Parallel Computational Approach to Gradient Based EM Optimization of Microwave Structures

by

Gongal Reddy Venu Madhav Reddy, M.Tech.

A book based on the thesis submitted to
the Faculty of Graduate and Postdoctoral Affairs
in fulfillment of the degree requirements of
Doctor of Philosophy

Ottawa-Carleton Institute for
Electrical and Computer Engineering

Department of Electronics
Carleton University
Ottawa, Ontario, Canada, 2016

© 2016

Gongal Reddy Venu Madhav Reddy

Abstract

Electromagnetic (EM) based optimization and design closure is an essential part of RF and microwave design cycle. The main objective of this thesis is to develop a new gradient based EM optimization technique which exploits parallel computations. The proposed EM optimization technique achieves speed up in the optimization process even if a coarse model is not available. Speed up in optimization has been achieved by using large and effective optimization updates in each iteration which resulted in fewer optimization iterations. The first contribution of the thesis is the development of a trust-region based optimization technique which uses parallel computational approach. In the proposed method, we deliberately increase the number of fine model evaluations without increasing the computation time by using a parallel computational approach. A large number of fine model evaluations allows us to build a surrogate model valid in a large neighborhood. These valid surrogate models are used to achieve large and effective optimization updates. The second major contribution of the thesis is the EM optimization decomposition approach to address the challenges in EM optimization with many design variables. This proposed method decomposes a single large EM optimization problem into multiple smaller sub-optimizations to improve the optimization efficiency. Multiple sub-optimizations are formulated to be independent so that the parallel computations can be exploited to evaluate concurrent sub-optimization updates. The resultant vector, a combined vector of sub-optimization updates, will be closer to the optimal solution. Furthermore, the optimization update of the proposed approach is much larger in comparison with the optimization update without decomposition.

To my parents

Ranga Laxmi and Subba Reddy

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Prof. Q. J. Zhang for the constant support and expert guidance during the course of this research work. His encouragement, motivation and thoughtful insights during my research made my journey in realizing my goal a memorable one. It was my honor to work under his supervision and guidance.

Specially, I'd like to gratefully acknowledge the members of my Ph.D. committee; Dr. Vicente E. Boria, Dr. Mustafa C.E. Yagoub, Dr. Pavan Gunupudi and Dr. Rony E. Amaya for their time, invaluable suggestions and corrections. I would also like to thank all present and former colleagues especially Dr. Yazhi Cao, Dr. Seyed Alireza Sadrossadat, S. Zhang, C. Zhang, F. Feng and W. Na in our research group for their enthusiasm and helpful discussions.

Finally, I wish to thank my parents, my brother, my sister and my sister-in-law for their love, belief, support and encouragement throughout my study. A special thanks to my friend Dr. Bhagya Lakshmi for her constant encouragement and motivation during my tough times.

Table of Contents

Abstract	iii
Acknowledgements	v
List of Figures	ix
List of Tables	xvii
1 Introduction	1
1.1 Introduction to EM Optimization	1
1.2 List of Contributions	3
1.2.1 Parallel Computation Approach to Gradient based EM optimization of Passive Microwave Circuits [18]	3
1.2.2 Parallel Decomposition Approach to Gradient based EM Optimization with Many Design Variables [19]	4
1.3 Thesis Organization	5
2 Literature Review	6

2.1	Optimization problem	7
2.2	Direct EM Optimization methods	7
2.2.1	Derivative-free Pattern Search Method	8
2.2.2	Gradient based Quasi-Newton Method	9
2.3	Space Mapping Optimization	10
2.3.1	The Space Mapping Concept	11
2.3.2	Input Space Mapping	14
2.3.3	Implicit Space Mapping	15
2.3.4	Output Space Mapping	16
2.3.5	Tuning Space Mapping	16
2.3.6	Coarse and Fine mesh Space mapping	17
2.4	Artificial Neural Networks (ANN)	18
2.4.1	Prior Knowledge Input	19
2.4.2	Knowledge Based Neural Networks	20
2.4.3	Neuro Space Mapping	21
2.5	Overview of Parallel Computing Architectures	21
2.6	Summary	25

3 Parallel Computational Approach to Gradient based EM Optimization of Passive Microwave Circuits **26**

3.1	Introduction	27
3.2	Proposed Parallel Optimization using Trust Region Framework . . .	29
3.2.1	Formulation of the Original Optimization Problem	29

3.2.2	Introduction to the Proposed Optimization Technique	30
3.2.3	Distribution of EM Samples in the n-dimensional Space . . .	31
3.2.4	Parallel Computation over Multiple EM Samples	34
3.2.5	Surrogate Modeling using Multiple Points across the Trust Region X_R	36
3.2.6	Surrogate Model Optimization and Trust Radius Update . .	41
3.2.7	Discussion	45
3.3	Application Examples	47
3.3.1	Optimization of Inter-digital Band-pass Filter	47
3.3.2	Optimization of Four-Pole Waveguide Filter	57
3.3.3	Optimization of a Dielectric Resonator Antenna	68
3.4	Conclusion	74
4	Parallel Decomposition Approach for Gradient based EM Opti- mization	75
4.1	Introduction	76
4.2	Formulation of the Optimization Problem	79
4.2.1	Original Optimization Problem	79
4.2.2	Sub-optimization Problem	80
4.3	Proposed Decomposition Approach to EM Optimization	82
4.3.1	Ideal Case of Decomposition Approach	83
4.3.2	Proposed Decomposition Technique using Second Order Deriva- tive Information	85

4.3.3	Identification of the Strongly Interacting Design Variables and Formation of Sub Design Vectors	87
4.4	Proposed Parallel EM Optimization using Sub Design Vectors . . .	94
4.4.1	Trust Radius for the i th Sub-optimization Problem	94
4.4.2	Surrogate Modeling over Multiple Samples in the i th Sub-Optimization	96
4.4.3	Optimization of the i th Sub-Problem and Trust Region Framework	97
4.4.4	Parallel Computation over Multiple Sub-Problems	98
4.4.5	Crossover and Overall Optimization Update using Multiple Sub-optimization Solutions	99
4.4.6	Discussion	111
4.5	Application Examples	111
4.5.1	Optimization of Four-Pole Waveguide Filter	111
4.5.2	Optimization of Iris Coupled Microwave Cavity Filter	124
4.6	Conclusion	137
5	Conclusions and Future Research	138
5.1	Conclusions	138
5.2	Future Research	140
	Bibliography	142

List of Figures

2.1	Space mapping concept following [31]	12
2.2	Mathematical representation of space mapping following [22]	13
2.3	Illustration of Implicit Space Mapping (ISM) following [24]	15
2.4	Prior Knowledge Input (PKI) neuromodeling concept following [45]	19
2.5	Knowledge Based Neural Networks (KBNN) neuromodeling concept following [46]	20
2.6	Space mapped neuromodeling concept following [47]	21
2.7	Structure of Hybrid Distributed-Shared Memory Architecture(HDSMA) following [50]	24
3.1	Illustration of orthogonal arrays (a specific type of DOE) sampling technique to generate multiple sample points ($\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{N_s}$) around the central point $\mathbf{x}^c$ in the trust region X_R . The central point $\mathbf{x}^c$ is updated after each iteration of the proposed optimization method. The central point $\mathbf{x}^c$ and the DOE samples around the central point move as the proposed optimization progresses from the current iteration to the next iteration	33

3.2	Flow chart showing the proposed optimization process.	44
3.3	Structure of a inter-digital band-pass filter with optimization variables $\mathbf{x} = [g \ s_1 \ s_2 \ s_3]^T$	47
3.4	Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 302 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after three iterations) (d) Fine model response using proposed optimization (after six iterations).	51
3.5	The values of objective function for the inter-digital band-pass filter using: the proposed method (o) and coarse and fine mesh SM method (x).	53
3.6	Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 227 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after three iterations) (d) Fine model response using proposed optimization (after seven iterations).	55
3.7	The values of objective function for the inter-digital band-pass filter using: the proposed method (o) and coarse and fine mesh SM method (x).	57
3.8	Structure of a four-pole waveguide filter with optimization variables $\mathbf{x} = [h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2}]^T$	58

3.9	Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 144 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after four iterations) (d) Fine model response using proposed optimization (after four iterations).	60
3.10	The values of objective function for the four-pole waveguide filter using: the proposed method (o) and coarse and fine mesh SM method (x).	62
3.11	Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 812 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after two iterations) (d) Fine model response using proposed optimization (after eight iterations).	64
3.12	The values of objective function for the four-pole waveguide filter using: the proposed method (o) and coarse and fine mesh SM method (x).	66

3.13	The value of objective function evaluated at each $\mathbf{x}^{sample}$ by changing τ . The value of objective function increases as τ increases in a small neighborhood ($\tau \leq 0.1$) which confirms $\mathbf{x}^{em}$ ($\tau = 0$) is a local minima. With large step size of the optimization update ($\tau = 1$), our proposed optimization can jump over the trap ($\tau = 0$) and find the new solution $\mathbf{x}^{new}$, where the value of the objective function is much lower than that at the local minima obtained using classical quasi-Newton method.	68
3.14	Structure of a dielectric loaded monopole antenna with optimization variables $\mathbf{x} = [h_1 \ h_2 \ r_1 \ r_2 \ r_a]^T$	69
3.15	Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 62 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after 1 iteration) (d) Fine model response using proposed optimization (after 6 iterations).	71
3.16	The values of objective function for the dielectric loaded monopole antenna using: the proposed method (o).	72
4.1	Graphical representation of the proposed EM optimization decomposition approach using two design variables with eccentricity (measure of interaction between design variables) equal to zero, i.e., $B_4 = 0$ in (4.5). In this case, the two variables x_1 and x_2 can be optimized independently.	83

4.2	Flow chart showing the proposed decomposition process. Through this process, the entire design vector $\mathbf{x}$ is sub-divided into multiple sub design vectors, $\tilde{\mathbf{x}}^{(i)} \forall i = 1, 2, \dots, L$.	93
4.3	Crossover mechanism to estimate the next overall optimization update for three groups. (a) Central point (big solid circle), base vector (small solid circle), and crossover vector (hashed circle). Crossover approach showing the formation of crossover vectors (hashed circles) $(\mathbf{u}_1, \mathbf{v}_1, \mathbf{w}_0)$, $(\mathbf{u}_1, \mathbf{v}_0, \mathbf{w}_1)$, $(\mathbf{u}_0, \mathbf{v}_1, \mathbf{w}_1)$, and $(\mathbf{u}_1, \mathbf{v}_1, \mathbf{w}_1)$ using the base vectors (small solid circles) $(\mathbf{u}_1, \mathbf{v}_0, \mathbf{w}_0)$, $(\mathbf{u}_0, \mathbf{v}_1, \mathbf{w}_0)$ and $(\mathbf{u}_0, \mathbf{v}_0, \mathbf{w}_1)$. (b) Resultant prospective design vectors for the next optimization update.	104
4.4	Flow chart showing the proposed EM optimization decomposition process.	108
4.5	Structure of a four-pole waveguide filter with optimization variables $\mathbf{x} = [d_{12} \ d_{23} \ h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2} \ w_p \ w_c]^T$	112

4.6	Comparison of fine model responses and the objective function values of the prospective updates in the first iteration of the proposed optimization method for a four-pole waveguide filter. (a) Fine model response and U of the base vector $\mathbf{z}_{10}$ evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$. (b) Fine model response and U of the base vector $\mathbf{z}_{01}$ evaluated using $\tilde{\mathbf{x}}^{\mu,(2)}$. (c) Fine model response and U of the crossover vector $\mathbf{z}_{11}$ evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$ and $\tilde{\mathbf{x}}^{\mu,(2)}$. As seen in the figure, $\mathbf{z}_{11}$ has the least value of objective function of 123.33 and is selected as the best solution $\mathbf{z}^{new}$ among $\mathbf{Z} = \{\mathbf{z}_{10}, \mathbf{z}_{01}, \mathbf{z}_{11}\}$ for the first iteration.	118
4.7	Comparison of four different optimization methods for a four-pole waveguide filter. (a) Fine model response at the starting point for all the four optimization methods. (b) Fine model response using direct EM optimization (after 763 iterations). (c) Fine model response using coarse and fine mesh SM optimization (after 2 iterations). (d) Fine model response using optimization method without decomposition method [18] (after 14 iterations). (e) Fine model response using proposed optimization with decomposition method (after 6 iterations).	120
4.8	(a) The values of objective function for the four-pole waveguide filter using: the proposed method with decomposition (o) and coarse and fine mesh SM method (x). (b) The values of objective function for the four-pole waveguide filter using: optimization method with decomposition (o) and optimization method without decomposition [18] (x).	123

4.11	Comparison of four different optimization methods for an iris coupled cavity filter. (a) Fine model response at the starting point for all the four optimization methods. (b) Fine model response using direct EM optimization (after 638 iterations). (c) Fine model response using coarse and fine mesh SM optimization (after 5 iterations). (d) Fine model response using optimization method without decomposition method [18] (after 13 iterations). (e) Fine model response using proposed optimization with decomposition method (after 6 iterations).	133
4.12	(a) The values of objective function for the iris coupled microwave cavity filter using: the proposed method with decomposition (o) and coarse and fine mesh SM method (x). (b) The values of objective function for the iris coupled microwave cavity filter using: optimization method with decomposition (o) and optimization method without decomposition [18] (x).	136

List of Tables

3.1	Comparison of CPU Time for Inter-digital Band-pass Filter with four design variables	52
3.2	Comparison of CPU Time for Inter-digital Band-pass Filter with three design variables	56
3.3	Comparison of CPU Time for a Four-pole Waveguide Filter using a Good Starting Point	61
3.4	Comparison of CPU Time for Four-pole Waveguide Filter using a Bad Starting Point	65
3.5	Confirming the final solution of the direct EM optimization is a local minima	67
3.6	Comparison of CPU Time for Dielectric Resonator Antenna	73
4.1	Representation of the Design Vectors $\mathbf{z}$ for the Three Groups in Each Iteration of the Overall Optimization	103
4.2	Comparison of the Validity Range (Expressed as Trust Radius) of the Surrogate Model with Decomposition and without Decomposition for a Four-pole Waveguide Filter	116

4.3	Comparison of Fine Model Evaluations Using the Optimization Method With and Without Decomposition for a Four-pole Waveguide Filter	119
4.4	Comparison of CPU Time for Optimization of a Four-pole Waveguide Filter	121
4.5	Comparison of the Validity Range (Expressed as Trust Radius) of the Surrogate Model with Decomposition and without Decomposition for an Iris Coupled Cavity Filter	128
4.6	Comparison of Fine Model Evaluations Using the Optimization Method With and Without Decomposition for an Iris Coupled Cavity Filter	132
4.7	Comparison of CPU Time for Optimization of an Iris Coupled Cavity Filter	134

List of Symbols

$\mathbf{A}$	Matrix representing a simple quadratic polynomial function to relate the coefficients γ_r (numerator or denominator) and the design vector $\mathbf{x}$ over multiple EM samples in the surrogate model range
$\mathbf{B}$	Input transformation matrix of size $\Re^{n \times n}$ that transforms $\mathbf{x}_f$ to $\mathbf{x}_c$ using space mapping techniques
$\mathbf{C}$	$n \times n$ matrix representing the summation of second order derivatives at all frequency points N_f for n design variables
C_{kl}	Measure of interaction between the k th design variable and the l th design variable
$\mathbf{c}$	Input shifting vector of size $\Re^{n \times 1}$ that transforms $\mathbf{x}_f$ to $\mathbf{x}_c$ using space mapping techniques
d_i	Number of design variables in the sub design vector of the i th group
d_{min}	Minimum number of design variables per group
d_{max}	Maximum number of design variables per group

e_{uq}	Error function for upper specification
e_{lq}	Error function for lower specification
g_r	Function representing the relation between γ_r and design vector $\mathbf{x}$
h	Finite step for perturbation for calculating sensitivity
J	Set of indices of the design variables in vector $\mathbf{x}$
J_i	Set of indices corresponding to the design variables of the i th group or the i th sub-optimization
J_s	Subset of set J
L	Number of groups/sub-optimization problems
L_{ini}	Initial number of groups
N_f	Number of frequency samples
N_s	Number of EM samples selected using design of experiments (DOE) for constructing the surrogate model in the trust region
$N_s^{(i)}$	Number of EM samples selected using DOE for constructing the i th surrogate model in the trust region during the i th sub-optimization
n	Number of design variables

n_{rem}	Number of remaining design variables to be grouped calculated after the formation of the i th group
P	Number of processors to evaluate the fine model responses in parallel
$\mathbf{P}_{SM}$	Mapping function that relates fine and coarse model design parameters
$\mathbf{Q}_{SM}$	Implicit space mapping function that relates fine, coarse model, and preassigned parameter parameters
Q_u	Index set for upper specification
Q_l	Index set for lower specification
$\mathbf{R}_c(\mathbf{x}_c)$	Vector of coarse model responses
$\mathbf{R}_f(\mathbf{x}_f)$	Vector of fine model responses
$R_f(\mathbf{x}, \omega)$	Fine model output response
$\tilde{R}_f(\tilde{\mathbf{x}}, \omega)$	Fine model response for the sub-optimization problem
$R_s(\mathbf{x}, \omega)$	Surrogate model output response
$\tilde{R}_s^{(i)}(\tilde{\mathbf{x}}^{(i)}, \omega)$	Surrogate model output response of the i th sub-optimization
r_a	Control index parameter that determines the expansion or contraction of the trust radius for the trust region X_R

S_p	Speed up ratio between optimization time of a sequential optimization process to that of the parallel optimization process using P processors.
S_{lq}	Set of lower specification
S_{uq}	Set of upper specification
$S_{q,a}$	Partial derivative of the fine model response $R_f(\mathbf{x}, \omega)$ with respect to the design variable x_a at the q th frequency sample
T_1	CPU time for sequential evaluation of the algorithm
T_2	CPU time for parallel evaluation of the algorithm
T_s	CPU time for estimating the sensitivities to determine the interval range of each design variable for the first iteration of the optimization process
T_p	CPU time per iteration to build a surrogate model and minimize the objective function U
T_m	Fine model evaluation time for an EM sample m in the optimization process
T_{ov}	Overhead CPU time (e.g., communication time between multiple processors) of each iteration for generating many fine model responses in parallel using P processors

U	Objective function represented as either a minimax or a generalized l_p function of e_{uq} or e_{lq}
$\mathbf{u}$	Sub design vector for group J_1 when the overall optimization problem is subdivided into three sub-optimization problems
$\mathbf{u}_0$	Central point for sub-optimization 1
$\mathbf{v}$	Sub design vector for group J_2 when the overall optimization problem is subdivided into three sub-optimization problems
$\mathbf{v}_0$	Central point for sub-optimization 2
W_u	Weighting factor for error function e_{uq}
W_l	Weighting factor for error function e_{lq}
$\mathbf{w}$	Sub design vector for group J_3 when the overall optimization problem is subdivided into three sub-optimization problems
$\mathbf{w}_0$	Central point for sub-optimization 3
X	Fine model design space
X_c	Coarse model design space
X_R	Trust region (bounded fine model design space) in each iteration of the overall optimization
$\tilde{X}$	Subspace of X and represents the search space for the sub-optimization problem

$\tilde{X}^{(i)}$	Trust region for the i th sub-optimization problem
$\boldsymbol{x}$	Vector of design variables
$\boldsymbol{x}_{aux}$	Preassigned parameters (auxiliary parameters)
$\boldsymbol{x}_f$	Fine model design parameters
$\boldsymbol{x}_c$	Coarse model design parameters
$\boldsymbol{x}_f^*$	Optimal fine model design parameters satisfying the design specifications
$\boldsymbol{x}_c^*$	Optimal solution after coarse model optimization
$\boldsymbol{x}^*$	Optimal design solution
$\boldsymbol{x}^c$	Central point in each iteration of the overall optimization
$\boldsymbol{x}^{em}$	Optimal point obtained using direct EM optimization
$\boldsymbol{x}^m$	Denotes an EM sample in the trust region X_R
$\boldsymbol{x}^{new}$	Optimization update (surrogate model optimal point)
$\tilde{\boldsymbol{x}}$	Sub design vector of $\boldsymbol{x}$
$\tilde{\boldsymbol{x}}^{(i)}$	Sub design vector for the i th group or the i th sub-optimization
$\tilde{\boldsymbol{x}}^{c,(i)}$	Central point for the i th sub-optimization problem

$\tilde{\mathbf{x}}^{m,(i)}$	Denotes an EM samples in the trust region $\tilde{X}^{(i)}$ for the i th sub-optimization
$\tilde{\mathbf{x}}^{\mu,(i)}$	Sub-optimization solution of the i th sub-optimization problem
$\mathbf{x}^+$	Resultant design vector from the perturbation of the design variable x_b by a finite step h in positive direction
$\mathbf{x}^-$	Resultant design vector from the perturbation of the design variable x_b by a finite step h in negative direction
$\mathbf{Z}$	Set of all possible prospective updates for the overall optimization problem
$\mathbf{z}$	Prospective solutions of $\mathbf{x}$ (intermediate solutions of $\mathbf{x}$)
$\mathbf{z}_{000}$	Central point of the overall optimization problem in each iteration
$\mathbf{z}^{new}$	Best solution among the set $\mathbf{Z}$
α	Vector of numerator coefficients for the Bi-linear transfer function in z -domain
β	Vector of denominator coefficients for the Bi-linear transfer function in z -domain
$\hat{\alpha}$	Numerator coefficient of the transfer function in rational functional format (s -domain)

$\hat{\beta}$	Denominator coefficient of the transfer function in rational functional format (s -domain)
γ_r	Vector representing the r th coefficient which may be either numerator or denominator coefficient
$\Delta \mathbf{R}$	Correction vector which is difference between fine model and the original space mapping at the current iteration
Δ^{max}	Maximum trust radius
$\Delta^{max,(i)}$	Maximum trust radius for the i th sub-optimization problem
δ	Trust radius in each iteration of the overall optimization
δ^{new}	Trust radius update
$\delta^{(i)}$	Trust radius for the i th sub-optimization problem
$\delta^{\mu,(i)}$	Prospective trust radius for the next iteration
$\delta^{new,(i)}$	Trust radius update of the i th sub-optimization problem for the next iteration
ϵ	User defined threshold (in the order of 10^{-4}) between two successive iterations of the optimization process
η	Parallel efficiency
λ_r	Vector of weighting vector (unknown factors) in the function g_r

λ^*	Optimal weighting vectors wherein the error between the fine model reponse and the surrogate model response is minimized
$\phi^{(i)}$	Vector, whose elements are the maximum value of absolute sensitivity of $\tilde{R}_f(\tilde{\mathbf{x}}, \omega)$ with respect to $x_j; j \in J_i$ evaluated at the initial central point $\tilde{\mathbf{x}}^{c,(i)}$, and the maximum value of absolute sensitivity is selected among that of all frequencies
$\psi^{(i)}$	User defined value for calculating the trust radius in the first iteration of the i th sub-optimization
Ω	Complete set of frequencies
ω	Frequency
ω_q	q th frequency sample
$\vee$	Logical OR operator
$\wedge$	Logical AND operator
$./$	Element-wise division of a vector

Chapter 1

Introduction

1.1 Introduction to EM Optimization

With the dramatic increase in the computer hardware performance, EM simulators are used to solve Maxwell equations for circuits and arbitrary geometrical shapes. Today's radio frequency (RF)/microwave design faces the challenges of increasing complexity, tighter tolerances, stringent design specifications and shorter design cycles. As a result, the demand for faster, more accurate and cost-effective computer aided design (CAD) techniques in the RF/microwave becomes more predominant. Advances in the technology of workstations and PCs over the last decade enable traditional EM optimization of some EM structures. However, the increasing complexity and the emphasis of the wireless industry on low cost and shorter time-to-market are creating increasing demands on computer-aided design (CAD) tools for RF/microwave circuits.

