(t) = \sum_{j \in Nbest, j \neq i}^N rand_j G_{ij}^d(t) \quad (4.17)$$

where $Nbest$ is the set of first N particles with the best fitness value and has the biggest mass.

Figure 4.1 presents the flowchart of the GSA to estimate the bladder boundary in the EIT. The solution candidates for estimating the bladder boundary are randomly initialized within the search space of the pelvic domain. The voltage for every particle is calculated and each fitness is evaluated. Based on this fitness the best and worst functions are updated. Later the mass, gravitational force, and acceleration of each particle are calculated. The velocity is updated for all the particles and the N_{best} particles are updated. After the total iteration is finished the best solution is presented as the optimum solution for the provided bladder boundary estimation. In this during every iteration gravitational constant is calculated and it decreased in each iteration.

4.3 Particle Swarm Optimization

Particle swarm optimization (PSO) is based on swarming theory. PSO was introduced in 1995 by Kennedy and Eberhart which was based on exploiting simple analogues of social interaction of a bird (Kennedy and Eberhart, 1995). PSO simulates the choreography of a bird flock and the potential solution is represented as a particle. PSO is initiated with a population of random solutions and then flown through the problem space. In each iteration, each particle keeps track of its position in the search space and the particle with the best fitness value is stored as $pbest$. A global best value so far obtained by any particles is also saved as $gbest$. Each particle's location is updated per iteration by changing the velocity towards $pbest$ and $gbest$ location until the iteration is exhausted.

Let's consider a PSO with N number of particles and the position of the i^{th} particle is defined by equation (4.4). The particle velocity and position are updated using equations (4.18) and (4.10) (Shi et al., 2001).

$$v_i^d(t) = w(t-1)v_i^d(t-1) + c_1rand_1(pbest_i^d - \lambda_i^d(t-1)) + c_2rand_2(gbest^d - \lambda_i^d(t-1)) \quad (4.18)$$

where $rand_j$ are random number $[0,1]$, c_j are positive constants, w is the inertia weight. $pbest_i$ and $gbest$ are the best previous position of the i^{th} particle and the best previous position among all the particles, respectively. The particle is updated using equation

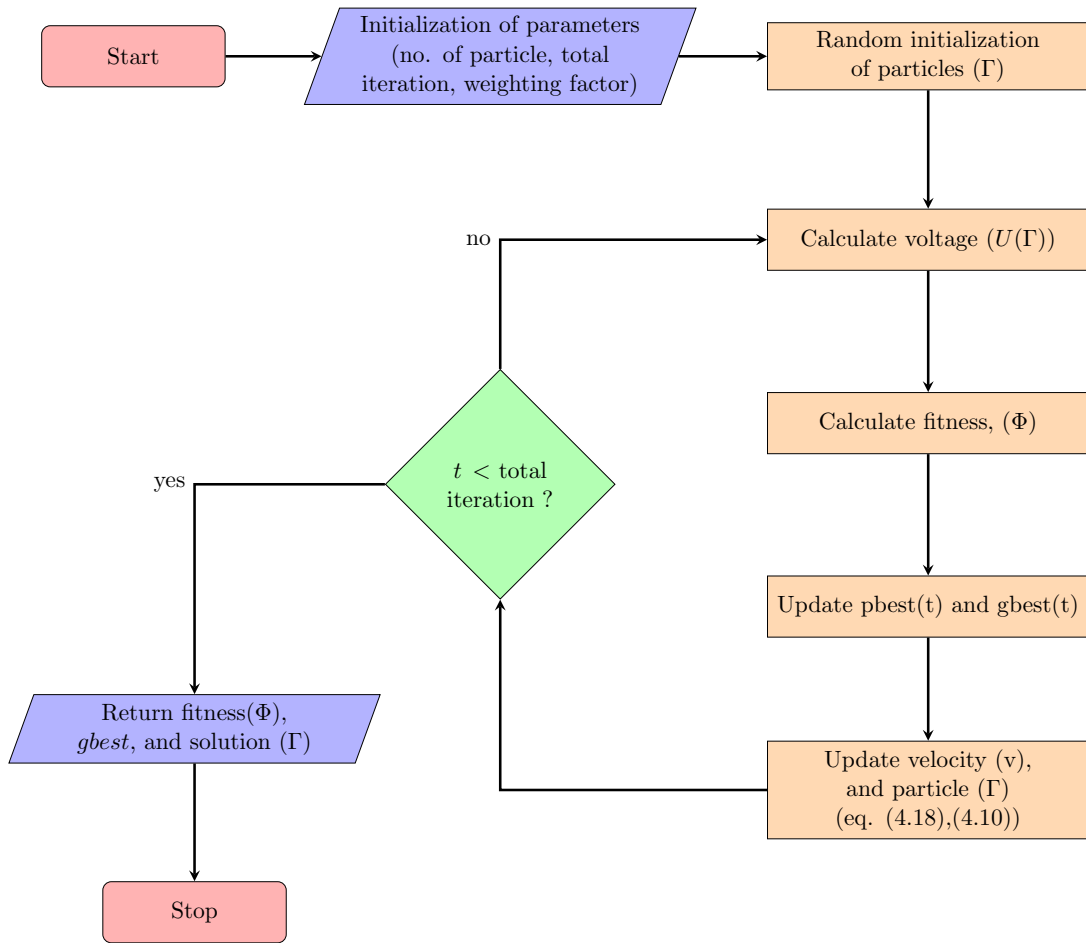


Figure 4.2: PSO flowchart for estimating the coefficient of Fourier series-defining bladder boundary.

(4.10).

Figure 4.2 presents the flowchart of the PSO algorithm. As previously stated PSO starts by randomly initializing the particle in the search space. The voltage for each particle is calculated and the fitness is evaluated using equation (3.1). The *pbest* and *gbest* are selected and the velocity is updated for each particle with equation (4.18). After updating the velocity the position of the particle is updated such that it moves towards the global minima. The process is repeated until the iteration is finished. After all the iteration is evaluated, the *gbest* particle and the optimum solution are returned by the algorithm.

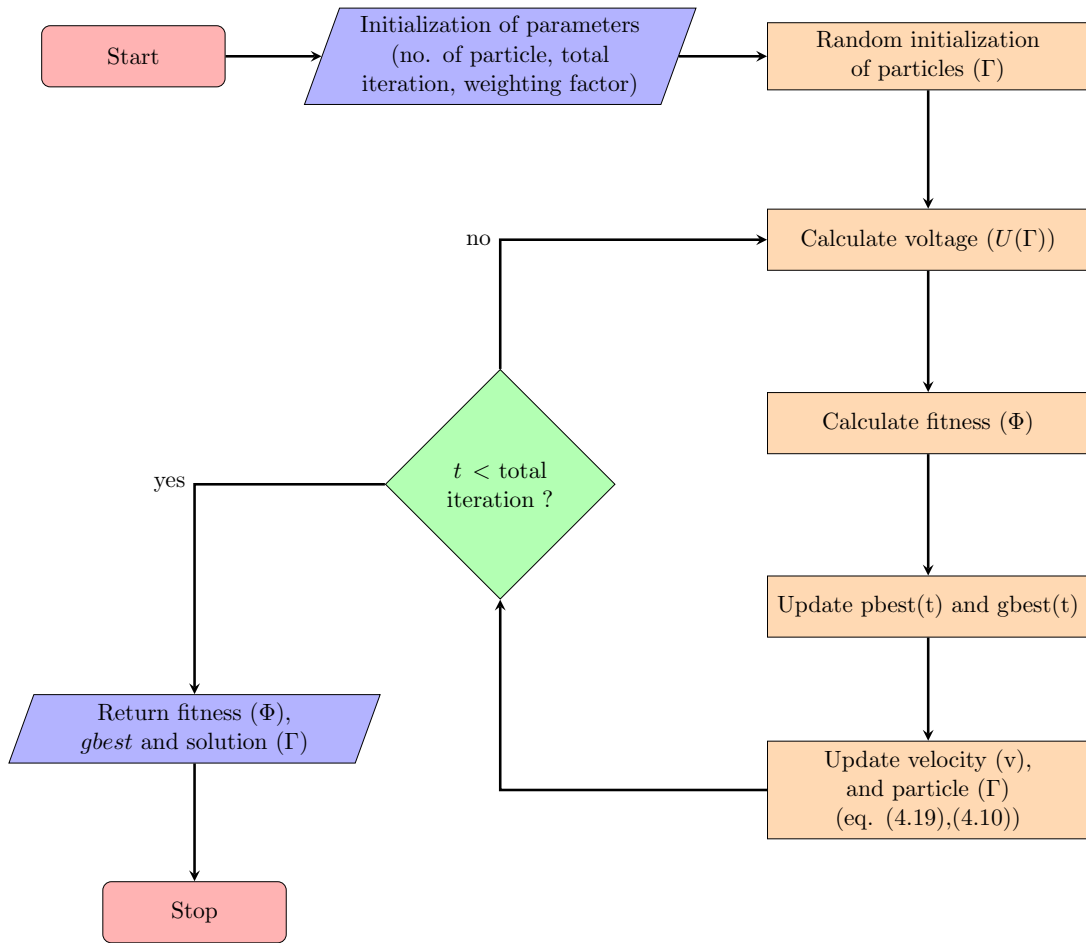


Figure 4.3: PSO-GSA flowchart for EIT estimation problem.

4.4 Hybrid PSO-GSA algorithm

A hybrid metaheuristic algorithm is the combination of two or more heuristic algorithms to perform as a single algorithm. There are mainly two types of classification of hybrid metaheuristic algorithm, i.e. low-level and high-level (Talbi, 2002). In a low-level metaheuristic algorithm, the internal workings of each combined algorithm have a direct relationship with each other. However, in the high-level metaheuristic algorithm, there is no direct relationship between the internal working of each algorithm. The hybrid PSO-GSA algorithm is a population-based metaheuristic algorithm that is a combination of PSO and GSA. The functionality of PSO and GSA is combined and run in parallel making it a low-level hybrid metaheuristic algorithm.

Since the PSO-GSA is the combination of PSO and GSA algorithm, the particles are defined using equation (4.4) having d dimensions. In PSO-GSA the ability of exploration

of PSO (*gbest*) is combined with the exploitation capability of GSA (Mirjalili and Hashim, 2010). The equation below combines the ability and capability of both algorithms in the same equation.

$$v_i^d(t) = w \times v_i^d(t-1) + c'_1 \times rand_1 \times a_i^d(t-1) + c'_2 \times rand_2 \times (gbest^d - \lambda_i^d(t-1)) \quad (4.19)$$

where $v_i^d(t)$ is the velocity of particle i at time t , c'_1 and c'_2 are the weighting factor, w is weighting function, $rand_1$ and $rand_2$ are the random numbers $[0,1]$, $a_i^d(t-1)$ is the acceleration of particle i , and *gbest* is the best solution so far achieved by algorithm.

Particles in the algorithm are randomly initialized and the forces of each particle are calculated using equation (4.17). The resultant forces between all particles, acceleration, and gravitational constant are calculated using equations (4.7), (4.8), and (4.11), respectively. In each iteration, the best solution is updated and velocities of all particles are updated using equation (4.19). The position of a particle is updated with the same equation as GSA i.e. equation (4.10).

Sine the PSOGSA is an iterative algorithm, the velocity and position of all the particles are updated per iteration. The best particle presenting the best solution so far is considered as per the fitness of the particles. The fitness is considered for updating the particles and considering the best solution so far. When the particles are near a good solution, it tends to move slower than the other particles. PSOGSA uses *gbest* to save the best solution found so far and the velocity of each particle is updated according to the found best solution.

The figure 4.3 presents the pictorial view of how the PSOGSA updates the particles in each iteration. Initialization of the parameters along with randomly assigning the particle position in the search space of the problem is the first step of PSOGSA. The voltage of each particle is calculated based on the position of the EIT and the fitness value is calculated. Based on this fitness value the best particle and the global best particles are selected. The velocity, acceleration, and position of the particles are updated based on this fitness value. The algorithm is iterated again and the position of the particles is updated. In the end, the algorithm returns the optimum position of the

best particles which is the optimum solution for the presented problem. The PSOGSA is used later in chapter 6 to estimate the defect geometry on the graphene sheet with EIT.

5 PSO-GSA tuned DNN

5.1 Deep neural network

A deep neural network (DNN) is the subset of machine learning which contains more hidden layers than another neural network. The DNN model is trained to update the weights of the node in the hidden layers of the model. The training is performed using the training dataset which contains the input and the output values. While training the DNN model, the back-propagation process suffers from three main difficulties, They are as follows

- Vanishing gradient
- Over-fitting
- Computational load

The error calculated at the output layer of the DNN model during training sometimes does not reaches the first hidden layer. This phenomenon is known as vanishing gradient as the error has vanished before it reaches all the nodes. The vanishing gradient problem is solved by using an activation function. RELU is the most widely used activation function as it transmits error better because it returns 0 for negative input value and provides positive same value which is in input. RELU function is shown in figure 5.1 expressed as

$$\Theta(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (5.1)$$

Over-fitting occurs when a neural network tries to map every features or pattern in the training data which gives a poor performance on unseen data. Due to this neural

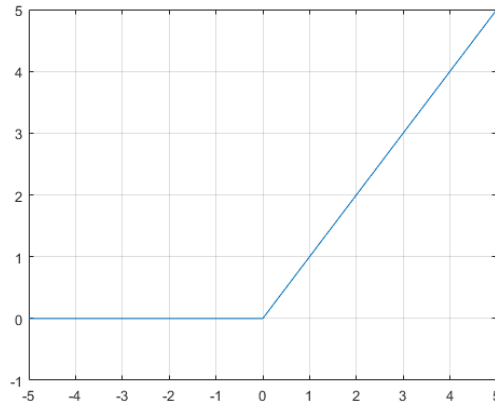


Figure 5.1: Rectified Linear Unit (ReLU) function.

network fails to generalize the features found in the training data. Regularization (L1, L2), and dropout methods are used to solve the over-fitting problem in a deep neural network. The concept of dropout is further illustrated in figure 5.2. Randomly selecting nodes and weights during the training process helps dropout to overcome the over-fitting problem.

Since the DNN model has more than 2 hidden layers, it requires more training data to avoid vanishing gradient. In using more training data the computational time for nodes' weights requirement increases geometrically. For better results from the model more computations are performed which takes a longer time to complete and is not suitable for the model development. The introduction of high-performance hardware, like GPU, and algorithms, like batch normalization, has provided aid to the computational load.

5.2 Training of DNN

DNN model is trained using a training dataset that contains input and output data. The relationship between input and output is learned by the model during the training process. The node's weights are updated with an activation function in the training process. In other words, the node's weights are updated during training to map the input and output data relationship.

During training nodes in the hidden layers are activated before it passes the output to the next layers. The input to the hidden layer is expressed as a vector (X) which is

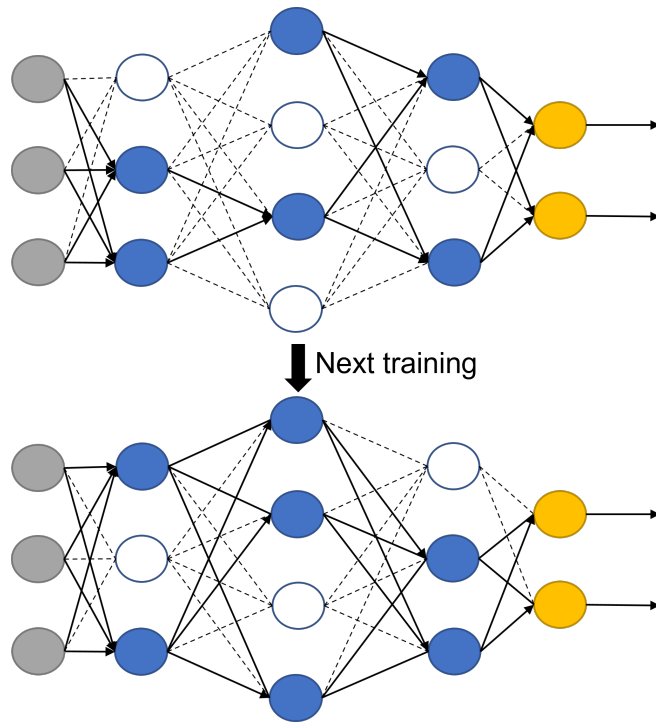


Figure 5.2: Dropout concept illustration where nodes are randomly selected and their outputs are set to zero to present as a deactivate nodes.

provided initially from the input layers. This vector is then combined with the weight (w) of the node and then the resultant of the product ($W \times X$) is passed to the activation function (Θ) along with the bias if available. The output from the first hidden layers can be mathematically expressed as

$$H_1(x) = \Theta(W_1X) \quad (5.2)$$

where (H_1) is the output vector from the first hidden layer and (W_1) is the weight matrix of nodes in the first hidden layers. An activation function is used to activate the neurons of the hidden layer in DNN. In other words, the activation function computes the input values of the layer to the output values. If the neuron is activated then it will provide input to the next layer's neurons. In this thesis, we have used rectified linear unit (RELU) as an activation function.

The set of parameters known as hyper-parameter is not learned by the model during training (Bergstra et al., 2011). However, these parameters have a major impact during the training of the model for a more accurate and efficient model. The following section

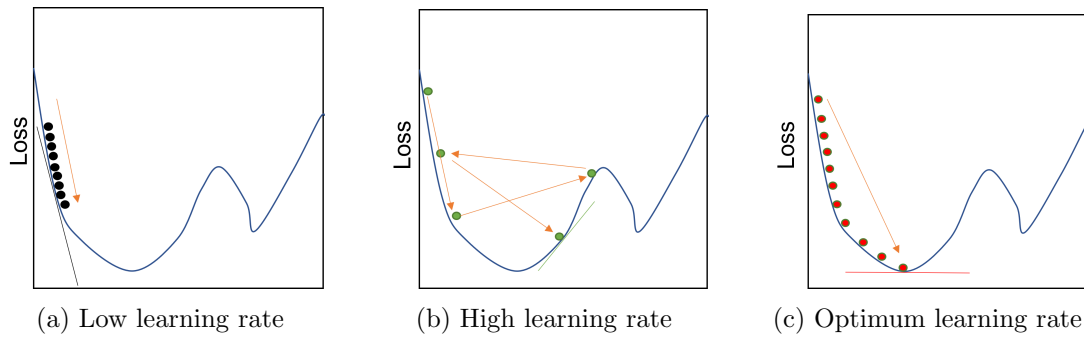


Figure 5.3: Learning rate illustration.

introduces a few hyper-parameters and methods to optimize them.

5.3 Hyper-parameter

Learning rate, number of hidden layers, number of nodes of different layers, and batch size are few important hyper-parameters required to be optimized for the better tuned DNN model. Learning rate defines the step size for the model to reach the accuracy level or the minimum loss function. If the learning rate is higher the model learns faster but may miss the minimum loss function. On the other hand, a lower learning rate has a higher chance to find the minimum loss function. However, it will take more time and computational resources to reach the minimum loss function. A good learning rate is a trade-off between minimum loss function with lower epochs and lower computational resources. The figure 5.3 illustrated the difference between higher, lower, and optimum learning rates.

The batch size is used to sample down the large training dataset to decrease the requirement of memory. If the training dataset is too large and it takes a longer time to train a model, which is solved by using batch size. Batch size helps to train the model fast by dividing the training dataset which required less computational resources.

The number of hidden layers in the DNN model is a very important hyper-parameter. The accuracy of the model will increase with more hidden layers but there is a problem of vanishing gradient during training. In other words, fewer layers will give under fitted model and a large number of layers will give over-fitted model. The number of nodes in the hidden layers also affects the accuracy of the model. The number of nodes per hidden layer traditionally is taken in a way that makes a funnel shape between input

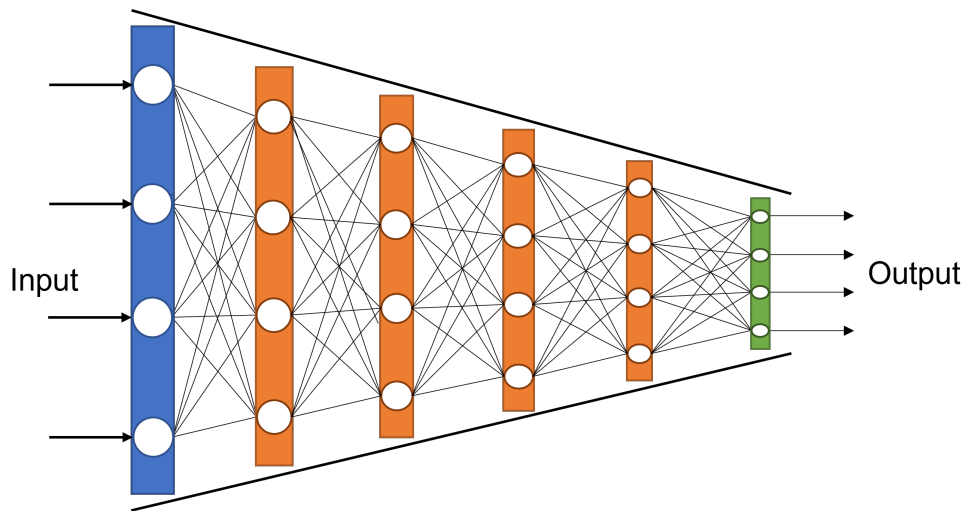


Figure 5.4: A schematic diagram of the DNN model designed as funnel.

and output layers as shown in figure 5.4. Choosing the right number of nodes per hidden layer is a very hard and time taking task.

5.4 Hyper-parameter tuning

As stated in the previous section, hyper-parameters are not learned by the model during the training. However, hyper-parameter tuning is required for optimizing the model before the training of the model is executed. We need to define how many layers and nodes are required to build the model before training the model. Different dataset requires a different set of hyper-parameters to predict with accuracy. Hyper-parameter tuning is very important as it helps to find the best sets of hyper-parameters for that particular DNN model for a specific dataset.

Manual attempts to initialize the hyper-parameter to the optimal value do not give a good-performing model. It is a very hard and almost impossible task to initialize the hyper-parameters for the complex DNN model. Alternatively, we can use the grid search method but it is time-consuming and are not suitable for time-bound predictions (Bergstra et al., 2011). These manual and grid search methods are prone to biases and suffer from scheduling problems. Bergstra and Bengio in 2012 introduced the random search algorithm to tune the hyper-parameter of the NN model. This algorithm randomly picks the combination of the hyper-parameter from the grid of options to minimize the NN cost function. However, the random search method suffers

from the fluctuating cost functions and slow convergence rate (Schumer and Steiglitz, 1968). Another optimization algorithm used was Bayesian optimization which was introduced in (Xia et al., 2017) had improved performance over manual, grid-search, and random methods. However, it was slow and was under-performing efficiently for dynamic problems.

The study conducted by Rere et al. showed that the meta-heuristic algorithm can be used to optimize the hyper-parameter of the CNN. The author used simulated annealing, different evolution, and harmony search algorithm to tune the CNN model. The author also showed that the meta-heuristic algorithm does not suffer from the problems which are faced by the above methods (Rere et al., 2016). Many studies were conducted to use a genetic algorithm (GA) as the optimizer for hyper-parameter of ANN model (Di Francescomarino et al., 2018; Wicaksono and Supianto, 2018; Xiao et al., 2020). Also, hyper-parameter of multi-layer perceptron (MLP) were searched by GA (Harvey, 2017).

In the deep learning network, the search space of hyper-parameters increases tremendously when compared to the simple NN. Due to this the training time of the DNN model is very long (Stamoulis et al., 2018). As the time required to train the DNN is long, hyper-parameter tuning with proper configuration is a major requirement. As stated earlier the grid-search optimizer can guarantee to find the best configuration for the hyper-parameter but it is very slow. This problem can be solved by using meta-heuristic approaches which are capable of finding a near-optimal combination of hyper-parameters in less time (Ojha et al., 2017). Inspired by this work PSOGSA a hybrid meta-heuristic algorithm is proposed and used to tune the DNN model hyper-parameters. PSOGSA is implemented to overcome the issue stated above. The hyper-parameter is tuned to an optimum value in a reasonable time and computational resources.

5.5 Tuning with PSOGSA

The most important hyper-parameter which affect the accuracy of the DNN model and also has a major impact on the time and computational resources needed to train

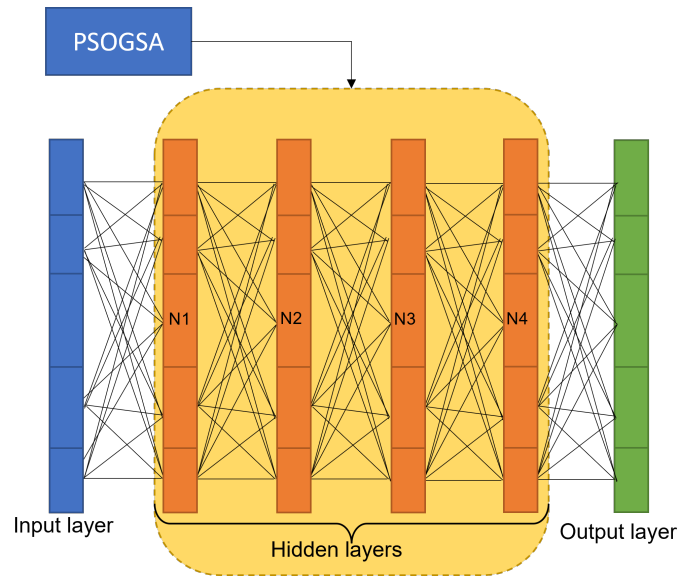


Figure 5.5: A schematic diagram of PSO-GSA tuned DNN model.

the model is the number of nodes in the layers. The number of nodes has a direct relationship with the time and computational resources required to train the model. If the model has a high number of nodes in the layer, it might be more accurate but it consumes more resources. A large number of nodes can also make the model over-fitted on the particular dataset used in the training.