There are three important reasons to simulate RF and microwave circuits and systems: to understand the physics of a complex system of interacting elements; to

test new concepts; and to optimize designs. Further, as the frequency of RF circuits increases from gigahertz to tens and hundreds of gigahertz, wavelengths becomes smaller. When the size of the device is comparable to the operating wavelength, the three-dimensional EM simulations becomes important for realizing the characteristics of the device. Electromagnetic (EM) simulation methods for high frequency structures developed recently brought the CAD for RF/microwave circuits to its current state of the art.

Steer et al [1] documented several fundamental issues for an EM-Oriented design. In the past decade, researches have reported several papers addressing these issues. Significant progress has been achieved in several areas such as EM simulation challenges with large mesh and complex geometries [2]-[4], [80]-[83], yield-driven design [5]-[8], in the circuit simulation area for capturing the dynamic behavior of nonlinear components and systems [9]-[15], space mapping [22]-[31], and adjoint sensitivity [15]-[17], [67], [90].

In this thesis, a trust-region based EM optimization technique using parallel computational approach [18] is developed. In the first part of the thesis, a robust optimization algorithm using integrated trust region framework is discussed. The proposed optimization addresses the situation when the coarse model is not available. The gradient based optimization method in commercial EM simulators such as HFSS, CST and ADS is sequential and evaluate the next optimization update using single-point EM evaluation per iteration. Parallel approach to gradient based EM optimization is explored such that surrogate model optimization will result in large and effective optimization update. Thereby, achieving speed up in the optimization

process. Several examples have been presented to verify the proposed optimization technique.

As a further advancement, EM optimization decomposition approach [19] is developed for the first time to address the scalability to EM optimization with many design variables. Single large EM optimization is decomposed into multiple smaller sub-optimizations to improve the optimization efficiency. Each sub-optimization includes generating multiple fine model responses for constructing the surrogate and its optimization using trust region. Multiple sub-optimizations are formulated to be independent so that the parallel computations can be exploited to evaluate concurrent sub-optimization updates. The resultant vector, a combined vector of sub-optimization updates will be closer to optimal solution. Several examples have been presented to verify the proposed optimization technique.

1.2 List of Contributions

The main objective of this thesis is to investigate and develop an efficient, scalable, and systematic EM optimization method which exploits parallel computations.

1.2.1 Parallel Computation Approach to Gradient based EM optimization of Passive Microwave Circuits [18]

In the first part of the thesis, the following significant contributions were made:

- A large number of fine model evaluations are used to achieve a speed up in the overall optimization by exploiting parallel computations.

- The speed up achieved due to the proposed parallel optimization is an additional speed up besides the speed up achieved due to parallel computations in existing EM simulators.
- Formulated a robust algorithm by the use of multiple EM samples and the use of trust region methods which guarantees convergence.
- Formulated a surrogate model using transfer functions and regression techniques which can effectively handle large region than that of the approximation functions used in classical quasi-Newton method and has a better chance of avoiding local minima traps.

1.2.2 Parallel Decomposition Approach to Gradient based EM Optimization with Many Design Variables [19]

The following contributions were made in the development of the new parallel decomposition method to EM optimization with many design variables:

- Formulated a new decomposition method to address the challenges of EM optimization with many design variables.
- Derived a method to measure the level of dependence between the design variables by using second order derivative information.
- Multiple sub-optimizations are formulated to be independent so that the parallel computations can be exploited to evaluate sub-optimization updates.

1.3 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 presents a literature review of popularly used EM optimization techniques for microwave design. An overview of classical space mapping methods such as implicit space mapping, output space mapping, tuning space mapping and the recently developed coarse and fine mesh space mapping is reviewed. Other known EM optimization techniques such as prior knowledge input, knowledge based neural networks and neurospace mapping methods are also discussed.

Chapter 3 presents an integrated trust region framework to gradient based EM optimization of microwave circuits using parallel computational approach. The proposed technique aims to increase the speed of gradient based EM optimization when no coarse model (e.g., empirical or equivalent circuits) is available. Three typical examples are used to illustrate the proposed technique.

Chapter 4 presents a new parallel decomposition technique for gradient based EM optimization with many design variables. Single large EM optimization is decomposed into multiple smaller sub-optimizations to improve the optimization efficiency. Two examples of EM optimization are used to illustrate the proposed technique.

Finally, Chapter 5 presents the conclusions and discussions on possible directions for future work.

Chapter 2

Literature Review

Engineers have been using optimization techniques for device, components and system modeling, and computer aided design (CAD) for decades [1]. Most of these techniques use simulations of the approximate models and any available derivatives to force relevant responses to satisfy the specifications subject to design constraints [20][21]. The higher the fidelity (accuracy) of the models the more expensive the computation using the classical methods. Recently, the increase in design complexity of electromagnetic (EM) structures coupled with stringent design specifications and shorter design cycles demand optimization techniques to be faster, accurate and automated.

The strength of the EM analysis, includes rigorous analysis of general microwave structures, makes the design of microwave circuits more reliable than the use of empirical or equivalent circuit models. However, design optimization of electromagnetic (EM) structures often require a massive amount of CPU time to find the optimum design space parameters. Methodologies based on exploitation of iteratively refined surrogates of accurate or high fidelity models have been developed to address this

issue[22]-[49]. Through the construction of a suitably accurate surrogate model (e.g. empirical or equivalent circuits), one can represent the high-fidelity (EM fine model) model over a region of the design space. Then, instead of optimizing the high fidelity model, we can optimize the surrogate which is further locally refined as increasingly accurate model data becomes available. Such methods are called surrogate-based methods.

2.1 Optimization problem

Let $R_f(\mathbf{x}_f, \omega)$ denote the response of the fine model under consideration. $\mathbf{x}_f \in \mathbb{R}^{n \times 1}$ is a vector of n design variables and ω represents the frequency. The original optimization problem is formulated as

$$\mathbf{x}_f^* = \arg \min_{\mathbf{x}_f \in X} U(R_f(\mathbf{x}_f, \omega)) \quad (2.1)$$

where $X \subseteq \mathbb{R}^n$ represents the fine model space and U is a given objective function. U represents the weighted error function of fine model response $R_f(\mathbf{x}, \omega)$ and the desired design specifications. $\mathbf{x}_f^*$ represents the optimal design vector satisfying the design specifications.

2.2 Direct EM Optimization methods

Recent advances in optimization methods made the EM based design more feasible for practical microwave structures. Direct EM optimization method refers to the existing built-in optimization method in the EM simulators. With the dramatic

increase in the computer hardware performance, EM simulator solves Maxwell equations for circuits and arbitrary geometrical shapes. Over the last decade, advances in the technology of workstations and computers have enabled the EM simulators to run the classical optimization methods on personal computers. The commonly used optimization methods are classified into two categories, global optimization methods and local optimization methods. Global optimization methods such as genetic algorithms [49][60][61], particle swarm optimization [54][84], simulated annealing [61] find the global solution(s) of a constrained optimization model (generally, in the presence of multiple local optima). Global optimization methods are usually heuristic in their approach for finding the global solution, therefore they are referred to as evolutionary algorithms.

Local optimization methods include gradient based optimization techniques and derivative-free methods. The most popularly used gradient based methods are quasi-Newton method [62][63], conjugate gradient method [62][63], and sequential non-linear programming [62]. Examples of derivative-free methods are pattern search [62] and Nelder-Mead method [62].

2.2.1 Derivative-free Pattern Search Method

Pattern search [62][63] belongs to a class of numerical optimization techniques that does not require the gradient of the problem to be optimized. The situations where the pattern search is useful is when the gradient for objective function is not defined at all points.

Pattern search has two moves: exploratory move and pattern move. The ex-

ploratory move explore the local behavior and the information of the objective function. The exploratory move is performed until there is an improvement in the value of objective function in each coordinate direction by increasing or decreasing the step size. Once, the exploratory move is performed in all directions, pattern movement is performed. The pattern move is included to move from the base point along the base line (formed using previous iterations) to a next base point. If the pattern move produces improvement in the value of the objective function, then we proceed to exploratory move for the next iteration, otherwise the step size for the exploratory move in the current iteration is reduced and the process is repeated.

2.2.2 Gradient based Quasi-Newton Method

Quasi-Newton method is based on Newton's method to find the stationary point of a objective function, where the gradient is zero. Gradient methods (conjugate gradient) assumes that the objective function can be locally approximated as a quadratic function near its optimal point. Newton method uses both first and second order derivatives to find the stationary point using least number of iterations compared to any other gradient method. Newton's method always converges to the minimum point (for a minimizing function). However,

- a) It requires the storing of $n \times n$ Hessian matrix.
- b) It is difficult to calculate the elements of the Hessian matrix directly.
- c) It requires inversion of the Hessian matrix in each iteration.

To overcome this case, quasi-Newton is developed. In the quasi-Newton method, the inverse of Hessian matrix is approximated using another matrix that is computed using only the first order derivatives of the objective function. There are several methods to approximate the Hessian matrix such as DFP (Davidson-Fletcher-Powell) [62] and BFGS (Broyden-Fletcher-Goldfarb-Shanno) [62] [21] rank-1 and rank-2 updates.

In general, existing EM simulation tools use the latest optimization methods for microwave design. A frequently used approach is the use of gradient optimization to drive EM simulation for microwave design. These methods are mostly sequential and evaluate the next optimization update using single-point EM evaluation to minimize the CPU time per iteration. One such gradient optimization method available in EM simulation tools is the direct EM optimization approach, which involves many iterations of optimization; therefore it has to perform that many single-point EM evaluations. Recent efforts on improving the speed of optimization include space mapping (SM) techniques [22]-[31] and artificial neural networks (ANN) [39]-[42].

2.3 Space Mapping Optimization

Space mapping technique [22]-[31] is the most popularly used surrogate based optimization technique. SM aims to use fewest possible fine model (EM model) evaluations by exploiting coarse models (e.g., empirical or equivalent circuits) during optimization, thereby increasing the speed of overall optimization. Space mapping (SM) is a recognized engineering optimization paradigm consisting of a number of efficient optimization approaches. The iterative optimization and updating of the

surrogate model (coarse model) replaces the direct EM optimization of an accurate, but computationally expensive fine model of interest. The formulation of SM optimization algorithm has been comprehensively explored in the recent years [22] and is shown in Fig. 2.1. Space mapping combines the computational efficiency of coarse models with the accuracy of fine models. The coarse models are typically empirical functions or equivalent circuits, which are computationally efficient but the accuracy is low. The fine models can be provided by an electromagnetic (EM) simulator, which is accurate, but computationally intensive. The space mapping establishes a mathematical link between the coarse and the fine models and directs the bulk of the CPU-intensive computations to the coarse models, while preserving the accuracy from the fine models. The mapped coarse model may be re-optimized to obtain a new design solution.

2.3.1 The Space Mapping Concept

The mathematical representation of space mapping methodology presented in [22] is recalled as follows. Let the coarse and fine model design parameters are denoted by $\mathbf{x}_c$ and $\mathbf{x}_f \in \Re^{n \times 1}$ respectively. Let $\mathbf{R}_f(\mathbf{x}_f) \in \Re^{N_f \times 1}$ represent a vector of fine model responses, e.g., $|S_{11}|$ at N_f selected frequency points ω . $\mathbf{R}_f(\mathbf{x}_f)$ is given by

Figure 2.1: Space mapping concept following [31]

$$\mathbf{R}_f(\mathbf{x}_f) = \begin{bmatrix} R_f(\mathbf{x}_f, \omega_1) \\ R_f(\mathbf{x}_f, \omega_2) \\ \vdots \\ R_f(\mathbf{x}_f, \omega_{N_f}) \end{bmatrix} \quad (2.2)$$

Similarly, $\mathbf{R}_c \in \mathbb{R}^{N_f \times 1}$ represents a vector of coarse model responses. Space mapping aims to find a mapping function $\mathbf{P}_{SM}$ that relates the fine and coarse model parameters as

Figure 2.2: Mathematical representation of space mapping following [22]

$$\mathbf{x}_c = \mathbf{P}_{SM}(\mathbf{x}_f) \quad (2.3)$$

such that

$$\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f)) \approx \mathbf{R}_f(\mathbf{x}_f) \quad (2.4)$$

in a region of interest.

Space mapping avoids direct EM optimization that requires us to solve (2.1) to find optimal design parameter $\mathbf{x}_f^*$. Let $\bar{\mathbf{x}}_f$ be a good estimate of $\mathbf{x}_f^*$ and is given by

$$\bar{\mathbf{x}}_f = \mathbf{P}_{SM}^{-1}(\mathbf{x}_c^*) \quad (2.5)$$

where $\mathbf{x}_c^*$ is the result of coarse model optimization.

Space mapping algorithms initially optimize the coarse model to obtain the optimal design $\mathbf{x}_c^*$. Subsequently, a mapped solution is found by minimizing the error between the fine model response and the coarse model response defined by

$$\mathbf{P}_{SM}(\mathbf{x}_f) = \arg \min_{\mathbf{x} \in X_c} \|\mathbf{R}_c(\mathbf{x}) - \mathbf{R}_f(\mathbf{x}_f)\| \quad (2.6)$$

where X_c is the coarse model space. Once the mapping parameters are extracted, $\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f))$ is optimized to find the solution of the optimization problem defined in (2.1). The surrogate model in this scenario is the coarse model along with mapping parameter $\mathbf{P}_{SM}$, i.e., $\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f))$. Thus, the optimization problem is rewritten as

$$\bar{\mathbf{x}}_f^* = \arg \min_{\mathbf{x}_f \in X} U(\mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f))) \quad (2.7)$$

where $\bar{\mathbf{x}}_f$ may be close to $\mathbf{x}_f^*$ if $\mathbf{R}_c$ is close enough to $\mathbf{R}_f$. Fig. 2.2 shows the mathematical representation of SM methodology presented in [22].

In the past decade, many efficient optimization algorithms such as aggressive space mapping [23], implicit space mapping (ISM) [24][25], output space mapping (OSM) [26], space mapping interpolating surrogates (SMISs) [27] that utilize surrogate models based on ISM and OSM has proven successful for difficult optimization problems.

2.3.2 Input Space Mapping

Input space mapping [22] aims at reducing misalignment between fine and coarse model by using an affine variable transformation based on the available fine model data. Input space mapping is also called original space mapping wherein Equation (2.3) is represented as

$$\mathbf{x}_c = \mathbf{P}_{SM}(\mathbf{x}_f) = \mathbf{B}\mathbf{x}_f + \mathbf{c} \quad (2.8)$$

Figure 2.3: Illustration of Implicit Space Mapping (ISM) following [24]

where matrices $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $\mathbf{c} \in \mathbb{R}^{n \times 1}$ are obtained using parameter extraction procedure in (2.6).

2.3.3 Implicit Space Mapping

Implicit space mapping (ISM) [24][25] explores the flexibility of the preassigned parameter such as dielectric constant, substrate height in the design optimization process. Let $\mathbf{x}_{aux}$ represent the preassigned parameters (auxiliary parameters), then the coarse model response vector is $\mathbf{R}_c(\mathbf{x}_c, \mathbf{x}_{aux})$. Therefore, implicit space mapping shown in Fig. 2.3 aims at establishing an implicit mapping $\mathbf{Q}_{SM}$ between the spaces $\mathbf{x}_f$, $\mathbf{x}_c$ and $\mathbf{x}_{aux}$, $\mathbf{Q}_{SM}(\mathbf{x}_f, \mathbf{x}_c, \mathbf{x}_{aux}) = 0$ such that $\mathbf{R}_f(\mathbf{x}_f) \approx \mathbf{R}_c(\mathbf{x}_c, \mathbf{x}_{aux})$ over a region in a parameter space. Implicit mapping ensures to produce a good match between the coarse model and fine model in the first iteration when the input space mapping alone cannot replicate a good match.

2.3.4 Output Space Mapping

Output space mapping technique has been proposed [27]. Output space mapping enhances the surrogate model by a correction term (residual) $\Delta \mathbf{R}$ which is the difference between the fine model and the original space mapping response at the current iteration so that a perfect match between these two models is ensured (zero-order consistency condition [32]-[34]), i.e.,

$$\Delta \mathbf{R} = \mathbf{R}_f(\mathbf{x}_f) - \mathbf{R}_c(\mathbf{P}_{SM}(\mathbf{x}_f)) \quad (2.9)$$

The surrogate model (coarse model along with mapping function) is further enhanced by using the Jacobian of the $\Delta \mathbf{R}$ (to satisfy first order consistency [32]-[34] between the surrogate model and fine model at the current design). If the misalignment between the fine and coarse models is not significant, SM-based optimization algorithms typically provide excellent results after only a few evaluations of the fine model.

2.3.5 Tuning Space Mapping

Tuning space mapping (TSM) [28] is a special type of SM technique that caters to tuning of EM structures. The surrogate model is replaced by a tuning model which introduces circuit components in to the fine model structure. The tuning model is optimized within a circuit simulator. With the optimal tuning parameters, thus obtained, they are mapped or transformed into the design variables using fast space-mapping surrogate or analytical formulas if available. Tuning models require a significant engineering expertise for a successful implementation of the optimization

process using TSM approach.

2.3.6 Coarse and Fine mesh Space mapping

In coarse and fine mesh SM method [30], the fine model uses a fine mesh for EM simulation and the coarse model uses coarse mesh for EM simulation. The accuracy of the fine model is achieved by the mesh convergence process. The EM simulation and mesh refinement are performed iteratively until the simulation results between successive iterations converge. The fine mesh simulation is accurate yet computationally expensive. The coarse model uses a coarse mesh EM simulation which only needs local meshing in EM simulation without the mesh convergence process. Also, the coarse model is computationally fast but less accurate. This method is useful when equivalent coarse model for the EM problem is not available, and when a conventional space mapping is not applicable.

Space mapping techniques (SM) [22]-[31] aims to use fewest possible fine model (EM model) evaluations by exploiting coarse models (e.g., empirical or equivalent circuits) during optimization, thereby increasing the speed of overall optimization. Recent advances in space mapping such as three level output space mapping [35], constrained parameter extraction using implicit space mapping [36], space mapping optimization using EM-based adjoint sensitivity [37], and fast EM modeling using shape-preserving response prediction and space mapping [38] focus on reducing the number of fine model evaluations.