In this thesis, we have used PSO-GSA to tune the number of nodes in hidden layers of the DNN model. PSO-GSA is used to find the optimum number of nodes for given hidden layers of the model. A schematic diagram of the proposed method to tune the hyper-parameter of the model is shown in figure 5.5. In the proposed method we tune one hyper-parameter for optimum value to achieve a lower cost function of the model. The node size of hidden layers of the model is tuned with the proposed method. The model is configured with four hidden layers and each layer's node is initialized by PSO-GSA based on the model cost function.

The figure 5.6 shows the pictorial view of the hyper-parameter tuning process. The tuning process is mainly divided into two-part. The first part is the training of the NN model where nodes' weights are updated using the dataset. In this process, the network is iterated to find the optimized node's weight of every hidden layer which gives the lower cost function. In the second part of the process, the cost function values are fed into the PSO-GSA algorithm which updates the position of the particle

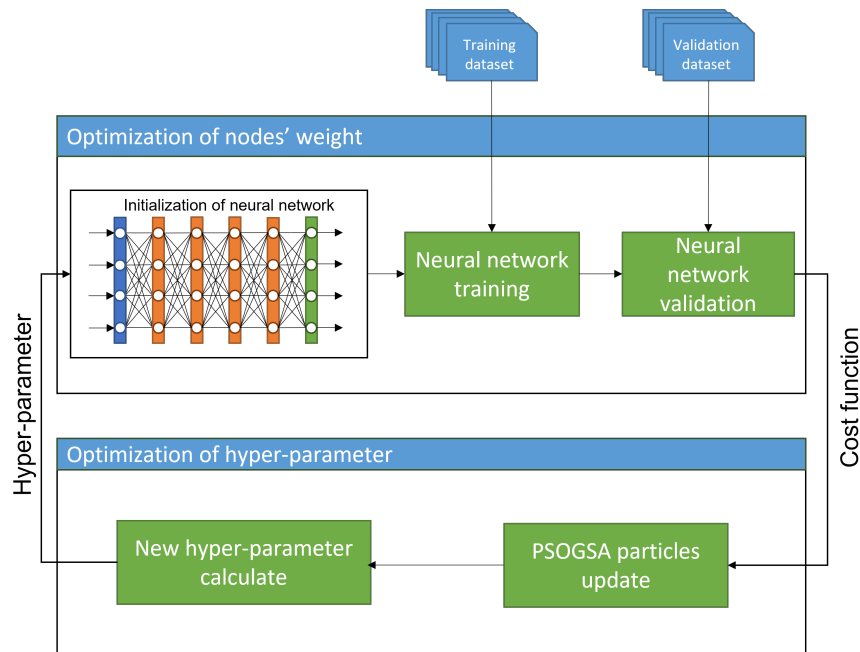


Figure 5.6: Hyper-parameter tuning flow-chart.

on this feedback. The PSO then produces the new sets of hyper-parameter which are used to train the model. During this process, we update the hyper-parameter to the optimized value which gives the low-cost function for that particular dataset. involve with the training of the NN and other parts of the process involve updating the hyper-parameter initialization.

Optimizing node size

In this work, the number of nodes in the hidden layers of the DNN model is tuned with PSO. The PSO is used to initialize the number of nodes and update it again based on the cost function returned by the model. The figure 5.7 shows the flow of how the PSO updates the required number of nodes for the DNN model. The particles provide the optimum value for configuring the hidden layer node size.

Initially, particles are randomly initialized within the search space and each particle is passed to the DNN model. The DNN model uses the solution provided by PSO as the optimum number of nodes in the hidden layers. A model is built with the initialized hyper-parameters and trained with the training dataset. For every particle provided by the PSO, the DNN model is trained and the cost function is passed back to the PSO algorithm after completion of the training of the DNN model.

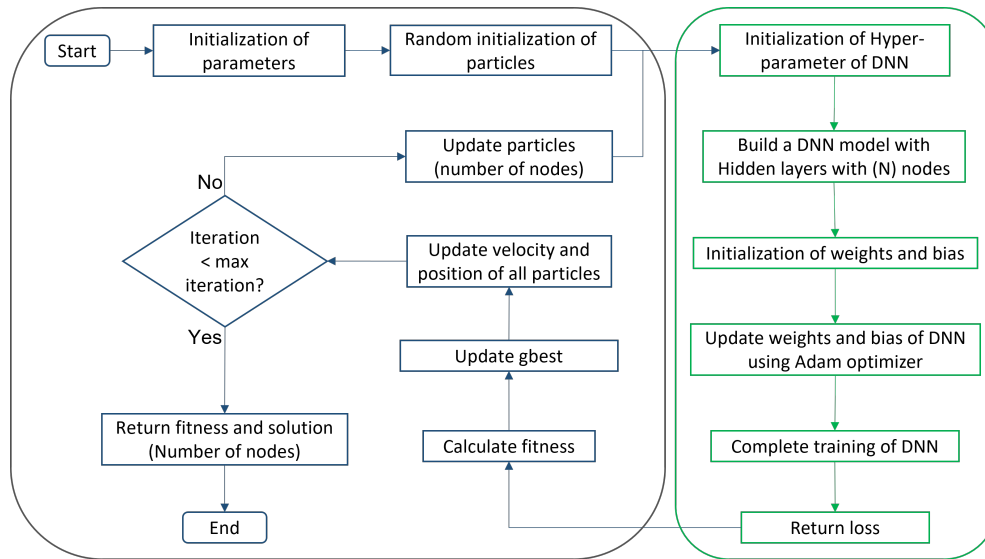


Figure 5.7: Flowchart of DNN with PSO-GSA tuning the hyper-parameter.

The cost function is used as a fitness of the particles in the PSO-GSA and based on the fitness value, the best particle is selected. The velocity and position for all the particles are updated and the updated particles are again used by the DNN model for configuration and trained again. These steps are repeated until the iteration of the PSO-GSA algorithm is finished. The PSO-GSA algorithm returns the best fitness value and the best particle. This particle is the best-optimized value for the hyper-parameter of the DNN model.

6 Bladder boundary reconstruction with GSA

This chapter presents the bladder boundary estimation by GSA. The boundary is defined by Fourier series coefficients assuming the conductivity inside the bladder is constant. The contrast of the conductivity between the organ and the surrounding tissues helps in estimating the bladder boundary. The following section defines how the bladder is represented and the later section presents the study.

6.1 Boundary representation of the bladder

A pelvis domain (Ω) is assumed to have the smooth boundary ($\partial\Omega$) which contains the bladder and neighboring organ. The Fourier series coefficients are used for describing the smooth boundary of the bladder and neighboring organ (Kolehmainen et al., 2001).

$$B_m(s) = \begin{pmatrix} x_m(s) \\ y_m(s) \end{pmatrix} = \sum_{f=1}^{F_\phi} \begin{pmatrix} \lambda_f^{x_m} \phi_f^x(s) \\ \lambda_f^{y_m} \phi_f^y(s) \end{pmatrix} \quad (6.1)$$

where $m = (1, 2, \dots, M)$, M is the number of organ in the pelvis domain Ω , $B_m(s)$ is the boundary of the bladder or the neighboring organ, F_ϕ is the order of the truncated Fourier series and $\phi_f(s)$ is the basis function which is periodic and represented as

$$\phi_1^\mu(s) = 1 \quad (6.2)$$

$$\phi_\theta^\mu(s) = \sin\left(2\pi\frac{\theta}{2}s\right), \quad \theta = 2, 4, 6, \dots, F_\phi - 1 \quad (6.3)$$

$$\phi_\theta^\mu(s) = \cos\left(2\pi\frac{(\theta-1)}{2}s\right), \quad \theta = 1, 3, 5, \dots, F_\phi \quad (6.4)$$

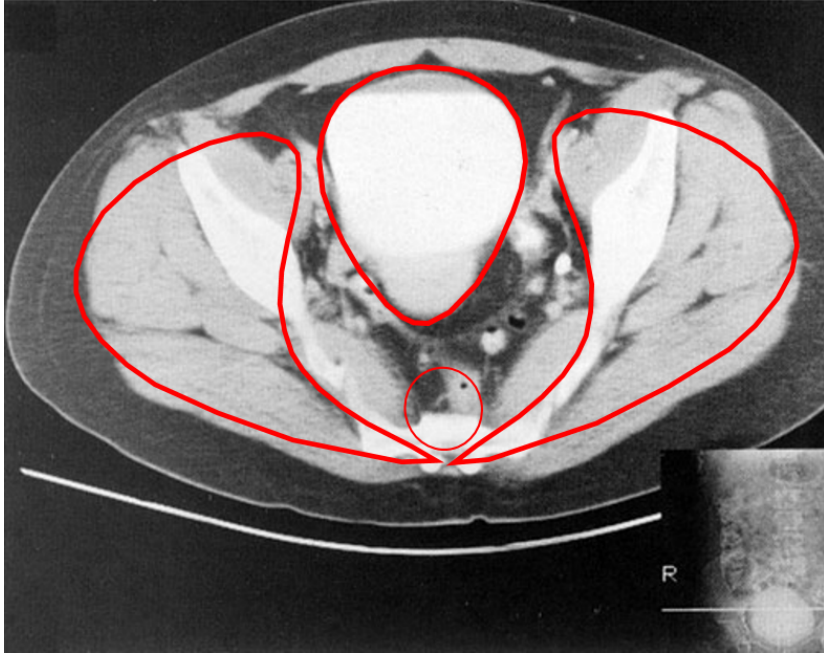


Figure 6.1: CT scan image of pelvis (Anjos et al., 2007). The bladder is the center top oval shape region with side muscles and bones included in two side lobes and circle represent the rectum.

where $s \in [0, 1]$ and $\mu = x$ or y . Using the above basis function (6.1), the organ boundary $B_m(s)$ can be expressed as shape coefficients (Γ) as

$$\Gamma = (\lambda_1^{\mu_1}, \dots, \lambda_{F_\phi}^{\mu_1}, \lambda_1^{\mu_s}, \dots, \lambda_{F_\phi}^{\mu_s})^T \quad (6.5)$$

6.2 Results

The conductivity values of the background, bladder, and neighboring organs are known, then the shape coefficients (Γ) of the bladder boundary can be estimated using the measured voltages obtained at the surface electrodes of the domain Ω . The cost function is used for measuring the discrepancy between measured and calculated voltages.

The particles are randomly initialized within the search space of the pelvic domain Ω . These particles represent the Fourier series coefficients which represent the bladder boundary. Each particle is used to calculate the voltage based on it. The fitness used to evaluate the particle for the GSA can be replaced with the cost function. Thus, the fitness is now defined by equation (3.1) and the GSA related equations are updated. The equations related to mass, best(t), and worst(t) (eq. (4.13), (4.15), and (4.16)) are

updated as

$$m_i(t) = \frac{\Phi_i - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \quad (6.6)$$

$$\text{best}(t) = \min_{i \in (1, \dots, N)} \Phi_i \quad (6.7)$$

$$\text{worst}(t) = \max_{i \in (1, \dots, N)} \Phi_i \quad (6.8)$$

$$\Gamma(t) = \Gamma(t - 1) + v(t) \quad (6.9)$$

where Φ_i is the cost function of the particle i . The shape coefficient is updated with the equation (6.9) where $v(t)$ is the velocity of particle defined by equation (4.9). The flowchart of GSA for estimating the bladder boundary is presented in figure 6.2. From the figure we can see the steps required by the GSA to optimize the value of the shape coefficient. After the parameters of GSA are initialized, particles are randomly initialized. Since a particle defines the shape coefficient it is used to calculate the conductivity distribution. A voltage is calculated for every particle and the fitness is evaluated. Based on the fitness evaluation, the velocity of all the particles are updated and the position of the particles are optimized. In every iteration the fitness is evaluated again and the particle with lower fitness is selected as the optimum solution. This optimum solution provides the best optimized Fourier series coefficients which presents the bladder boundary.

Bladder boundary estimation is done firstly conducted in the numerical simulation for different test conditions then later on phantom experiments. GSA is used as an inverse problem solver algorithm to estimate the shape coefficients of the bladder boundary. For the geometry of the pelvis-shaped domain, a CT scan image is taken as a reference and the image is presented in figure 6.1. For numerical calculation, finite element mesh is used which is shaped as a pelvic. A constant current is applied to the surface electrodes with a magnitude of 1 mA and an opposite current injection method is used. In this study, 16 electrodes are attached to the surface of the domain. Using these 16 electrodes with opposite current injection method generates 128 independent voltage readings.

A total of 16 surface electrodes are used and 128 independent voltage readings are

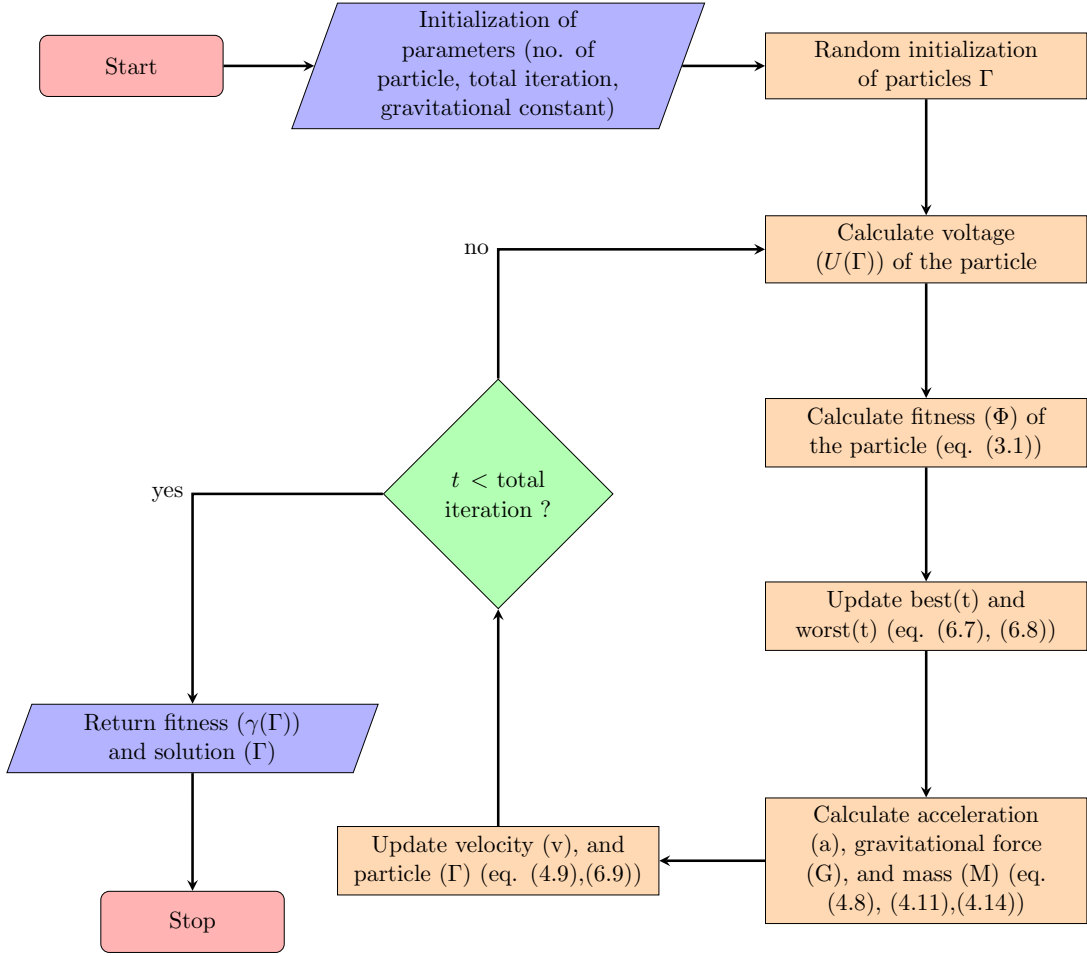


Figure 6.2: Flowchart for the gravitational search algorithm for estimating the bladder boundary.

measured from this configuration. As a performance metric root mean square error (RMSE) of estimated Fourier coefficients is used. The RMSE (Meng et al., 2010) is defined as

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left[\frac{(\Gamma_i^t - \Gamma_i)^T (\Gamma_i^t - \Gamma_i)}{(\Gamma_i^t)^T (\Gamma_i^t)} \right]} \quad (6.10)$$

where N is the number of test data, Γ_i^t is the shape coefficient matrix defining the actual bladder boundary and Γ_i is the estimated shape coefficient matrix.

6.2.1 Numerical simulation

For numerical simulation, the EIDORS framework (Adler and Lionheart, 2006) has been used in Matlab. In the forward problem, FEM is used for simulating the voltage measurement data based on the shape of the bladder and the neighboring organs and

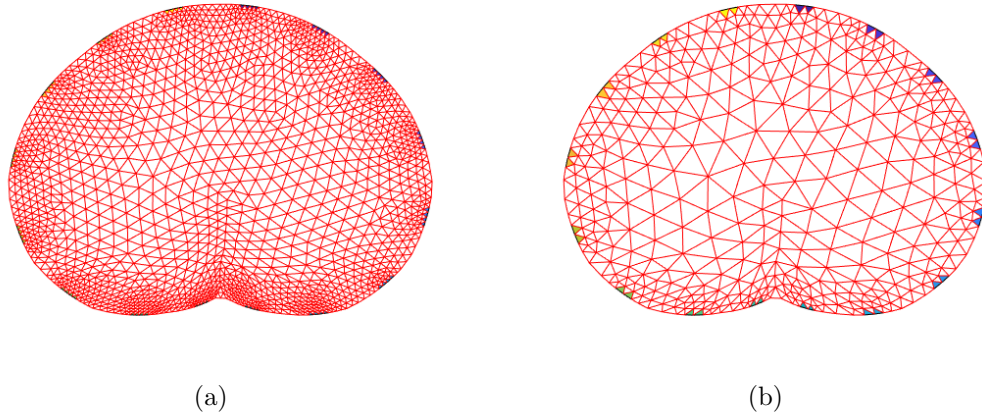


Figure 6.3: Mesh of the pelvic domain for estimating the bladder boundary. (a) Fine mesh used for the forward problem. (b) Coarse mesh used for inverse problem.

tissues. Two different mesh have been used to avoid inverse crime (Wirgin, 2004). One for the forward problem with 1714 nodes and 3220 elements and another for the inverse problem with 455 nodes and 805 elements which are presented in figure 6.3. Four different targets, i.e., background, bladder, side lobes, and rectum are defined inside a pelvis domain with each having different conductivity values of 0.0034 S/cm, 0.021 S/cm, 0.056 S/cm, and 0.0012 S/cm, respectively (Gabriel et al., 1996a,b). The conductivity of the bladder is considered higher than the background region due to the presence of salt in urine inside the bladder. The rectum is represented by a circle, the bladder is represented as an oval region and the two side lobes are represented by a complex shape region. For representing the shape of the bladder, the order of the Fourier series needed is four ($F_\phi = 4$). Therefore, the total numbers of shape coefficients for estimating are 8. The side lobes are considered to include all the extra side muscles, other inner organs, and bones as shown in figure 6.1. In the numerical simulation, the generated voltage data are added with a 2% relative noise to account for instrumental and environmental error (Eyuboglu et al., 1994).

Different cases are performed for the numerical simulation before using the phantom experimental data. A big bladder size is considered the first case which represents a bladder filled with the urine to the maximum extent. The other case represents a smaller bladder size in which the urine is in very low quantity. Two different scenarios are considered with and without neighboring tissues. At first, a single bladder

target is considered inside the pelvis domain without neighboring tissues and its shape coefficients are estimated with GSA and PSO. A total of 20 random initial solutions are defined in the search space for GSA and PSO estimation. The initial guesses for numerical simulation as well as for the phantom experiments are chosen randomly within the search space of the pelvis domain boundary based on the prior information from the CT scan. The search range for the coefficients is defined individually. The parameter R_{ij} in equation (4.6) is defined as Euclidean distance between two particles and parameter $g_c(t_0)$ in equation (4.11) is a constant value of 100. The convergence graph of GSA for full bladder case 1 without neighboring tissues is given in figure 6.4. From the figure, it can be noticed that GSA has good convergence by the 25th iteration. GSA and PSO are applied for boundary estimation of case 1 where both the methods are conducted with 75 iteration steps and 20 random initialized solutions.

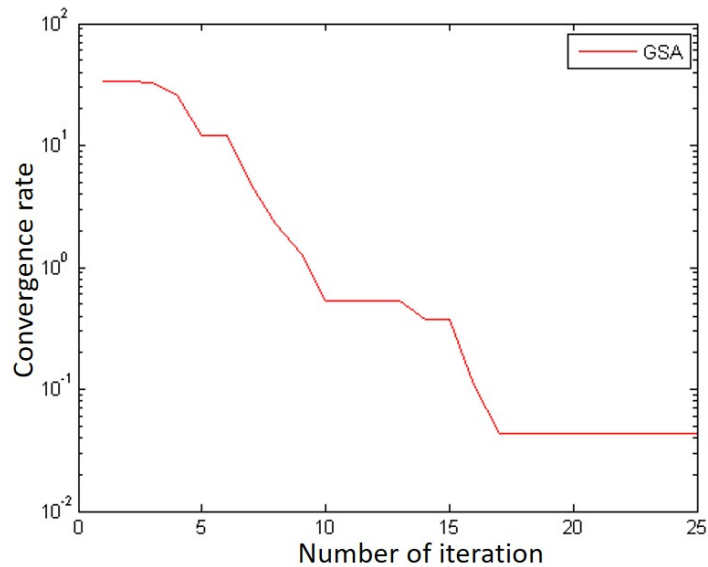


Figure 6.4: Convergence graph of GSA for estimating the bladder boundary in the pelvic domain

Figure 6.5a shows the reconstruction boundary result for the big bladder case. As it is noticed, GSA has a better estimation of bladder shape than PSO. Figure 6.5b shows the reconstructed boundary result for the small bladder case. In the case of a small bladder, the bladder tissue is located close to the center and the sensitivity is low as compared to a full bladder case. In a small bladder case (figure 6.5b) the estimated bladder boundary with GSA is close to the true boundary when compared to PSO.

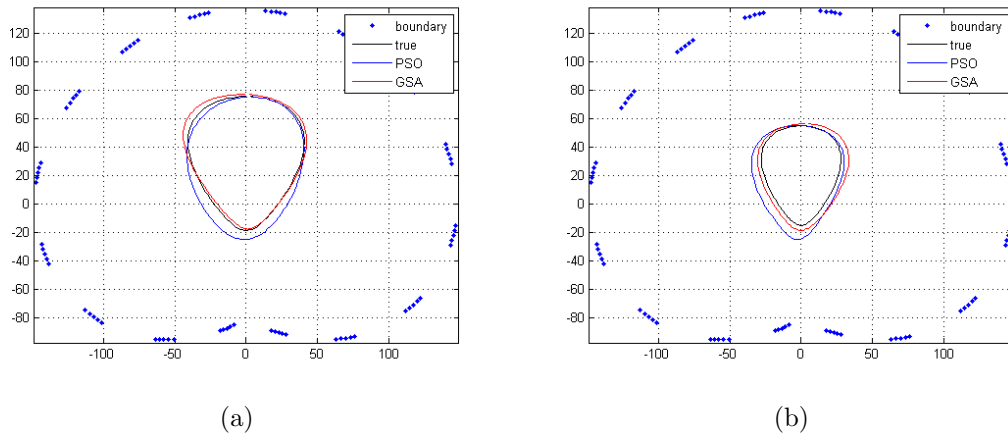


Figure 6.5: A simple numerical case for bladder estimated by GSA and PSO. The axis units are on a millimeter scale. (a) Big bladder case, and (b) small bladder case in the pelvic domain.

The normal pelvis domain is a complex structure, which consists of muscles, bones, and tissues along with the bladder. Thus, numerical simulations are performed with a more realistic scenario. In this, all the different neighboring organs and tissues are considered along with the bladder. The simulated voltage data is generated, and shape coefficients of bladder boundary are estimated with GSA. Different bladder sizes and shapes were considered in a multi-target scenario for numerical simulation. There are three cases in the multi-target scenario involving neighboring tissues. The first one corresponds to a full bladder where the bladder is assumed to be filled with the urine to the maximum extent. The other two cases represent a smaller bladder size in which the urine is in very low quantity. Bladder location is slightly different from person to person and the small bladder is mostly covered up by other neighboring tissues making it hard to estimate.

The reconstruction result for the full big bladder case considering neighboring tissues is given in figure 6.6. GSA could estimate the location, size, and shape of the bladder with good accuracy when compared to PSO. The shape estimated using PSO is different when compared to the true shape. The RMSE is computed for the estimated bladder coefficients and is shown in Table 6.1. From the RMSE it is seen that in the case of a full bladder with neighboring tissues, GSA has lower RMSE than PSO.

Figure 6.7 shows the small bladder case with neighboring tissues. The estimated bladder boundary with GSA has good accuracy when compared with PSO results. Figure 6.8

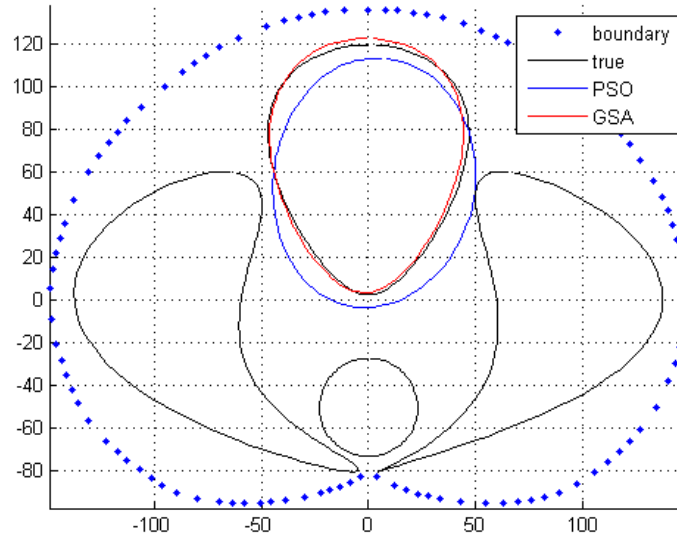


Figure 6.6: Numerical results for shape estimating of a big bladder case (case 1) using GSA for a multi-target scenario. The axis units are on a millimeter scale.