2.4 Artificial Neural Networks (ANN)

Another method to speed up the optimization process is neural network based parametric modeling and optimization [39]-[42]. Artificial neural networks has been widely used for modeling microwave devices and circuits in several innovative ways. A complete review on ANNs for microwave modeling is described in [39][41]. In neural network based parametric modeling and optimization [39][40] the computation burden of fine model evaluations are shifted from online optimization to offline neural network training. The trained neural network model is used for optimization.

The most common strategy for optimizing microwave structures using neural networks consists of generating a neural model (surrogate model) with a certain training region of the design parameters and then applying conventional optimization to the neuromodel to find the optimal solution that yields the desired response. The computational cost of developing the neuromodel is dependent on the number of design variables (dimensionality) which results in large number of training samples and training region of the design space. An alternative to reduce the size of the training set for the training region is to carefully select the sample points using the design of experiments methodology to ensure adequate parameter coverage [43]. Another way to speed up the training process was proposed in [44] by means of preliminary neural clusterization of similar responses using the self-organizing feature map (SOM) approach. Innovative strategies have been proposed to reduce the learning data and improve the capabilities of ANN by incorporating empirical models: neural EM-based design exploiting prior knowledge input (PKI) method [45],

Figure 2.4: Prior Knowledge Input (PKI) neuromodeling concept following [45]

knowledge based neural networks (KBNN) [46], and neural space mapping (NSM) approach [47] are developed.

2.4.1 Prior Knowledge Input

In the Prior knowledge input (PKI) method [45], the coarse model responses are used as inputs for the neural network model in addition to the design variables and frequency as shown in Fig. 2.4. In this case, the neural network is trained such that its output response is as close to the fine model response at all the training data samples.

Figure 2.5: Knowledge Based Neural Networks (KBNN) neuromodeling concept following [46]

2.4.2 Knowledge Based Neural Networks

Knowledge based neural networks (KBNN) [46] incorporates the available knowledge of the EM structure to build a neural network model for optimization. By inserting the microwave empirical formulas into the neural network structure, they can be refined or adjusted as part of the neural network training process. KBNNs are used extensively for developing models of microwave circuits, once trained they can be used as an accurate and inexpensive model for conventional optimal design. The neuromodeling of KBNN structures is shown in Fig. 2.5.

Figure 2.6: Space mapped neuromodeling concept following [47]

2.4.3 Neuro Space Mapping

A conglomerate of the both SM and ANN has also been proposed and developed in [47]-[49]. NSM optimization follows a SM approach wherein the mapping function from fine to the coarse model parameter space is implemented using neural networks. The mapping function in (2.3) is determined by neural network training at multiple samples points. Once the neural network is trained, the space mapped neuromodel is an accurate representation of the fine model for further higher level simulation and optimization.

2.5 Overview of Parallel Computing Architectures

Traditional computer software has been written for sequential computation. An algorithm is implemented as a serial stream of instructions. These instructions are

executed on a Central Processing Unit (CPU) on one computer. A target problem is solved by executing instructions sequentially. Parallel computing techniques execute instructions simultaneously by multiple processing elements, which are accomplished by breaking the problem into independent parts so that each processing element can execute its own part concurrently with the others [50]. The processing elements include a variety of computational resources, such as a single computer with multiple processors, several interconnected computers, graphics processing units, or any combination of the above.

To determine the effect of parallel computing algorithm, we measure the parallel speed up against the corresponding sequential algorithm. Let T_1 be the time of executing algorithm sequentially. For parallel computing, let P be the number of processors, then the execution time of parallel algorithm with P processors is T_2 . The speed up S_p is measured as

$$S_p = \frac{T_1}{T_2} \quad (2.10)$$

Ideally, a linear speed up $S_p = P$ is expected to obtain when using P processors. However, in most cases, the ideal speed up cannot be achieved due to several factors such as design of the algorithm, competition for the shared memory, communication and synchronization, e.t.c. Parallel efficiency is introduced to show how the processors are well-utilized to execute the parallel algorithm. The parallel efficiency (η) is defined as

$$\eta = \frac{S_p}{P} \quad (2.11)$$

In this thesis, Hybrid Distributed-Shared Memory Architecture (HDSMA) is used for parallel computation. The structure of Hybrid distributed-shared memory architecture (HDSMA) is shown in Fig. 2.7. In this architecture, the system consists of multiple interconnected computers called nodes. Each node has its private local memory that is uniformly shared by multiple processors. On one node, multiple processors can operate independently but share the same memory resources on that node. The communications between nodes are performed by message passing [51]. The most typical network is Ethernet. HDSMA contains the advantage of Shared Memory Architecture (SMA) that the unified global memory address space provides a fast data sharing between tasks and a user-friendly programming perspective. This architecture exploits the maximum CPU capacity in the computer cluster. Compared with the SMA, HDSMA expands the performance on a single computer with SMA computer to a group of computers. Compared with Distributed Memory Architecture (DMA), HDSMA makes full use of all available processors on each node and provides extra computational capability.

On the top level, tasks are distributed among all available nodes in the same way as DMA by Message Passing Interface (MPI) [51], which is the only message passing library that is considered as a standard. On the lower level, the master thread on each node will fork a group of slave threads to execute tasks that distributed to that node simultaneously by Open Multiprocessing (OpenMP) [52]. HDSMA reaches the maximum speed up against sequential computation. It is a highly

Figure 2.7: Structure of Hybrid Distributed-Shared Memory Architecture(HDSMA) following [50]

scalable architecture that allows user easily adding more nodes to the entire system to further increase the overall computational capability.

Recently, parallel computation has been researched to speed up intensive computational processes and utilize computers number crunching ability more effectively [50][53]. Parallel computation methods has been used for global optimization such as genetic algorithms (GA) [49] and particle swarm optimization (PSO) [54], where the evaluations of multiple chromosomes of a population (GA) or multiple particles of a swarm (PSO) are computed in parallel. Parallel automatic model generation technique is proposed in [55], using parallel adaptive sampling and parallel data generation to save model development time. Distributed fine model evaluation techniques has been proposed in [56][57], where the fine model frequency range is sub-divided into different frequency bands and evaluated on parallel processors. A

parallel space mapping approach [58] is used for EM optimization, where the fine model evaluations and training of the surrogate model uses parallel computational techniques. In [37][38][30] a coarse mesh EM simulation is used instead of an equivalent circuit model for space mapping techniques based on sequential computational approach. Recently, a preliminary work on EM optimization with no coarse models is presented in [64] where the use of parallel computational techniques for EM optimization was briefly explored.

2.6 Summary

In this chapter, a literature review of popularly used EM optimization techniques for microwave design has been presented. An overview of classical space mapping methods such as implicit space mapping, output space mapping, tuning space mapping and the recently developed coarse and fine mesh space mapping is reviewed. Other known EM optimization techniques using artificial neural networks such as prior knowledge input, knowledge based neural networks and neurospace mapping methods are also discussed. Further, an overview of the parallel architecture, Hybrid Distributed-Shared Memory Architecture (HDSMA), used in this thesis for parallel computations is also presented. Most of the EM optimization techniques presented require coarse model for surrogate based optimization. In the next chapter, a new EM optimization technique that exploits parallel computations and require no coarse model is presented.

Chapter 3

Parallel Computational Approach to Gradient based EM Optimization of Passive Microwave Circuits

Conventional EM optimization aims to use fewest possible fine model evaluations to increase the speed of optimization. In this chapter, we propose to use a large number of fine model evaluations to achieve an overall speedup. A large number of fine model evaluations allows us to build a surrogate model valid in a large neighborhood. In the proposed technique, these valid surrogate models are used to achieve large and effective optimization updates, thereby resulting in fewer iterations of the optimization process. Valid surrogate models uses many fine model evaluations which are realized in parallel using hybrid distributed shared memory computing platforms. Parallel computation of large number of fine model evaluations reduces the major computational time required for constructing a surrogate model. Furthermore, we exploit trust region algorithms to guarantee convergence

and to re-define the fine model evaluation range in each iteration of the proposed optimization algorithm. The proposed technique aims to increase the speed of gradient based EM optimization when no coarse model (e.g., empirical or equivalent circuits) is available. Three typical examples are used to illustrate the proposed technique.

3.1 Introduction

Design and optimization of electromagnetic (EM) structures often require a massive amount of CPU time to find the optimum design space parameters. The strength of the EM analysis, includes rigorous analysis of general microwave structures, makes the design of microwave circuits more reliable than the use of empirical or equivalent circuit models. Recent advances in optimization methods, especially gradient based methods, made the EM based design more feasible for practical microwave structures. These methods are mostly sequential and evaluate the next optimization update using single-point EM evaluation to minimize the CPU time per iteration. The gradient optimization method involves many iterations of optimization; therefore it has to perform that many single-point EM evaluations. Recent efforts on improving the speed of optimization include space mapping techniques and artificial neural networks which are discussed in literature review.

Most of the space mapping techniques [22]-[31] described earlier in literature review require the availability of a coarse model of the microwave structure. When the coarse model is not available, parallel approach to gradient based EM optimization still remains an open subject. Recently, a preliminary work on EM optimization

with no coarse models is presented in [64] where the use of parallel computational techniques for EM optimization was briefly explored. This chapter provides a comprehensive discussion on parallel EM optimization using an integrated trust region framework with a systematic parallel approach. A specific type of design of experiments (DOE), i.e., orthogonal arrays, for sampling multiple EM points over the predefined surrogate model range are explored. Without the loss of generality, trust region algorithms [32][65] are tailored such that the convergence properties are satisfied and the speed of optimization is increased compared to [64] by using adaptive large optimization updates. Compared to [58] the proposed optimization technique requires no coarse models. In comparison to [39]-[42], the surrogate model development proposed in the present chapter doesn't involve an overhead cost of training, thereby speeding the optimization process. Also, the proposed parallel optimization can be used even when the EM simulator uses fast simulation feature or frequency interpolating simulation feature.

In the proposed technique, we deliberately use many fine model evaluations (multi-point function evaluations) within each iteration of optimization. The large number of fine model evaluations are used to construct a surrogate model in a relatively large neighborhood around a central point. Central point refers to the best solution from optimizing the surrogate model in each iteration. Each iteration has a set of fine model evaluations that are generated in parallel on a hybrid distributed-shared memory computational architecture. The surrogate model developed provides rich information in estimating the direction for the next optimization update. Further, the central point is updated after each iteration using trust region

framework, thereby allowing the trust radius to change dynamically from iteration to iteration. Since the surrogate model is valid in a large neighborhood, this results in a large and effective optimization update. Therefore, the proposed method takes fewer iterations and achieves speedup in the optimization process. Three typical examples are illustrated to verify the proposed optimization technique.

This chapter is organized as follows. Section 3.2 outlines the original optimization problem followed by a detailed description of the proposed parallel optimization technique. Various aspects of the proposed technique such as sampling methods, parallel distribution strategy, surrogate modeling, and optimization update using trust region are elaborated. In Section 3.3, we demonstrate the EM optimization using three microwave examples including an inter-digital band-pass filter, waveguide filter and a dielectric resonator loaded antenna. In Section 3.4, we provide conclusions.

3.2 Proposed Parallel Optimization using Trust Region Framework

3.2.1 Formulation of the Original Optimization Problem

Let $R_f(\mathbf{x}, \omega)$ denote the response of the fine model under consideration, $\mathbf{x}$ denote a vector of n design variables and ω represents the frequency. The original optimization problem is formulated as

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbf{X}} U(R_f(\mathbf{x}, \omega)) \quad (3.1)$$

where $X \subseteq \mathbb{R}^n$ represents the design variable space and U is a given objective function. U represents the weighted error function of fine model response $R_f(\mathbf{x}, \omega)$ and the desired design specifications. $\mathbf{x}^*$ represents the optimal design vector satisfying the design specifications. Let S_{uq} and S_{lq} represent the set of upper and lower specifications, where q is the index for frequency. Let Q_u and Q_l represent the index sets of the frequency points for upper specification and lower specification respectively. The error functions for upper and lower specifications are defined as

$$\begin{aligned} e_{uq} &= W_u(R_f(\mathbf{x}, \omega_q) - S_{uq}), \quad q \in Q_u \\ e_{lq} &= W_l(S_{lq} - R_f(\mathbf{x}, \omega_q)), \quad q \in Q_l \end{aligned} \tag{3.2}$$

where W_u and W_l are the weighting factors for Q_u and Q_l respectively. ω_q represents the q th frequency sample. The objective function U defined in (3.1) may be either a minimax or a generalized l_p function [20] of the error functions defined in (3.2).

3.2.2 Introduction to the Proposed Optimization Technique

Conventional EM optimization of the EM structure is generally based on single fine model evaluations per iteration. In the proposed method, we use a set of fine model evaluations in a relatively larger neighborhood around a central point to provide rich information in estimating the direction for the next optimization update. These fine model evaluations along with the central point are used to predict the overall behavior of the device in the region of interest. The central point is updated after each iteration of the proposed optimization using trust region framework.

The proposed parallel optimization technique is formulated as distribution of

multiple data samples using sampling techniques, computation using parallel processors, surrogate modeling over multiple EM samples and optimization update using trust region framework.

3.2.3 Distribution of EM Samples in the n -dimensional Space

In the proposed technique, large and effective optimization updates per iteration are used to increase the overall speed of the optimization process. The large and effective optimization updates are possible only if the surrogate model is valid in a relatively large neighborhood. For such a surrogate model to be valid, a large number of fine model evaluations are necessary. Therefore, the very first step for the successful surrogate model is the distribution of the samples spanning across the large neighborhood. The most commonly used sample distributions are full-grid distribution, star distribution and orthogonal distribution.

The full-grid distribution sampling method is feasible when the number of design variables n is small and the number of levels (subspace divisions) are small. However, when n or number of levels becomes larger, the full-grid distribution leads to an exponential increase of sample points. Furthermore, such a huge number of sample points may require millions of fine model evaluations which are not feasible. Alternatively, star distribution of sample points around a central point are used in [47]. In star distribution, when the number of design variables n becomes large, the number of sample points grows linearly. However, the surrogate model constructed using star distribution is valid in a relatively small neighborhood around the central point. In the proposed technique, orthogonal sampling [66] is used for

generating multiple sample points where the subspace divisions are sampled with the same density and are orthogonal. Orthogonal sampling around the central point enables the surrogate model to be valid in relatively large neighborhood compared to star distribution and also uses far fewer sampling points compared to full-grid distribution.

Let n be the number of optimization variables, i.e., the dimension of vector $\mathbf{x}$. Let $\mathbf{x}^c = [x_1^c \ x_2^c \ \cdots \ x_n^c]^T$ be the best solution of $\mathbf{x}$ in the previous iteration. $\mathbf{x}^c$ will be referred to as the central point in the current iteration for the subsequent descriptions. Around the neighborhood of this central point $\mathbf{x}^c$, we define the region of interest X_R using trust region algorithm as

$$X_R = \{ \mathbf{x} \mid x_j^c - \delta_j \leq x_j \leq x_j^c + \delta_j; \quad \text{where } j = 1, 2, \dots, n \} \quad (3.3)$$

where δ_j determines the interval of each design variable in the current iteration and is initialized in the first iteration by a user defined value. $\boldsymbol{\delta} = [\delta_1 \ \delta_2 \ \cdots \ \delta_n]^T$ determines the interval range of the surrogate model for the current iteration, where the surrogate model is presumed to be valid. δ_j is selected as the percentage deviation from the central point. The percentage deviation for each design variable is determined according to the fine model sensitivities estimated through fine model evaluations [67] in the first iteration. For the later stages of optimization, $\boldsymbol{\delta}$ is updated using trust region framework.

Once the trust region X_R is defined, design of experiments (DOE) [66] sampling strategy e.g., orthogonal sampling, is used to generate a set of samples around the

Figure 3.1: Illustration of orthogonal arrays (a specific type of DOE) sampling technique to generate multiple sample points $(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{N_s})$ around the central point $\mathbf{x}^c$ in the trust region X_R . The central point $\mathbf{x}^c$ is updated after each iteration of the proposed optimization method. The central point $\mathbf{x}^c$ and the DOE samples around the central point move as the proposed optimization progresses from the current iteration to the next iteration

central point $\mathbf{x}^c$ in each iteration of the optimization process. Fig. 3.1 shows design of experiments (DOE) sampling strategy used to generate multiple sample points around a central point $\mathbf{x}^c$ for the current iteration. Let $\mathbf{x}^m = [x_1^m \ x_2^m \ \dots \ x_n^m]^T$ denote one such sample in the trust region X_R , where $m \in \{1, 2, \dots, N_s\}$ and represents the sample number in the current iteration. N_s represents the total number of sample points selected using DOE. When the optimization process moves, the central point $\mathbf{x}^c$ moves to a new optimization update. All the other DOE sample points $(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{N_s})$ move along with the central point when the optimization progresses from current iteration to next iteration. Therefore, the values of $\mathbf{x}$ samples change from iteration to iteration.

3.2.4 Parallel Computation over Multiple EM Samples

In the proposed technique, multi-point fine model evaluations in the trust region X_R is the major computational burden for constructing a surrogate model. Sequential fine model evaluations of the samples require N_s times the computational time of one fine model evaluation. Therefore, to reduce the overall computational time, we propose to use parallel computational approach for fine model evaluations over multiple samples.

Parallel computational approach is implemented to accelerate data generation on hybrid distributed-shared memory computational architecture, i.e., a cluster consisting multiple computers with multiple processing cores on each computer. A systematic and scalable job distribution is used to evaluate multiple fine model responses (jobs). Firstly, fine model evaluations are distributed across all available computers. Each computer is designed to get equal or similar number of jobs so that the workload on each computer over the entire cluster is balanced. In the proposed technique, we set independent simulation environment in each processor where the input-output files in each environment are separated. Each processor executes a job, and then requests another job, until no jobs remain to be executed. Fine model response $R_f(\mathbf{x}^m, \omega)$ is generated for the set of samples $m = 1, 2, \dots, N_s$, in the trust region X_R simultaneously by simulating multiple EM structures in parallel.

$$\begin{aligned}
& \{R_f(\mathbf{x}^m, \omega) | m = 1, 2, \dots, N_s\} \\
& = \{R_f(\mathbf{x}^1, \omega), R_f(\mathbf{x}^2, \omega), \dots, R_f(\mathbf{x}^{N_s}, \omega)\} \quad (3.4)
\end{aligned}$$

Let S_p be the speed up ratio between optimization time of a sequential optimization process to that of the parallel optimization process using P processors, and is defined as

$$S_p = \frac{T_s + \left(\sum_{m=1}^{N_s} T_m + T_p \right) i_{total}}{T_s + \left(\frac{N_s}{P} \max_{1 \leq m \leq N_s} \{T_m\} + T_p + T_{ov} \right) i_{total}} \quad (3.5)$$

where T_s represents the CPU time for estimating the sensitivities to determine the interval range of each design variable for the first iteration of the optimization process. T_p is the CPU time per iteration to build a surrogate model and minimize the objective function U and is non-parallel. T_m is the fine model evaluation time for a sample m in the optimization process. T_{ov} is the overhead CPU time (e.g., communication time between multiple processors) of each iteration for generating many fine model responses in parallel using P processors. i_{total} represents the number of iterations of the proposed optimization and N_s represents the number of samples per iteration in the optimization process. Parallel efficiency (η) is defined as the speedup achieved S_p divided by the total number of processors P ,

$$\eta = \frac{S_p}{P} \quad (3.6)$$

It is observed that maximum parallel efficiency (η) is achieved when the overhead CPU time and surrogate model generation and optimization CPU time per iteration are minimized. In the proposed technique, the number of samples N_s derived from typical DOE distributions are comparable to the number of processors of a typical or moderate computer cluster, therefore we set the number of processors to be equal to the number of samples, i.e., $P = N_s$. This simplifies the description while maximizing practical parallel efficiency of our algorithm.

3.2.5 Surrogate Modeling using Multiple Points across the Trust Region X_R

In existing EM optimization, surrogate models are constructed using single point fine model data, or accumulated fine model data from previous iterations [22][31]. In [58] multiple samples using star distribution are used for constructing surrogate model. However, the availability of a fast coarse model or problem-dependent prior knowledge is a pre-requisite for these techniques. In reality, not all designs can satisfy this pre-requisite. Therefore, the proposed technique aims to address EM optimization without the use of coarse models. In this way, the proposed technique is useful for EM optimization even when the empirical or equivalent circuits models for the EM structure are not available. In the proposed technique, we use widely accepted transfer functions to represent the EM behavior for constructing the surrogate.