Table 6.1: Comparison of RMSE for the estimated Fourier coefficients by PSO and GSA in multi-target numerical scenarios.

Method	Case 1	Case 2	Case 3
PSO	0.1159	0.1878	0.2944
GSA	0.0349	0.0464	0.059

shows the estimated result of another case of a small bladder positioned in a different location. This case has a bladder located more towards the center. Even in this situation, it can be observed that GSA could estimate the bladder boundary with good accuracy whereas with PSO the estimated bladder size is very small, and the shape is quite different from the true shape. As it is noticed from the computed RMSE values GSA has a very low RMSE of estimated Fourier coefficients when compared to PSO. Further, the factors affecting the performance such as accuracy and convergence of GSA to bladder boundary estimation are analyzed. The effect of a different number of particles or solution and iteration steps are considered by performing numerical simulation using a single bladder target. Five different particle sizes (5, 10, 20, 75, 100) are considered in estimating the single bladder target. Figure 6.9a shows the reconstruction result for the bladder boundary using GSA with a different particles

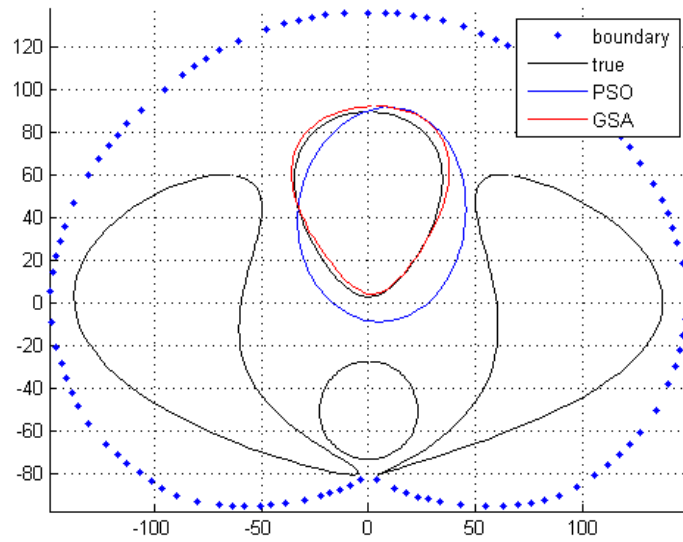


Figure 6.7: Numerical results for shape estimating of a small bladder case (case 2) using GSA for a multi-target scenario. The axis units are on a millimeter scale.

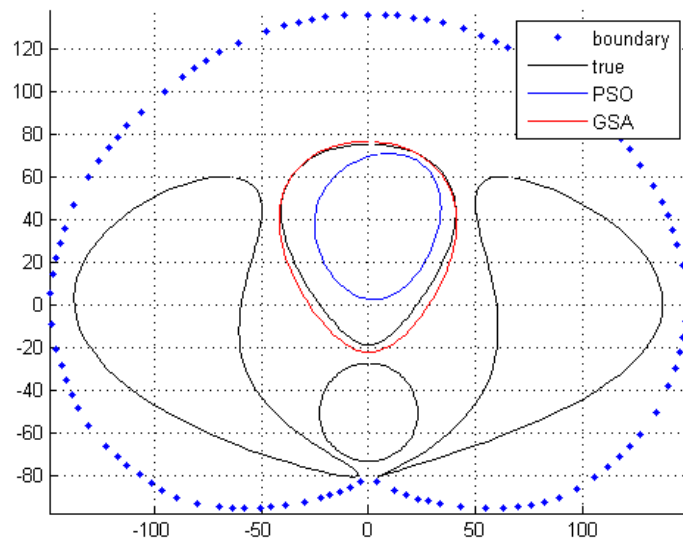
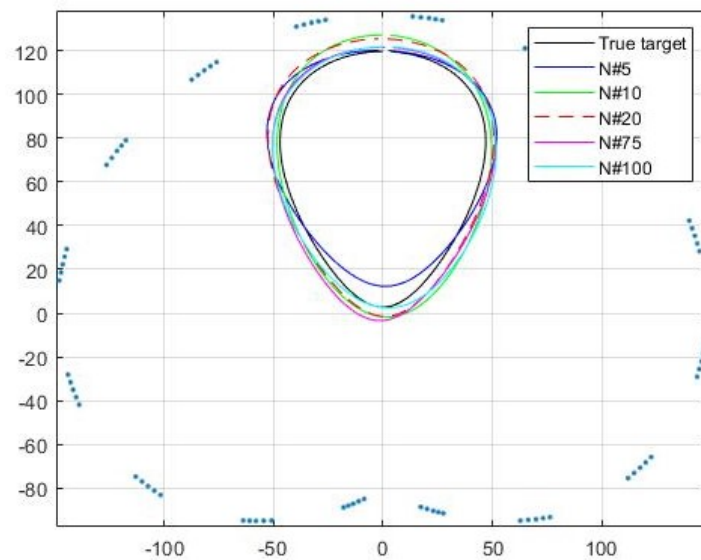
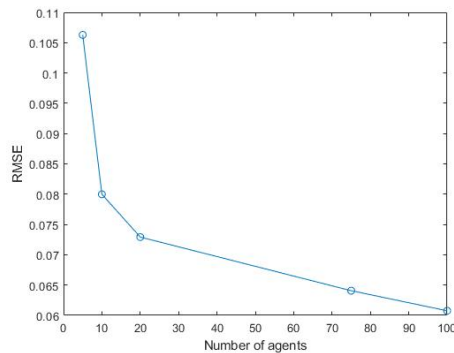


Figure 6.8: Numerical results for shape estimating of a small bladder case (case 3) at a different location using GSA for a multi-target scenario. The axis units are on a millimeter scale.

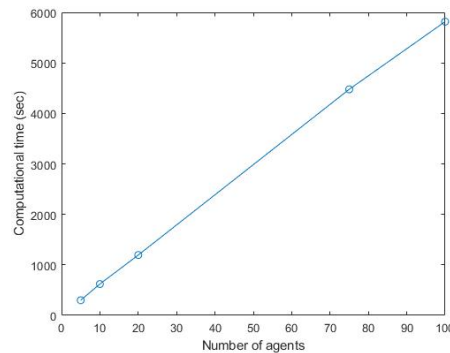
size. With 100 particles, the reconstructed boundary is close to the true target. RMSE of Fourier coefficients is computed for the single bladder case with a different number of particles and is plotted in figure 6.9b. By increasing the number of particles, the accuracy is improved. Also, figure 6.9c is the graph plot between computational time and the number of particles for the GSA. The graph is linearly increasing, as we increase the number of particles, the computational time also increases. From figures 6.9a, 6.9b, and 6.9c we can say that increasing the particle size increases the accuracy of the estimated results, but it also increases the computational time, therefore, there should be a trade-off when considering.



(a)



(b)



(c)

Figure 6.9: Comparison of estimated numerical results for GSA with a different number of particles assigned (5, 10, 20, 75, 100). (a) True bladder boundary and the estimated bladder boundary by GSA, (b) Corresponding RMSE of the estimate result by GSA, and (c) The computational time of algorithm with a different number of particles.

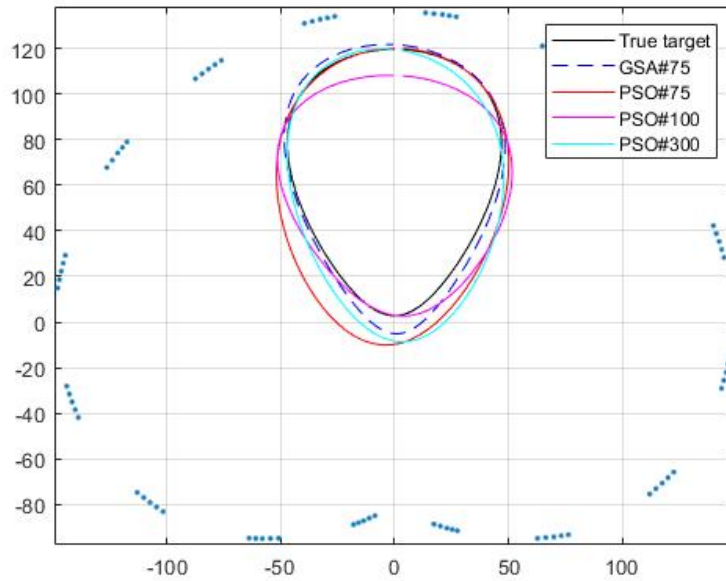


Figure 6.10: Comparison between GSA estimation result and PSO estimation result with the different iteration number. GSA#75 is the GSA estimate result with 75 iteration steps, PSO#75 is PSO estimate result with 75 iteration steps, PSO#100 is PSO estimate result with 100 iteration steps, and PSO#300 is PSO estimate result with 300 iteration steps.

PSO is also analyzed for the different iteration steps and compared with the GSA result. The comparison of PSO with different iteration steps is presented in figure 6.10. The iteration steps were 75, 100, and 300 and the estimation result of PSO was compared with the GSA estimation result which is estimated in 75 iteration steps. The RMSE for estimated Fourier coefficients with GSA (75 iteration steps) and PSO with a different number of iterations is shown in Table 6.2. It can be noticed that GSA has a lower RMSE than PSO with a higher number of iterations. The estimated result of the PSO with large iteration steps has good accuracy. However, the GSA can achieve similar results with a lower number of iterations.

6.2.2 Phantom experiment

For phantom experiments, four different targets are defined inside a pelvis domain with each having different conductivity values. The conductivity of the bladder and background are 0.0133S/cm and 0.0043S/cm, (Gabriel et al., 1996a,b) respectively, and are used as the a priori information. The phantom with 16 electrodes that had the

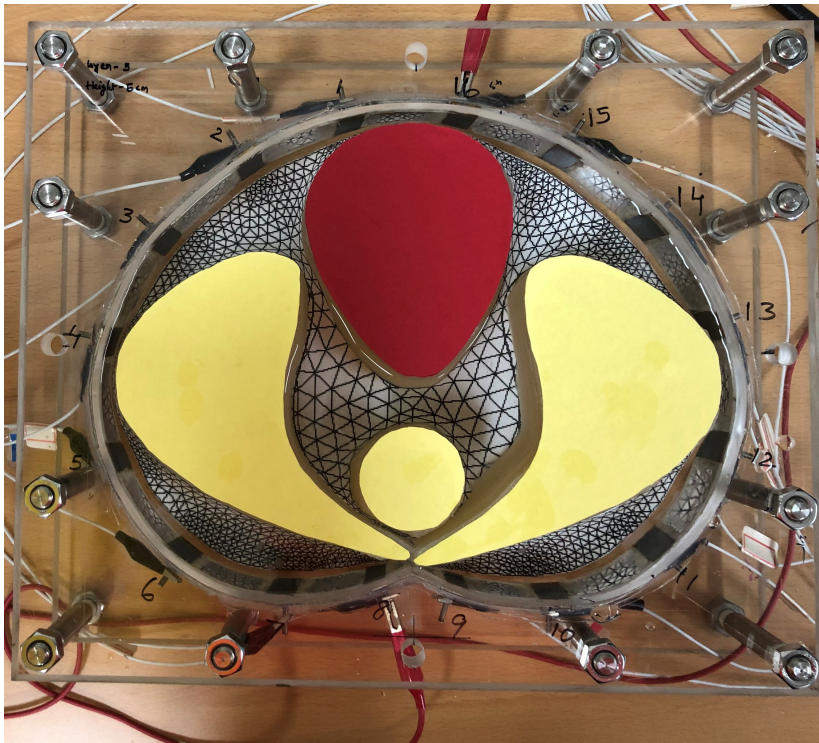
Table 6.2: Comparison of RMSE of the estimated result of PSO with different iteration steps with GSA estimated result with 75 iteration steps.

Method	Iteration	RMSE
GSA	75	0.0641
PSO	75	0.1170
	100	0.1046
	300	0.1004

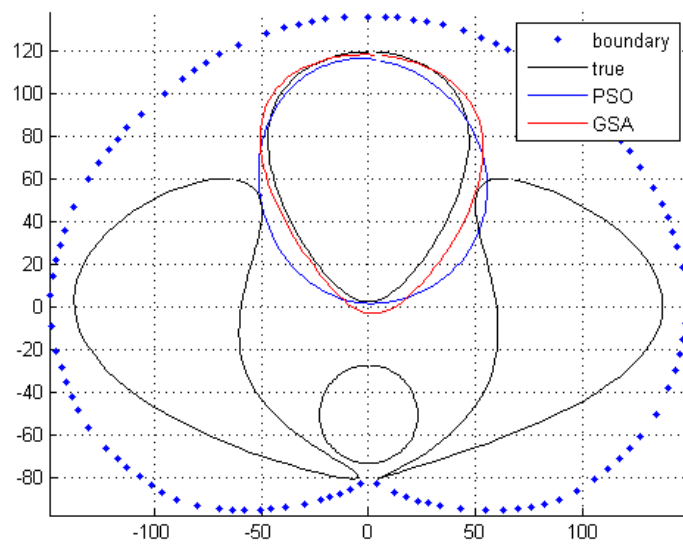
shape of the pelvis cross-sectional of a human body is used for experimental studies. The bladder, side lobes, and rectum are made from gelatin with different conductivities, and the background is made from saline water. Gelatin is cut based on the layout from the CT scan. The size, shape, and location of tissues are the same as those considered in the numerical simulation. EIT experiment is done with Agilent Precision LCR meter used for injecting constant current to the electrodes attached to the pelvis shaped domain and National Instruments PXI-1042Q used for measuring the voltage from the surface electrodes. A current of 1mA magnitude is applied to the phantom in an opposite injection pattern and the resulting voltages are measured on the surface electrodes.

Three different cases are considered in the phantom experiments. The first case is a big bladder case in which the bladder is assumed to be filled with urine and expanded to the full stretch while the second and third cases are medium and small size bladders, respectively. The true experimental setup and the corresponding estimated result are given in figure 6.11. From the figure 6.11b it can be noticed that GSA is successful in estimating the location and shape of the bladder with an experimental full bladder case whereas with PSO, the bladder location is estimated but the shape is different. Figure 6.12 shows the estimated result for medium size bladder along with the real experimental setup. The estimated result for the bladder boundary by GSA and PSO is present in figure 6.12b. In the case of medium size bladder, GSA is found to reconstruct the boundary size and shape close to the true object when compared to PSO.

Figure 6.13 shows the small bladder case, where the bladder is assumed to be filled with a very less amount of urine, and the bladder size is smaller than in the other two cases. Figure 6.13a shows the real experimental phantom layout for the small

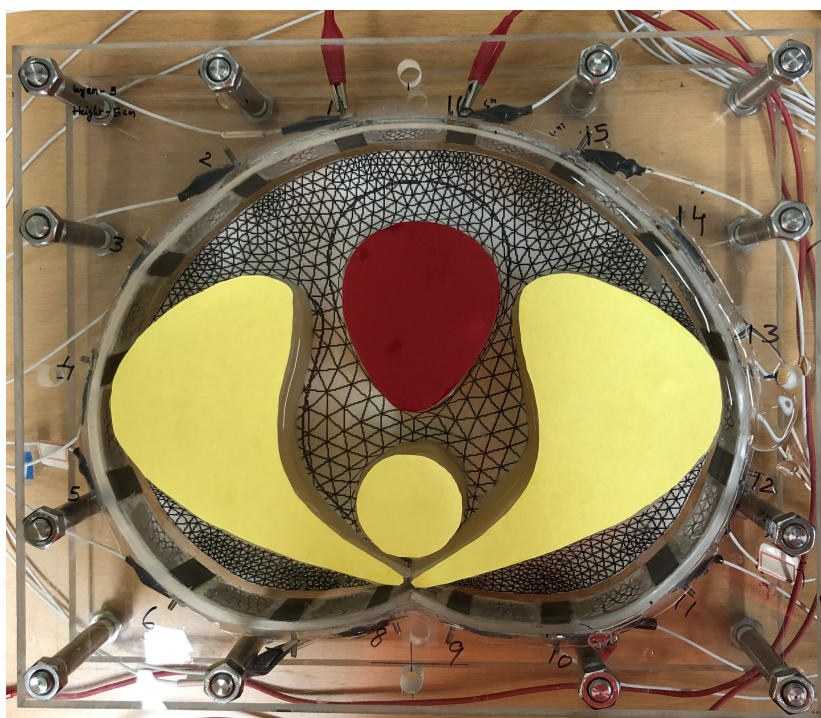


(a)

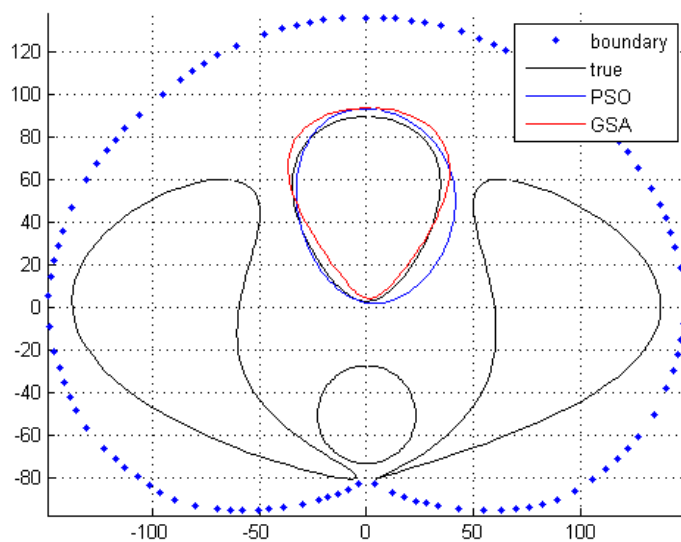


(b)

Figure 6.11: (a) Phantom experiment setup for big bladder (case 1). The red target is the bladder and yellow target regions are side lobes and rectum. (b) Phantom experiment estimation result for big bladder cases with GSA and PSO. The axis units are on a millimeter scale.

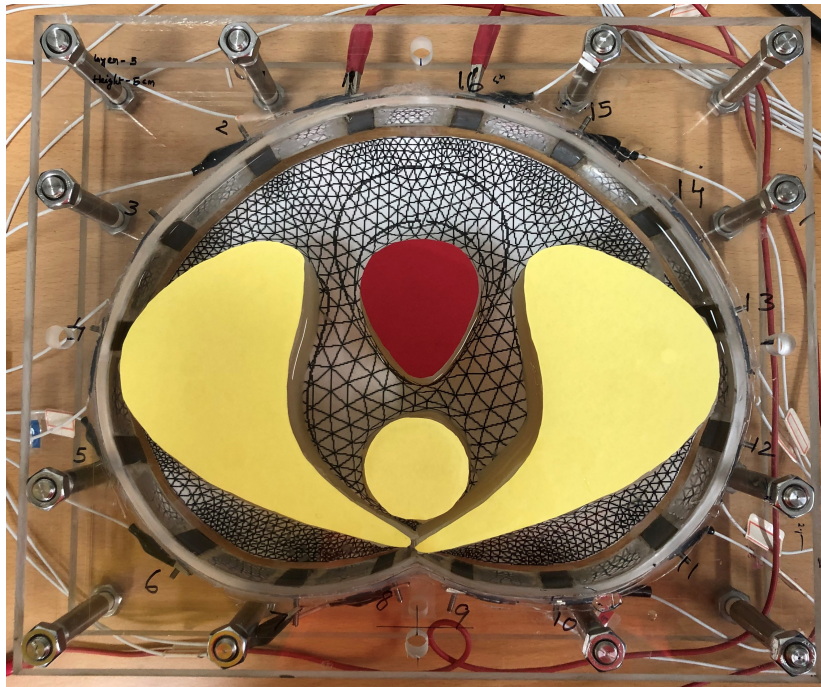


(a)

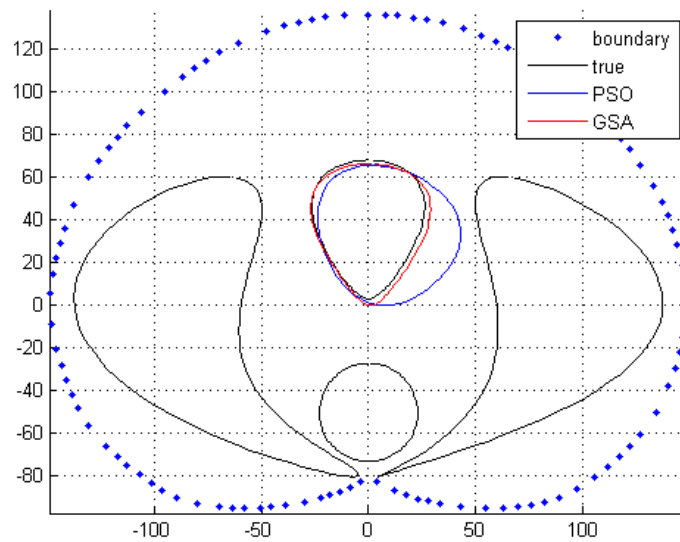


(b)

Figure 6.12: (a) Phantom experiment setup for medium bladder size (case 2). The red target is the bladder and yellow target regions are side lobes and rectum. (b) Phantom experiment estimation results for medium bladder size case with GSA and PSO. The axis units are on a millimeter scale.



(a)



(b)

Figure 6.13: (a) Phantom experiment setup for small bladder size (case 3). The red target is the bladder and yellow target regions are side lobes and rectum. (b) Phantom experiment estimation results for medium bladder size case with GSA and PSO. The axis units are on a millimeter scale.

Table 6.3: Comparison of RMSE for the estimated Fourier coefficients by PSO and GSA in phantom experiments.

Method	Case 1	Case 2	Case 3
PSO	0.1371	0.1074	0.2684
GSA	0.0805	0.0741	0.0606

size bladder case and figure 6.13b shows the estimated result of GSA and PSO. GSA estimated the small size of the bladder located close to the center of the pelvis with good accuracy although sensitivity is low for this situation. In the case of PSO, it could locate the position of the bladder, but the shape cannot be estimated. RMSE of Fourier coefficients for medium and small size bladder can be seen from Table 6.3 and shows the superior performance of GSA over PSO. In all three cases, GSA has a very good estimation result with good accuracy in the reconstructed image than the PSO. The computational time for phantom experiments using GSA is 1024 seconds and PSO is 1608 seconds (Intel(R) Core (TM) i7-6700 CPU at 3.40GHz, 3.50GB RAM, Windows 7, Matlab).

6.3 Conclusion

A study on shape estimation for the bladder boundary is done by the GSA using the EIT. A pelvis-shaped domain is considered, where the bladder, side lobes, and rectum boundaries are assumed to be smooth and described by using a truncated Fourier series. The coefficients of the Fourier series, which are used to determine the shape and size of the bladder, are estimated using the GSA algorithm.

Three cases (big, medium, and small) are considered to correspond to different bladder sizes. From the numerical simulation and phantom studies, it is found that GSA is successful in estimating the bladder shape and location with good accuracy. The better performance of GSA can be noticed from the RMSE of the estimated shape coefficients. Compared with the PSO, GSA has better estimation performance using the same initial conditions. The particles or solutions are randomly assigned in the search space in both PSO and GSA. The particles are updated differently in both algorithms. In PSO, particles are updated using the two best positions, i.e., pbest

and gbest. However, in GSA, particles are updated based on the overall force of all particles. When updating the velocity (v) of the solution, PSO uses pbest and gbest solutions, but, in GSA, the particle's last position is used. In PSO, pbest contains the information of all the previous iteration particle best positions. However, in GSA, only the particle's previous step position is considered to update, making GSA memoryless compared with PSO. The distance between particles is considered in GSA, but, in PSO, it is not considered. Moreover, GSA does not require the computation of Jacobian as it is a derivative-free algorithm. Different numbers of particles are used to analyze the estimation results. Using a higher number of particles shows better performance, but more computational time is required. The RMSE value for GSA has a lower value compared with PSO in all the cases of numerical study and phantom experiment. The estimated result of PSO with higher iteration steps is comparable with the estimated result of GSA with the lower iteration steps. Also, the computing time taken by the GSA is less than the PSO.

7 Defect detection in graphene with PSO-GSA

This chapter deals with the study conducted to locate the defect on graphene sheets with PSO-GSA. The defect geometry is defined by the Fourier series coefficients as defined previously for the bladder boundary (see section 6.1). However, in this case, the background conductivity of the graphene is also estimated by the PSO-GSA. For defining the background conductivity one Fourier series coefficient is used. The shape coefficient and the background conductivity are defined as

$$\begin{bmatrix} \Gamma_i \\ \sigma_i^b \end{bmatrix} = (\lambda_i^1, \lambda_i^2, \dots, \lambda_i^D, \lambda_i^b) \quad (7.1)$$

where σ_i^b is the background conductivity of the graphene sheet and one dimension of the particle defines the conductivity (λ_i^b). The update equation of velocity is also modified to adjust for the background conductivity as

$$\begin{aligned} v_i(t) = & w \times v_i(t-1) + c_1 \times rand \times (pbest_i - \begin{bmatrix} \Gamma_i(t-1) \\ \sigma_i^b(t-1) \end{bmatrix}) \\ & + c_2 \times rand \times (gbest - \begin{bmatrix} \Gamma_i(t-1) \\ \sigma_i^b(t-1) \end{bmatrix}) \end{aligned} \quad (7.2)$$

where $v_i(t)$ is the velocity of i^{th} particle at t iteration, w is the weighting function, c_k is a weighting factor, $rand$ is random number $[0,1]$, $\Gamma_i(t)$ is the current position of particle i at iteration t , $pbest_i$ is the best particle in that iteration, and $gbest$ is the best solution till that iteration. After every iteration the shape coefficient and

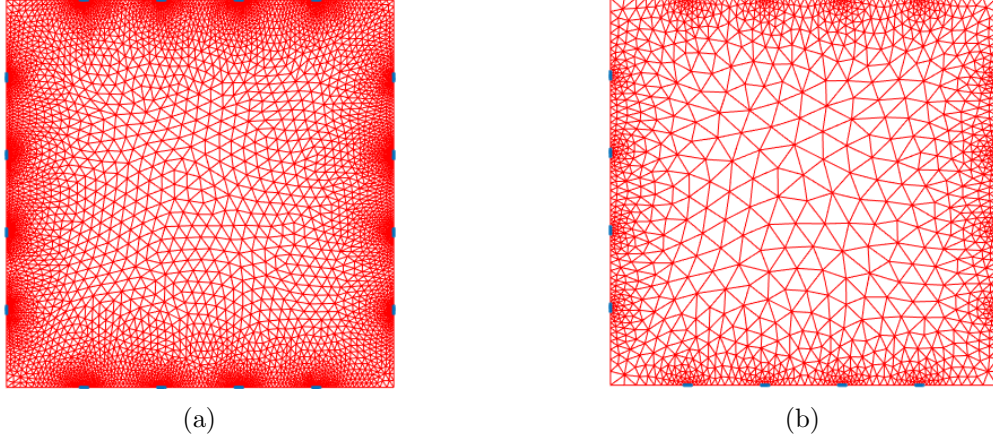


Figure 7.1: Fine and coarse mesh of graphene used in the study. The boundary electrodes are represented in blue colour attached to the mesh boundary.

the background conductivity is optimized based on the velocity of the particles. The particles are updated as

$$\begin{bmatrix} \Gamma_i(t) \\ \sigma_i^b(t) \end{bmatrix} = \begin{bmatrix} \Gamma_i(t-1) \\ \sigma_i^b(t-1) \end{bmatrix} + v_i(t) \quad (7.3)$$

The solutions are updated based on the fitness of each particle and the cost function is defined as

$$\gamma(\Gamma_i, \sigma_i^b) = \frac{\|U(\Gamma_i, \sigma_i^b) - V\|^2}{2} \quad (7.4)$$

where $U(\Gamma_i, \sigma_i^b)$ is the calculated voltage with FEM which depends on the estimated Fourier series coefficients and the background conductivity of the graphene sheet, and V is the measured voltage recorded from the electrical impedance system. The flowchart of PSO-GSA for estimating the shape of the defect on the graphene and the background conductivity is presented in figure 7.2.

7.1 Results

A numerical simulation and experimental studies are presented in this section in which external defect location and shape on graphene sheet are estimated with PSO-GSA. In this study, the conductivity of the graphene sheet is also estimated along with the defect details. The same geometry of the graphene sheet has been used for the numerical and

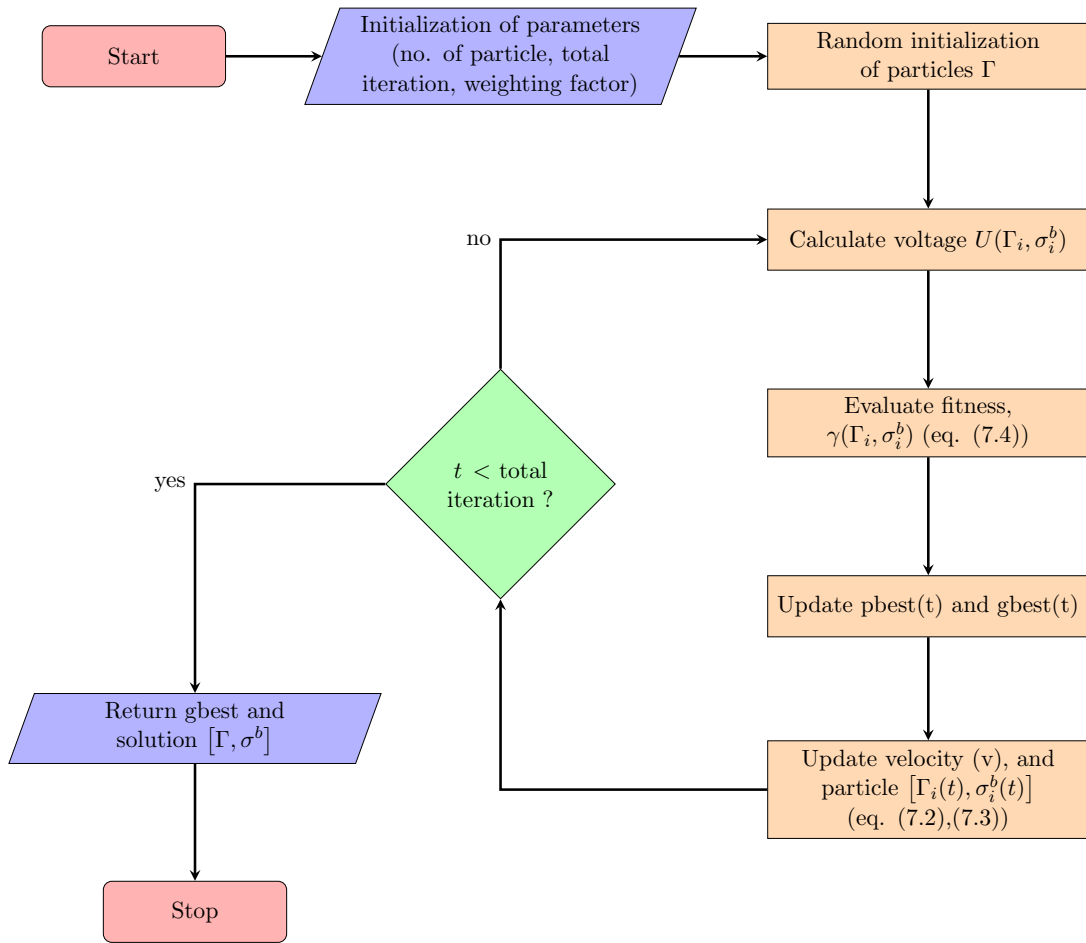


Figure 7.2: Flowchart of the PSO-GSA algorithm for estimating defect geometry and the background conductivity of the graphene sheet.

experimental cases and the preparation of the graphene sheet is also described later in the experimental study section. As GSA has good performance in estimating the higher-order Fourier series coefficients in the complex problem of EIT, hence it is used to compare the estimated result from PSO-GSA for all the cases. EIDORS (Adler and Lionheart, 2006) framework is used to compute the forward solution of the EIT in the Matlab software. To avoid the inverse crime, two separate mesh is used for the forward and inverse problems which are presented in figure 7.1. A mesh configured with 9648 elements and 5121 nodes is used in the forward problem, while the mesh in the inverse problem used 2412 elements and 1355 nodes. Both PSO-GSA and GSA were configured with the same search space, and particle size and run for equal iterations. The particles are updated per iteration and the best solution is updated which has a lower value cost function.

The robustness of the proposed algorithm PSOGSA is studied with the help of Monte Carlo simulation. In this, each run of PSOGSA is assigned with a different set of particles and a different noise seed. The Monte Carlo simulation is executed only on numerical study for $\eta = 20$ runs for estimating the Fourier series coefficient. The mean value of the Fourier series coefficient is used as the estimated value from the PSOGSA algorithm. Mean ($\bar{\lambda}$) is one of the statistical parameter analyzed by Monte Carlo simulation along with mean absolute error (\overline{MAE}), mean error square (\overline{MES}), error standard deviation (E_{sd}), and the mean root mean square error (\overline{RMSE}). \overline{MAE} and \overline{RMSE} provide the magnitude of estimation error for the corresponding true value. The dispersion of the error (e_i) for estimated parameters is calculated by error standard deviation (E_{sd}) and the square of the bias of the error is given by \overline{MES} . The mathematical definitions of each parameter are defined below as

$$\bar{\lambda} = \frac{1}{\eta} \sum_{i=1}^{\eta} \lambda_i \quad (7.5)$$

$$\overline{MES} = \left(\frac{1}{\eta} \sum_{i=1}^{\eta} (\lambda_{true} - \lambda_i) \right)^2 = \left(\frac{1}{\eta} \sum_{i=1}^{\eta} e_i \right)^2 = (E[e_i])^2 = (\bar{e})^2 \quad (7.6)$$

$$\overline{MAE} = \frac{1}{\eta} \sum_{i=1}^{\eta} |\lambda_{true} - \lambda_i| = \frac{1}{\eta} \sum_{i=1}^{\eta} |e_i| \quad (7.7)$$

$$\overline{MSE} = \frac{1}{\eta} \sum_{i=1}^{\eta} (\lambda_{true} - \lambda_i)^2 = \frac{1}{\eta} \sum_{i=1}^{\eta} e_i^2 = E[e_i^2] \quad (7.8)$$

$$E_{sd} = \sqrt{\frac{1}{\eta} \sum_{i=1}^{\eta} (e_i - \bar{e})^2} = \sqrt{E[e_i^2] - (E[e_i])^2} = \sqrt{\overline{MSE} - \overline{MES}} \quad (7.9)$$

$$RMSE = \sqrt{\frac{(\lambda - \lambda_t)^T (\lambda - \lambda_t)}{\lambda_t^T \lambda_t}} \quad (7.10)$$

$$\overline{RMSE} = \frac{1}{\eta} \sum_{i=1}^{\eta} RMSE_i \quad (7.11)$$

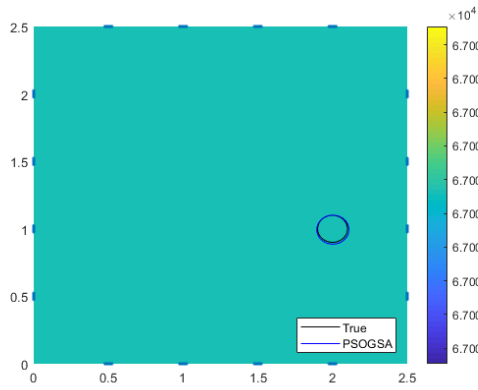
where λ and λ_t are the estimated and true values of the respective Fourier series coefficients, respectively. The statistical parameters for the numerical cases which are reported which are tabled in a later section that provides the information about the PSOGSA algorithm regarding stability and accuracy. For the boundary estimation of

the defect the calculated statistical parameters are $\bar{\lambda}$, \overline{MAE} , \overline{MES} , and E_{sd} .

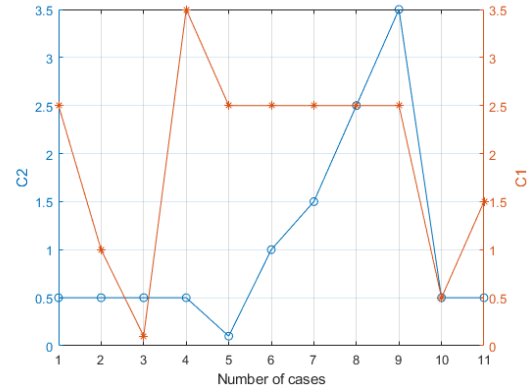
7.1.1 Numerical study

There are 16 electrodes attached to the boundary of the graphene sample where each side has four electrodes. The conductivity of the background and the defect on the graphene sheet is assumed to be homogeneous and is used as a known priori information for estimating the Fourier series coefficient. The defects are resistive to the current flow therefore a very low conductivity value of $5 \times 10^{-9} mS/cm$ is assumed in numerical simulations and of the background to be $6.7 \times 10^4 mS/cm$, respectively. The background conductivity of the graphene sheet is also estimated along with the Fourier series coefficients. A 1% relative white Gaussian noise is added to generated voltage data to account for instrumental and environmental noise.

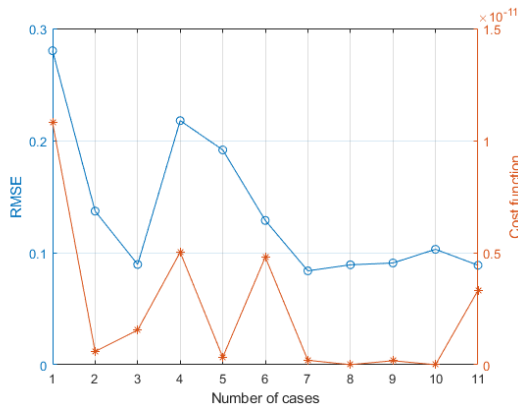
A single circular defect on a graphene sheet is considered a test case for estimating the optimum parameter values for estimating defect boundary and background conductivity of the square-shaped domain. This case is presented with a circular defect estimated by PSO-GSA, see figure 7.3a. The parameters such as the number of particles, and the number of iteration is analyzed for both single defect and two defects on the graphene sheet, however, weighting factors are analyzed only for single defect case. The weighting factors (C'_j) different configuration effect on the performance of the PSO-GSA algorithm is initially analyzed. A total of 11 cases of different weighting factors configuration are used presented in figure 7.3b and the corresponding RMSE, and cost function are shown in figure 7.3c. Out of 11 cases of weighting factors configuration only 3 cases are preferred which have the lower RMSE and cost function compared to other configurations. The 7th configuration of weighting factors gives the low RMSE of an estimated Fourier series coefficients along with a lower cost function. PSO-GSA performance is compared with different iteration levels. By this analysis, we want to check if there will be any improvement in the performance of the algorithm with more iterations. Five different cases for the iteration are presented in the figure 7.3d. We can see that in almost all cases the cost function converged before 20th iteration and there is no future improvement further. Thus, it is a waste of computational resources to run the PSO-GSA algorithm beyond 25th iteration.



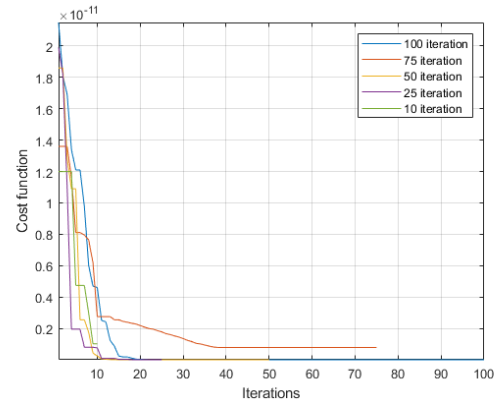
(a)



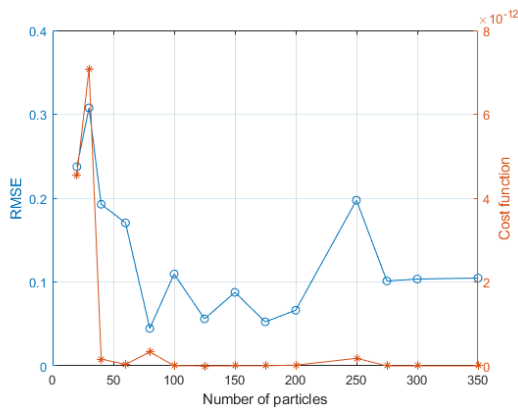
(b)



(c)



(d)



(e)

Figure 7.3: Numerical case of single defect on graphene sheet for parameter optimization. (a) True defect location along with the estimated result from PSO-GSA. (b) Weighting factors (C'_j) combination, (c) RMSE and cost function value for different cases of weighting factor combination, (d) cost function value for a different level of iteration, (e) RMSE and cost function with a different number of particles in PSO-GSA.

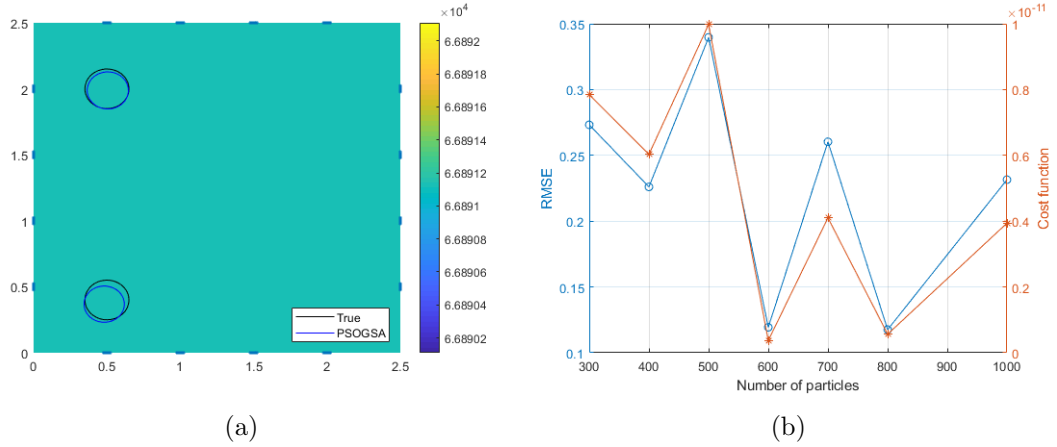


Figure 7.4: Numerical case of two defects on the graphene sheet for parameter optimization. (a) True defect location along with the estimated result from PSO-GSA. The black circle represents the true position of the defect. The estimated location of a defect by PSO-GSA is represented by a blue circle. (b) Corresponding RMSE and cost function with a different number of particles in PSO-GSA.

The impact of a different number of particles on the PSO-GSA algorithm estimation is also analyzed. The number of particles is initialized in the range of 20 and 350 with 14 different values. As the number of particles is increased the cost function decreases but the RMSE values are fluctuating after the number of particles is increased beyond 40. The cost function is lower for 5 cases where the number of particles is from 100 to 200, in which RMSE of the Fourier series coefficients is low for 175 particles. The comparison of the cost function and the RMSE with a different number of particles for the single defect case is presented in figure 7.3e.

A single defect in the graphene sheet is defined with six Fourier series coefficients which represent a simple shape, i.e., circular and ellipse. Two defects on the graphene sheet are considered to analyze the proposed algorithm for the estimation of the defect boundary. The two defects needed 12 coefficients to define their simple geometry. As the unknowns to be estimated by the proposed algorithm have also doubled in number, we need to increase the number of particles of PSO-GSA to achieve the lower cost function. For this two defects scenario, presented in figure 7.4a, we have analyzed the proposed algorithm with a different number of particles. Now the number of particles is in the range of 300 to 1000. The algorithm configured with 600 and 900 particles have shown good result with lower RMSE and cost function. The RMSE of Fourier series coefficient and cost

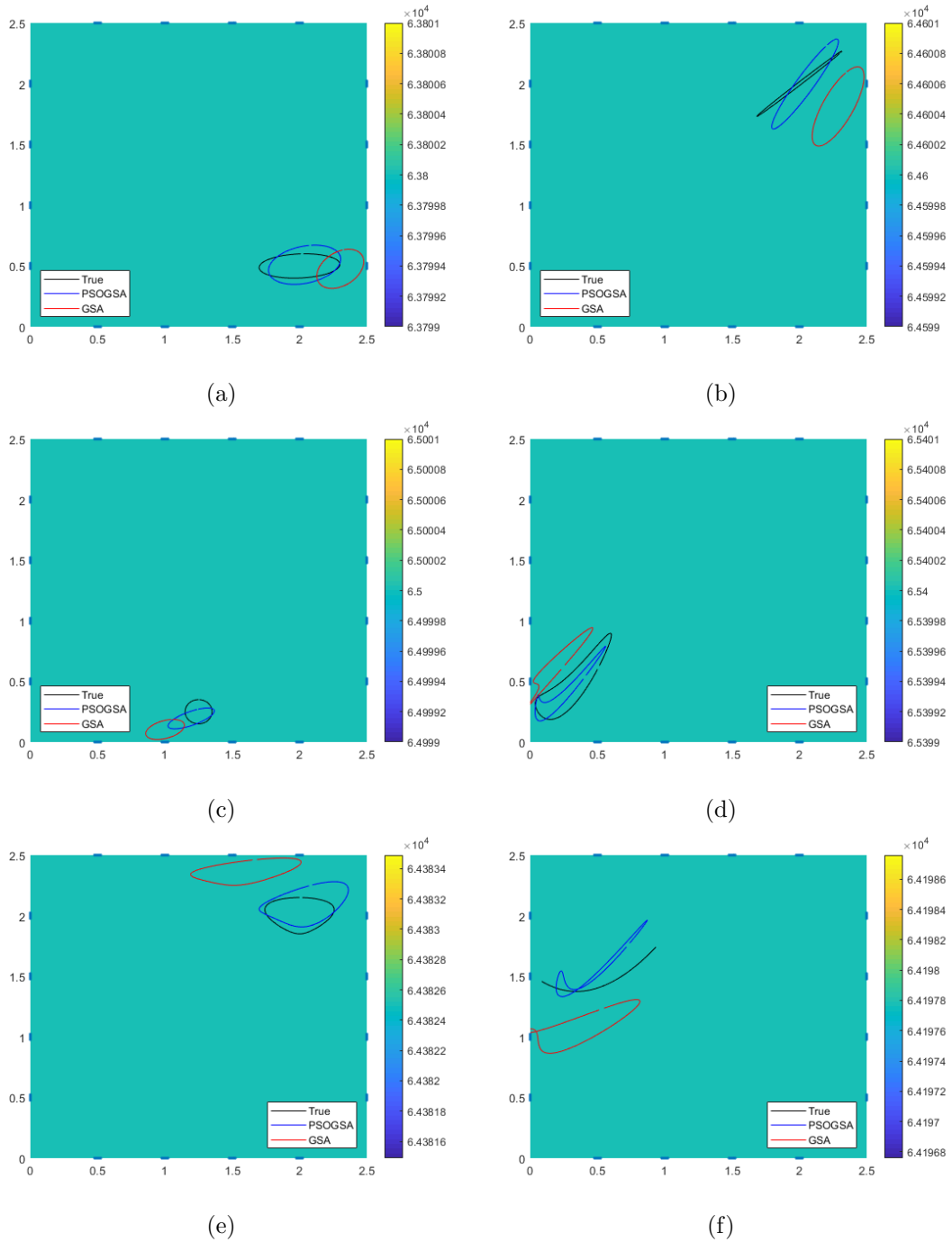


Figure 7.5: Numerical results for cases 1-6 with a single defect on graphene surface by PSO and GSA. The black circle represents the true position of the defect. The estimated location of the defect by PSO and GSA is represented by a blue circle and red circle, respectively.

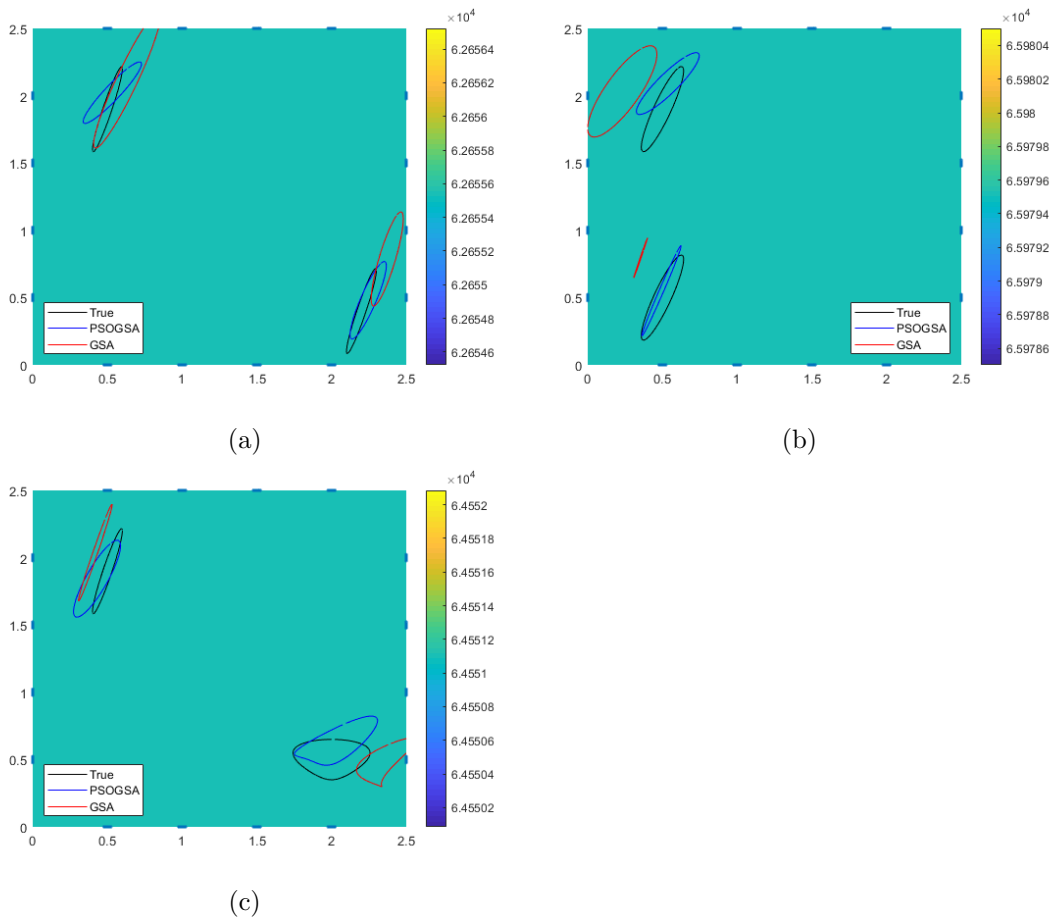


Figure 7.6: Numerical results for cases 7-9 with two defects on graphene surface by PSO-GSA and GSA. The black circle represents the true position of the defect. The estimated location of a defect by PSO-GSA and GSA is represented by a blue circle and red circle, respectively.

function for the corresponding number of particles are presented in figure 7.4b. From the figure, we can see that the PSO-GSA algorithm configured with 600 particles has a lower cost function and a similar RMSE of Fourier series coefficient when compared to PSO-GSA with 900 particles. Since the algorithm configured with the lower number of particles required less computational resources, PSO-GSA configured with 600 particles is used to estimate the two defects on the graphene sheet.