The surrogate model, that is valid across the entire trust region (DOE sample space), provides the output response $R_s(\mathbf{x}, s)$ as a function of frequency ($s = j\omega$)

and the design variables,

$$R_s(\mathbf{x}^m, s) \approx R_f(\mathbf{x}^m, s) \quad (3.7)$$

The surrogate model response $R_s(\mathbf{x}, s)$ is expressed as a transfer function in the rational function format [68] as shown in (3.8). This can be achieved by using a popular vector-fitting method [69]-[71].

$$R_s(\mathbf{x}^m, s) = \frac{\sum_{l=0}^M \hat{\alpha}_l^m s^l}{1 + \sum_{k=1}^N \hat{\beta}_k^m s^k} \quad (3.8)$$

where M, N represents the highest order of the numerator and denominator of the transfer function respectively. If the order of the transfer function in (3.8) is very high, then the transfer function has a very large dynamic range because of the power series $\{s^0, s^1, s^2, \dots, s^M\}$. A high dynamic range will cause the transfer function value to be very sensitive to the coefficients $\hat{\alpha}_l^m, \hat{\beta}_k^m$. Consequently, it becomes a challenge to develop a good surrogate model. In order to resolve this challenge the power series $\{s^0, s^1, s^2, \dots, s^M\}$ are transformed to $\{e^0, e^{-j\theta}, e^{-j2\theta}, \dots, e^{-jM\theta}\}$ with a constant magnitude and a linear-phase. The numerator and denominator coefficients are transformed from s -domain to z -domain using pascal matrix [72]. Therefore, the surrogate model response is now expressed in z -domain as shown in (3.9),

$$R_s(\mathbf{x}^m, \hat{z}) = \frac{\sum_{l=0}^M \alpha_l^m \hat{z}^l}{1 + \sum_{k=1}^N \beta_k^m \hat{z}^k} = H(\boldsymbol{\alpha}^m, \boldsymbol{\beta}^m, \hat{z}) \quad (3.9)$$

where $\hat{z} = (\frac{1-s}{1+s})$, $\boldsymbol{\alpha}^m$ and $\boldsymbol{\beta}^m$ are vectors of numerator and denominator coefficients for the bilinear transfer function (3.9) in z -domain. From equations (3.7)-(3.9), it is distinctly clear that the coefficients of the transfer function $H(\boldsymbol{\alpha}^m, \boldsymbol{\beta}^m, \hat{z})$ are dependent only on the design variables and are independent of frequency. Therefore, each coefficient of the numerator or denominator for fine model response can now be expressed as a function of the design variables i.e.,

$$\gamma_r^m = g_r(\boldsymbol{\lambda}_r, \mathbf{x}^m) \quad (3.10)$$

where $\gamma_r^m \in \{\boldsymbol{\alpha}^m, \boldsymbol{\beta}^m\}$ and $r = 1, 2, 3, \dots, M+N$. r represents the index of coefficient (either numerator or denominator). The total number of coefficients for the current iteration in (3.9) is $M+N$. g_r is a function representing the relation between γ_r and design vector space $\mathbf{x}$. $\boldsymbol{\lambda}_r$ represents a vector of weighting factors (unknown factors) in the function g_r . The values of the weighting factors are optimized such that the error between fine model response and the surrogate model response is minimized.

Let $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^T \boldsymbol{\lambda}_2^T \dots \boldsymbol{\lambda}_{M+N}^T]^T$ be a complete set of unknown weighting vectors for both numerator and denominator coefficients, i.e., for all $(M+N)$ coefficients. The new weighting factors $\boldsymbol{\lambda}^*$ for the current iteration are defined as

$$\begin{aligned}
\boldsymbol{\lambda}^* &= \arg \min_{\boldsymbol{\lambda}} \sum_{m=1}^{N_s} \sum_{\omega \in \Omega} \|R_s(\mathbf{x}^m, \hat{z}) - R_f(\mathbf{x}^m, s)\| \\
&= \arg \min_{\boldsymbol{\lambda}} \sum_{m=1}^{N_s} \sum_{\omega \in \Omega} \|H(\mathbf{g}(\mathbf{x}^m, \boldsymbol{\lambda}), \hat{z}) - R_f(\mathbf{x}^m, s)\|
\end{aligned} \tag{3.11}$$

where $\mathbf{g} = [g_1 \ g_2 \ \cdots \ g_{M+N}]^T$, N_s represents the number of samples in the trust region X_R and Ω represent the complete set of frequency points used for the fine model response. In (3.11), the surrogate model is expressed as a function of numerator and denominator coefficients, frequency and the design vector space $\mathbf{x}$.

A simple quadratic polynomial function is used to represent the relation between γ_r and design vector space $\mathbf{x}$ in a relatively large neighborhood. Therefore, γ_r^m can be re-written as follows,

$$\begin{aligned}
\gamma_r^m &= \lambda_r^0 + \lambda_r^1 x_1^m + \lambda_r^2 x_2^m + \cdots + \lambda_r^n x_n^m + \lambda_r^{(n+1)} (x_1^m)^2 + \lambda_r^{(n+2)} (x_1^m x_2^m) \\
&+ \cdots + \lambda_r^{\left(n + \frac{n(n+1)}{2}\right)} (x_n^m)^2
\end{aligned} \tag{3.12}$$

where γ_r^m is a generic representation of the r^{th} coefficient (either $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$) of the transfer function in (3.9) and is evaluated at data sample m of the current iteration in the trust region X_R . Using this approach, the weighting vectors $\boldsymbol{\lambda}_r$ are formulated as

$$\boldsymbol{\lambda}_r = \left[\lambda_r^0 \ \lambda_r^1 \ \cdots \ \lambda_r^n \ w_r^{(n+1)} \ \cdots \ \lambda_r^{\left(n + \frac{n(n+1)}{2}\right)} \right]^T \tag{3.13}$$

$$\begin{bmatrix} \gamma_r^1 \\ \gamma_r^2 \\ \vdots \\ \vdots \\ \gamma_r^m \\ \vdots \\ \gamma_r^{N_s} \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_n^1 & (x_1^1)^2 & \cdots & (x_n^1)^2 \\ 1 & x_1^2 & x_2^2 & \cdots & x_n^2 & (x_1^2)^2 & \cdots & (x_n^2)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^m & x_2^m & \cdots & x_n^m & (x_1^m)^2 & \cdots & (x_n^m)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{N_s} & x_2^{N_s} & \cdots & x_n^{N_s} & (x_1^{N_s})^2 & \cdots & (x_n^{N_s})^2 \end{bmatrix} \cdot \begin{bmatrix} \lambda_r^0 \\ \lambda_r^1 \\ \vdots \\ \lambda_r^n \\ \lambda_r^{(n+1)} \\ \vdots \\ \lambda_r^{(n+\frac{n(n+1)}{2})} \end{bmatrix} \quad (3.14)$$

$$\text{or} \quad \boldsymbol{\gamma}_r = \mathbf{A} \times \boldsymbol{\lambda}_r \quad (3.15)$$

The number of unknowns in (3.13) is dependent on the size of design vector space in the design problem. The number of samples selected using DOE must be greater than the number of unknown weighting parameters for a good modeling accuracy. Also, a good modeling accuracy can be achieved by refining $\boldsymbol{\delta}$. Re-writing (3.12) in matrix form as shown in (3.14) over the set of fine model responses in the trust region X_R will result in a system of linear equations. $\mathbf{A}$ represents the matrix in (3.14) which is determined from N_s samples of $\mathbf{x}$ in the trust region for the current iteration. $\boldsymbol{\gamma}_r$ in (3.15) is a vector $[\gamma_r^1 \ \gamma_r^2 \ \cdots \ \gamma_r^{N_s}]^T$, i.e., the r^{th} coefficient for all samples ($m = 1, 2, \dots, N_s$) in the trust region X_R . Solve the linear system of equations in (3.15) using simple regression techniques to evaluate the unknown weighting factors. The weighting vector $\boldsymbol{\lambda}$ is calculated using (3.11) by minimizing the error between fine model response and the surrogate model response. A good surrogate model match is obtained not only at the DOE samples points in the

trust region, but also across the entire trust region.

3.2.6 Surrogate Model Optimization and Trust Radius Update

The surrogate model generated in section 3.2.5 is used to estimate the next prospective central point (optimization update). The valid surrogate model in the trust region X_R is optimized using optimization routines such as minimax or quasi-newton to obtain the new central point $\mathbf{x}^{new}$ and is formulated as

$$\mathbf{x}^{new} = \arg \min_{\mathbf{x} \in X_R} U(R_s(\mathbf{x}, s)) \quad (3.16)$$

where U is a given objective function, e.g., minimax or a generalized l_p function of the error in (3.2) between surrogate model response $R_s(\mathbf{x}, s)$ and the design specifications. $\mathbf{x}^{new}$ represents the surrogate model optimal point. To determine whether $\mathbf{x}^{new}$ is the new prospective central point for the next iteration we use a trust region approach [32]-[65].

A trust region framework is used to improve the convergence of the proposed parallel technique. Control index parameter r_a determines the ratio of actual reduction in the value of objective function to the predicted reduction in the value of objective function. The value of r_a is set to -1, if $U(R_f(\mathbf{x}^{new}, s)) > U(R_f(\mathbf{x}^c, s))$, i.e., the surrogate model in the trust region X_R is not accurate and will not predict the proper direction for the next optimization update. Otherwise, r_a can be calculated as

$$r_a = \frac{U(R_f(\mathbf{x}^c, s)) - U(R_f(\mathbf{x}^{new}, s))}{U(R_f(\mathbf{x}^c, s)) - U(R_s(\mathbf{x}^{new}, s))} \quad (3.17)$$

In the proposed technique, the trust region range X_R of the surrogate model changes from iteration to iteration based on trust radius. The control index parameter r_a determines whether the trust radius δ^{new} has to be expanded or contracted from the previous iteration or remain unchanged using the following conditions,

$$\delta^{new} = \begin{cases} c_1 \delta, & r_a < 0.1 \\ \min(c_2 \delta, \Delta^{max}), & r_a > 0.75 \\ \delta, & \text{Otherwise} \end{cases} \quad (3.18)$$

where Δ^{max} sets the maximum limit for each design variable. Expansion and contraction of trust radius depends on the values of c_1 and c_2 in (3.18). In this chapter, we use $c_1 = 0.69$ and $c_2 = 1.3$.

To determine whether the surrogate model optimal point $\mathbf{x}^{new}$ is the next prospective central point, we evaluate the condition $U(R_f(\mathbf{x}^{new}, s)) < U(R_f(\mathbf{x}^c, s))$, i.e., the value for the objective function U decreases (converges) from the previous iteration. If the condition is satisfied, then the optimization update $\mathbf{x}^{new}$ (surrogate model optimal point) from the current iteration will be the new central point $\mathbf{x}^c$ for the next iteration. Also, the trust radius update for the next iteration is δ^{new} , hence changing the surrogate model range. This completes one iteration of the proposed optimization technique. Otherwise, $\mathbf{x}^c$ remains unchanged and the trust radius for the current iteration is updated as δ^{new} . The algorithm is terminated if one of the conditions in (3.19) or (3.20) is satisfied; i.e., the absolute difference between subse-

quent iteration points is sufficiently small or if the desired design specifications are satisfied.

$$\|\Delta \mathbf{x}^c\| \leq \varepsilon \quad (3.19)$$

$$U(R_f(\mathbf{x}^{new}, s)) \leq 0 \quad (3.20)$$

where $\Delta \mathbf{x}^c$ is the absolute difference between the central point of the current iteration and the central point of the next iteration. ε is a user defined value (in the order of 10^{-4}). Fig. 4.4 shows a complete flowchart of our proposed optimization. The proposed algorithm is summarized below

Step 1: Set initial central point (starting point) $\mathbf{x}^c$, Δ^{max} and δ for the first iteration.

Step 2: Use design of experiments (DOE) sampling strategy to generate a set of samples $\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m, \dots, \mathbf{x}^{N_s}\}$ around the central point $\mathbf{x}^c$ over the range defined by (3.3) and is shown in Fig. 3.1.

Step 3: Evaluate EM fine model responses $R_f(\mathbf{x}^m, s)$ at all the DOE sample points using parallel hybrid distributed-shared memory computing platform for $m = 1, 2, \dots, N_s$ as in (3.4).

Step 4: Construct a surrogate model utilizing transfer functions by extracting the coefficients γ_r^m using vector fitting in (3.10)

Step 5: Find the unknown weighting parameters in (3.14) i.e.,

$$\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \sum_{m=1}^{N_s} \sum_{\omega \in \Omega} \|R_s(\mathbf{x}^m, \hat{z}) - R_f(\mathbf{x}^m, s)\|$$

Figure 3.2: Flow chart showing the proposed optimization process.

Step 6: Solve and find the surrogate model optimal point $\mathbf{x}^{new}$ using (3.16).

Step 7: Calculate the control index parameter r_a using (3.17) and update the trust radius using (3.18).

Step 8: If one of the termination conditions in (3.19) or (3.20) is satisfied then go to Step 12 else go to next step.

Step 9: If $U(R_f(\mathbf{x}^{new}, s)) < U(R_f(\mathbf{x}^c, s))$ is satisfied then go to Step 10 else go to Step 11.

Step 10: Set the optimization update (prospective central point) $\mathbf{x}^{new}$ from the current iteration as the central point for the next iteration i.e., $\mathbf{x}^c = \mathbf{x}^{new}$. Update the trust radius $\delta = \delta^{new}$ for the next iteration. Go to Step 2.

Step 11: Keep the central point $\mathbf{x}^c$ unchanged and set $\delta = \delta^{new}$. Go to Step 2.

Step 12: Stop the optimization process.

3.2.7 Discussion

The robustness of the proposed algorithm is enhanced by the use of multiple EM samples and the use of trust region methods. Multiple EM samples are used to create a very good approximation of the surrogate model in a large neighborhood. The large neighborhood around the central point for constructing a surrogate model results in a large step size for the optimization update. The large optimization update provides a better possibility of avoiding being trapped in local minima over classical quasi-Newton method. In this way, the quality of the optimization solution obtained using the proposed method can be better than that from classical quasi-Newton method. Further, by exploiting the fact that the transfer function coefficients are smoother than the S-parameters with respect to the changes in the

design parameters [68], we formulate the surrogate model using transfer function and regression techniques which can effectively handle larger region than that of the approximation functions used in the classical quasi-Newton method. This large neighborhood allows the proposed optimization to reach the optimization solution quickly. Furthermore, the large neighborhood allows the proposed algorithm to handle different starting points more effectively.

The other factor attributing to the robustness is the guaranteed convergence of the proposed algorithm. This is achieved by the use of trust region framework. The surrogate model range in each iteration of the proposed algorithm can be adaptively controlled using the control parameter r_a which determines the trust radius. The adaptive trust radius ensures the convergence properties of the trust region algorithm are satisfied.

Compared to parallel optimization methods such as PSO and GA which are non-gradient based, our proposed method is gradient based. For non-gradient methods the optimum solution is found usually at the expense of substantially more computational time compared to gradient based approach. The proposed parallel gradient method has higher speed of convergence over global optimization methods, better quality of the optimization solution and increased robustness over existing classical quasi-Newton method.

Figure 3.3: Structure of a inter-digital band-pass filter with optimization variables $\mathbf{x} = [g \ s_1 \ s_2 \ s_3]^T$.

3.3 Application Examples

3.3.1 Optimization of Inter-digital Band-pass Filter

Consider a standard inter-digital band-pass filter [64] as illustrated in Fig. 3.3. Assume equal spacing (g) between each end of the resonator and the cavity wall. Coupling ratio between resonators are adjusted by tuning the values of spacing (s_1), spacing (s_2), and spacing (s_3) between resonators. Each resonator is of length 43.18 mm, width 5 mm, thickness 0.5 mm, and the structure is enclosed in an cavity of height 10 mm. The design space vector $\mathbf{x} = [g \ s_1 \ s_2 \ s_3]^T$ for the example is chosen based on the sensitivity information.

Fine model evaluation is performed by HFSS EM simulator using fast simulation feature. The desired filter specifications for the above structure are defined as

$|S_{11}| \leq -30 \text{ dB}$ in the frequency range of 1.3 GHz to 1.8 GHz. The initial central point is selected based on designer experience and knowledge of the problem which results in the corresponding fine model response not too far away from the optimal solution. The initial central point may also be selected based on the past design of similar filters. For this example, the initial central point is the same starting point used in the reference [64]. $\mathbf{x}^c = [4.23 \ 1.25 \ 2.47 \ 2.32]^T$ (all values are in mm) is selected as a starting point. The neighborhood around the central point for each iteration X_R is defined in (3.3). δ for the first iteration is a user defined initial trust radius selected based on the sensitivity of the fine model data and is chosen as $[0.423 \ 0.125 \ 0.247 \ 0.232]^T$ (all values in mm). Fine model sensitivity information for the above example can be extracted in two ways. Firstly, using the in-built feature of the EM simulator to extract sensitivity information. This involves a significant overhead in the computational cost because of the necessity for discrete frequency sweep. Secondly, we can use perturbation techniques to extract sensitivity information using fast simulation feature without the additional overhead involved in discrete frequency sweep. However, perturbation techniques require $2n+1$ fine model responses to extract sensitivity information of the design variables. Using nine (i.e., $2n+1$) samples of data forming a star distribution, nine (i.e., $2n+1$) fine model responses are generated using parallel processors simultaneously. Now, we extract the sensitivity information using finite difference method for each design variable with minimum overhead cost. Using this method, one fine model evaluation time is sufficient to obtain all sensitivities. Δ^{max} for this structure is chosen as $[0.6345 \ 0.1875 \ 0.3705 \ 0.3480]^T$ (all values in mm) which is 15% of the central point

$\mathbf{x}^c$. Δ^{max} gives the relaxed value until the surrogate model no longer predicts the model responses accurately. Typically, it is user defined value and is selected based on the sensitivity of each design variable.

In this example, the optimization problem space has four design variables. The size of unknown weighting vector $\boldsymbol{\lambda}$ in (3.14) depends on the type of regression technique. For this example, using quadratic polynomial will result in $(1 + n + n(n + 1)/2)$ unknown weighting vectors. Therefore, using regression techniques the size of weighting vector $\boldsymbol{\lambda}$ is calculated to be 15. For a good surrogate model, the number of data samples is selected to be greater than the size of weighting vector $\boldsymbol{\lambda}$. Using DOE [66] with four variables and five levels (subspace divisions), 25 different geometrical samples are obtained which are far fewer and feasible for parallel computation compared to full-grid distribution requiring 1024 samples (4^5). These 25 data samples along with the central point are distributed on multiple computers with multiple processors such that the workload on each processor is balanced. We use a cluster of Dell PowerEdge computers with Intel Xeon X5680 processor with each computer having eight processing cores. Using this cluster, 25 fine model evaluations are executed in parallel to obtain 25 fine model responses.

The set of 25 fine model responses is used to generate a surrogate model valid in the region X_R using Steps 4 and 5 in the proposed algorithm. Vector fitting method uses just enough number of poles so as to avoid numerical instability in (3.14) while estimating the weighting vectors $\boldsymbol{\lambda}$. For this example, 12 poles are sufficient to get an accurate vector fitting for the 25 fine model responses. The coefficients γ_r^m in (3.10) are extracted from the transfer function using (3.8) and (3.9). Later, using

regression techniques Equation (3.11) is solved to estimate the weighting vectors λ for all the coefficients. Once the weighting vectors are determined, the surrogate model is optimized using NeuroModeler-plus program in the trust region X_R using (3.16) to extract the surrogate model optimal point $\mathbf{x}^{new}$.

To determine whether the surrogate model optimal point $\mathbf{x}^{new}$ is the next prospective optimization update, we evaluate the condition $U(R_f(\mathbf{x}^{new}, s)) < U(R_f(\mathbf{x}^c, s))$, where U , in this example, uses a one-sided l_2 objective function of the error in (3.2). If the condition is satisfied then evaluate (3.17) and (3.18) to update the next prospective central point and subsequently update the trust radius. Otherwise, the central point remains unchanged and the trust radius is updated using (3.18). The above process is repeated iteratively until one of the termination conditions in (3.19) and (3.20) is satisfied.

Using the proposed technique, the final optimal solution $\mathbf{x}^* = [3.0128 \ 0.728381 \ 1.92592 \ 2.28048]^T$ (all values are in mm) is obtained after six iterations. Fig. 4.6(a) and Fig. 4.6(d) show the fine model response at the initial central point (starting point) and final optimal solution respectively. For comparison purposes, we also use coarse and fine mesh space mapping optimization method [30] to optimize this filter. Fig. 4.6(c) shows the final optimal point obtained after three iterations. For a further comparison, we performed EM optimization of this filter using HFSS internal optimization feature. For convenience, we refer the HFSS internal optimization as direct EM optimization method because the optimization algorithm is applied directly to the EM fine model. The direct EM optimization method uses HFSS's built-in gradient based quasi-Newton optimization algorithm. In SM techniques

Figure 3.4: Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 302 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after three iterations) (d) Fine model response using proposed optimization (after six iterations).

the optimization is applied to the coarse model which are iteratively mapped to the EM fine model (indirect optimization). Fig. 4.6(b) shows the final optimal point obtained after 302 fine model evaluations using direct EM optimization. Table 3.1 and Fig. 4.6 shows that direct EM optimization doesn't converge to the optimal solution whereas our proposed technique takes 29.24 min using 25 parallel processors (one processing core for one fine model evaluation) to reach an optimal point.