A total of 6 cases of a single defect on a graphene sheet are analyzed for estimation of the defect boundary with PSO-GSA. In the first case, the defect is located in the lower right corner which is presented in figure 7.5a. In the figure, we can see that the PSO-GSA algorithm has estimated the defect location and boundary with good accuracy when compared with the estimation by the GSA. Similarly in the second case, the defect is

Table 7.1: Statistical parameter for numerical case 1 (figure 7.5a): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO-GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	2.0000	2.0377	0.3417	0.0014	0.1428	0.3760	2.3034
λ_2	0.5000	0.5100	0.3399	0.0001	0.1377	0.3710	0.4763
λ_3	0.3000	0.2662	0.2235	0.0011	0.0519	0.2254	0.1734
λ_4	0.0100	0.0270	0.0238	0.0002	0.0008	0.0240	0.0443
λ_5	0.0100	0.0324	0.0292	0.0005	0.0012	0.0273	0.0161
λ_6	0.1000	0.1590	0.0788	0.0034	0.0074	0.0628	0.1550

Table 7.2: Statistical parameter for numerical case 2 (figure 7.5b): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO-GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	2.0000	2.0424	0.3969	0.0018	0.1784	0.4202	2.2882
λ_2	2.0000	1.9974	0.3271	0.0000	0.1355	0.3681	1.8128
λ_3	0.1000	0.1954	0.1496	0.0091	0.0400	0.1759	0.1850
λ_4	0.1000	0.2040	0.1951	0.0108	0.0567	0.2143	0.1748
λ_5	0.3000	0.1527	0.2297	0.0217	0.0592	0.1937	0.0540
λ_6	0.2500	0.3090	0.2089	0.0035	0.0496	0.2147	0.2753

Table 7.3: Statistical parameter for numerical case 3 (figure 7.5c): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	1.2500	1.1949	0.2325	0.0030	0.0588	0.2361	1.0010
λ_2	0.2500	0.1948	0.1597	0.0030	0.0273	0.1558	0.1035
λ_3	0.1000	0.1671	0.0978	0.0045	0.0090	0.0668	0.1365
λ_4	0.0000	0.0449	0.0471	0.0020	0.0026	0.0236	0.0085
λ_5	0.0000	0.0424	0.0445	0.0018	0.0025	0.0256	0.0452
λ_6	0.1000	0.0734	0.0794	0.0007	0.0066	0.0767	0.0822

located in the top right corner, however, the defect width is very thin when compared to the first case. The second case is presented in figure 7.5b which also contains the estimation done by PSO GSA and GSA algorithm. In this case, the estimation of GSA deviated from the true defect location towards the graphene sheet boundary. However, we can see that the defect location estimated by PSO GSA is in close agreement with the true location of the defect. But both algorithms failed to estimate the shape of the defect. In case 3 the single defect is a small size defect when compared with the previous two cases. In this scenario, the defect is located on the lower part of the graphene sheet between 14th and 15th electrodes. This small size defect is harder to estimate but PSO GSA has estimated with a low error which can be seen in the figure 7.5c along with the estimation result from the GSA. In these three cases, defects are defined with 6 Fourier series coefficients due to which the shapes are ellipses.

To evaluate the performance of PSO GSA for more complex shapes, defects were defined with more shape coefficients. Eight Fourier coefficients were used to define a single defect on a graphene sheet and were estimated. Cases 4, 5, and 6 of single defect with higher Fourier series coefficients are analyzed which are presented in figures 7.5d, 7.5e, 7.5f. We can see that PSO GSA could estimate the defect geometry with good accuracy than GSA. However, when the defect on the graphene sheet is very thin as in figure 7.5f PSO GSA estimated result has the intersecting boundaries along with GSA. Even if the boundaries are intersecting PSO GSA presents the estimated result with close

Table 7.4: Statistical parameter for numerical case 4 (figure 7.5d): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	0.3000	0.2335	0.2432	0.0044	0.0615	0.2388	0.1320
λ_2	0.5000	0.4344	0.2741	0.0043	0.0773	0.2702	0.5420
λ_3	0.2000	0.1970	0.1568	0.0000	0.0274	0.1655	0.1891
λ_4	0.3000	0.1998	0.1770	0.0100	0.0385	0.1688	0.2957
λ_5	0.2000	0.1606	0.1768	0.0016	0.0324	0.1756	0.0976
λ_6	0.1000	0.0855	0.0940	0.0002	0.0091	0.0942	0.0569
λ_7	0.0200	0.0741	0.0754	0.0029	0.0102	0.0854	0.1280
λ_8	0.1000	0.1499	0.0909	0.0025	0.0084	0.0766	0.1352

resemblance to the true shape.

All the single defect cases are estimated by the PSO GSA algorithm with a low error than the GSA algorithm. The statistical parameters for the numerical cases of the single defect are reported in the tables 7.1-7.6. Case 6 has the highest error standard deviation (0.6160) and case 1 achieved the lower error standard deviation (0.0240). Case 6 has a higher standard deviation error in almost all the coefficients due to the complex shape. In this case, the boundary overlaps with each other representing an arc that is very hard to estimate with PSO GSA and also with GSA. Even though the error is high in this case, still, PSO GSA estimated the location of the defect with good accuracy.

From the 6 cases of single defect, we can say that the PSO GSA has good performance over the GSA algorithm with the same parameter configurations. The RMSE of the estimated Fourier series coefficients from PSO GSA and GSA is presented in table 7.7. The table of RMSE for Fourier series coefficients shows that PSO GSA has a lower RMSE value which suggests better estimation performance.

Two defects on the graphene sheet cases were also studied and the performance of the PSO GSA was evaluated against GSA. Three different cases are presented for this study. Case 7 is the simplest two defects case study where one defect is on the top left and

Table 7.5: Statistical parameter for numerical case 5 (figure 7.5e): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	2.0000	2.0168	0.3204	0.0003	0.1370	0.3698	1.5766
λ_2	2.0000	2.0798	0.3052	0.0064	0.1146	0.3290	2.3546
λ_3	0.2400	0.3119	0.2126	0.0052	0.0483	0.2077	0.3752
λ_4	0.0000	0.0388	0.0407	0.0015	0.0039	0.0493	0.0301
λ_5	0.0000	0.0524	0.0550	0.0027	0.0055	0.0529	0.0735
λ_6	0.1500	0.1661	0.0717	0.0003	0.0059	0.0748	0.1045
λ_7	0.0500	0.0537	0.0504	0.0000	0.0028	0.0523	0.0837
λ_8	0.0000	0.0436	0.0458	0.0019	0.0044	0.0498	0.0173

Table 7.6: Statistical parameter for numerical case 6 (figure 7.5f): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	0.5000	0.4536	0.3665	0.0022	0.1645	0.4029	0.2893
λ_2	1.5000	1.5728	0.5735	0.0053	0.3848	0.6160	1.0723
λ_3	0.3000	0.1919	0.2535	0.0117	0.0640	0.2287	0.3251
λ_4	0.1000	0.1265	0.0951	0.0007	0.0092	0.0923	0.0254
λ_5	0.3000	0.2613	0.2090	0.0015	0.0481	0.2160	0.2226
λ_6	0.1000	0.1697	0.1032	0.0049	0.0099	0.0713	0.1500
λ_7	0.0100	0.0945	0.0957	0.0071	0.0136	0.0806	0.1359
λ_8	0.1000	0.1812	0.0978	0.0066	0.0091	0.0505	0.1044

Table 7.7: RMSE of the estimated Fourier series coefficients by PSO GSA and GSA for single defect cases.

Algorithm	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
PSO GSA	0.040	0.0757	0.0958	0.2212	0.0475	0.1214
GSA	0.1611	0.2496	0.2303	0.3231	0.2030	0.3085

Table 7.8: Statistical parameter for numerical case 7 (figure 7.6a): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	0.5000	0.5328	0.2860	0.0011	0.0988	0.3125	0.6321
λ_2	1.9000	2.0224	0.3173	0.0150	0.1178	0.3207	2.1183
λ_3	0.0000	0.1695	0.1695	0.0287	0.0338	0.0712	0.1763
λ_4	0.1000	0.1454	0.1162	0.0021	0.0166	0.1207	0.2676
λ_5	0.1000	0.0966	0.0961	0.0000	0.0093	0.0965	0.1388
λ_6	0.3000	0.1760	0.1348	0.0154	0.0238	0.0920	0.4312
λ_7	2.2000	2.2452	0.3452	0.0020	0.1309	0.3590	2.3727
λ_8	0.4000	0.4801	0.2586	0.0064	0.0807	0.2725	0.7868
λ_9	0.0000	0.0901	0.0901	0.0081	0.0158	0.0878	0.0885
λ_{10}	0.1000	0.1010	0.0903	0.0000	0.0085	0.0924	0.1402
λ_{11}	0.1000	0.0813	0.0853	0.0003	0.0081	0.0879	0.0608
λ_{12}	0.3000	0.2699	0.2298	0.0009	0.0557	0.2341	0.3202

another one is on the bottom right corner respectively. The estimation of the defect boundary by PSO GSA is in close agreement with the true geometry when compared with the estimation result from the GSA which is presented in figure 7.6a. Another two defects scenario on the graphene sheet is labeled as case 8 in which both defects are on the side side of the graphene, i.e., the left side which is seen in the figure 7.6b. In this case, the GSA estimated the defect location with poor accuracy when compared with PSO GSA. We can see that the PSO GSA has estimated both defect locations on the graphene sheet with good accuracy than GSA. In both these cases, i.e. cases 7 and 8, each defect is defined with 6 Fourier series coefficients. To include a complex-shaped defect in case 9 one defect is defined with 6 coefficients but another defect is defined with 8 coefficients. In this case, there are now 14 coefficients to be estimated by the algorithm. The true defect geometry along with the estimated geometry by PSO GSA and GSA are presented in figure 7.6c. From the figure, we can see that the PSO GSA could estimate the location of the defect with good accuracy than the GSA. But the complex shape of the defect was not estimated with good accuracy. In all cases of

Table 7.9: Statistical parameter for numerical case 8 (figure 7.6b): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	0.5000	0.5361	0.2474	0.0013	0.0868	0.2924	0.2311
λ_2	1.9000	2.0901	0.3471	0.0361	0.1334	0.3119	2.0321
λ_3	0.1000	0.1803	0.0903	0.0064	0.0083	0.0427	0.1965
λ_4	0.1000	0.1257	0.1206	0.0007	0.0176	0.1301	0.1284
λ_5	0.1000	0.1082	0.0897	0.0001	0.0088	0.0932	0.1228
λ_6	0.3000	0.1927	0.1488	0.0115	0.0276	0.1267	0.3139
λ_7	0.5000	0.4933	0.3100	0.0000	0.1036	0.3218	0.3538
λ_8	0.5000	0.5504	0.2739	0.0025	0.0898	0.2954	0.7949
λ_9	0.1000	0.0888	0.0908	0.0001	0.0085	0.0917	0.0426
λ_{10}	0.1000	0.1940	0.1523	0.0088	0.0264	0.1326	0.1406
λ_{11}	0.1000	0.0976	0.0883	0.0000	0.0085	0.0922	0.0182
λ_{12}	0.3000	0.2750	0.2046	0.0006	0.0498	0.2218	0.0483

two defects scenarios (cases 7, 8, and 9), PSO GSA could estimate the geometry of the defect with good accuracy but GSA failed to estimate the defect location and shape. Table 7.8, 7.9, and 7.10 presents the statistical parameters for the two defects cases. The standard deviation of error from the table shows PSO GSA estimated with good accuracy even though the unknowns has increased. This was achieved by configuring the PSO GSA with optimum parameter value especially by increasing the number of particles from 250 to 600. Table 7.11 shows each defect error by both algorithms for the two defects cases. From the table, we can see that in two defects cases PSO GSA has estimated each defect with good accuracy than the estimation done by GSA. But for case 9, the RMSE value of both estimated defects by PSO GSA is higher than GSA. However, if we look at the estimated boundary of the defects in figure 7.6c we can see that the estimated shape of the defect with PSO GSA has a low error. The accumulated RMSE for all the coefficients estimated by PSO GSA for cases 7, 8, and 9 are 0.0939, 0.1601, and 0.2889 respectively. Also, the RMSE for the estimated result from GSA for those particular cases is 0.1917, 0.2484, and 0.1603 respectively.

Table 7.10: Statistical parameter for numerical case 9 (figure 7.6c): mean $\bar{\lambda}$, mean absolute error \overline{MAE} , mean error squared \overline{MES} , mean square error \overline{MSE} , and standard deviation E_{sd} of Fourier series coefficient λ_i estimated using PSO GSA

Fourier Coef.	True value	$\bar{\lambda}$	\overline{MAE}	\overline{MES}	\overline{MSE}	E_{sd}	GSA
λ_1	0.5000	0.4304	0.2595	0.0048	0.0789	0.2721	0.4187
λ_2	1.9000	1.8440	0.2939	0.0031	0.1030	0.3159	2.0373
λ_3	0.0000	0.1311	0.1311	0.0172	0.0246	0.0864	0.0929
λ_4	0.1000	0.1220	0.1125	0.0005	0.0162	0.1253	0.2633
λ_5	0.1000	0.0871	0.0829	0.0002	0.0078	0.0872	0.0638
λ_6	0.3000	0.2582	0.1691	0.0017	0.0315	0.1725	0.2452
λ_7	2.0000	2.0075	0.3275	0.0001	0.1381	0.3716	2.3562
λ_8	0.5000	0.6092	0.4100	0.0119	0.2491	0.4870	0.4586
λ_9	0.2400	0.2524	0.2085	0.0002	0.0479	0.2186	0.1505
λ_{10}	0.0000	0.0758	0.0758	0.0057	0.0080	0.0480	0.0974
λ_{11}	0.0000	0.0658	0.0658	0.0043	0.0065	0.0469	0.0221
λ_{12}	0.1500	0.1492	0.0762	0.0000	0.0070	0.0834	0.1516
λ_{13}	0.0500	0.0563	0.0457	0.0000	0.0023	0.0474	0.0805
λ_{14}	0.0000	0.0510	0.0510	0.0026	0.0055	0.0534	0.0249

Table 7.11: RMSE of the estimated Fourier series coefficients by PSO GSA and GSA for two defects cases.

Algorithm	Case 7		Case 8		Case 9	
	Defect 1	Defect 2	Defect 1	Defect 2	Defect 1	Defect 2
PSO GSA	0.1252	0.0591	0.0740	0.3936	0.2544	0.3069
GSA	0.1898	0.1932	0.1590	0.5438	0.1280	0.1849

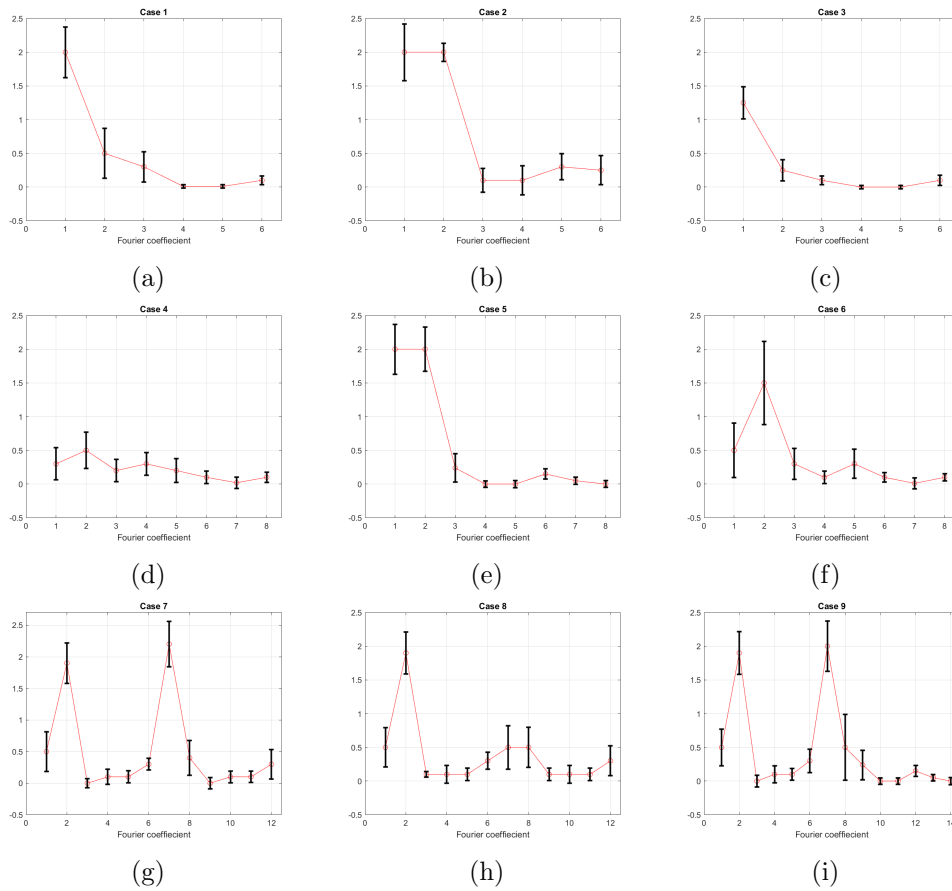


Figure 7.7: Standard deviation error of the estimated Fourier series coefficients of the all 9 numerical cases.

Thus, in all 9 numerical cases, PSO-GSA estimated the defect on the graphene sheet with good accuracy when compared to GSA. The statistical parameters of PSO-GSA shown in table 7.1 - table 7.10 show that the performance of PSO-GSA is better than that of GSA for most of the cases. The standard deviation error of the estimated Fourier series coefficients for all the 9 numerical cases is plotted. From the graph, we can see that the error is a little high for the coefficients defining the defect location when compared to others as shown in figure 7.7. Along with the unknown location of the defects in the graphene sheet, the background conductivity is also estimated. The true value of the background is set to $6.7 \times 10^4 mS/cm$ and both algorithms estimated the background conductivity of the graphene sheet. The estimated background conductivity of the graphene sheet by PSO-GSA is presented along with the estimated result.

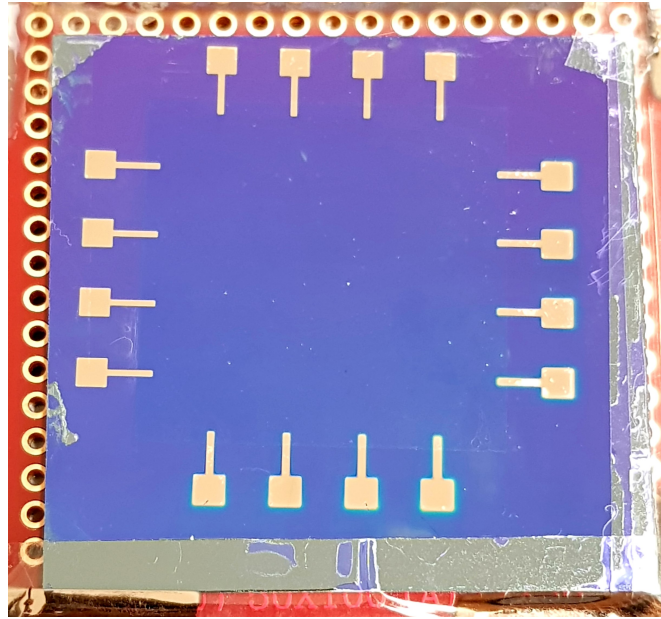


Figure 7.8: Electrode coated graphene sample used for the experimental study.

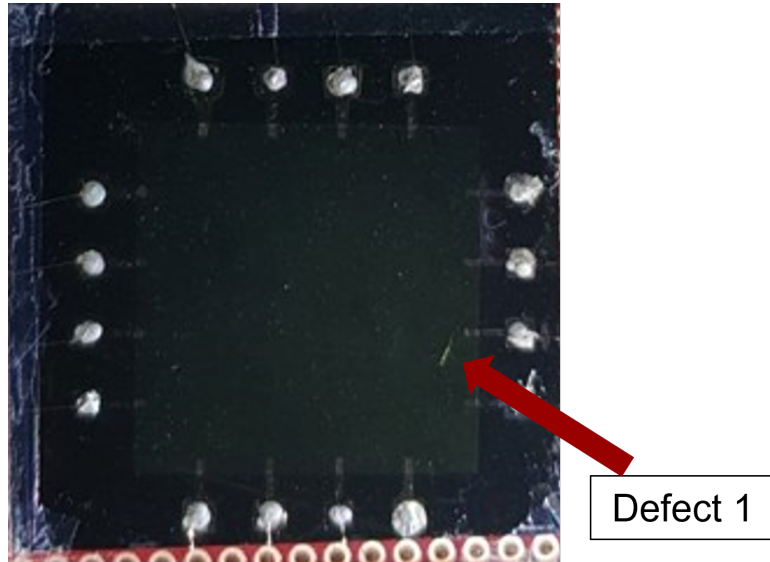
7.1.2 Experimental study

The EIT system setup used for the experiment consists of a constant current source (Agilent 4284A precision LCR meter) and a data acquisition system (NI PXI-1042Q) to measure the resultant voltage readings.

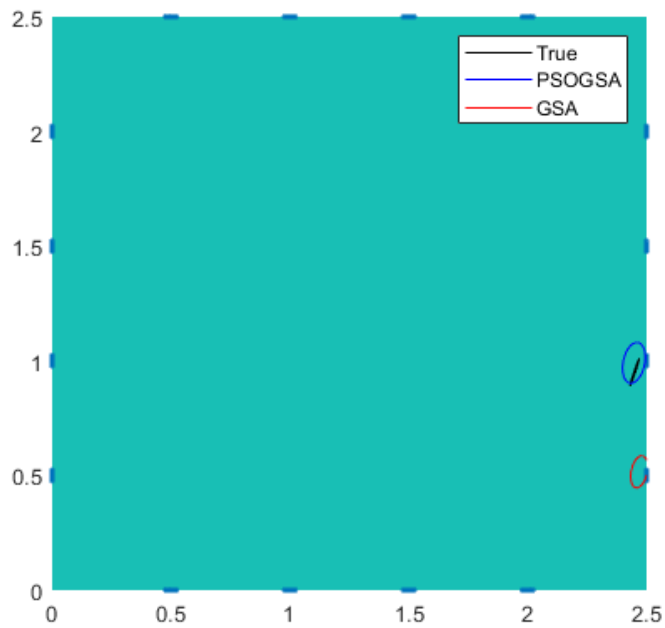
Preparation of graphene sample

In this section, a graphene sample on the SiO_2 substrate preparation is explained. A CVD processed graphene sample was purchased from Graphene Square Inc. For a mechanical supporter during the transfer process poly (methyl-methacrylate) PMMA (950 PMMA A4) was spin-coated at 300 RPM for 30 seconds on graphene synthesized copper foil and annealed on a hot plate at 100°C for 5 minutes. After the supporter is coated, the sample was put into an ammonium persulfate solution to remove the copper foil. PMMA/Graphene was cleaned with deionized water and was later transferred to SiO_2/Si wafer, which is the target substrate used for the experiment, after which PMMA was dissolved using acetone.

A graphene sample of size 2.5×2.5 cm is used in this study. A total of 16 copper electrodes are coated on the graphene sample using the sputtering method with a mask which is designed with the electrode location and shape. Each side of the graphene



(a)



(b)

Figure 7.9: Experimental study for case 1 with defect identification on graphene of size 2.5×2.5 cm. (a) Graphene with a single defect. (b) Experimental result for defect location on the graphene sheet. The estimated location of the defect by PSOGSA and GSA is represented by a blue circle and red circle, respectively.

sample has 4 electrodes placed equidistantly and are separated by a gap of 0.5 cm as shown in figure 7.8. A gold wire, connecting the EIT measurement system to the graphene sample, was attached to the electrodes with the help of silver paste. A current of the constant amplitude of 0.1 mA is injected into the graphene sheet using a crossed injection pattern. For 16 electrode setup, a total of 128 voltage measurement reading is obtained from the boundary electrodes.

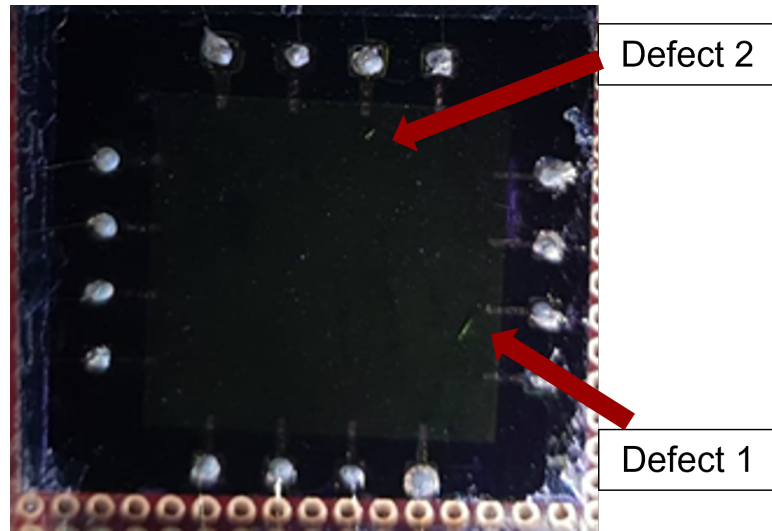
Result

As a first experimental case a single defect, created with a pointed knife, on the surface of a graphene sheet is presented in figure 7.9a where we can see the defect is located near 7th electrode. For estimating this single defect scenario, PSO-GSA is configured with 450 particles and iterated for 25 iterations.

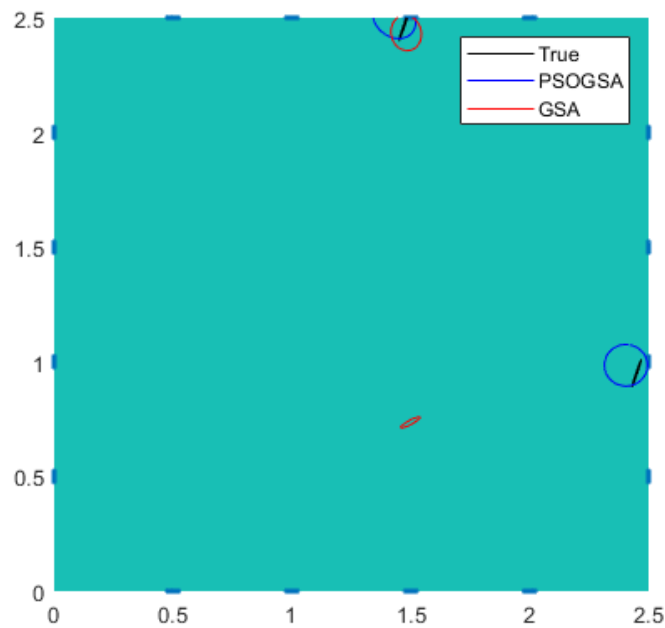
Figure 7.10a presents a true scenario where two defects are present on a graphene sheet. In this scenario, one defect is located on the lower right side of the graphene sheet near the 7th electrode and the second defect is located on the top side of the sample near 3rd electrode. Both defects are made with a pointed knife on the surface of the graphene sheet. The presence of two defects has increased the number of unknown parameters to be estimated. As the number of unknowns is increased, thus the number of particles in the algorithm is increased to 750 for better estimation results.

Figure 7.9b shows the estimated result of the single defect location by PSO-GSA and GSA together with the true location. We can see that the estimated location of the single defect by PSO-GSA is in close agreement with the true location whereas the estimated location by GSA is far away from the true location. The estimated location of defects for the second experiment case of two defects on graphene is presented in figure 7.10b. In this, we can see that the PSO-GSA has estimated the location of both defects successfully with good accuracy whereas GSA could only estimate one defect location with good accuracy but failed to estimate the 2nd defect on graphene. Also, the background conductivity of graphene is estimated for both experimental cases along with defect location. The estimated background conductivity of the graphene sheet can be seen in table 7.12.

From both the numerical and experimental studies, we see that the PSO-GSA performed



(a)



(b)

Figure 7.10: Experimental study for case 2 with defects identification on graphene of size 2.5×2.5 cm. (a) Graphene with two defects. (b) Experimental result for defect location on the graphene sheet. The estimated location of a defect by PSO GSA and GSA is represented by a blue circle and red circle, respectively.

Table 7.12: Estimated conductivity (mS/cm) of graphene sheet by PSO GSA and GSA

Case	PSO GSA	GSA
Single defect	62464.28	65795.49
Two defects	66846.33	61000.00

better in all the scenarios when compared with GSA. The estimated conductivity values by PSO-GSA are presented in the color bar of all the estimated results. In all the cases, PSO-GSA has a very good estimation result with good accuracy for estimating the Fourier series coefficients describing the geometry of the defect along with background conductivity of graphene sheet than GSA.

7.2 Conclusion

In this study, estimation of the defect boundary along with the background conductivity of graphene is done by PSO-GSA. The performance of the proposed algorithm for EIT is analyzed with different numerical cases for a single defect and two defects. Monte Carlo simulation is used for analyzing the statistical parameters which also verifies the robustness of the proposed algorithm. The effect of different parameter configurations (weighting factors, particle size, and total iteration) on PSO-GSA is analyzed. Based on this analysis the initial conditions of PSO-GSA are configured for better performance and the iteration size is configured to 25 steps. The number of unknowns to be estimated by PSO-GSA is directly linked with the particle size. Thus, for the two defects scenario, a higher particle size is required.

Nine different cases are considered in the numerical study to estimate the location of the defects and the background conductivity of the graphene sheet. The estimation error is measured with RMSE for both algorithms for all the cases. In the two defects numerical cases, GSA failed to estimate the defect location and its shape. Whereas, PSO-GSA has a more accurate estimation than the other algorithm. In the experimental study proposed algorithm has estimated defects with good accuracy whereas GSA failed to estimate the defect location. The better estimation result is obtained from the PSO-GSA as it has the exploration capability of PSO, which makes it better than GSA.

8 Interlayer boundary estimation by PSO-GSA-DNN

This chapter presents the estimation of the interlayer of the subsurface. The two different boundaries are used to separate the layers of the subsurface. Both boundaries are estimated by using a DNN which is tuned by PSO-GSA. Firstly the hyperparameter of DNN is tuned by PSO-GSA. The number of nodes in the hidden layers is tuned by PSO-GSA. After the DNN is tuned, the model is used for estimating the interlayer boundaries of the subsurface. The following section defines the mathematical representation of the interlayer boundary and the later section presents the study.

8.1 Interlayer boundary representation

When the conductivity values of the subsurface are known a priori then the shape and location of the interlayer boundary become the unknown to be estimated. Considering the subsurface is layered, the open boundary between the two layers was approximated as discrete front points $B_\gamma(\gamma = 1, 2, \dots, \Gamma)$ located on the boundary (Khambampati et al., 2013; Kim et al., 2007b). Here defines the total number of front points, which describes the interlayer boundary. The front point location is given by

$$B_\gamma = (x_\gamma, y_\gamma), \gamma = 1, 2, \dots, \Gamma \tag{8.1}$$

where x_γ is the reference point located on the surface of the domain as shown in the figure 8.1. The front point is defined as the vertical distance from the reference point

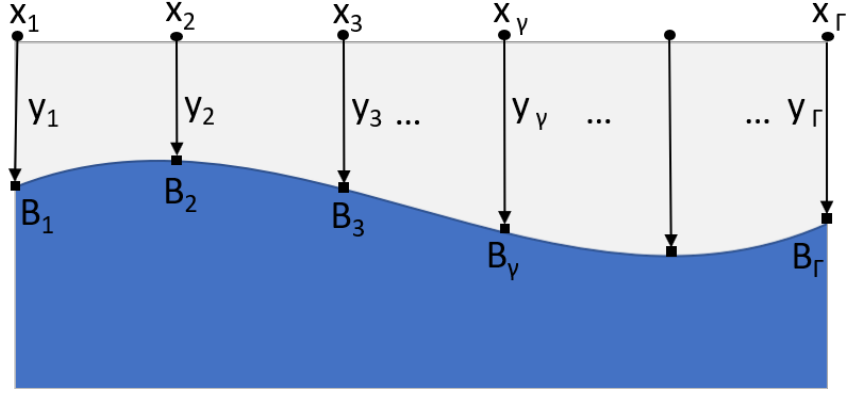


Figure 8.1: Interlayer boundary parameterized with the discrete front points.

$(, 0)$ located on the surface of the domain and the unknown parameter is

$$y = (y_1, y_2, \dots, y_\Gamma)^T \quad (8.2)$$

8.2 Results

In this work, the interlayer boundary's front points inside the subsurface domain are estimated by the DNN model. It estimates the shape, size, and location of the interlayer boundary, which maps the non-linear relationship. In this, we assume that the internal conductivity of the layer is constant and is known prior with the help of the borehole method. The interlayer boundary of the subsurface is described by the front points method. The dataset is used to train and validate the model which contains measured voltages and related interlayer boundary front points. To estimate the interlayer boundary front points by the DNN model, we define the input layer neurons as surface voltage measurements and the output layer neurons as front points. The hidden layers are used to extract the features from the training dataset, which is used to map the relationship between input and output data. This relationship mapping is done using a mapping function (H) defined as

$$H_\theta(V) = \sum_{k=0}^m W^k V^k = W^T(V) \quad (8.3)$$

where W is the weight of the node, V is the measurement voltage reading, and m is the total number of data samples. The DNN model is trained to learn the weights (W) of the nodes of the hidden layers. The training of the model for learning weights is done by minimizing the cost function which expresses the sum of the errors between the estimated output of the DNN model and the desired output. The least-square cost function is minimized to determine the weights of the node, which is expressed as

$$\theta = \underset{\theta}{\operatorname{argmin}} \sum_{k=1}^m \| H(V^k) - y_{\gamma}^k \|^2 \quad (8.4)$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (8.5)$$

where α is the learning rate of the training. The DNN model updates the weights of each node which minimizes the cost function. The update of the weights is executed using Adam's optimization algorithm, see equation (8.5) (Kingma and Ba, 2014). The weights θ are updated using equation (8.5). The weights, θ , are updated by the DNN model during training which uses the training dataset. The trained DNN model is evaluated with unseen validation data.

Dataset preparation

DNN model is designed, trained, and evaluated for estimating the interlayer boundary front points inside the subsurface domain. Before the training of the model, preprocessing of the data sample is necessary. The model learns from the dataset and is heavily depends on data samples (Holzinger, 2019). If the dataset is not optimized, it is very hard to train the DNN model for optimized estimation. Data samples in the dataset should not be repeating and the bad data samples should be removed. In this work, the uniqueness of data sample is considered and is optimized before it is used to train the model. The dataset contains the measured voltage reading as input and the front points as output. The input voltage is normalized before it is used in the training, testing, or estimating the output by the DNN model. As the input data sample are different from each other, the range of it may be also different. It will make training of the model very difficult and model may not find the global minima due to it. To

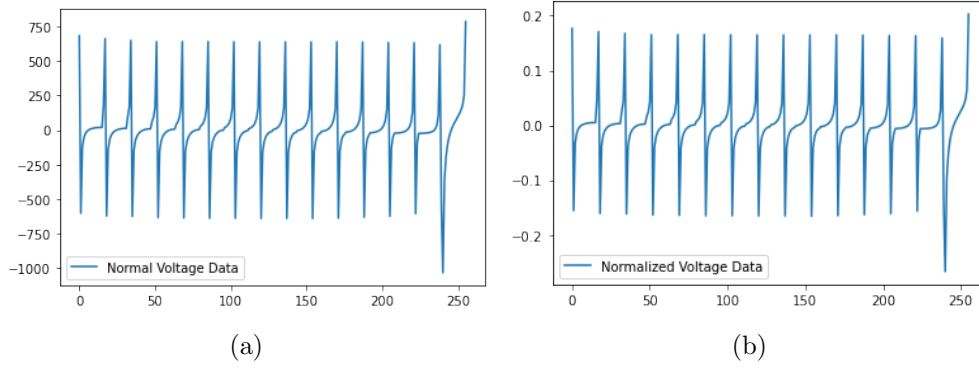


Figure 8.2: A sample of the input voltage reading from the dataset. (a) Normal voltage reading, (b) Normalized voltage reading.

increase the effectiveness of the learning rate, normalization is done. In this work we have used the Min-Max normalization method. In this method, the data is scaled in the range of $[0, 1]$ or $[-1, 1]$ (Bhanja and Das, 2018). The input value v of the voltage V is converted to normalized voltage v_{norm} of the range $[v_{max}, v_{min}]$ defined as

$$v_{norm} = \frac{(v_{max} - v_{min}) \times (v - minV)}{maxV - minV} \quad (8.6)$$

where $minV$ and $maxV$ are the minimum and maximum values of the voltage V in the data set. We have used scikit library to normalized the input data samples of the dataset (Pedregosa et al., 2011). The voltage reading samples of the dataset is presented in the figure 8.2. From the figure we can see that the information of the voltage is not changed, however the magnitude is change to the range of $[-1, 1]$.

Numerical study

The domain used for the study has a size of $10m$ in-depth and $20m$ in width. A total of 16 surface electrodes are attached to the domain, illustrated in figure 8.3. These electrodes are located on top of the subsurface domain representing a surface electrode placed $1m$ apart. An adjacent current pattern is used in this study which generates 256 independent voltage readings. EIT forward problem is solved using the finite element method to compute the boundary voltages for the resistivity profile with multi-layer boundaries represented using front points. The three layers in the subsurface domain are assumed as alluvium, clay, and argillite. The conductivity of those layers

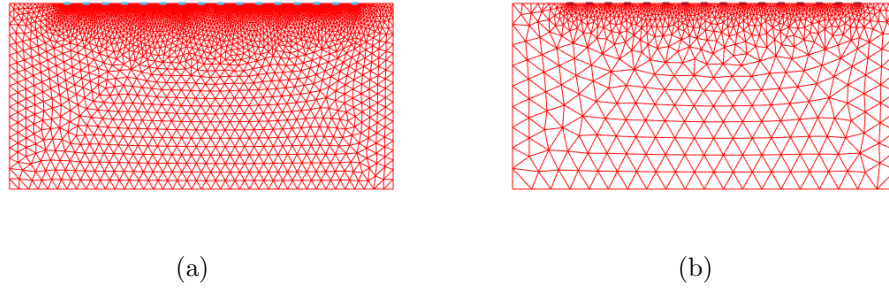


Figure 8.3: Subsurface mesh used for the calculation of the voltage based on the boundaries of the interlayer.

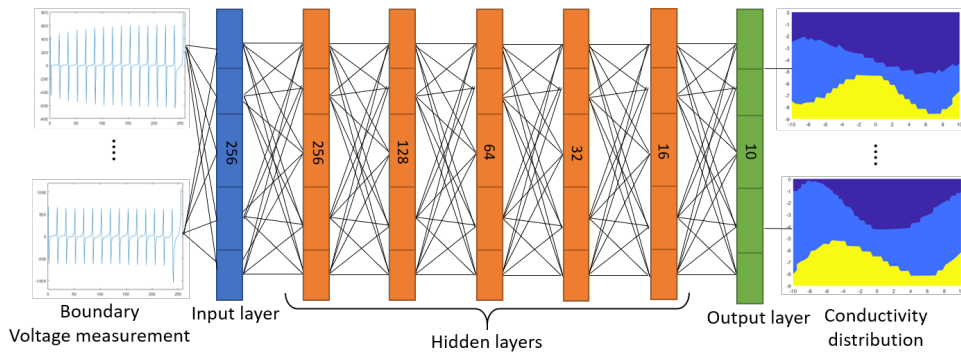


Figure 8.4: Training scheme of the deep neural network model for estimating the front points of interlayer boundaries. The number represents the number of nodes for each layer.

are $0.0013S/m$, $0.01S/m$, and $0.04S/m$, respectively (Katsube et al., 2004; McNeill, 1980). The three layers are separated by two interlayer boundaries that have different shapes and thicknesses. Each layer boundary is represented using five front points with linear interpolation. The interlayer boundaries are considered noncrossing boundaries, and layers are considered piecewise with constant conductivity. The bottom layer is considered argillite, the middle layer is clay and the top layer is alluvium. A continuous amplitude current of $1Amp$ is injected into the domain, and the surface voltage is measured across all surface electrodes. EIDORS is used for FEM numerical calculation which generates voltage measurement reading based on the corresponding interlayer boundaries front-points of the subsurface domain. These voltage measurement readings and the corresponding front points make a dataset (training and validating). The training dataset is used to train the DNN model for learning the relationship between measured voltage reading and the interlayer boundary front points. The testing dataset is used to validate the training of the DNN model.

In this work, the input node is 256 and the output node is 10. The DNN model is designed in such a way that the number of nodes decreases in a sequential manner known as funneling. The number of nodes in the hidden layers is 256, 128, 64, 32, and 16. The one training dataset sample consists of 256 independent boundary voltage measurements as input, and the corresponding output consists of 10 front points reading corresponding to the interlayer boundaries. The training scheme for the model is illustrated in figure 8.4. Tensorflow (Abadi et al., 2016) library in python is used to implement the DNN algorithm with mean square error (MSE) as loss function and Adam optimization algorithm for optimization on a workstation configured with Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 8GB RAM, NVIDIA GeForce GT730 GPU, Windows 10. The model is trained on a dataset of 15120 samples with 164 epochs and a batch size of 100. The learning rate used during the training of the DNN model is $1e^{-3}$. A validation dataset is used to validate the training process, and it contains 3780 separate data samples. This DNN model is used for comparing with the PSO-GSA tuned DNN model. The model learns the weight of the nodes from the dataset but the hyperparameter such as the number of nodes in the hidden layer is not learned. The optimum value of the hyperparameter makes the DNN model optimized for the presented problem. Optimizing the hyperparameter manually is not a suitable solution which was explained in section 5.4.

Analysis of PSO-GSA for tuning DNN

The PSO-GSA algorithm is analyzed with different particle size for this study. Seven different particle sizes are configured to PSO-GSA which will provide the optimized node size of the hidden layers of the DNN model. Particle sizes range from 5 to 125 and the fitness for each are evaluated. Figure 8.5a shows the different particles size used for PSO-GSA for determining the node size of the hidden layers. As the particle size increases, the execution time also increases. The ratio by which the execution time increases is linear for lower particle size whereas it increases exponentially for higher size as shown in figure 8.5b. The fitness of different particle size is compared and presented in figure 8.5c. From the figure, we can see that 85 particle size has the lowest error when compared with others.

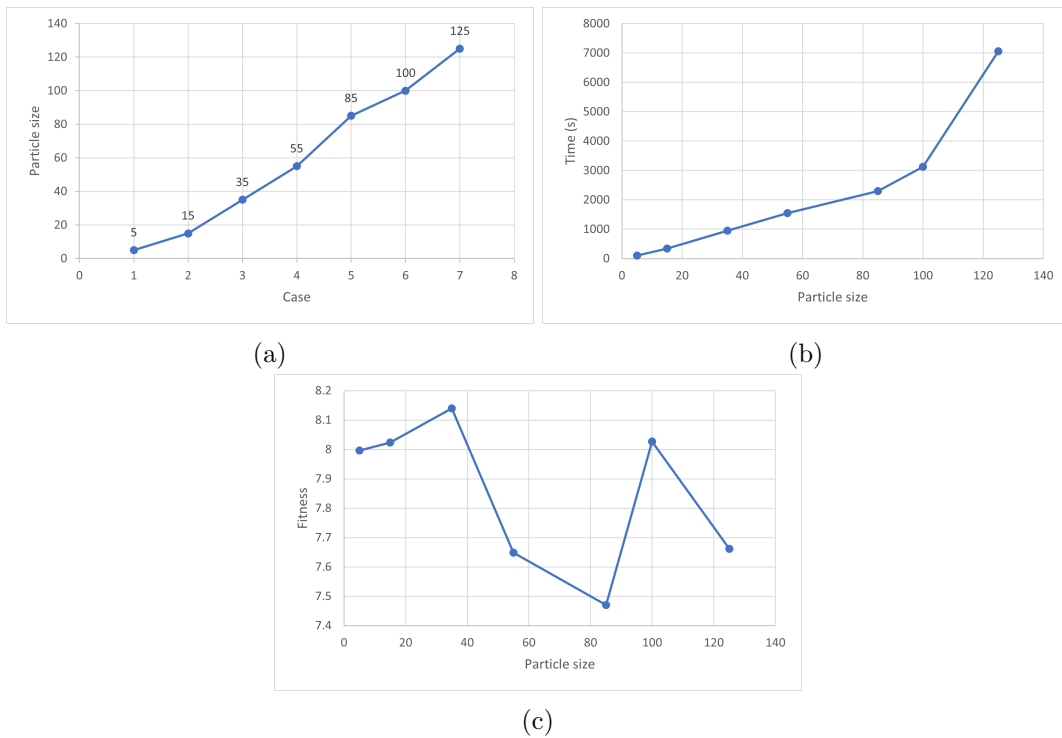


Figure 8.5: Analysis of PSO-GSA with different number of particle size to tune DNN model. (a) Different particle size. (b) Per iteration execution time. (c) Fitness of PSO-GSA.

Figure 8.6 presents the optimized value for the node size to be used in hidden layers of the DNN model. Each value is the optimum node size for the hidden layers provided by the PSO-GSA algorithm with a different particle size. The node size estimated by the PSO-GSA with 85 particles have estimated the optimum node size. The optimum node size of the hidden layers are 227, 237, 246, 243, and 159 respectively. This optimized DNN model is trained with the same database as described previously.

Analysis of tuned DNN

The PSO-GSA tuned DNN model is further optimized by analyzing other hyperparameters (batch size, learning rate, and activation functions). Initially, the different batch size is configured and the DNN model is trained for each configuration. The error of the predicted output of the DNN model is measured with MSE and the accuracy is measured with RMSE. The training loss and RMSE for the different batch size is presented in figure 8.8. This figure also presents the validation loss and RMSE of the DNN model for the validation dataset.

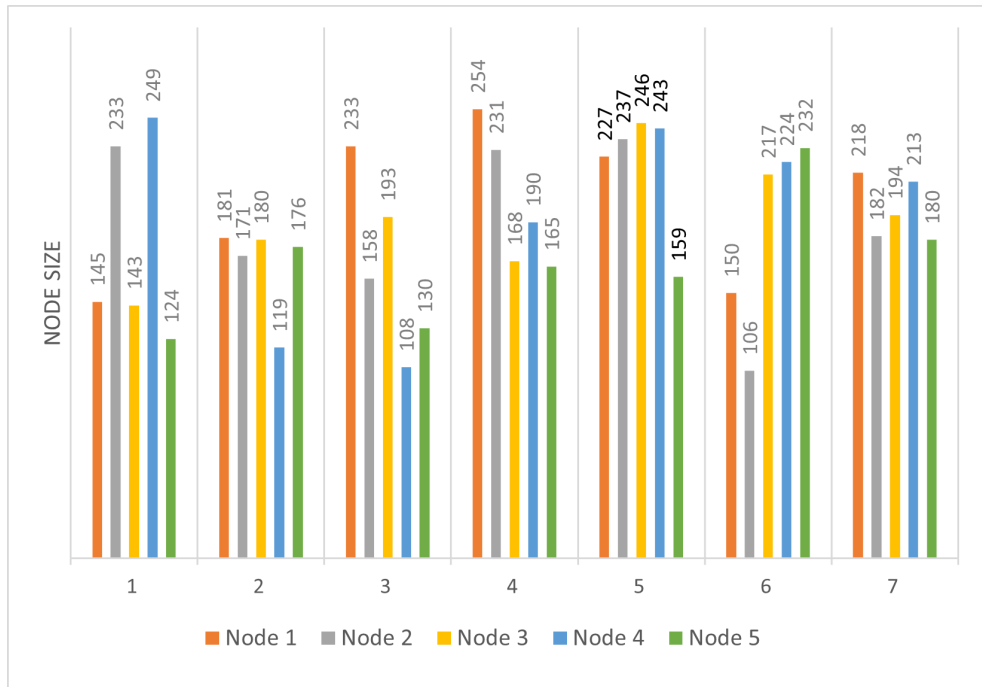


Figure 8.6

Figure 8.7: Optimum node size for hidden layer of the DNN model estimated by PSO-GSA.

The training and validation loss for the batch size of 25 has the lowest loss of the trained DNN model but the validation loss is high, thus the model is not generalized. The training loss for the batch size 100 and the validation loss is similar. Thus the model is more generalized for the unseen dataset and is used for the estimating of the interlayer boundaries of the subsurface.

With the configuration of the batch size of 100, the DNN model is trained with the different learning rates. Four different learning rates are analyzed and the corresponding loss and RMSE of training and validation are presented in figure 8.9. From the figure, we can see that the learning rate of $1e^{-3}$ gives the lowest possible loss for the DNN model. This learning rate is used in the DNN model for further analysis.

Next, the activation function analysis is conducted. Two activation function is used for this part. Tanh and RELU are configured to the hidden layers in a different configuration. The combination of the RELU and Tanh activation function gives the model a great improvement in the estimation (Li et al., 2020). Tanh takes any real value as input and outputs values in the range $[-1, 1]$. The larger input value has the

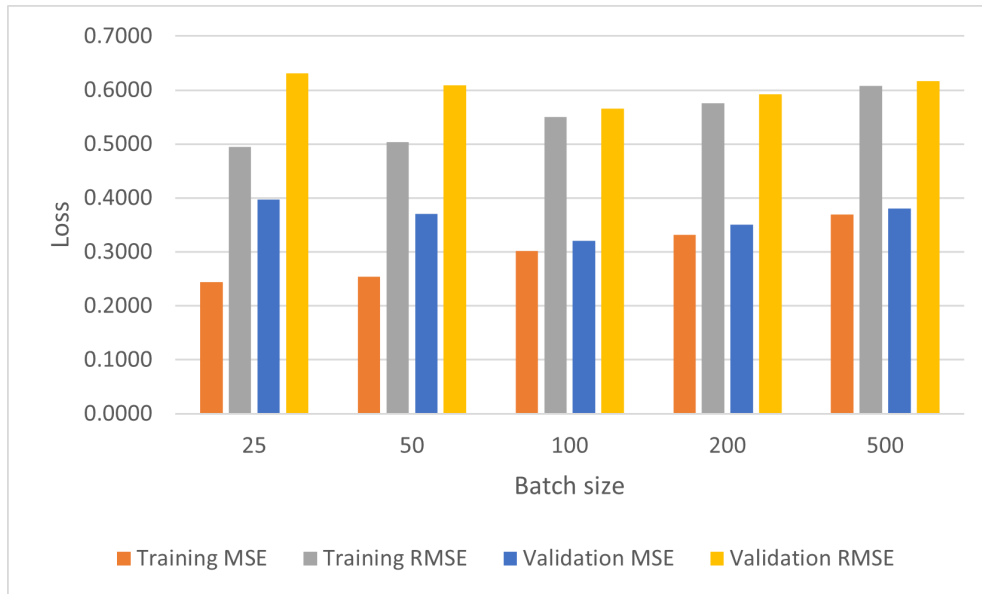


Figure 8.8: Training and validation loss and RMSE the PSO-GSA tuned DNN model with different batch size

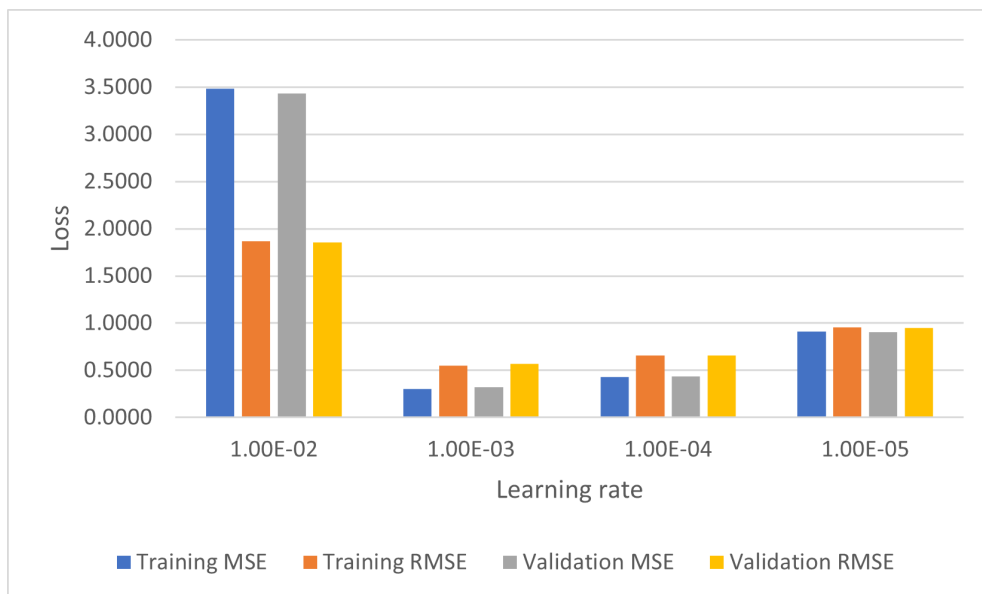


Figure 8.9: Training and validation loss and RMSE the model PSO-GSA tuned DNN model with different learning rate.

output value close to 1, whereas the smaller input has the output close to -1. Tanh activation is mathematically defined as

$$\Theta(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (8.7)$$

The different configuration of the activation function along with the training and validation loss and RMSE of the DNN model is presented in figure 8.10. The related training and validation loss and RMSE is also plotted in this figure. A total of 28 different configuration of the activation function is analyzed. The second configuration of the activation function gives the lower MSE of training and validation for the DNN model whereas some configuration shows over fitted model which is seen in the figure. The best configuration of the activation function is RELU, RELU, RELU, RELU, and Tanh for the five hidden layers.