Fig. 4.8 shows the values of the objective function U for the proposed opti-

Table 3.1: Comparison of CPU Time for Inter-digital Band-pass Filter with four design variables

Optimization	Direct EM optimization using 8 cores	Coarse and fine mesh SM optimization	Proposed optimization
Fine model evaluation time for each sample	2m 4s	2m 4s	2m 53s
No. of fine model samples per iteration	1	1	25
No. of iterations	302	3	6
Time for sensitivity estimation	-	-	2m 53s
Time for evaluating all fine models	$2.07\text{m} \times 302$	$2.07\text{m} \times 4$	$2.88\text{m} \times 7$
Time for surrogate model generation and optimization	0	-	$1\text{m } 2\text{s} \times 6$
No. of coarse models evaluations	-	242	-
Time for coarse model evaluations	-	$35\text{s} \times 242$	-
Total time	624.11m*	149.43m*	29.24m

* - Design specifications are not satisfied.

mization and coarse and fine mesh SM optimization. Using coarse and fine mesh SM method, the value of objective function decreases for three iterations and on further optimization the value of objective function doesn't improve even after six

Figure 3.5: The values of objective function for the inter-digital band-pass filter using: the proposed method (o) and coarse and fine mesh SM method (x).

iterations. Fig. 4.8 also shows that, the proposed optimization converges quickly to reach the termination condition in (3.20) (satisfies the design specification) compared to coarse and fine mesh SM optimization.

Since we have 25 data samples (N_s), we use 25 parallel processors (P) to generate 25 fine model responses. Using proposed parallel technique (3.4) the CPU time (T_k) for 25 fine model evaluations is 2.88 min, whereas the CPU time for the sequential fine model evaluations is 72 min ($25 \times 2.88 \text{ min}$). The CPU time T_s for estimating the sensitivities of each design variable is 2.88 min. Surrogate model generation and optimization takes 1.033 min CPU time (T_p). The overhead CPU time T_{ov} (for e.g., communication time between multiple processors) is considerably small. The speedup (S_p) factor of 16.74 is calculated using (3.6) which results in a parallel efficiency (η) of about 66.9%. It is interesting to note that for direct

EM optimization we use the built-in capability of HFSS to utilize multiple cores for evaluating the fine model response. The proposed parallel technique, even though increases the total number of fine model evaluations, takes a large and effective optimization update and converges to the optimal solution faster.

As a further experiment, we use a new case of different number of design variables to demonstrate the capability of our proposed design strategy. The design vector space has three design variables $\mathbf{x} = [s_1 \ s_2 \ s_3]^T$ while the spacing between resonator and the cavity wall is fixed at 3 mm. We choose a different starting point i.e., $\mathbf{x}^c = [1.5 \ 2.8 \ 2.2]^T$ (all values are in mm) to re-run our proposed method. Using the same initial trust radius δ and Δ^{max} , the proposed method reaches the final optimal solution $\mathbf{x}^* = [0.7225 \ 1.924 \ 2.27507]^T$ (all value in mm) after seven iterations. Fig. 3.6(a) and Fig. 3.6(d) shows the fine model response at the initial central point (starting point) and final optimal solution respectively. Fig. 3.6(a) shows that the fine model response at the initial point for the new case which is a different starting point from Fig. 4.6(a) for the optimization process.

For comparison purposes, we use coarse and fine mesh space mapping optimization method [30] to optimize this filter. Fig. 3.6(c) shows the final optimal point is obtained after three iterations. We also performed direct EM optimization of this filter using quasi-Newton method for a further comparison. Fig. 3.6(b) shows the final optimal point is obtained after 227 fine model evaluations. Table 3.2 and Fig. 3.6 shows that direct EM optimization doesn't converge to the optimal solution whereas our proposed technique takes 33.15 min using 16 parallel processors (one processing core for one fine model evaluation) to reach an optimal point. Fig.

Figure 3.6: Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 227 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after three iterations) (d) Fine model response using proposed optimization (after seven iterations).

3.7 shows values of the objective function U for the proposed optimization and coarse and fine mesh SM optimization. Fig. 3.7 shows that, the proposed optimization converges quickly to reach the termination condition in (3.20) (satisfies the design specifications) compared to coarse and fine mesh SM optimization. The speedup (S_p) factor of 11.11 is calculated using (3.6) using 16 parallel processors and a parallel efficiency (η) of about 69.4%. The proposed method converges to the optimal solution and has high parallel efficiency even with different number of

Table 3.2: Comparison of CPU Time for Inter-digital Band-pass Filter with three design variables

Optimization	Direct EM optimization using 8 cores	Coarse and fine mesh SM optimization	Proposed optimization
Fine model evaluation time for each sample	2m 4s	2m 4s	2m 53s
No. of fine model samples per iteration	1	1	16
No. of iterations	227	3	7
Time for sensitivity estimation	-	-	2m 53s
Time for evaluating all fine models	$2.07\text{m} \times 227$	$2.07\text{m} \times 4$	$2.88\text{m} \times 8$
Time for surrogate model generation and optimization	0	-	$1\text{m } 2\text{s} \times 7$
No. of coarse models evaluations	-	328	-
Time for coarse model evaluations	-	$35\text{s} \times 328$	-
Total time	461.49m*	199.61m*	33.15m

* - Design specifications are not satisfied.

design variables proving the capability of the proposed design strategy.

Figure 3.7: The values of objective function for the inter-digital band-pass filter using: the proposed method (o) and coarse and fine mesh SM method (x).

3.3.2 Optimization of Four-Pole Waveguide Filter

This example shows a four-pole waveguide filter [73] with tuning elements as the posts of square cross section placed at the center of each cavity and each coupling window, as shown in Fig. 4.5. Height (h_1), height (h_2), and height (h_3) are the heights of the tuning posts in the coupling windows. Heights (h_{c1}), and (h_{c2}) are the heights of the square cross section placed in the center of the resonator cavities. The input and output waveguides, as well as the resonant cavities, are standard WR-75 waveguides ($a=19.05$ mm and $b=9.525$ mm). The thickness of the coupling windows is set to 2 mm. The design space vector $\mathbf{x} = [h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2}]^T$ for this example are chosen based on the sensitivity information.

Fine model evaluation is performed by HFSS EM simulator using the fast simulation feature. The desired filter specifications for the above structure are defined

Figure 3.8: Structure of a four-pole waveguide filter with optimization variables $\mathbf{x} = [h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2}]^T$

as a standard four-pole chebyshev curves of 300 MHz bandwidth, $|S_{11}| \leq -25 \text{ dB}$ and centered around 11 GHz. The starting point for this example is selected as the same starting point in the reference [73]. $\mathbf{x}^c = [3.3 \ 4.4 \ 4.0 \ 3.28 \ 2.91]^T$ (all values are in mm) is selected as the starting point. The neighborhood around the central point for each iteration X_R is defined using (3.3). δ for the first iteration is a user defined initial trust radius selected based on the sensitivity of the fine model data and is chosen as $[0.132 \ 0.176 \ 0.16 \ 0.0656 \ 0.0582]^T$ (all values in mm). Δ^{max} for the this example is chosen as $[0.132 \ 0.176 \ 0.16 \ 0.0656 \ 0.0582]^T$ (all values in mm) which is $[4 \ 4 \ 4 \ 2 \ 2]^T$ percent of the central point $\mathbf{x}^c$. Δ^{max} gives the relaxed value until the surrogate model no longer predicts model responses accurately. Typically, it is a user defined value and is selected based on the sensitivity of each design variable.

In this example, the optimization problem space has five design variables. The size of unknown weighting vector $\boldsymbol{\lambda}$ in (3.14) depends on the type of regression technique. For this example, using quadratic polynomial will result in $(1 + n +$

$n(n+1)/2$ unknown weighting vectors. Therefore, using regression techniques the size of weighting vector $\boldsymbol{\lambda}$ is calculated to be 21. For a good surrogate model, the number of data samples is selected to be greater than the size of weighting vector $\boldsymbol{\lambda}$. Using DOE [66] with five variables and five levels (subspace divisions), 25 different geometrical samples are obtained which are far fewer and feasible for parallel computation compared to full-grid distribution requiring 5125 samples (5^5). These 25 data samples are distributed on multiple computers with multiple processors and are evaluated in parallel to obtain 25 fine model responses.

The set of 25 fine model responses are used to generate a surrogate model valid in the region X_R using Steps 4 and 5 in the proposed algorithm. Vector fitting method of order 10 is used to get an accurate vector fitting for the 25 fine model responses. Later, using regression techniques Equation (3.11) is solved to estimate the weighting vectors $\boldsymbol{\lambda}$ for all the coefficients. Once the weighting vectors are determined, the surrogate model optimal point $\boldsymbol{x}^{new}$ is estimated using Step 6. Subsequently, using Steps 7 - 10 the next optimization update (prospective central point) and the new trust radius are calculated. The proposed optimization process is terminated if one of the conditions in (3.19) and (3.20) is satisfied.

Using the proposed technique, the final optimal solution $\boldsymbol{x}^* = [3.68227 \ 4.43935 \ 3.94449 \ 3.18003 \ 2.93364]^T$ (all values are in mm) is obtained after four iterations. Fig. 4.10(a) and Fig. 4.10(d) shows the fine model response at the initial central point (starting point) and final optimal solution respectively. Fig. 4.10(a) shows that the initial central point is a very good starting point for optimization. For comparison purposes, we also use coarse and fine mesh space mapping optimization

Figure 3.9: Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 144 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after four iterations) (d) Fine model response using proposed optimization (after four iterations).

method [30] to optimize this filter. Fig. 4.10(c) shows the final optimal point obtained after four iterations. For a further comparison, we performed direct EM optimization (quasi-Newton) using HFSS internal optimization feature. Fig. 4.10(b) shows the final optimal point is obtained after 144 fine model evaluations. Table 3.3 and Fig. 4.10 shows that direct EM optimization doesn't converge to the optimal solution whereas our proposed technique takes 21.23 min using 25 parallel processors (one processing core for one fine model evaluation) to reach an optimal point.

Table 3.3: Comparison of CPU Time for a Four-pole Waveguide Filter using a Good Starting Point

Optimization	Direct EM optimization using 8 cores	Coarse and fine mesh SM optimization	Proposed optimization
Fine model evaluation time for each sample	2m 5s	2m 5s	2m 49s
No. of fine model samples per iteration	1	1	25
No. of iterations	144	4	4
Time for sensitivity estimation	-	-	2m 49s
Time for evaluating all fine models	$2.083\text{m} \times 144$	$2.083\text{m} \times 5$	$2.82\text{m} \times 5$
Time for surrogate model generation and optimization	0	-	$1\text{m } 5\text{s} \times 4$
No. of coarse models evaluations	-	710	-
Time for coarse model evaluations	-	$27\text{s} \times 710$	-
Total time	300m*	329.91m*	21.23m

* - Design specifications are not satisfied.

The values of the one-sided l_2 objective function U for the proposed optimization and coarse and fine mesh SM optimization are shown in Fig. 4.12. Using coarse and fine mesh SM method, the value of objective function decreases for three iterations

Figure 3.10: The values of objective function for the four-pole waveguide filter using: the proposed method (o) and coarse and fine mesh SM method (x).

and on further optimization the value of objective function doesn't improve significantly. Fig. 4.12 shows that, the proposed optimization converges quickly to reach the termination condition in (3.20) (satisfies the design specifications) compared to coarse and fine mesh SM optimization.

In this example, we have 25 data samples (N_s), therefore we use 25 parallel processors (P) to generate 25 fine model responses. Using proposed parallel technique (3.4) the CPU time (T_k) for 25 fine model evaluations is 2.82min, whereas the CPU time for the sequential fine model evaluations is 70.5 min ($25 \times 2.82 \text{ min}$). The CPU time T_s for estimating the sensitivities of each design variable is 2.82 min. Surrogate model generation and optimization takes 1.083 min CPU time (T_p). The overhead CPU time T_{ov} (communication time between multiple processors) is considerably small. The speedup (S_p) factor of 15.69 is calculated using (3.6) which results in a

parallel efficiency (η) of about 62.75%. The proposed parallel technique takes much less CPU time compared to the conventional gradient based EM optimization to reach the optimal solution.

As a further experiment about the convergence of the proposed optimization for this filter example, we use a much worse starting point (bad starting point) $\mathbf{x}^c = [3.0 \ 4.1 \ 3.5 \ 3.3 \ 3]^T$ (all values are in mm) to re-run our proposed method. Using the same initial trust radius δ and Δ^{max} , the proposed method reaches the final optimal solution $\mathbf{x}^* = [3.51594 \ 4.222617 \ 3.71936 \ 3.25666 \ 2.96285]^T$ (all value in mm) after eight iterations. Fig. 3.11(a) and Fig. 3.11(d) shows the fine model response at the initial central point (starting point) and final optimal solution respectively. Fig. 3.11(a) shows that the initial point is much worse starting point than Fig. 4.10(a) for the optimization process.

For comparison purposes, we use coarse and fine mesh space mapping optimization method [30] to optimize this filter. Fig. 3.11(c) shows the final optimal point is obtained after two iterations. We also performed direct EM optimization of this filter using HFSS internal optimization feature (quasi-Newton) for a further comparison. Fig. 3.11(b) shows the final optimal point is obtained after 812 fine model evaluations. Table 3.4 and Fig. 3.11 shows that direct EM optimization doesn't converge to the optimal solution whereas our proposed technique takes 39.69 min using 25 parallel processors (one processing core for one fine model evaluation) to reach an optimal point. Fig. 3.12 shows values of the objective function U for the proposed optimization and coarse and fine mesh SM optimization. Using coarse and fine mesh SM method, the value of objective function decreases quickly for

Figure 3.11: Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 812 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after two iterations) (d) Fine model response using proposed optimization (after eight iterations).

two iterations and on further optimization the value of objective function doesn't improve significantly even after eight iterations. Fig. 3.12 shows that, the proposed optimization converges quickly to reach the termination condition in (3.20) (satisfies the design specifications) compared to coarse and fine mesh SM optimization. The speedup (S_p) factor of 16.9 is calculated using (3.6) which results in a parallel efficiency (η) of about 67.61%. Even with a much worse starting point the proposed technique reached the final optimal point, thereby showing the robustness

Table 3.4: Comparison of CPU Time for Four-pole Waveguide Filter using a Bad Starting Point

Optimization	Direct EM optimization using 8 cores	Coarse and fine mesh SM optimization	Proposed optimization
Fine model evaluation time for each sample	2m 5s	2m 5s	2m 30s
No. of fine model samples per iteration	1	1	25
No. of iterations	812	2	8
Time for sensitivity estimation	-	-	2m 49s
Time for evaluating all fine models	$2.083\text{m} \times 812$	$2.083\text{m} \times 3$	$2.82\text{m} \times 9$
Time for surrogate model generation and optimization	0	-	$1\text{m } 5\text{s} \times 8$
No. of coarse models evaluations	-	526	-
Time for coarse model evaluations	-	$27\text{s} \times 526$	-
Total time	28h 19m*	242.95m*	$36.86\text{m} + 2.82\text{m}^\perp = 39.69\text{m}$

* - Design specifications are not satisfied

$^\perp$ - unused fine model evaluation time in the proposed technique

Figure 3.12: The values of objective function for the four-pole waveguide filter using: the proposed method (o) and coarse and fine mesh SM method (x).

with respect to different starting points for the proposed technique.

To further examine the robustness of the algorithm with respect to the quality of the optimum solution, we consider the optimization solution obtained using classical quasi-Newton approach shown in the Fig. 3.11(b). The optimal point obtained using direct EM optimization $\mathbf{x}^{em} = [3.30572 \ 3.939 \ 3.49032 \ 3.33195 \ 2.9937]^T$ (all values are in mm) is a local minima at which the design specifications are still not satisfied. The value of the objective function at $\mathbf{x}^{em}$ is 39.028. On further optimization, the value of the objective function doesn't improve using existing gradient based approach in the EM simulators. To verify the optimization solution obtained using direct EM optimization $\mathbf{x}^{em}$ is a local minima, we perturb each design variable of $\mathbf{x}$ by 0.1% of $\mathbf{x}^{em}$ in both positive and negative directions. Table 3.5 shows that all the neighboring points obtained by perturbation have a higher U value than the U

Table 3.5: Confirming the final solution of the direct EM optimization is a local minima

	Value of objective function U	
Optimization solution of direct EM optimization $\mathbf{x}^{em}$	39.028	
	+ve direction	-ve direction
Perturb x_1 of $\mathbf{x}^{em}$	43.2310	43.9502
Perturb x_2 of $\mathbf{x}^{em}$	44.8211	43.7092
Perturb x_3 of $\mathbf{x}^{em}$	44.5872	44.3367
Perturb x_4 of $\mathbf{x}^{em}$	49.3292	45.6096
Perturb x_5 of $\mathbf{x}^{em}$	48.2208	48.2649

at $\mathbf{x}^{em}$. This confirms that $\mathbf{x}^{em}$ is a local minima. Therefore, classical quasi-Newton method cannot proceed further because the gradient of U increases even for a small step size of the next optimization update. Using the proposed method, we perform optimization using $\mathbf{x}^{em}$ as the starting point. The value of objective function decreases to 10.0457 after one iteration of the proposed method. The optimization update after first iteration $\mathbf{x}^{new}$ is $[3.40103 \ 4.04834 \ 3.51521 \ 3.29829 \ 2.97951]^T$. In order to demonstrate the fact the proposed method has a better possibility of avoiding being trapped in local minima we use a discrete parameter τ , where $0 \leq \tau \leq 1$. By varying τ , fine model responses are evaluated at the sample points $\mathbf{x}^{sample}$ generated using

$$\mathbf{x}^{sample} = \mathbf{x}^{em} + \tau.(\mathbf{x}^{new} - \mathbf{x}^{em}) \quad (3.21)$$

As shown in Fig. 3.13, the value of the objective function increases as τ increases

Figure 3.13: The value of objective function evaluated at each $\mathbf{x}^{sample}$ by changing τ . The value of objective function increases as τ increases in a small neighborhood ($\tau \leq 0.1$) which confirms $\mathbf{x}^{em}$ ($\tau = 0$) is a local minima. With large step size of the optimization update ($\tau = 1$), our proposed optimization can jump over the trap ($\tau = 0$) and find the new solution $\mathbf{x}^{new}$, where the value of the objective function is much lower than that at the local minima obtained using classical quasi-Newton method.

within a small neighborhood ($\tau \leq 0.1$). When τ is further increased, the value of the objective function jumps over the local minima trap, $\mathbf{x}^{em}$. This demonstrates the fact the proposed algorithm with large and effective optimization updates has a better chance of avoiding local minima traps than classical quasi-Newton which uses smaller optimization updates. This feature increases the robustness of our proposed algorithm compared to classical quasi-Newton method.

3.3.3 Optimization of a Dielectric Resonator Antenna

The third example is a dielectric resonator loaded antenna example [74]-[76] as shown in Fig. 3.14. The monopole antenna is loaded with dielectric resonator discs to increase the operating range of the monopole. The height of the monopole

Figure 3.14: Structure of a dielectric loaded monopole antenna with optimization variables $\mathbf{x} = [h_1 \ h_2 \ r_1 \ r_2 \ r_a]^T$

from the ground plane is fixed at 8.75 mm and therefore limits the lower end of the operating frequency range of the antenna. Height (h_1) and height (h_2) are the heights of the dielectric resonator loaded disc on the monopole. The dielectric resonators have inner radius (r_a) and outer radii (r_1), (r_2) for each disc respectively. The ground plane size is chosen to be 35 mm * 35 mm. The design space vector $\mathbf{x} = [h_1 \ h_2 \ r_1 \ r_2 \ r_a]^T$ for the example are chosen based on the sensitivity information.

Fine model evaluation is performed by HFSS EM simulator using interpolated simulation feature. Fast simulation feature cannot be used because of the wide frequency range. The desired antenna specifications for the above structure are defined as $|S_{11}| \leq -15 \text{ dB}$ in the frequency range of 7.5 GHz to 20.5 GHz. The initial central point is selected based on the values given in [76] while the values for the other set of parameters (fixed parameters such as ground plane size, length and radius of the monopole) are assigned using designer experience and knowledge. $\mathbf{x}^c = [2.6 \ 2.5 \ 4.4 \ 1.4 \ 3.0]^T$ (all values are in mm) is used as the starting point

for the optimization process. The neighborhood around the central point for each iteration X_R is defined in (3.3). δ for the first iteration is a user defined initial trust radius selected based on the sensitivity of the fine model data and is chosen as $[0.13 \ 0.125 \ 0.22 \ 0.07 \ 0.15]^T$ (all values in mm). Δ^{max} for this structure is chosen as $[0.52 \ 0.5 \ 0.88 \ 0.28 \ 0.6]^T$ (all values in mm) which is 20% of the central point $\mathbf{x}^c$. Δ^{max} gives the relaxed value until the surrogate model no longer predicts the model responses accurately. Typically, it is a user defined value and is selected based on the sensitivity of each design variable.

In this example, the optimization problem space has five design variables. Using regression techniques the size of weighting vector $\boldsymbol{\lambda}$ is calculated to be 21 (i.e., $1 + n + n(n + 1)/2$). For a good surrogate model, the number of data samples is selected to be greater than the size of weighting vector $\boldsymbol{\lambda}$. Using DOE [66] with five variables and five levels (subspace divisions), 25 different geometrical samples are obtained which are far fewer and feasible for parallel computation compared to full-grid distribution requiring 5125 samples (5^5). These 25 data samples are distributed on multiple computers (computer cluster) with eight processing cores for each computer to evaluate 25 fine model responses in parallel.

The set of 25 fine model responses are used to generate a surrogate model valid in the region X_R using Steps 4 and 5 in the proposed algorithm. Vector fitting method uses just enough number of poles so as to avoid numerical instability in (3.14) while estimating the weighting vectors $\boldsymbol{\lambda}$. For this example, 12 poles are sufficient to get an accurate vector fitting for the 25 fine model responses. Later, using regression techniques Equation (3.11) is solved to estimate the weighting vectors $\boldsymbol{\lambda}$ for all

Figure 3.15: Comparison of three different optimization methods (a) Fine model response at the starting point for all the three optimization methods (b) Fine model response using direct EM optimization (after 62 iterations) (c) Fine model response using coarse and fine mesh SM optimization (after 1 iteration) (d) Fine model response using proposed optimization (after 6 iterations).

the coefficients. Once the weighting vectors are determined, the surrogate model optimal point $\mathbf{x}^{new}$ is calculated by optimizing the surrogate model in the trust region X_R . Using Steps 7 -10 the next optimization update as well as the new trust radius are determined. The proposed optimization is terminated if one of the conditions in (3.19) and (3.20) is satisfied.

Using the proposed technique, the final optimal solution $\mathbf{x}^* = [2.42686 \ 2.33676 \ 4.808 \ 1.33145 \ 2.85884]^T$ (all values are in mm) is obtained after six iterations. Fig.

Figure 3.16: The values of objective function for the dielectric loaded monopole antenna using: the proposed method (o).

3.15(a) and Fig. 3.15(d) shows the fine model response at the initial central point (starting point) and final optimal solution respectively. The values of the objective function U for the proposed optimization are shown in Fig. 3.16. Fig. 3.16 shows that, the proposed optimization satisfies the design specifications in six iterations.

For comparison purposes, we use coarse and fine mesh space mapping optimization method [30] to optimize this filter. Fig. 3.15(c) shows the final optimal point is obtained after one iteration. We also performed direct EM optimization of the antenna using HFSS internal optimization feature (quasi-Newton optimization algorithm) for a further comparison. Fig. 3.15(b) shows the final optimal point is obtained after 62 fine model evaluations. Table 3.6 and Fig. 3.15 shows that direct EM optimization takes 16h 32min and coarse and fine mesh SM optimization method takes 323.99 min whereas our proposed technique takes 170.16 min using 25 parallel processors (one processing core for one fine model evaluation) to reach

Table 3.6: Comparison of CPU Time for Dielectric Resonator Antenna

Optimization	Direct EM optimization using 8 cores	Coarse and fine mesh SM Optimization	Proposed optimization
Fine model evaluation time for each sample	16m	16m	19m 25s
No. of fine model samples per iteration	1	1	25
No. of iterations	62	1	6
Time for sensitivity estimation	-	-	20m 20s
Time for evaluating all fine models	$16\text{m} \times 62$	$16\text{m} \times 2$	$20.33\text{m} \times 7$
Time for surrogate model generation and optimization	0	-	$1\text{m } 15\text{s} \times 6$
No. of coarse models evaluations	-	48	-
Time for coarse model evaluations	-	$6\text{m } 5\text{s} \times 48$	-
Total time	16h 32m	323.99m	170.16m

an optimal point using frequency interpolating simulation feature.

In this example, we have 25 data samples (N_s), therefore we use 25 parallel processors (P) to generate 25 fine model responses. Using proposed parallel technique (3.4) the CPU time (T_k) for 25 fine model evaluations is 19.42 min, whereas the

CPU time for the sequential fine model evaluations is 485.42 min ($25 \times 19.42min$). The CPU time T_s for estimating the sensitivities of each design variable is 19.42 min. Surrogate model generation and optimization takes 1.25 min CPU time (T_p). The overhead CPU time T_{ov} (communication time between multiple processors) is considerably small. The speedup (S_p) factor of 20.5 is calculated using (3.6) which results in a parallel efficiency (η) of about 81.98%. The proposed parallel technique deliberately increases the total number of fine model evaluations and results in a large and effective optimization updates. Also, it is observed from the three examples that the parallel efficiency of the proposed parallel technique increases dramatically when the fine model evaluations CPU time is much greater than surrogate model generation and optimization CPU time.

3.4 Conclusion

A trust-region based optimization technique has been proposed for microwave structures using parallel computational approach. The proposed method achieves speedup in the optimization process even if a coarse model is not available. Speedup in optimization has been achieved by using large and effective optimization updates in each iteration which resulted in fewer optimization iterations. In the proposed technique, we deliberately increase the number of fine model evaluations without increasing the computation time by using a parallel computational approach. This method has high parallel efficiency using fast simulation feature and interpolating simulation feature in available EM simulation tools. The proposed technique makes EM optimization more practical for microwave circuit design.

Chapter 4

Parallel Decomposition Approach for Gradient based EM Optimization

In this chapter, for the first time, a decomposition method is formulated to address the challenges of EM optimization with many design variables. Single large EM optimization is decomposed into multiple smaller sub-optimizations to improve the optimization efficiency. In practical EM optimizations there are no two design variables whose effect on the output response are completely independent. We propose a decomposition method which uses second order derivative information to measure the level of dependence (measure of interaction) between the design variables and thereby identify, separate and group the design variables to form multiple smaller sub-optimizations. Each sub-optimization includes generating multiple fine model responses for constructing the surrogate model and its optimization using trust region. Multiple sub-optimizations are formulated to be independent so that the parallel computations can be exploited to evaluate concurrent sub-optimization

updates. The resultant vector, a combined vector of sub-optimization updates, will be closer to the optimal solution. Furthermore, the optimization update of the proposed approach is much larger in comparison with the optimization update without decomposition. Using this approach, the overall large optimization update decreases the value of the objective function more efficiently and quickly. Two examples of EM optimization are used to illustrate the proposed technique.

4.1 Introduction

Electromagnetic (EM) based optimization and design closure is one of the most important steps in the design of passive microwave circuits. Recent advances in EM optimization aims at automated adjustment of design variables through gradient based numerical optimization. However, with the increasing complexity of microwave structures, the number of design variables becomes large. Consequently, it becomes more difficult to automatically adjust the geometrical parameters for an optimal design vector in a large search space due to the increased likelihood of stopping at an undesired local optimum using conventional gradient based methods. Other difficulties, such as sequential optimization updates and increased number of optimization iterations will result in many EM evaluations, thereby requiring a massive amount of CPU time to find the optimal design.

Decomposition is an important vehicle to solve large-scale problems. Topological decomposition of circuits [78][79] has been used to address the challenges in large-scale circuit simulations. For EM, structural domain decomposition methods (DDM) [80]-[83] are been used extensively to address EM simulation challenges

with complex geometries and large mesh problems. DDMs require accurate representation/mapping for the interface between the substructures for an internally decomposed EM structure. In [80], neural networks are used to represent the EM behavior at the interface between substructures (may or may not be EM components). In [81][83], modular neural networks are investigated for microwave filter design by incorporating filter equivalent circuit model as the interface for mapping the sub-models. However, these decomposition techniques do not directly alleviate the challenges of EM optimization with many design variables.

Recently, parallel computation has been researched with respect to surrogate based optimization. Parallel automatic model generation technique is proposed in [55], using parallel adaptive sampling and parallel data generation to save model development time. A parallel space mapping approach [58] is used for EM optimization, where the fine model evaluations and training of the surrogate model uses parallel computational techniques. However, the method in [58] requires the availability of a coarse model. When the coarse model is not available, parallel approach to gradient-based EM optimization is explored in [18]. This method uses a moderate number of design variables for EM optimization. A limitation with this technique is the scalability to EM optimization with many design variables. In search based optimization methods [84], the probability of generating a sample in the optimality region (a small volume of search space surrounding the optimal solution) decreases exponentially as the dimensionality of the search space increases. Therefore, it is significantly harder to find the optimum solution of an EM optimization problem with many design variables compared with a low-dimensional problem with similar

topology. Using conventional gradient based optimization methods, the increase in the volume of the search space will result in the increased likelihood of stopping at a undesired local minima.

With an increase in the number of design variables, the number of possible design variations that are needed to predict the EM behavior increases dramatically. For an accurate surrogate model the validity range of the surrogate model is decreased with an increase in the dimension of the search space (volume of search space). Further, the decrease in the surrogate model range will result in an increase in the number of optimization iterations for reaching the optimal solution using the method of [18].

Optimization decomposition techniques using equivalent circuits have been addressed in the literature [85][86] with manageable computational CPU time. However, EM simulations are computationally much more expensive than circuit simulations. Therefore, the same optimization decomposition techniques that are applied for equivalent circuits are not directly suitable. In this chapter, we investigate possible ways to formulate decomposition between EM design variables to exploit parallel computations. In this way, we address gradient based EM optimization and decomposition challenges with many design variables.

Reference [18] shows that, with fewer design variables the surrogate model can be built accurately in a relatively larger neighborhood for EM optimization. In this chapter, we propose a decomposition technique to split the design space vector into multiple sub-design vectors so that the sub-optimization can exploit the efficiency of optimization with fewer design variables. In EM simulations, the output

response is effected by many design variables. If the effect of one design variable on the output response is influenced by another design variable, then we refer such effect as the interaction between the two design variables. Furthermore, we call the design variables involved in such effects as interacting design variables. In this chapter, we propose a measure of interaction (eccentricity) between design variables for decomposition. The proposed decomposition method reduces the original EM optimization into several smaller sub-optimization problems which results in the increase of the trust region for sub-optimization and therefore leads to a larger optimization update. Subsequently, the large optimization updates will result in fewer iterations of optimization. Moreover, multiple sub-optimizations are formulated to be independent so that parallel computations can be exploited to evaluate concurrent sub-optimization updates. The resultant vector, a combined vector of sub-optimization updates, will be closer to the optimal solution. Using the proposed method, EM optimization with many design variables reaches the optimal solution efficiently and quickly. Two typical EM optimization examples are illustrated to validate the proposed decomposition and parallel gradient-based EM optimization.

4.2 Formulation of the Optimization Problem

4.2.1 Original Optimization Problem

Let $\mathbf{x}$ represents a vector of n design variables $[x_1 \ x_2 \ \cdots \ x_n]^T$, $\mathbf{x} \in \Re^n$. The fine model output response, denoted as $R_f(\mathbf{x}, \omega)$ is a function of design vector $\mathbf{x}$ and frequency ω . The original optimization problem is formulated as

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in X} U(R_f(\mathbf{x}, \omega)) \quad (4.1)$$

where $X \subseteq \mathbb{R}^n$ represents the design variable space and U is a given objective function. U represents the weighted error function of fine model response $R_f(\mathbf{x}, \omega)$ and the desired design specifications. $\mathbf{x}^*$ represents the optimal design vector satisfying the design specifications. Let S_{uq} and S_{lq} represent the set of upper and lower specifications, where q is the index for frequency. Let Q_u and Q_l represent the index sets of the frequency points for upper specification and lower specification respectively. The error functions for upper and lower specifications are defined as

$$\begin{aligned} e_{uq} &= W_u(R_f(\mathbf{x}, \omega_q) - S_{uq}), \quad q \in Q_u \\ e_{lq} &= W_l(S_{lq} - R_f(\mathbf{x}, \omega_q)), \quad q \in Q_l \end{aligned} \quad (4.2)$$

where W_u and W_l are the weighting factors for Q_u and Q_l respectively. ω_q represents the q th frequency sample. The objective function U defined in (4.1) may be either a minimax or a generalized l_p function [20] of the error functions defined in (4.2).

Let the set J contain the indices of the design variables in vector $\mathbf{x}$ and is defined as

$$J \triangleq \{1, 2, \dots, n\} \quad (4.3)$$

4.2.2 Sub-optimization Problem

In our EM optimization decomposition approach, we attempt to reach the overall solution by solving a set of sub-optimization problems iteratively. If the sub-

optimization problems are formulated as independent of other sub-problems then the overall solution is a combined vector of the optimal solutions from each sub-optimization problem. Furthermore, the independent sub-optimization problems can be executed in parallel to reach the overall optimal solution faster and more efficiently. For a n -dimensional problem in (4.1), a typical sub-optimization problem is defined as

$$\tilde{\mathbf{x}}^* = \arg \min_{\tilde{\mathbf{x}} \in \tilde{X}} U \left(\tilde{R}_f(\tilde{\mathbf{x}}, \omega) \right) \quad (4.4)$$

where $\tilde{\mathbf{x}}$ represents a sub design vector of $\mathbf{x}$, i.e., $\tilde{\mathbf{x}}$ contains the elements x_j , $j \in J^s$. Here, J^s is a subset of J . $\tilde{R}_f(\tilde{\mathbf{x}}, \omega)$ represents the fine model response $R_f(\mathbf{x}, \omega)$ as a function of only a subset of variables $\tilde{\mathbf{x}}$ while the other design parameters (x_j , $j \notin J^s$) remain fixed at $\mathbf{x}^c$, where $\mathbf{x}^c$ refers to a value of the design vector in the overall optimization before sub-optimization is performed. In other words, $\tilde{R}_f(\tilde{\mathbf{x}}, \omega)$ is the fine model for the sub-optimization problem. $\tilde{X}$ represents a subspace of X with the design variables in other subspaces fixed at a constant value.

For EM optimization with many design variables, the most important task for a good decomposition method is to identify and exploit the interaction patterns amongst the design variables x_j , $j \in J$.

4.3 Proposed Decomposition Approach to EM Optimization

Decomposition of design variables is employed in many large-scale global optimization problems. The recently developed and popularly used random grouping (RG) [87], delta grouping (DG) [88], differential grouping [89] methods are used extensively in numerous meta-heuristic algorithms such as cooperative co-evolutionary algorithms. Random grouping is based on the equal probability that the strongly interacting design variables are grouped in one sub-problem. However, there is no systematic way of identifying the interacting variables in [87]. Delta grouping in [88] is based on the magnitude of change in the dependent design variable for one generation during the evolution of sub-problems. The magnitude of change is used to identify the strongly interacting design variables after every generation. This method is less efficient when the optimization problem has more interacting variables [88]. In [89], a differential grouping, an automatic way of decomposing an optimization problem into a set of few or no inter-dependencies between the sub-problems was proposed. However, using differential grouping in EM optimization approach, the optimum solution is found usually at the expense of substantially more computational time compared to EM based gradient approach. In this chapter, we propose an decomposition approach to gradient based EM optimization using second order derivative information to identify, segregate and group the design variables.

Figure 4.1: Graphical representation of the proposed EM optimization decomposition approach using two design variables with eccentricity (measure of interaction between design variables) equal to zero, i.e., $B_4 = 0$ in (4.5). In this case, the two variables x_1 and x_2 can be optimized independently.

4.3.1 Ideal Case of Decomposition Approach

As a special case for the illustration of decomposition, let us consider $R_f(\mathbf{x}, \omega)$ as a function of two design variables, i.e., $\mathbf{x} = [x_1 \ x_2]^T$. Using second order Taylor series expansion of $R_f(\mathbf{x}, \omega)$ at a fixed frequency ω_0 , $R_f(\mathbf{x}, \omega_0)$ is expressed as

$$\begin{aligned}
 R_f(\mathbf{x}, \omega_0) = & B_0 + B_1x_1 + B_2x_2 + B_3x_1^2 \\
 & + B_4x_1x_2 + B_5x_2^2
 \end{aligned} \tag{4.5}$$

where $\{B_1, B_2\}$ and $\{B_3, B_4, B_5\}$ represent the first order and second order partial derivatives with respect the design variables $\mathbf{x}$. In Equation (4.5), B_4 represents the eccentricity (measure of interaction) between the design variables x_1 and x_2 . In an ideal scenario, $B_4 = 0$, i.e., $\frac{\partial^2 R_f(\mathbf{x}, \omega_0)}{\partial x_1 \partial x_2} = 0$, then the two design variables x_1 and x_2 are independent and separable. Fig. 4.1 shows the contours of $R_f(\mathbf{x}, \omega_0)$ as a function of design variables $\mathbf{x}$ with zero eccentricity. Let $\mathbf{x}^c = [x_1^c \ x_2^c]^T$ represent the design vector at the point O in Fig. 4.1 (starting point of the optimization process). Since, design variables x_1 and x_2 are separable, the original optimization problem in (4.1) is divided into two independent sub-optimization problems. Sub-optimization of each sub-problem with the other design variable preset to a fixed value, will result in the sub-optimal solutions x_1^* and x_2^* in their respective design spaces. The sub-optimization problems are defined as

$$x_1^* = \arg \min_{x_1 \in \tilde{X}^{(1)}} U \left(\tilde{R}_f(x_1, \omega_0) \right) \quad (4.6)$$

$$x_2^* = \arg \min_{x_2 \in \tilde{X}^{(2)}} U \left(\tilde{R}_f(x_2, \omega_0) \right) \quad (4.7)$$

where $\tilde{X}^{(1)}$ represents a sub-domain of X with x_1 as design variable and x_2 fixed at x_2^c . Similarly, $\tilde{X}^{(2)}$ represents a sub-domain of X with x_2 as design variable and x_1 fixed at x_1^c respectively. Fig. 4.1 shows a graphical representation of the proposed approach to reach the overall optimal solution $\mathbf{x}^*$ using a decomposition method.

The sub-optimal solutions x_1^* and x_2^* are estimated by concurrent optimization of two independent sub-optimization problems. Since, $B_4 = 0$, the overall solution $\mathbf{x}^*$ is a vector obtained by concatenating the two sub-optimal solutions as shown by the dashed line in Fig. 4.1.

$$\mathbf{x}^* = [x_1^* \ x_2^*]^T \quad (4.8)$$

4.3.2 Proposed Decomposition Technique using Second Order Derivative Information

An automatic decomposition approach to large-scale microwave systems was proposed in [85] wherein the relation between the design variables with the error function groups are explored. However, the method in [85] is applied to equivalent circuit models for decomposition wherein the decoupling is between the circuit error functions and the design variables. In the present chapter, for the first time, we propose EM optimization decomposition approach wherein the EM design variables have non-zero eccentricity with respect to the common objective function defined in (4.1). In our proposed work, a systematic way to decouple the design variables with respect to a single objective function U is proposed. For EM optimization decomposition with many design variables, it is important to identify and exploit the interaction patterns and the level of interaction amongst the design variables.

To identify the level of interaction, firstly, we perform the sensitivity analysis of the fine model response with respect to the design variables. Sensitivity information can be obtained either directly through derivatives that are available from EM

simulator or indirectly using a finite difference method [90]. The derivative of fine model response with respect to the design variable is represented as

$$S_{q,a} = \frac{\partial R_f(\mathbf{x}, \omega_q)}{\partial x_a} \quad (4.9)$$

where a may take any value in the set J , i.e., $a \in J$. Sensitivity is evaluated at each frequency sample q , where $q = 1, 2, \dots, N_f$, and N_f represents the number of frequency samples. To study the interaction among the design variables, we use the second order derivative information. Second order derivatives, which are not available explicitly from the EM simulators, are computed using perturbation techniques such as finite difference method [90]

$$\begin{aligned} \frac{\partial S_{q,a}}{\partial x_b} &= \frac{\partial^2 R_f(\mathbf{x}, \omega_q)}{\partial x_a \partial x_b} \\ &\approx \frac{S_{q,a}|_{\mathbf{x}=\mathbf{x}^+} - S_{q,a}|_{\mathbf{x}=\mathbf{x}^-}}{2h} \end{aligned} \quad (4.10)$$

where a and b may take any value in the set J , i.e., $a \in J$ and $b \in J$. $\mathbf{x}^+$ and $\mathbf{x}^-$ are the resultant design vectors from the perturbation of the design variable x_b by a finite step h in positive and negative directions respectively. The second order derivative in (4.10) is computed using the first order derivatives provided from EM simulators. The examination of various interaction patterns using second order derivatives in (4.10) will result in the breakdown of the design vectors into multiple sub design vectors with index sets $J_1, J_2, \dots, J_L$, where L represents the number

of groups. The index set J of the original design variables $\mathbf{x}$ is related to sub-index sets as

$$J = J_1 \cup J_2 \cup J_3 \cdots \cup J_L \quad (4.11)$$

In EM optimization problems, there are no two design variables that are completely independent with respect to a single objective function. Therefore, the eccentricity (measure of interaction between design variables, e.g., B_4 in (4.5)) is not exactly equal to zero. Non-zero eccentricity is a major property of a non-separable problems. However, a relatively low value of eccentricity will lead to the identification of the weakly interacting design variables. A relatively small value of interaction indicates that the design variables can be segregated into groups so as to optimize them separately. We use this approach to segregate the weakly interacting design variables into separate groups for subsequent sub-optimizations.