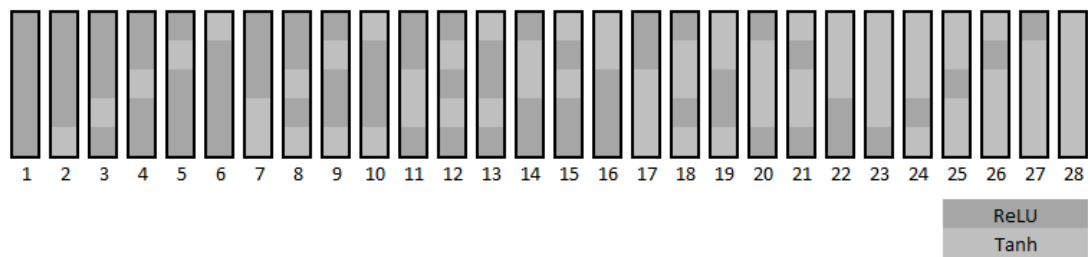
Boundary estimation

These all parameters are used for training the DNN model to estimate the subsurface interlayer boundaries. The estimation result of the tuned DNN model is compared with the 7-layer funnel shaped DNN model which was described previously. RMSE (see equation (6.10)) and Pearson correlation coefficient (PCC) are evaluated the accuracy of the estimated boundary by the models. RMSE is also used for checking the error between estimated front-points with the actual front-points of the interlayer. Also, the Pearson correlation coefficient (PCC) (Lee Rodgers and Nicewander, 1988) is used to find the correlation between the estimated front points and the true front points that represent interlayer boundaries. PCC is defined as

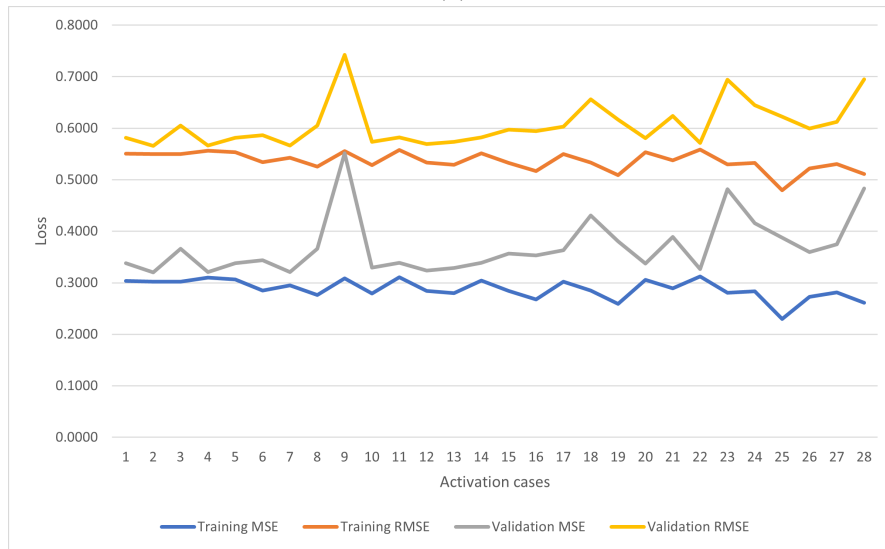
$$r = \frac{\sum_{i=1}^N (y_i - \bar{y}) \sum_{i=1}^N (y_i^t - \bar{y}^t)}{\sqrt{\sum_{i=1}^N (y_i - \bar{y})^2 \sum_{i=1}^N (y_i^t - \bar{y}^t)^2}} \quad (8.8)$$

where y and \bar{y} is the estimated and mean of estimated front points, y^t and \bar{y}^t is the true value and mean of true value of the front points respectively, and N is the total number of front points.

A total of 7 numerical cases are analyzed for this work where the PSOGSA tuned DNN



(a)



(b)

Figure 8.10: (a) Different activation function configuration for the hidden layers. (b) Training and validation loss and RMSE the model for the same.

estimation is compared with untuned funnel shaped DNN. In these cases, rocks are added to the subsurface domain which is not present in the training dataset. These rocks act as anomalies and are randomly generated for each case, considered noise in the dataset, representing a more realistic scenario. We tried to consider all types of rocks in the study that are found in the subsurface domain. The conductivity of these rocks are considered within a range of $2e^{-4}S/m$ to $2e^{-7}S/m$ (McNeill, 1980).

The first two cases of the interlayer boundary represent the linear boundary case. Case 1 is presented in figure 8.11 which shows the true conductivity distribution of the subsurface along with the estimated boundaries of the subsurface by the DNN and PSO-GSA tuned DNN. From the figure, we can see that the PSO-GSA tuned DNN model estimated both interlayer boundaries with good accuracy. Case 2 estimation is also reflecting the same accuracy of the PSO-GSA tuned DNN over the other model. Case 3 presented in figure 8.13a shows the linear and non-linear boundaries scenario.

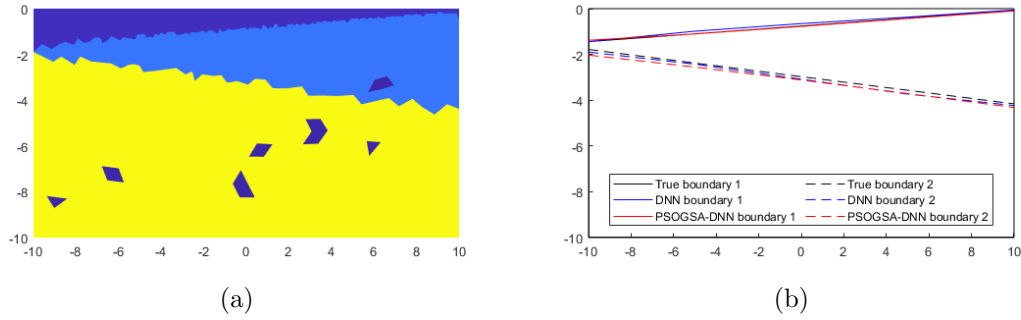


Figure 8.11: Numerical results with case 1 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

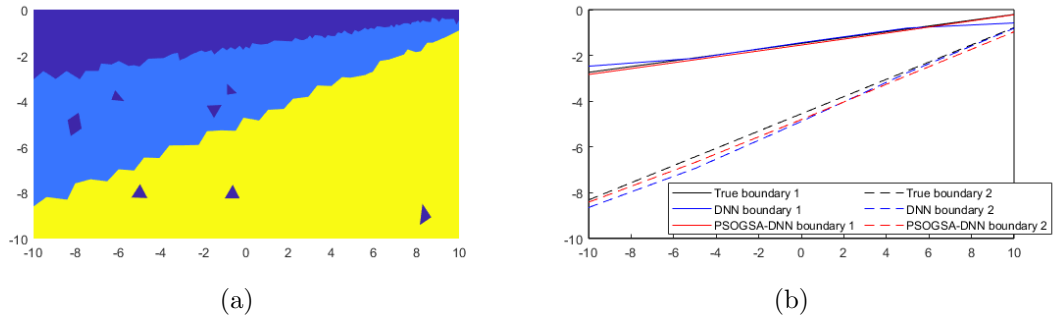


Figure 8.12: Numerical results with case 2 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

The estimation result by both DNN and PSO-GSA tuned DNN is presented in figure 8.13b which shows that the estimated boundaries by DNN have deviated from the true shape. However, the estimation by the PSO-GSA tuned DNN model has to close resemblance to the true shape and location of the interlayer boundaries.

Case 4 presents the equally distributed subsurface layers whose true conductivity distribution is presented in the figure 8.14a. From the figure, we can see that the rocks are more in the bottom layers. The DNN model failed to estimate the shape of both boundaries, however, PSO-GSA tuned DNN model could estimate boundaries close to the true shape which is presented in figure 8.14b. Case 5 presents another case where the bottom layer is thicker than the other two layers. In which PSO-GSA tuned DNN could estimate the boundary shape and location close to the true boundaries when compared with DNN as shown in figure 8.15.

Case 6, and 7 show a scenario of the interlayer subsurface where the lower layer is

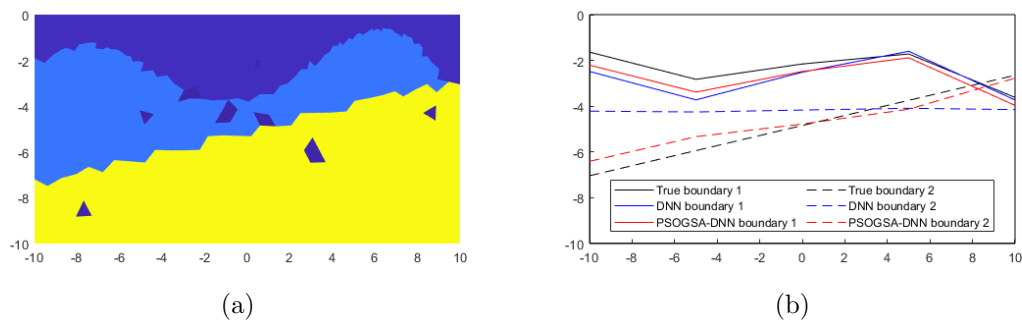


Figure 8.13: Numerical results with case 3 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

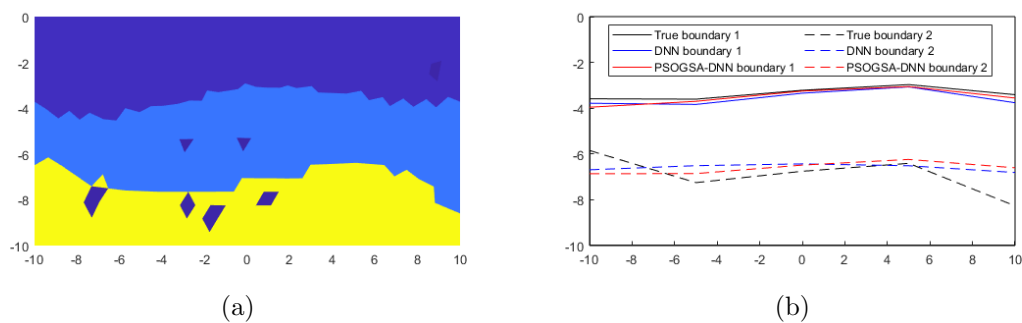


Figure 8.14: Numerical results with case 4 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

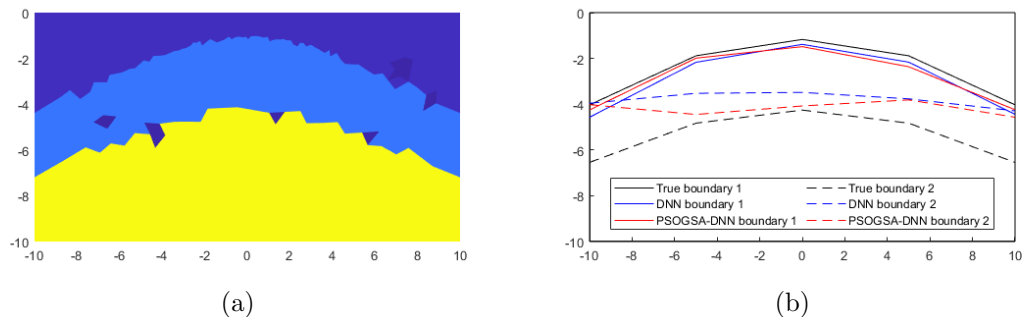


Figure 8.15: Numerical results with case 5 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

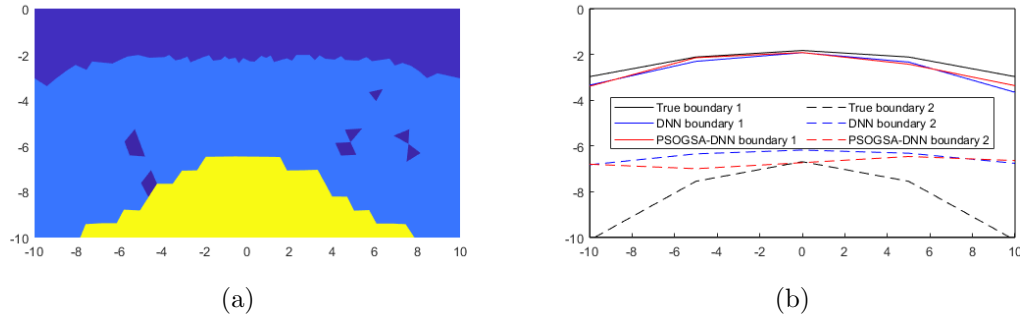


Figure 8.16: Numerical results with case 6 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

very small when compared to other layers. The true conductivity distribution of these cases are presented in figures 8.16a and 8.17a. In cases 6 and 7 the middle layer is bigger than the other layers and most of the rocks are concentrated on it. In both cases PSO-GSA tuned DNN estimated the upper boundary shape and location close to the true boundary when compared with the DNN model. However, the lower boundary in both cases the model estimation result were similar which can be seen in figure 8.16b and 8.17b.

From these cases, the estimation of the interlayer boundaries front-point by PSO-GSA tuned DNN model has done better estimation than that of DNN model. In the complex scenario of the non-linear boundaries, both the models produced a similar result as shown by the figures. The RMSE and PCC of all the cases are tabulated in table 8.1. From the table, we can see that PSO-GSA tuned DNN model has lower RMSE for most of the cases presented in the studies. Hence, the estimation done by PSO-GSA tuned DNN model is more accurate to the true boundary. The RMSE and PCC of the estimated interlayer boundaries of the subsurface by DNN and PSO-GSA-tuned DNN is presented in the figure 8.18.

The trained DNN model for estimating the front points of open interlayer boundaries in the subsurface domain has very good accuracy with the true scenario. However, there are limitations to this approach. The number of layers in the subsurface is assumed to be known a priori. The conductivity distribution is assumed to be uniform within each layer and the value of the conductivity of the layer is known beforehand also. The training of the DNN model takes time, but once the model is trained estimation

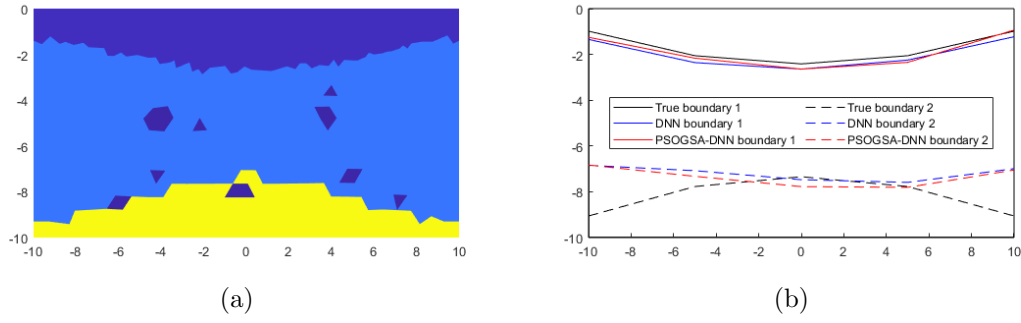


Figure 8.17: Numerical results with case 7 for interlayer boundary estimation of subsurface. (a) True conductivity profile. (b) Reconstructed boundaries using PSO-GSA tuned DNN and DNN model.

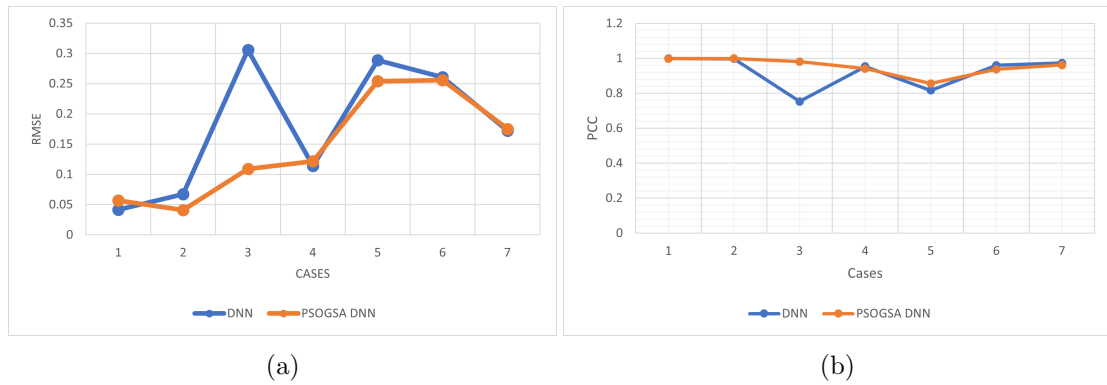


Figure 8.18: RMSE and PCC for the estimated front points by PSO-GSA-tuned DNN and DNN model.

of inter-layer boundaries is done in a few seconds. Also, the PSO-GSA tuning of the hyper-parameters consumes more resources than training the DNN model only. The tuned model works very well for the opened interlayer boundary but with some limitations. Both the models are not trained for the closed interlayer boundaries and the horizontal opened boundaries. Also, layer boundaries need to be end to end and do not intersect with each other.

8.3 Conclusion

In this work, we have tuned the DNN model using PSO-GSA. PSO-GSA is used for optimizing the number of nodes in the hidden layers of the DNN model. The DNN model is used for estimating the interlayer boundaries of the subsurface using electrical impedance tomography. The conductivity inside the subsurface is assumed to be known, and the DNN model estimates the interlayer boundary. The layers are opened and

Table 8.1: Comparison of RMSE and PCC for the estimated front points by PSO-GSA tuned DNN and DNN model for interlayer boundaries in subsurface domain.

Cases	RMSE		Correlation coefficient	
	DNN	PSO-GSA-DNN	DNN	PSO-GSA-DNN
1	0.0415	0.0570	0.9990	0.9985
2	0.0675	0.0409	0.9973	0.9994
3	0.3060	0.1093	0.7528	0.9813
4	0.1142	0.1216	0.9534	0.9414
5	0.2886	0.2540	0.8164	0.8555
6	0.2607	0.2561	0.9604	0.9371
7	0.1724	0.1753	0.9728	0.9616

are present across the domain. This DNN model is not trained for the closed and/or horizontal interlayer boundaries. Discrete front points describe the interlayer boundary. An inverse problem of estimating the interlayer boundary front points is done with a DNN model.

PSO-GSA was initially analyzed for various particle size configurations and the optimized parameter for hidden layers is used for training the DNN model. The DNN model is trained using the pair of boundary voltage readings and the corresponding front points. The measured voltage readings are used as an input, and interlayer boundary front points are used as output, making the dataset. The dataset is divided into 2 parts, i.e., training and validation, containing non-repeated data samples. A noisy measured boundary voltage is fed to a trained DNN model to estimate the corresponding interlayer boundary front points.

The interlayer boundaries for subsurface estimation are done with PSO-GSA-tuned DNN model, and the performance of the proposed model is compared against 7 layer funnel-shaped DNN model. The training of both models is done with the same configuration. The PSO-GSA-tuned DNN model is 7 layer model and the estimation done by it was better than that of funnel-shaped DNN model.

9 Conclusion

In this study, estimation of the boundary of an anomaly inside the domain has been done. A gravitational search algorithm (GSA) is used as a reconstruction algorithm as it has the advantages of strong exploitation performance and fast convergence speed. In the boundary estimation approach, the conductivity of the regions is known as prior information. The conductivity distribution inside the domain is assumed to be homogeneous. The boundary to be estimated is of two types, i.e. open and close boundary.

The closed boundary of the anomaly is defined by the coefficients of the truncated Fourier series. In this, the boundary is assumed to be closed and smooth. The bladder boundary in the pelvic domain and the defect boundary on the graphene sheet domain is defined using the Fourier series coefficients. The bladder boundary is estimated by GSA and the algorithm performance is compared with the particle swarm optimization (PSO) algorithm. A different scenario of bladder size is used in the study for both numerical simulation and phantom studies. The RMSE of the estimated coefficients is measured for the accuracy of GSA. From the RMSE we see that the GSA estimation result is more accurate than that of the PSO in all the cases of numerical study and phantom experiment of the bladder. The GSA was found to have a poor exploration performance during the search process due to the unavailability of diversity information. This information is available with PSO during the optimization process. Hence, to enhance the performance of GSA, the algorithm was combined with PSO to make a new hybrid algorithm known as PSOGSA. The performance of the proposed algorithm PSOGSA is analyzed for the estimation of the defect(s) boundary on the graphene sheet. The Fourier series coefficients which define the defect geometry on the graphene sheet are estimated along with the background conductivity. Monte

Carlo simulation is used for analyzing those statistical parameters which also verifies the robustness of the proposed algorithm PSO-GSA. The effect of different parameter configurations on PSO-GSA is analyzed and based on this initial conditions of PSO-GSA are configured. Nine different cases are considered in the numerical study to estimate the location of the defects and the background conductivity of the graphene sheet. For the experimental study, the graphene sheet is prepared with equally spaced copper electrodes. A graphene sheet with a single defect and two defects are studied. In both the experimental study PSO-GSA have estimated defect geometry with good accuracy. We have used PSO-GSA to tune the hyperparameter of the DNN model which is not learned during the model training. In this work, PSO-GSA is used for optimizing the node size of the hidden layer in the DNN model. This optimized DNN model is trained to estimate the interlayer boundaries of the subsurface domain which are defined by the discrete front points. The interlayer boundaries for subsurface estimation are done with PSO-GSA tuned DNN model, and the estimation result of the proposed model is compared against funnel-shaped DNN model. The training of both models is done with the same configuration and the same training dataset. The PSO-GSA-tuned DNN model estimation is better than that of the funnel-shaped DNN model. The presented work is focused on two-dimensional boundary estimation. The work on bladder and graphene is conducted in the control experimental environment which can be future expanded. The bladder boundary estimation in is future need to be studied in the living being. The estimation of defect boundary on graphene is studied on the single-layer graphene sample which can be future expanded to the multilayer graphene wafers. The subsurface interlayer boundary work is a numerical study that future work on the in-situ site. This boundary estimation using EIT can be future expanded to other tomography techniques.

Summary

Electrical impedance tomography (EIT) is a noninvasive imaging technique. It reconstructs the cross-sectional image of the internal conductivity distribution of an electrically conducting object. In EIT an array of electrodes are attached to the boundary of the domain under study. A constant current is applied from the external current source through the electrodes and the excited voltages are measured on the surface of the electrode. Using this current-voltage relationship the internal conductivity distribution is reconstructed. Reconstruction of internal conductivity distribution from EIT suffers from poor spatial resolution. This is due to an ill-posed and highly non-linear problem. If the internal conductivity values of the disjoint region are known prior then estimating the boundary of the anomaly inside the object becomes the problem. This approach is known as boundary estimation or shape estimation.

In this work, gravitational search algorithm (GSA) based optimization is used to estimate the boundary of the anomaly in the object. When the boundary of the anomaly is closed and smooth, it can be defined by the Fourier series coefficient. The boundary of the bladder is closed and smooth, thus defined by Fourier series coefficients. GSA estimated the bladder boundary with the good accuracy to the true shape and size.

However, GSA tends to stuck in the local minima as it has poor exploration ability. The exploration capability of GSA is increased after it was combined with particle swarm optimization (PSO) algorithm. This new hybrid algorithm is used for estimating the defect boundary on the graphene sheet. This algorithm also estimated the background conductivity of the graphene sheet along with Fourier series coefficients which defines the defect boundary. The PSOGSA is also used for tuning the hyper-parameter of the DNN model, where the node size of the hidden layers in the model used optimized. The tuned DNN model is used for the estimation of the interlayer boundary. The interlayer

boundary of the subsurface is assumed to be open boundary scenario and was defined by the Front point method. The PSOGSA tuned DNN model was trained to estimate those front points.

The present work is on the 2-D boundary estimation. Fourier series and Front point methods are used to define the close and open boundary of the anomaly, respectively. GSA based optimization algorithm are used for estimating the anomaly boundary in the domain. These work can be future expanded to the 3-D estimation problem and also towards other tomography techniques.

List of publications

S.K. Sharma, S.K. Konki, A.K. Khambampati, and K.Y. Kim. PSOGSA based detection of graphene defects with EIT, *IEEE Access*, 2022 Under review.

S.K. Sharma, A.K. Khambampati, and K.Y. Kim. A study on estimating the interlayer boundary of the subsurface using a artificial neural network with electrical impedance tomography. *Journal of IKEEE* 25 (4), 650-663, 2021.

S.K. Sharma, S.K. Konki, A.K. Khambampati, and K.Y. Kim, Bladder boundary estimation by gravitational search algorithm using electrical impedance tomography. *IEEE Transactions on Instrumentation and Measurement*, 69(12), pp.9657-9667. 2020.

S.K. Sharma, A. K. Khambampati, and K. Y. Kim. Estimating aquifer location using deep neural network with electrical impedance tomography. *Journal of IKEEE*, 24 (4):982–990, 2020.