4.3.3 Identification of the Strongly Interacting Design Variables and Formation of Sub Design Vectors

Let d_i represent the number of design variables in the sub design vector of the i th group. The number of groups L that are possible using decomposition can be related to the number of design variables n . To identify the possible number of design variables in each group, we follow a set of initial criteria that must be satisfied while selecting the number of groups L and the number of design variables d_i in each group. The initial criteria are listed as follows

- 1: With many interacting design variables, if the decomposition results in too many groups (smaller d_i per group) then we may be ignoring many interactions between design variables, thereby degrading the performance of the optimization algorithm. If the problem is decomposed into too few groups, then a large value of d_i doesn't utilize the power of decomposition approach to its full potential. Since the EM optimization problem at hand is usually not an ideally separable problem, it is essential to maintain the strongly interacting design variables together. Therefore, we set the maximum number of groups to be less than $\sqrt{n}$ and the minimum number of groups as two, i.e., $2 \leq L < \sqrt{n}$. For an EM optimization with many interacting design variables, the initial number of groups L_{ini} is selected as $L_{ini} = \lfloor \sqrt{n-1} \rfloor$. For a completely separable problem, L is equal to n , which results in the optimal solution as the combined vector of optimal solution in each dimension.
- 2: Using the initial number of groups L_{ini} , the minimum and maximum number of design variables per group are calculated as

$$\begin{aligned}
d_{min} &= \left\lfloor \frac{n}{L_{ini}} \right\rfloor \\
d_{max} &= n - d_{min}(L_{ini} - 1)
\end{aligned} \tag{4.12}$$

where d_{min} is used to identify the minimum number of design variables per group based on the L_{ini} , and d_{max} ensures that the strongly interacting design variables (high value of eccentricity) are intact. If $d_{max} = d_{min}$, we set $d_{max} = d_{min} + 1$. This condition provides the flexibility in grouping strongly interacting

design variables even when the number of design variables is a multiple of L_{ini} .

Once, the initial values of L_{ini} , d_{min} and d_{max} are determined, we can iteratively identify the design variables that are to be grouped from the pool of n design variables. To do so, various interaction patterns are examined using a $n \times n$ matrix $\mathbf{C}$ whose (a, b) th component is calculated as

$$C_{ab} = \sum_{q=1}^{N_f} \left\| \frac{\partial S_{q,a}}{\partial x_b} \right\| = \sum_{q=1}^{N_f} \left\| \frac{\partial^2 R_f(\mathbf{x}, \omega_q)}{\partial x_a \partial x_b} \right\| \quad (4.13)$$

where a and b represent the row and the column indices of the matrix $\mathbf{C}$ respectively. $\mathbf{C}$ matrix represents the summation of the second order derivatives at all frequency points (N_f frequency points) for n design variables. Once, the second order derivative information is determined, we identify the design variables with strong interactions and group them together into one sub-problem so that the design variables corresponding to high eccentricity stay intact.

The first step, in the decomposition process is to identify and group the design variables with strongest interaction. Let J_i represent the set of indices of the i th group and $\tilde{\mathbf{x}}^{(i)}$ represent the i th sub design vector. $\tilde{\mathbf{x}}^{(i)}$ contains the design variables $\tilde{x}_j^{(i)}$, $j \in J_i$, where J_i is determined by an iterative process. Initially, we set all the groups $J_1, J_2, \dots, J_i, \dots, J_{L_{ini}}$ to be null sets and their corresponding sub design vectors $\tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \dots, \tilde{\mathbf{x}}^{(L_{ini})}$ to be null vectors. Since, $\mathbf{C}$ is a symmetrical matrix, the upper triangular matrix of $\mathbf{C}$ is sufficient to identify the interaction between the design variables. To form an i th group, the strongest interaction between the

design variables is identified as

$$C_{kl} = \begin{cases} \max_{\substack{1 \leq a \leq n \\ a < b \leq n}} C_{ab}, & a \notin J_i \vee b \notin J_i \end{cases} \quad (4.14)$$

where a and b may take any value from the complete set J . The symbol $\vee$ represents a logical OR operator and the symbol $\wedge$ represents a logical AND operator in the subsequent equations. C_{kl} represents a strong measure of interaction (eccentricity) between k th design variable and l th design variable. Therefore, the sub design vector is updated as

$$\tilde{\mathbf{x}}^{(i)} = \begin{cases} \begin{bmatrix} \tilde{\mathbf{x}}^{(i)} \\ x_k \\ x_l \end{bmatrix}, & \text{if } k \notin J_i \wedge l \notin J_i \wedge (d_i < d_{min}) \\ \begin{bmatrix} \tilde{\mathbf{x}}^{(i)} \\ x_k \end{bmatrix}, & \text{if } k \notin J_i \wedge l \in J_i \wedge (d_i < d_{max}) \\ \begin{bmatrix} \tilde{\mathbf{x}}^{(i)} \\ x_l \end{bmatrix}, & \text{if } k \in J_i \wedge l \notin J_i \wedge (d_i < d_{max}) \end{cases} \quad (4.15)$$

In Equation (4.15), $\tilde{\mathbf{x}}^{(i)}$ in the right hand side of (4.15) is updated by concatenating with either x_k or x_l or both. Further, J_i is updated with the k th and l th

design variables as follows

$$J_i = \begin{cases} J_i \cup \{k\} \cup \{l\}, & \text{if } k \notin J_i \wedge l \notin J_i \wedge (d_i < d_{min}) \\ J_i \cup \{k\}, & \text{if } k \notin J_i \wedge l \in J_i \wedge (d_i < d_{max}) \\ J_i \cup \{l\}, & \text{if } k \in J_i \wedge l \notin J_i \wedge (d_i < d_{max}) \end{cases} \quad (4.16)$$

Equations (4.14) - (4.16) are repeated iteratively until the number of design variables in the i th sub design vector is greater than or equal to d_{min} , i.e., $d_i \geq d_{min}$. If this condition is satisfied, we check the condition $k \in J_i$ or $l \in J_i$. If either $k \in J_i$ or $l \in J_i$ is satisfied, then the corresponding k th or l th design variable is strongly interacting with one of the design variables in J_i . Therefore, the k th or the l th design variable will be grouped into the same i th group provided the number of design variables in the i th group doesn't exceed d_{max} . If neither the k th nor the l th design variable are interacting with any of the design variables in J_i , then we stop the formation of the i th group. Also, if d_i reaches maximum possible number of design variables ($d_i \geq d_{max}$), then we stop the formation of the i th group. Once the group J_i is formed, update the $\mathbf{C}$ matrix by replacing the entire rows and columns whose indices belong to J_i with zeros. The new $\mathbf{C}$ matrix is used to identify the design variables that are to be grouped together in the next group J_{i+1} . This process is repeated until there is no more further decomposition possible based on termination condition

$$2 < n_{rem} \leq (d_{max} + 2) \quad (4.17)$$

where n_{rem} represents the number of remaining design variables to be grouped which is calculated after the formation of the i th group as

$$n_{rem} = n - \sum_{t=1}^i d_t \quad (4.18)$$

where d_t represents the number of design variables in the t th group. The termination condition in (4.17) ensures, that the last group has at least three design variables.

Once the termination condition for the decomposition process is satisfied, the total number of groups L is equal to $i + 1$, i.e., $L = i + 1$. The last group J_L is formed with the indices of the remaining design variables

$$J_L = J - \left\{ \bigcup_{t=1}^i J_t \right\} \quad (4.19)$$

The last sub design vector $\tilde{\mathbf{x}}^{(L)}$ is formed with the remaining design variables as $\tilde{x}_j^{(L)}$, for all $j \in J_L$. The total number of groups L may not be necessarily equal to L_{ini} . Fig. 4.2 shows a complete flowchart of the decomposition method.

In this decomposition method, we use static grouping, i.e., the number of groups as well as the number of design variables in the group are constant throughout the optimization process. The balance between the number of groups and the number of design variables in each group is dependent on the value of eccentricity at which the original problem is divided into sub-problems. A large number of groups usually corresponds to a high value of eccentricity between sub-problems, therefore this

Figure 4.2: Flow chart showing the proposed decomposition process. Through this process, the entire design vector $\mathbf{x}$ is sub-divided into multiple sub design vectors, $\tilde{\mathbf{x}}^{(i)} \forall i = 1, 2, \dots, L$.

decomposition will be at the expense of ignoring many interactions between the sub-problems, which may degrade the performance of the entire optimization process. On the contrary, if we use too few groups, the advantage of using decomposition approach is lost. In order to fully utilize the power of the decomposition approach we

use d_{min} and d_{max} to divide the original vector into multiple sub design vectors. In the proposed method, the decomposition starts with the grouping of design variables with strongest interaction, followed by the next strongest and so-on until a suitable balance between the number of groups and the number of design variables per group is reached.

4.4 Proposed Parallel EM Optimization using Sub Design Vectors

The proposed parallel EM optimization technique is sub-divided into various sections such as decomposition technique to identify, segregate and group the design variables to form multiple sub design vectors, initial trust radius computation for each sub-optimization problem, surrogate modeling over multiple EM samples for each sub-problem, sub-optimizations and trust radius computation for sub-problems, crossover, and overall optimization update using trust region framework.

4.4.1 Trust Radius for the i th Sub-optimization Problem

Before sub-optimization, let the value of the design vector for the overall optimization be denoted as $\mathbf{x}^c$, which is referred to as the central point of the overall optimization for the subsequent descriptions. In the first iteration, $\mathbf{x}^c$ is the initial point of the overall optimization. For the i th sub-optimization problem, let $\tilde{\mathbf{x}}^{(i)}$ represent the sub design vector, where $\tilde{\mathbf{x}}^{(i)}$ is a subset of $\mathbf{x}$. The remaining design variables, $x_j, j \notin J_i$, remain fixed at x_j^c during the i th sub-optimization process. Let $\tilde{\mathbf{x}}^{c,(i)}$, which is a subset of $\mathbf{x}^c$, represent the central point for the i th sub-optimization

problem. Around the neighborhood of this central point $\tilde{\mathbf{x}}^{c,(i)}$, we define the region of interest $\tilde{X}^{(i)}$, also known as trust region for the i th sub-optimization problem,

$$\tilde{X}^{(i)} = \{ \tilde{\mathbf{x}}^{(i)} \mid \tilde{\mathbf{x}}^{c,(i)} - \boldsymbol{\delta}^{(i)} \leq \tilde{\mathbf{x}}^{(i)} \leq \tilde{\mathbf{x}}^{c,(i)} + \boldsymbol{\delta}^{(i)} \} \quad (4.20)$$

where $\boldsymbol{\delta}^{(i)}$ is called the trust radius. $\boldsymbol{\delta}^{(i)}$ determines the interval range of the i th group and is calculated in the first iteration by a user defined variable $\psi^{(i)}$. For the later stages of optimization, $\boldsymbol{\delta}^{(i)}$ is updated using trust region framework.

For the first iteration, $\tilde{\mathbf{x}}^{c,(i)}$ represents the initial central point of the i th sub-optimization. Let $\boldsymbol{\phi}^{(i)}$ represent a vector, whose elements are the maximum value of absolute sensitivity of $\tilde{R}_f(\tilde{\mathbf{x}}^{(i)}, \omega)$ with respect to x_j , $j \in J_i$ evaluated at the initial central point $\tilde{\mathbf{x}}^{c,(i)}$, and the maximum value of absolute sensitivity is selected among that of all frequencies. The initial trust radius $\boldsymbol{\delta}^{(i)}$ is calculated as

$$\boldsymbol{\delta}^{(i)} = \psi^{(i)} ./ \boldsymbol{\phi}^{(i)} \quad (4.21)$$

where the operator $./$ denotes element-wise division of a vector, i.e., the division is applied to each and every element of the vector. $\psi^{(i)}$ is a user defined value for calculating the initial trust radius.

4.4.2 Surrogate Modeling over Multiple Samples in the i th Sub-Optimization

Once the trust radius $\delta^{(i)}$ for the i th sub-problem is determined, design of experiments (DOE) [66] sampling strategy is used to generate multiple sample points around the central point $\tilde{\mathbf{x}}^{c,(i)}$ in the trust region $\tilde{X}^{(i)}$. Let $\tilde{\mathbf{x}}^{m,(i)}$ denote one such sample in the trust region $\tilde{X}^{(i)}$ for the i th group, where m represents the sample number in the i th sub-problem, $m = 1, 2, \dots, N_s^{(i)}$, and $N_s^{(i)}$ represents the total number of sample points selected using DOE in the i th sub-optimization problem.

Let $\tilde{R}_s^{(i)}(\tilde{\mathbf{x}}^{(i)}, \omega)$ represent the i th surrogate model defined as a function of only the sub design vector $\tilde{\mathbf{x}}^{(i)}$. A good surrogate model match (w.r.t. fine model response) is obtained at the DOE samples points in the trust region by minimizing the error between the fine model response and surrogate model response [18] as

$$\boldsymbol{\lambda}^{*,(i)} = \arg \min_{\boldsymbol{\lambda}^{(i)}} \sum_{m=1}^{N_s^{(i)}} \sum_{\omega \in \Omega} \left\| \tilde{R}_s^{(i)}(\tilde{\mathbf{x}}^{m,(i)}, \boldsymbol{\lambda}^{(i)}, \omega) - \tilde{R}_f(\tilde{\mathbf{x}}^{m,(i)}, \omega) \right\| \quad (4.22)$$

where Ω represents the complete set of frequency points used for the fine model response. $\boldsymbol{\lambda}^{(i)}$ represents the internal weighting vectors in the surrogate model. In (4.22), the surrogate model response is expressed as a function of pole/residues and frequency. Further, the pole/residues are expressed as a quadratic function of sub design vector $\tilde{\mathbf{x}}^{(i)}$. Regression techniques are used to obtain the optimal internal weighting vectors $\boldsymbol{\lambda}^{*,(i)}$ in the quadratic function of the i th sub-problem such that the error between fine model response and the surrogate model response is minimized.

4.4.3 Optimization of the i th Sub-Problem and Trust Region Framework

Let $\tilde{\mathbf{x}}^{\mu,(i)}$ represent the sub-optimization solution of the i th group. Using trust region framework, the i th sub-optimization, defined in (4.4), in the trust region $\tilde{X}^{(i)}$ will result in a sub-optimization solution $\tilde{\mathbf{x}}^{\mu,(i)}$,

$$\tilde{\mathbf{x}}^{\mu,(i)} = \arg \min_{\tilde{\mathbf{x}}^{(i)} \in \tilde{X}^{(i)}} U \left(\tilde{R}_s^{(i)}(\tilde{\mathbf{x}}^{(i)}, \omega) \right) \quad (4.23)$$

where $\tilde{X}^{(i)}$, defined in (4.20), is calculated using trust radius $\delta^{(i)}$ and is controlled by trust region framework [18].

Let $\delta^{\mu,(i)}$ represent the prospective trust radius update for the next iteration. This update $\delta^{\mu,(i)}$ is determined using the following conditions,

$$\delta^{\mu,(i)} = \begin{cases} c_1 \delta^{(i)}, & \text{if } r_a^{(i)} < 0.1 \\ \min(c_2 \delta^{(i)}, \Delta^{\max,(i)}), & \text{if } r_a^{(i)} > 0.75 \\ \delta^{(i)}, & \text{Otherwise} \end{cases} \quad (4.24)$$

where $\Delta^{\max,(i)}$ sets the maximum limit for each sub design variable in the i th sub-optimization problem. Expansion and contraction of trust radius in the i th group depend on the values of c_1 and c_2 in (4.24). In the present chapter, we use $c_1 = 0.69$ and $c_2 = 1.3$. $r_a^{(i)}$ is a control index parameter, defined as

$$r_a^{(i)} = \frac{U\left(\tilde{R}_f(\tilde{\mathbf{x}}^{c,(i)}, \omega)\right) - U\left(\tilde{R}_f(\tilde{\mathbf{x}}^{\mu,(i)}, \omega)\right)}{U\left(\tilde{R}_f(\tilde{\mathbf{x}}^{c,(i)}, \omega)\right) - U\left(\tilde{R}_s^{(i)}(\tilde{\mathbf{x}}^{\mu,(i)}, \omega)\right)} \quad (4.25)$$

The prospective trust radius $\delta^{\mu,(i)}$ is later on used to determine the trust radius update. Convergence of the i th sub-optimization is ensured by embedding the entire sub-optimization process discussed in section 4.4.2 and 4.4.3 into the trust region framework [18]. However, unlike [18] the trust radius update for the next iteration of the overall optimization is dependent on the prospective trust radius of the sub-problem as well as the overall optimization update.

4.4.4 Parallel Computation over Multiple Sub-Problems

The proposed decomposition technique discussed earlier ensures that the sub-problems are formed with minimum interactions. For such decomposed sub-problems, parallel computational approach is well suited to further accelerate data generation (for surrogate model training) of multiple sub-problems. In other words, we consider the generation of multiple fine model responses using multiple EM samples for multiple sub-problems simultaneously using multiple computers with multiple processors. EM data generation for multiple sub-problems in parallel is given by

$$\left\{ \left\{ \tilde{R}_f(\tilde{\mathbf{x}}^{m,(i)}, \omega) \mid m = 1, 2, \dots, N_s^{(i)} \right\} \mid i = 1, 2, \dots, L \right\} \quad (4.26)$$

where $\tilde{R}_f(\tilde{\mathbf{x}}^{m,(i)}, \omega)$ represents the m th EM sample in the i th sub-optimization. $N_s^{(i)}$

fine model responses for all sub-problems, i.e., $i = 1, 2, \dots, L$, are generated in parallel.

Various blocks of the sub-optimization problem such as surrogate model construction and sub-optimization solution using trust region framework are performed in parallel. Surrogate model construction involves parallel fine model evaluations given in (4.26) and optimization of the internal weighting vectors $\boldsymbol{\lambda}^{(i)}$ to obtain a good match between the fine model response and the surrogate model given in (4.22) for all $i = 1, 2, \dots, L$. The surrogate model responses in the sub-optimization problems defined as in (4.23) are simultaneously optimized using multiple processors to obtain multiple sub-optimization solutions,

$$\{\tilde{\boldsymbol{x}}^{\mu,(i)} | i = 1, 2, \dots, L\} = \{\tilde{\boldsymbol{x}}^{\mu,(1)}, \tilde{\boldsymbol{x}}^{\mu,(2)}, \dots, \tilde{\boldsymbol{x}}^{\mu,(L)}\} \quad (4.27)$$

4.4.5 Crossover and Overall Optimization Update using Multiple Sub-optimization Solutions

In a gradient based EM optimization of microwave circuits with many design variables, the interaction amongst the design variables cannot be completely ignored. Using the proposed decomposition approach, we prune the interaction between the sub-problems based on the relative value of eccentricity. However, the small value of eccentricity still may have a non-negligible effect on the performance of the resultant optimal solution vector. Cooperative Particle Swarm Optimization (CPSO) [84] decomposes the n -dimensional problem into multiple sub-dimensional problems with a fixed dimensions throughout the optimization process. The sub-optimal solutions

reached after each iteration are concatenated with each other, called as context vector in [84] and then the context vector is used to evaluate the objective function. However, in EM based problems all the design variables are interacting and therefore concatenating the sub-optimization solutions to form a single optimal vector may not always be the optimal solution. In this chapter, we propose a more diverse approach for concatenation of sub design vectors from different sub-optimizations using a combinatorial method (crossover). Our proposed crossover allows us to directly jump to a new prospective update of $\mathbf{x}$ without iteratively searching for it. This crossover is achieved by combining the sub-optimization solution of one subspace with the sub-optimization solutions of one or more other subspaces. Crossover is an interesting technique to cover the entire optimization space effectively and efficiently. By using this approach, the overall optimization update has a better chance of decreasing the value of the objective function efficiently and quickly.