A.K. Khambampati, **S.K. Sharma**, and K.Y. Kim. Estimation of electrode contact impedance and conductivity of single-layer graphene using electrical impedance tomography. *Measurement Science and Technology*, 2022.

S.K. Konki, A.K. Khambampati, **S.K. Sharma**, and K.Y. Kim. A deep neural network for estimating the bladder boundary using electrical impedance tomography. *Physiological Measurement* 41 (11), 115003, 2020.

A.K. Khambampati, S.A. Rahman, **S.K. Sharma**, W.Y. Kim, and K.Y. Kim. Imaging Conductivity Changes in Monolayer Graphene Using Electrical Impedance Tomography. *Micromachines* 11 (12), 1074, 2020.

A.K. Khambampati, S.K. Konki, Y.J. Han, **S.K. Sharma**, and K.Y. Kim.
An efficient method to determine the size of bladder using electrical impedance tomography. *TENCON 2018-2018 IEEE Region 10 Conference*, 1933-1936, 2018.

Bibliography

- M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.
- A. Adler and W. R. Lionheart. Uses and abuses of eidors: an extensible software base for eit. *Physiological measurement*, 27(5):S25, 2006.
- A. Adler, R. Amyot, R. Guardo, J. Bates, and Y. Berthiaume. Monitoring changes in lung air and liquid volumes with electrical impedance tomography. *Journal of Applied Physiology*, 83(5):1762–1767, 1997.
- D. A. Anjos, E. C. Etchebehere, C. D. Ramos, A. O. Santos, C. Albertotti, and E. E. Camargo. 18f-fdg pet/ct delayed images after diuretic for restaging invasive bladder cancer. *Journal of Nuclear Medicine*, 48(5):764–770, 2007.
- A. P. Bagshaw, A. D. Liston, R. H. Bayford, A. Tizzard, A. P. Gibson, A. T. Tidswell, M. K. Sparkes, H. Deghani, C. D. Binnie, and D. S. Holder. Electrical impedance tomography of human brain function using reconstruction algorithms based on the finite element method. *NeuroImage*, 20(2):752–764, 2003.
- L. E. Baker. Principles of the impedance technique. *IEEE engineering in medicine and biology magazine*, 8(1):11–15, 1989.
- D. Barber. Bh brown applied potential tomography j. *Phys. E*, 17:723–733, 1984.
- R. Barker and J. Moore. The application of time-lapse electrical tomography in groundwater studies. *The Leading Edge*, 17(10):1454–1458, 1998.

- R. H. Bayford. Bioimpedance tomography (electrical impedance tomography). *Annu. Rev. Biomed. Eng.*, 8:63–91, 2006.
- E. Beretta, S. Micheletti, S. Perotto, and M. Santacesaria. Reconstruction of a piecewise constant conductivity on a polygonal partition via shape optimization in eit. *Journal of Computational Physics*, 353:264–280, 2018.
- J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.
- J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.
- S. Bhanja and A. Das. Impact of data normalization on deep neural network for time series forecasting. *arXiv preprint arXiv:1812.05519*, 2018.
- K. I. Bolotin, K. J. Sikes, Z. Jiang, M. Klima, G. Fudenberg, J. Hone, P. Kim, and H. Stormer. Ultrahigh electron mobility in suspended graphene. *Solid state communications*, 146(9-10):351–355, 2008.
- B. Brandstatter. Jacobian calculation for electrical impedance tomography based on the reciprocity principle. *IEEE transactions on magnetics*, 39(3):1309–1312, 2003.
- B. H. Brown. Medical impedance tomography and process impedance tomography: a brief review. *Measurement Science and Technology*, 12(8):991, 2001.
- J. S. Bunch, A. M. Van Der Zande, S. S. Verbridge, I. W. Frank, D. M. Tanenbaum, J. M. Parpia, H. G. Craighead, and P. L. McEuen. Electromechanical resonators from graphene sheets. *Science*, 315(5811):490–493, 2007.
- H. Chen, J. Yao, L. Yang, K. Liu, B. Chen, J. Li, and M. Takei. Development of a portable electrical impedance tomography device for online thrombus detection in extracorporeal-circulation equipment. *IEEE Sensors Journal*, 21(3):3653–3659, 2020.
- M. Cheney, D. Isaacson, and J. C. Newell. Electrical impedance tomography. *SIAM review*, 41(1):85–101, 1999.

- K. Cheng, D. Isaacson, J. Newell, and D. Gisser. Error comparison for different current patterns in electrical impedance imaging. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 279–280. IEEE, 1988.
- K.-S. Cheng, D. Isaacson, J. Newell, and D. G. Gisser. Electrode models for electric current computed tomography. *IEEE Transactions on Biomedical Engineering*, 36(9):918–924, 1989.
- V. Cherepenin, A. Karpov, A. Korjenvsky, V. Kornienko, A. Mazaletskaya, D. Mazourov, and D. Meister. A 3d electrical impedance tomography (eit) system for breast cancer detection. *Physiological measurement*, 22(1):9, 2001.
- M. H. Choi, T.-J. Kao, D. Isaacson, G. J. Saulnier, and J. C. Newell. A reconstruction algorithm for breast cancer imaging with electrical impedance tomography in mammography geometry. *IEEE transactions on biomedical engineering*, 54(4):700–710, 2007.
- W. Daily, A. Ramirez, D. LaBrecque, and J. Nitao. Electrical resistivity tomography of vadose water movement. *Water Resources Research*, 28(5):1429–1442, 1992.
- G. D’Antona, A. Ferrero, M. Lazzaroni, R. Ottoboni, and E. Samarani. Active monitoring apparatus for underground pollutant detection based on electrical impedance tomography. In *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No. 00CH37276)*, volume 1, pages 577–579. IEEE, 2002.
- J. C. de Munck, T. J. Faes, and R. M. Heethaar. The boundary element method in the forward and inverse problem of electrical impedance tomography. *IEEE transactions on Biomedical Engineering*, 47(6):792–800, 2000.
- C. Di Francescomarino, M. Dumas, M. Federici, C. Ghidini, F. M. Maggi, W. Rizzi, and L. Simonetto. Genetic algorithms for hyperparameter optimization in predictive business process monitoring. *Information Systems*, 74:67–83, 2018.

- F. Dickin and M. Wang. Electrical resistance tomography for process applications. *Measurement Science and Technology*, 7(3):247, 1996.
- A. Dijkstra, B. Brown, A. Leathard, N. Harris, D. Barber, and D. Edbrooke. Review clinical applications of electrical impedance tomography. *Journal of medical engineering & technology*, 17(3):89–98, 1993.
- B. H. Doerstling. *A 3-D reconstruction algorithm for the linearized inverse boundary value problem for Maxwell's equations*. PhD thesis, Rensselaer Polytechnic Institute, 1995.
- M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.
- B. Eyuboglu, T. Pilkington, and P. Wolf. Estimation of tissue resistivities from multiple-electrode impedance measurements. *Physics in Medicine & Biology*, 39(1):1, 1994.
- J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1-3):187–204, 1986.
- I. Frerichs. Electrical impedance tomography (eit) in applications related to lung and ventilation: a review of experimental and clinical activities. *Physiological measurement*, 21(2):R1, 2000.
- I. Frerichs, G. Hahn, W. Golisch, M. Kurpitz, H. Burchardi, and G. Hellige. Monitoring perioperative changes in distribution of pulmonary ventilation by functional electrical impedance tomography. *Acta anaesthesiologica scandinavica*, 42(6):721–726, 1998.
- I. Frerichs, J. Hinz, P. Herrmann, G. Weisser, G. Hahn, M. Quintel, and G. Hellige. Regional lung perfusion as determined by electrical impedance tomography in comparison with electron beam ct imaging. *IEEE transactions on medical imaging*, 21(6):646–652, 2002.
- J.-N. Fuchs and M. O. Goerbig. Introduction to the physical properties of graphene. *Lecture notes*, 10:11–12, 2008.

- C. Gabriel, S. Gabriel, and y. E. Corthout. The dielectric properties of biological tissues: I. literature survey. *Physics in medicine & biology*, 41(11):2231, 1996a.
- S. Gabriel, R. Lau, and C. Gabriel. The dielectric properties of biological tissues: II. measurements in the frequency range 10 hz to 20 ghz. *Physics in medicine & biology*, 41(11):2251, 1996b.
- D. Gu, D. Liu, D. Smyl, J. Deng, and J. Du. Supershape recovery from electrical impedance tomography data. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.
- D. Halliday, R. Resnick, and J. Walker. *Fundamentals of physics*. John Wiley & Sons, 2013.
- R. J. Halter, A. Hartov, and K. D. Paulsen. A broadband high-frequency electrical impedance tomography system for breast imaging. *IEEE Transactions on biomedical engineering*, 55(2):650–659, 2008.
- D. K. Han and A. Prosperetti. A shape decomposition technique in electrical impedance tomography. *Journal of computational physics*, 155(1):75–95, 1999.
- M. Harvey. Let’s evolve a neural network with a genetic algorithm. *Coastline automation*, posted on April, 2017.
- D. S. Holder. Electrical impedance tomography (eit) of brain function. *Brain topography*, 5(2):87–93, 1992.
- A. Holzinger. Big data calls for machine learning. In R. Narayan, editor, *Encyclopedia of Biomedical Engineering*, pages 258–264. Elsevier, Oxford, 2019. ISBN 978-0-12-805144-3.
- D. B. Hoover, D. P. Klein, D. C. Campbell, and E. du Bray. Geophysical methods in exploration and mineral environmental investigations. *Preliminary compilation of descriptive geoenvironmental mineral deposit models: USGS Open-File Report*, 95 (831):19–27, 1995.

- H. Hu. Multi-slice helical ct: Scan and reconstruction. *Medical physics*, 26(1):5–18, 1999.
- U. Z. Ijaz, A. K. Khambampati, M. C. Kim, S. Kim, J. S. Lee, and K. Y. Kim. Particle swarm optimization technique for elliptic region boundary estimation in electrical impedance tomography. In *AIP Conference Proceedings*, volume 914, pages 896–901. American Institute of Physics, 2007.
- D. Isaacson, J. Mueller, J. Newell, and S. Siltanen. Imaging cardiac activity by the d-bar method for electrical impedance tomography. *Physiological Measurement*, 27(5):S43, 2006.
- B. Jayasena and S. Subbiah. A novel mechanical cleavage method for synthesizing few-layer graphenes. *Nanoscale research letters*, 6(1):1–7, 2011.
- M. Jin, H.-K. Jeong, T.-H. Kim, K. P. So, Y. Cui, W. J. Yu, E. J. Ra, and Y. H. Lee. Synthesis and systematic characterization of functionalized graphene sheets generated by thermal exfoliation at low temperature. *Journal of Physics D: Applied Physics*, 43(27):275402, 2010.
- O. C. Jones, J.-T. Lin, L. Ovacik, and H. Shu. Impedance imaging relative to gas-liquid systems. *Nuclear engineering and design*, 141(1-2):159–176, 1993.
- T. J. Katsube, R. A. Klassen, Y. Das, R. Ernst, T. Calvert, G. Cross, J. Hunter, M. Best, R. DiLabio, and S. Connell. Prediction and validation of soil electromagnetic characteristics for application in landmine detection. In *Detection and Remediation Technologies for Mines and Minelike Targets VIII*, volume 5089, pages 1219–1230. International Society for Optics and Photonics, 2003.
- T. J. Katsube, P. K. Keating, H. McNairn, Y. Das, R. DiLabio, V. Singhroy, S. Connell-Madore, M. E. Best, J. Hunter, R. Klassen, et al. Soil moisture and electrical conductivity prediction and their implication for landmine detection technologies. In *Detection and Remediation Technologies for Mines and Minelike Targets IX*, volume 5415, pages 691–704. International Society for Optics and Photonics, 2004.

- P. Kearey, M. Brooks, and I. Hill. *An introduction to geophysical exploration*, volume 4. John Wiley & Sons, 2002.
- J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.
- I. R. Kenyon. *General relativity*. 1990.
- A. Khambampati, S. Konki, Y. Han, S. Sharma, and K. Kim. An efficient method to determine the size of bladder using electrical impedance tomography. In *TENCON 2018-2018 IEEE Region 10 Conference*, pages 1933–1936. IEEE, 2018.
- A. K. Khambampati, B. A. Lee, K. Y. Kim, and S. Kim. An analytical boundary element integral approach to track the boundary of a moving cavity using electrical impedance tomography. *Measurement Science and Technology*, 23(3):035401, 2012.
- A. K. Khambampati, Y. J. Hong, K. Y. Kim, and S. Kim. A boundary element method to estimate the interfacial boundary of two immiscible stratified liquids using electrical resistance tomography. *Chemical Engineering Science*, 95:161–173, 2013.
- A. K. Khambampati, K. Y. Kim, Y.-G. Lee, and S. Kim. Boundary element method to estimate the time-varying interfacial boundary in horizontal immiscible liquids flow using electrical resistance tomography. *Applied Mathematical Modelling*, 40(2):1052–1068, 2016.
- V. S. Khoo, D. P. Dearnaley, D. J. Finnigan, A. Padhani, S. F. Tanner, and M. O. Leach. Magnetic resonance imaging (mri): considerations and applications in radiotherapy treatment planning. *Radiotherapy and Oncology*, 42(1):1–15, 1997.
- D. H. Kim, A. Abraham, and J. H. Cho. A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Information Sciences*, 177(18):3918–3937, 2007a.
- S. Kim, U. Z. Ijaz, A. K. Khambampati, K. Y. Kim, M. C. Kim, and S. I. Chung. Moving interfacial boundary estimation in stratified flow of two immiscible liquids

- using electrical resistance tomography. *Measurement Science and Technology*, 18(5):1257, 2007b.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- V. Kolehmainen, A. Voutilainen, and J. P. Kaipio. Estimation of non-stationary region boundaries in eit—state estimation approach. *Inverse Problems*, 17(6):1937, 2001.
- S. K. Konki, A. K. Khambampati, S. K. Sharma, and K. Y. Kim. A deep neural network for estimating the bladder boundary using electrical impedance tomography. *Physiological Measurement*, 41(11):115003, 2020.
- E. Koomen, S. Janssen, and B. J. Anderson. Use of ultrasound bladder monitoring in children after caudal anaesthesia. *Pediatric Anesthesia*, 12(8):738–741, 2002.
- S. P. Kumar, N. Sriraam, P. Benakop, and B. Jinaga. Reconstruction of brain electrical impedance tomography images using particle swarm optimization. In *2010 5th International Conference on Industrial and Information Systems*, pages 339–342. IEEE, 2010.
- A. Lazar. Heuristic knowledge discovery for archaeological data using genetic algorithms and rough sets. In *Heuristic and optimization for knowledge discovery*, pages 263–278. IGI Global, 2002.
- J. Lee Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.
- X. Li, Y. Zhu, W. Cai, M. Borysiak, B. Han, D. Chen, R. D. Piner, L. Colombo, and R. S. Ruoff. Transfer of large-area graphene films for high-performance transparent conductive electrodes. *Nano letters*, 9(12):4359–4363, 2009.
- X. Li, Z. Hu, and X. Huang. Combine relu with tanh. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 51–55, 2020. doi: 10.1109/ITNEC48623.2020.9084659.

- Y. Li, L. Rao, R. He, G. Xu, Q. Wu, W. Yan, G. Dong, and Q. Yang. A novel combination method of electrical impedance tomography inverse problem for brain imaging. *IEEE Transactions on Magnetism*, 41(5):1848–1851, 2005.
- X. Liang, B. A. Sperling, I. Calizo, G. Cheng, C. A. Hacker, Q. Zhang, Y. Obeng, K. Yan, H. Peng, Q. Li, et al. Toward clean and crackless transfer of graphene. *ACS nano*, 5(11):9144–9153, 2011.
- W. R. Lionheart. Eit reconstruction algorithms: pitfalls, challenges and recent developments. *Physiological measurement*, 25(1):125, 2004.
- D. Liu, A. K. Khambampati, S. Kim, and K. Y. Kim. Multi-phase flow monitoring with electrical impedance tomography using level set based method. *Nuclear Engineering and Design*, 289:108–116, 2015.
- D. Liu, A. K. Khambampati, and J. Du. A parametric level set method for electrical impedance tomography. *IEEE transactions on medical imaging*, 37(2):451–460, 2017.
- D. Liu, D. Gu, D. Smyl, J. Deng, and J. Du. B-spline-based sharp feature preserving shape reconstruction approach for electrical impedance tomography. *IEEE transactions on medical imaging*, 38(11):2533–2544, 2019.
- K. Liu, Y. Wu, S. Wang, H. Wang, H. Chen, B. Chen, and J. Yao. Artificial sensitive skin for robotics based on electrical impedance tomography. *Advanced Intelligent Systems*, 2(4):1900161, 2020.
- H. G. Madersbacher. Neurogenic bladder dysfunction. *Current opinion in urology*, 9(4):303–307, 1999.
- J. Maillol, M.-K. Seguin, O. Gupta, H. Akhauri, and N. Sen. Electrical resistivity tomography survey for delineating uncharted mine galleries in west bengal, india. *Geophysical prospecting*, 47(2):103–116, 1999.
- R. Mann, F. Dickin, M. Wang, T. Dyakowski, R. A. Williams, R. Edwards, A. Forrest, and P. Holden. Application of electrical resistance tomography to interrogate mixing processes at plant scale. *Chemical engineering science*, 52(13):2087–2097, 1997.

- R. Mansouri, F. Nasser, and M. Khorrami. Effective time variation of g in a model universe with variable space dimension. *Physics Letters A*, 259(3-4):194–200, 1999.
- J. McNeill. Electrical conductivity of soils and rocks. geonics limited. *Mississauga, Ontario, Technical Note TN-5*, 1980.
- L. N. Meads, L. R. Bentley, and C. A. Mendoza. Application of electrical resistivity imaging to the development of a geologic model for a proposed edmonton landfill site. *Canadian Geotechnical Journal*, 40(3):551–558, 2003.
- R. G. Mendoza and J. E. C. Lope. Reconstructing images in electrical impedance tomography using hybrid genetic algorithms. *Science Diliman*, 24(2), 2012.
- Z. Meng, Z. Huang, B. Wang, H. Ji, H. Li, and Y. Yan. Air–water two-phase flow measurement using a venturi meter and an electrical resistance tomography sensor. *Flow Measurement and Instrumentation*, 21(3):268–276, 2010.
- S. Mirjalili and S. Z. M. Hashim. A new hybrid psogsa algorithm for function optimization. In *2010 international conference on computer and information application*, pages 374–377. IEEE, 2010.
- M. Moskowitz, T. Ryan, K. Paulsen, and S. Mitchell. Clinical implementation of electrical impedance tomography with hyperthermia. *International journal of hyperthermia*, 11(2):141–149, 1995.
- J. L. Mueller, D. Isaacson, and J. C. Newell. Reconstruction of conductivity changes due to ventilation and perfusion from eit data collected on a rectangular electrode array. *Physiological measurement*, 22(1):97, 2001.
- K. Najarian and R. Splinter. *Biomedical signal and image processing*. CRC press, 2005.
- V. K. Ojha, A. Abraham, and V. Snášel. Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60:97–116, 2017.
- P. Ola, L. Päiväranta, and E. Somersalo. An inverse boundary value problem in electrodynamics. *Duke Mathematical Journal*, 70(3):617–653, 1993.

- K. Osterman, T. Kerner, D. Williams, A. Hartov, S. Poplack, and K. Paulsen. Multifrequency electrical impedance imaging: preliminary in vivo experience in breast. *Physiological measurement*, 21(1):99, 2000.
- H. Park, H. Lee, K. Park, S. Mo, and J. Kim. Deep neural network approach in electrical impedance tomography-based real-time soft tactile sensor. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7447–7452. IEEE, 2019.
- H. Park, K. Park, S. Mo, and J. Kim. Deep neural network based electrical impedance tomographic sensing methodology for large-area robotic tactile sensing. *IEEE Transactions on Robotics*, 2021.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- R. Philp and P. Crisp. Surface geochemical methods used for oil and gas prospecting—a review. *Journal of Geochemical Exploration*, 17(1):1–34, 1982.
- P. Pinheiro, W. Loh, and F. Dickin. Smoothness-constrained inversion for two-dimensional electrical resistance tomography. *Measurement Science and Technology*, 8(3):293, 1997.
- A. Plaskowski, M. Beck, R. Thorn, and T. Dyakowski. *Imaging industrial flows: applications of electrical process tomography*. CRC Press, 1995.
- N.-W. Pu, C.-A. Wang, Y. Sung, Y.-M. Liu, and M.-D. Ger. Production of few-layer graphene by supercritical co₂ exfoliation of graphite. *Materials Letters*, 63(23): 1987–1989, 2009.
- E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.
- A. Reina, X. Jia, J. Ho, D. Nezich, H. Son, V. Bulovic, M. S. Dresselhaus, and

- J. Kong. Large area, few-layer graphene films on arbitrary substrates by chemical vapor deposition. *Nano letters*, 9(1):30–35, 2009.
- L. Rere, M. I. Fanany, and A. M. Arymurthy. Metaheuristic algorithms for convolution neural network. *Computational intelligence and neuroscience*, 2016, 2016.
- S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. 2002.
- M. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, 1968.
- F. Schwierz. Industry-compatible graphene transistors. *Nature*, 472(7341):41–42, 2011.
- C. Seif, B. Herberger, E. Cherwon, F. Martinez Portillo, M. Molitor, T. Stieglitz, G. Böehler, S. Zandler, K. Jünemann, and P. Braun. Urinary bladder volumetry by means of a single retrosymphysically implantable ultrasound unit. *Neurourology and Urodynamics: Official Journal of the International Continence Society*, 23(7):680–684, 2004.
- S. K. Sharma, A. K. Khambampati, and K. Y. Kim. Estimating aquifer location using deep neural network with electrical impedance tomography. *Journal of IKEEE*, 24(4):982–990, 2020.
- Y. Shi and R. C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1945–1950. IEEE, 1999.
- Y. Shi et al. Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, volume 1, pages 81–86. IEEE, 2001.
- D. Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- M. Soleimani, W. Lionheart, and O. Dorn. Level set reconstruction of conductivity and permittivity from boundary electrical measurements using experimental data. *Inverse problems in science and engineering*, 14(2):193–210, 2006.

- E. Somersalo, M. Cheney, and D. Isaacson. Existence and uniqueness for electrode models for electric current computed tomography. *SIAM Journal on Applied Mathematics*, 52(4):1023–1040, 1992.
- B. Spies and R. G. Ellis. Cross-borehole resistivity tomography of a pilot-scale, in-situ vitrification test. *Geophysics*, 60(3):886–898, 1995.
- R. Stacey, K. Li, and R. N. Horne. Electrical impedance tomography (eit) technique for real-time saturation monitoring. In *SPE Annual Technical Conference and Exhibition*. OnePetro, 2006.
- D. Stamoulis, E. Cai, D.-C. Juan, and D. Marculescu. Hyperpower: Power-and memory-constrained hyper-parameter optimization for neural networks. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 19–24. IEEE, 2018.
- K. Sudha, M. Israil, S. Mittal, and J. Rai. Soil characterization using electrical resistivity tomography and geotechnical investigations. *Journal of Applied Geophysics*, 67(1):74–79, 2009.
- E.-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of heuristics*, 8(5):541–564, 2002.
- K.-S. Tang, K.-F. Man, S. Kwong, and Q. He. Genetic algorithms and their applications. *IEEE signal processing magazine*, 13(6):22–37, 1996.
- P. Tarasewich and P. R. McMullen. Swarm intelligence: power in numbers. *Communications of the ACM*, 45(8):62–67, 2002.
- O.-P. Tossavainen, M. Vauhkonen, V. Kolehmainen, and K. Y. Kim. Tracking of moving interfaces in sedimentation processes using electrical impedance tomography. *Chemical Engineering Science*, 61(23):7717–7729, 2006.
- P. J. Van Laarhoven and E. H. Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

- B. Vasić, A. Zurutuza, and R. Gajić. Spatial variation of wear and electrical properties across wrinkles in chemical vapour deposition graphene. *Carbon*, 102:304–310, 2016.
- M. Vauhkonen, P. Karjalainen, and J. Kaipio. A kalman filter approach applied to the tracking of fast movements of organ boundaries. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)*, volume 2, pages 1048–1051. IEEE, 1998.
- J. G. Webster. *Electrical impedance tomography*. Taylor & Francis Group, 1990.
- A. S. Wicaksono and A. A. Supianto. Hyper parameter optimization using genetic algorithm on machine learning methods for online news popularity prediction. *International Journal of Advanced Computer Science and Applications*, 9(12):263–267, 2018.
- A. Wirgin. The inverse crime. *arXiv preprint math-ph/0401050*, 2004.
- Y. Xia, C. Liu, Y. Li, and N. Liu. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241, 2017.
- X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan. Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. *arXiv preprint arXiv:2006.12703*, 2020.
- G. Xu, H. Wu, S. Yang, S. Liu, Y. Li, Q. Yang, W. Yan, and M. Wang. 3-d electrical impedance tomography forward problem with finite element method. *IEEE transactions on magnetics*, 41(5):1832–1835, 2005.
- Z. Xu, J. Yao, Z. Wang, Y. Liu, H. Wang, B. Chen, and H. Wu. Development of a portable electrical impedance tomography system for biomedical applications. *IEEE Sensors Journal*, 18(19):8117–8124, 2018.
- G. A. Ybarra, Q. H. Liu, G. Ye, K. H. Lim, J.-H. Lee, W. T. Joines, and R. T. George. Breast imaging using electrical impedance tomography (eit). *Emerging Technologies in Breast Imaging and Mammography*, 2007.

- W. Zhu, T. Low, V. Perebeinos, A. A. Bol, Y. Zhu, H. Yan, J. Tersoff, and P. Avouris.
Structure and electronic transport in graphene wrinkles. *Nano letters*, 12(7):
3431–3436, 2012.
- O. C. Zienkiewicz and R. L. Taylor. *Finite Element Method: Vol. 3: Fluid Dynamics*.
Elsevier Science & Technology Books, 2000.