For ease of understanding, let us consider the case where the overall optimization problem is divided into three sub-optimization problems. Therefore, the index set J is decomposed into three groups J_1 , J_2 and J_3 , i.e., $J = J_1 \cup J_2 \cup J_3$. Let $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ represent the sub design vectors of J_1 , J_2 and J_3 respectively after the decomposition technique described in section 4.3.2 and 4.3.3. For convenience of description, we use $\mathbf{z}$ to represent the prospective updates of $\mathbf{x}$ (intermediate solutions of $\mathbf{x}$) during

the overall optimization process. The vector $\mathbf{z}$ is represented as

$$\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{x}}^{(1)} \\ \tilde{\mathbf{x}}^{(2)} \\ \tilde{\mathbf{x}}^{(3)} \end{bmatrix} \quad (4.28)$$

The actual update (optimization update) in the previous iteration of the overall optimization process is used as the central point for the current iteration. Using the proposed decomposition technique, the central point in the current iteration is divided into three sub design vectors $\mathbf{u}_0$, $\mathbf{v}_0$ and $\mathbf{w}_0$. In the current iteration, $\mathbf{u}_0$, $\mathbf{v}_0$ and $\mathbf{w}_0$ refer to the central point of the sub-design vectors $\tilde{\mathbf{x}}^{(1)}$, $\tilde{\mathbf{x}}^{(2)}$, and $\tilde{\mathbf{x}}^{(3)}$ respectively. Therefore, the central point in the current iteration of the overall optimization is represented as

$$\mathbf{z}_{000} = \begin{bmatrix} \tilde{\mathbf{x}}^{c,(1)} \\ \tilde{\mathbf{x}}^{c,(2)} \\ \tilde{\mathbf{x}}^{c,(3)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \\ \mathbf{w}_0 \end{bmatrix} \quad (4.29)$$

For sub-optimization of group 1, $\mathbf{u}$ is the design vector with the central point as $\mathbf{u}_0$, and the values of the sub design vectors $\mathbf{v}$ and $\mathbf{w}$ remain fixed at $\mathbf{v}_0$ and $\mathbf{w}_0$ (unchanged). Similarly, for sub-optimization of group 2 and group 3, the design vectors are $\mathbf{v}$ and $\mathbf{w}$ respectively, while the other sub design vectors remain unchanged. Furthermore, sub-optimization of group 1, group 2 and group 3 will result in the sub-optimization solutions $\mathbf{u}_1$, $\mathbf{v}_1$ and $\mathbf{w}_1$ respectively. The sub-optimization solution along with the central point of all other sub-optimizations is referred as the prospective updates for $\mathbf{z}_{000}$. Therefore, sub-optimization of three groups will

lead to three prospective updates $\mathbf{z}_{100}$, $\mathbf{z}_{010}$ and $\mathbf{z}_{001}$ in the current iteration of the overall optimization, defined as

$$\mathbf{z}_{100} = \begin{bmatrix} \tilde{\mathbf{x}}^{\mu,(1)} \\ \tilde{\mathbf{x}}^{c,(2)} \\ \tilde{\mathbf{x}}^{c,(3)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{v}_0 \\ \mathbf{w}_0 \end{bmatrix} \quad (4.30)$$

$$\mathbf{z}_{010} = \begin{bmatrix} \tilde{\mathbf{x}}^{c,(1)} \\ \tilde{\mathbf{x}}^{\mu,(2)} \\ \tilde{\mathbf{x}}^{c,(3)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{v}_1 \\ \mathbf{w}_0 \end{bmatrix} \quad (4.31)$$

$$\mathbf{z}_{001} = \begin{bmatrix} \tilde{\mathbf{x}}^{c,(1)} \\ \tilde{\mathbf{x}}^{c,(2)} \\ \tilde{\mathbf{x}}^{\mu,(3)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{v}_0 \\ \mathbf{w}_1 \end{bmatrix} \quad (4.32)$$

The prospective updates $\mathbf{z}_{100}$ (or $\mathbf{z}_{010}$ or $\mathbf{z}_{001}$) is the projection of sub-optimization solution $\tilde{\mathbf{x}}^{\mu,(1)}$ (or $\tilde{\mathbf{x}}^{\mu,(2)}$ or $\tilde{\mathbf{x}}^{\mu,(3)}$) on to the design space vector $\mathbf{x}$. For convenience of ongoing description, we call these prospective updates, i.e., $\mathbf{z}_{100}$, $\mathbf{z}_{010}$, and $\mathbf{z}_{001}$ as the base update vectors (base vectors in short). These base vectors form the basis to produce new prospective updates by crossover mechanism. Fig. 4.3 shows the crossover mechanism using the base vectors from the three groups. Crossover replaces the central points of one or more sub-optimizations in a base vector with

Table 4.1: Representation of the Design Vectors $\mathbf{z}$ for the Three Groups in Each Iteration of the Overall Optimization

Type		Design vector $\mathbf{z}$
Central point or overall optimization update from previous iteration ($\mathbf{z}_{000}$)		$[\mathbf{u}_0^T \mathbf{v}_0^T \mathbf{w}_0^T]^T$
Prospective updates of $\mathbf{x}$	Base update vectors ($\mathbf{z}_{100}, \mathbf{z}_{010}, \mathbf{z}_{001}$)	$[\mathbf{u}_1^T \mathbf{v}_0^T \mathbf{w}_0^T]^T, [\mathbf{u}_0^T \mathbf{v}_1^T \mathbf{w}_0^T]^T, [\mathbf{u}_0^T \mathbf{v}_0^T \mathbf{w}_1^T]^T$
	Crossover update vectors ($\mathbf{z}_{011}, \mathbf{z}_{101}, \mathbf{z}_{110}, \mathbf{z}_{111}$)	$[\mathbf{u}_1^T \mathbf{v}_1^T \mathbf{w}_0^T]^T, [\mathbf{u}_1^T \mathbf{v}_0^T \mathbf{w}_1^T]^T, [\mathbf{u}_0^T \mathbf{v}_1^T \mathbf{w}_1^T]^T, [\mathbf{u}_1^T \mathbf{v}_1^T \mathbf{w}_1^T]^T$

the solutions of other sub-optimizations to form a crossover update vector (crossover vector in short) of $\mathbf{z}$. The number of crossover vectors are given by $2^L - (L + 1)$. The crossover update vectors of $\mathbf{z}_{000}$ are represented as $\mathbf{z}_{110}, \mathbf{z}_{101}, \mathbf{z}_{011}$ and $\mathbf{z}_{111}$. For example $\mathbf{z}_{110}$ is given as

$$\mathbf{z}_{110} = \begin{bmatrix} \tilde{\mathbf{x}}^{\mu,(1)} \\ \tilde{\mathbf{x}}^{\mu,(2)} \\ \tilde{\mathbf{x}}^{c,(3)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{v}_1 \\ \mathbf{w}_0 \end{bmatrix} \quad (4.33)$$

Table 4.1 and Fig. 4.3(a) shows the prospective updates of $\mathbf{x}$ obtained for the current iteration of the overall optimization using three groups. The four crossover update vectors (hashed circles) $\mathbf{z}_{110}, \mathbf{z}_{101}, \mathbf{z}_{011}$ and $\mathbf{z}_{111}$ are used to evaluate the fine

Figure 4.3: Crossover mechanism to estimate the next overall optimization update for three groups. (a) Central point (big solid circle), base vector (small solid circle), and crossover vector (hashed circle). Crossover approach showing the formation of crossover vectors (hashed circles) (u_1, v_1, w_0) , (u_1, v_0, w_1) , (u_0, v_1, w_1) , and (u_1, v_1, w_1) using the base vectors (small solid circles) (u_1, v_0, w_0) , (u_0, v_1, w_0) and (u_0, v_0, w_1) . (b) Resultant prospective design vectors for the next optimization update.

model responses $R_f(z, \omega)$ in parallel. Fig. 4.3(b) shows the resultant prospective design vectors for the overall optimization update. The best solution z^{new} is selected as

$$z^{new} = \arg \min_{z \in Z} U(R_f(z, \omega)) \quad (4.34)$$

where Z is a set of all possible crossover update vectors and base update vectors, defined as

$$Z = \{z_{001}, z_{010}, z_{100}, z_{011}, z_{101}, z_{110}, z_{111}\} \quad (4.35)$$

In (4.34), U refers to a given objective function, e.g., minimax and is defined in

(4.1). $\mathbf{z}^{new}$ represents the design vector with the minimum value of the objective function U among all the prospective updates in the set $\mathbf{Z}$ defined in (4.35).

The optimization update $\mathbf{x}^{new}$ is given by the best solution $\mathbf{z}^{new}$ found in (4.34) as

$$\mathbf{x}^{new} = \mathbf{z}^{new} \quad (4.36)$$

To further clarify, sub-optimization solutions from some of the sub-problems may be a part of $\mathbf{z}^{new}$ ($\tilde{\mathbf{x}}^{\mu,(i)} \subset \mathbf{z}^{new}$). Sub-optimization solutions for some of the other sub-problems may not be a part of $\mathbf{z}^{new}$ ($\tilde{\mathbf{x}}^{\mu,(i)} \not\subset \mathbf{z}^{new}$). The trust radius update $\delta^{new,(i)}$ for the i th sub-optimization depends on the prospective trust radius $\delta^{\mu,(i)}$ in (4.24) as well as the overall optimization update $\mathbf{x}^{new}$. If the sub-optimization solution ($\tilde{\mathbf{x}}^{\mu,(i)}$) is a part of the design vector $\mathbf{z}^{new}$, then the trust radius update for the i th sub-problem is equal to the prospective trust radius of the i th sub-optimization. Otherwise, the trust radius of the i th sub-optimization remains unchanged for the next iteration. This condition ensures the number of unused fine model evaluations for the i th sub-problem in the later stages of optimization process is minimized. More specifically, the trust radius update $\delta^{new,(i)}$ for the i th group is calculated as

$$\delta^{new,(i)} = \begin{cases} \delta^{\mu,(i)}, & \text{if } \tilde{\mathbf{x}}^{\mu,(i)} \subset \mathbf{z}^{new} \\ \delta^{(i)}, & \text{if } \tilde{\mathbf{x}}^{\mu,(i)} \not\subset \mathbf{z}^{new} \end{cases} \quad (4.37)$$

The entire process of parallel computation of sub-optimizations and their subsequent evaluation of the optimization update represent one iteration of the overall optimization in our proposed method, i.e., DOE based data generation and (4.22) - (4.37). The algorithm is terminated if one of the conditions in (4.38) or (4.39) is sat-

isfied, i.e., the absolute difference between subsequent iteration points is sufficiently small or if the desired design specifications are satisfied:

$$\|\mathbf{x}^c - \mathbf{x}^{new}\| \leq \varepsilon \quad (4.38)$$

$$U(R_f(\mathbf{x}^{new}, \omega)) \leq 0 \quad (4.39)$$

where ε is a user defined value (in the order of 10^{-4}).

If the termination conditions are not satisfied, then the optimization update $\mathbf{x}^{new}$ from the current iteration will be the new central point $\mathbf{x}^c$ for the next iteration, i.e.,

$$\mathbf{x}^c = \mathbf{x}^{new} \quad (4.40)$$

Subsequently, the central point of the sub design vector $\tilde{\mathbf{x}}^{c,(i)}$ for i th sub-optimization problem in the next iteration will be a subset of this new central point. The trust radius of the i th sub-optimization is updated as $\delta^{(i)} = \delta^{new,(i)}$. The overall optimization proceeds to the next iteration by repeating the above process starting from section 4.4.2. This iterative process continues until the termination condition in (4.38) or (4.39) is satisfied.

Convergence of the proposed EM optimization decomposition approach is guaranteed by the independent trust region frameworks embedded in the concurrent sub-optimizations. The value of objective function decreases (converges) after the sub-optimization process. Therefore, the value of objective function for the resul-

tant base update vectors that are given in $\mathbf{Z}$ are less than the value of the objective function of the central point $\mathbf{z}_{000}$ for each iteration of the overall optimization. Therefore, the value of the objective function corresponding to the best solution obtained using (4.34) always converges (doesn't increase).

Fig. 4.4 shows a complete flowchart of our proposed optimization. The proposed algorithm is summarized below

Figure 4.4: Flow chart showing the proposed EM optimization decomposition process.

- Step 1: Set initial central point $\mathbf{x}^c$ and initial number of groups L_{ini} . For all i , $i = 1, 2, \dots, L_{ini}$, J_i is set to null set and $\tilde{\mathbf{x}}^i$ is set to null vector.
- Step 2: Compute $\mathbf{C}$ matrix in (4.13), d_{min} and d_{max} using L_{ini} in (4.12) which sets the limits for minimum and maximum number of sub design variables per group.
- Step 3: Form the group J_i and the sub design vector $\tilde{\mathbf{x}}^{(i)}$ using (4.14), (4.15), and (4.16) until $d_i \geq d_{min}$. If the condition is satisfied, then go to Step 4. Otherwise concatenate the group J_i using (4.16) and repeat Step 3.
- Step 4: Check the condition $k \notin J_i \wedge l \notin J_i$. If the condition is satisfied then go to Step 5. Otherwise, the corresponding k th or l th design variable are grouped in the same group J_i using (4.14), (4.15), and (4.16) provided the condition $d_i < d_{max}$ is satisfied. If the condition $d_i < d_{max}$ is not satisfied then go to Step 5.
- Step 5: Check if the termination condition for the decomposition process given in (4.17) is satisfied or not. If the condition is satisfied, stop the decomposition process and go to Step 7. Otherwise, go to Step 6.
- Step 6: Set $i = i + 1$ and update the $\mathbf{C}$ matrix by replacing the entire rows and columns whose indices belong to J_i with zeros. Go to Step 3 and repeat the process to form the $i + 1$ group.
- Step 7: The last group is formed using (4.19). Total number of groups L is calculated as $L = i + 1$. The L groups are formed with the indices $J_1, J_2, \dots, J_L$, and the corresponding design vectors are given by $\tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \dots, \tilde{\mathbf{x}}^{(L)}$.

- Step 8: Execute the Steps 9 - 12 for all sub-problems, i.e., for all i , $i = 1, 2, \dots, L$, in parallel.
- Step 9: For a given $\mathbf{x}^c$, calculate the trust region around the neighborhood of the central point $\tilde{\mathbf{x}}^{c,(i)}$ in the i th sub-optimization using the trust radius $\boldsymbol{\delta}^{(i)}$ as shown in (4.20).
- Step 10: Evaluate EM fine model responses $\tilde{R}_f(\tilde{\mathbf{x}}^{m,(i)}, \omega)$ at all the DOE sample points ($m = 1, 2, \dots, N_s^{(i)}$) generated around the central point $\tilde{\mathbf{x}}^{c,(i)}$ using parallel computations.
- Step 11: Construct a pole-residue surrogate model $R_s^{(i)}(\tilde{\mathbf{x}}^{(i)}, \omega)$ using regression techniques whose internal weighting vectors $\boldsymbol{\lambda}^{(i)}$ are calculated using (4.22).
- Step 12: Solve and find the sub-optimization solution $\tilde{\mathbf{x}}^{\mu,(i)}$ using (4.23), and the prospective trust radius $\boldsymbol{\delta}^{\mu,(i)}$ using (4.24) for the i th sub-problem.
- Step 13: Use solutions of various sub-problems to find the overall optimization update $\mathbf{x}^{new}$ using (4.36) and the trust radius update $\boldsymbol{\delta}^{new,(i)}$ using (4.37) for all i , $i = 1, 2, \dots, L$.
- Step 14: If one of the termination conditions in (4.38) or (4.39) is satisfied then stop the optimization process.
- Step 15: Set $\mathbf{x}^c = \mathbf{x}^{new}$ and $\boldsymbol{\delta}^{(i)} = \boldsymbol{\delta}^{new,(i)}$ for all $i = 1, 2, \dots, L$ for the next iteration and go to Step 8.

4.4.6 Discussion

This section gives a detailed interpretation of the vector $\mathbf{z}$ (used earlier in Equations (4.28) - (4.37)) relative to vector $\mathbf{x}$. Both vectors $\mathbf{z}$ and $\mathbf{x}$ are of same dimensions, i.e., $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^n$. Vector $\mathbf{z}$ contains the same elements as vector $\mathbf{x}$ except that the sequence of elements may be different between $\mathbf{z}$ and $\mathbf{x}$. The sequence of elements in $\mathbf{x}$ is user defined. The sequence of elements of $\mathbf{z}$ is defined such that it starts with all variables in sub-optimization one, followed by all variables in sub-optimization two and so-on. In other words, $\mathbf{z}$ is formed by concatenating all the sub design vectors. Therefore, by changing the sequence of design variables, we can obtain the vector $\mathbf{x}$ from vector $\mathbf{z}$ and vice versa.

4.5 Application Examples

4.5.1 Optimization of Four-Pole Waveguide Filter

This example shows a four-pole waveguide filter [73] with tuning elements as the posts of square cross section placed at the center of each cavity and each coupling window, as shown in Fig. 4.5. h_1 , h_2 , and h_3 are the heights of the tuning posts in the coupling windows. h_{c1} and h_{c2} are the heights of the square cross section placed in the center of the resonator cavities. w_c and w_p are the widths of the square cross section placed at the center of the resonant cavity and the coupling window respectively. d_{12} and d_{23} represent the diameters of resonant cavities. The input and output waveguides, as well as the resonant cavities, are standard WR-75 waveguides ($a = 19.05$ mm and $b = 9.525$ mm). The design space vector $\mathbf{x} =$

$[d_{12} \ d_{23} \ h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2} \ w_p \ w_c]^T$ for this example are chosen based on the sensitivity information.

Figure 4.5: Structure of a four-pole waveguide filter with optimization variables $\mathbf{x} = [d_{12} \ d_{23} \ h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2} \ w_p \ w_c]^T$

Fine model evaluation is performed by HFSS [93] EM simulator. The desired filter specifications for the above structure are defined as a standard four-pole Chebyshev curves of 300 MHz bandwidth, $|S_{11}| \leq -25$ dB and centered around 11 GHz. $\mathbf{x}^c = [10.5 \ 13.3 \ 3.0 \ 4.1 \ 3.5 \ 3.3 \ 3 \ 2.0 \ 4.0]^T$ (all values are in mm) is selected as the initial central point (starting point) for the optimization process. The starting point for this example is selected as the same starting point in reference [73]. The vector $\mathbf{x}$ has nine design variables ($n = 9$). In order to solve the optimization problem in (4.1), we decompose the original optimization problem into multiple sub-optimization problems defined in (4.4). The first step in the proposed optimization is to identify, segregate and group the interacting design variables to form multiple sub-optimization problems. To do so, we examine various interaction patterns among the design variables $\mathbf{x}$ using the second order derivatives in (4.10). First

order derivatives are obtained directly from the HFSS simulator. However, second order derivatives are not available explicitly from HFSS. Therefore, we use perturbation techniques to generate 18 ($2n$) fine model responses and their derivatives to extract second order derivatives w.r.t. the design variables. For this example, 18 ($2n$) fine model responses using discrete simulation feature are generated using parallel processors simultaneously. Now, we extract the second order derivatives using the finite difference method. Using this method, the total time required to obtain the second order derivative information for all design variables is equivalent to one fine model evaluation time. To examine the interaction patterns, $\mathbf{C}$ matrix is formed using (4.13). Since, $\mathbf{C}$ is symmetrical matrix, the upper triangular matrix of normalized $\mathbf{C}$ is sufficient and is calculated as

$$\mathbf{C} = \begin{pmatrix} -14.3 & 5.8 & 5.4 & 1.5 & 41 & 44.1 & 17.8 & 14.9 \\ - & - & 6.0 & 14.4 & 5.1 & 46.4 & \underline{127.2} & 33.7 & 32.3 \\ - & - & - & 4.4 & 1.5 & 28.1 & 38 & 14 & 11.5 \\ - & - & - & - & 1.4 & 20.6 & 39.6 & 13.5 & 11.5 \\ - & - & - & - & - & 14.2 & 25.7 & 8.5 & 7.4 \\ - & - & - & - & - & - & \underline{162.6} & 65.3 & 53.6 \\ - & - & - & - & - & - & - & \underline{122.0} & \underline{114.6} \\ - & - & - & - & - & - & - & - & 69.7 \\ - & - & - & - & - & - & - & - & - \end{pmatrix}$$

Using the proposed decomposition technique, the initial number of groups (L_{ini}) for this example is chosen to be two ($2 \leq L_{ini} < \sqrt{n}$). d_{min} and d_{max} are calculated

using (4.12) as four and five respectively. The decomposition algorithm is used to identify and group the design variables. In the $\mathbf{C}$ matrix, the strongest value of eccentricity is found as 162.6 using (4.14). Therefore, the sixth design variable h_{c1} and the seventh design variable h_{c2} are grouped together using (4.15) and (4.16). The number of elements in J_1 is less than four (d_{min}). Therefore, the process is repeated until the condition ($d_1 \geq 4$) is satisfied. At this point $J_1 = \{6, 7, 2, 8\}$, the next strongest element 114.6 is extracted using (4.14) and corresponds to the sixth design variable and the ninth design variable. However, the sixth design variable exists in group J_1 . Therefore, we group the ninth design variable in the same group J_1 using (4.15) and (4.16) which means that the strongly interactive design variables are to stay intact in the same group. Once, $d_i \geq d_{max}$, the formation of the group $J_1 = \{6, 7, 2, 8, 9\}$ along with the sub design vectors $\tilde{\mathbf{x}}^{(1)}$ is complete. The underline elements in the matrix $\mathbf{C}$ are used to represent the group J_1 . After group J_1 is formed, we check the the termination condition in (4.17). If the termination condition is satisfied, then we form the last group and stop the decomposition process. Using (4.19) the last group is formed as $J_2 = \{1, 3, 4, 5\}$. The total number of groups in this example is $L = L_{ini} = 2$. Using the decomposition algorithm strongly interacting design variables which includes d_{23} , h_{c1} , h_{c2} , w_p and w_c are grouped together. The weakly interacting design variables are d_{12} , h_1 , h_2 , and h_3 . The sub design vector $\tilde{\mathbf{x}}^{(1)} = [d_{23} \ h_{c1} \ h_{c2} \ w_c \ w_p]^T$ represents the first group, sub design vector $\tilde{\mathbf{x}}^{(2)} = [d_{12} \ h_1 \ h_2 \ h_3]^T$ represents the second group and vector $\mathbf{z} = [(\tilde{\mathbf{x}}^{(1)})^T \ (\tilde{\mathbf{x}}^{(2)})^T]^T$.

Sub-optimization of the first group involves the optimization of the sub design

vector $\tilde{\mathbf{x}}^{(1)}$ with the remaining design variables, i.e., the design variables in $\tilde{\mathbf{x}}^{(2)}$, fixed at a value in $\mathbf{x}^c$. The initial central point for the first sub-optimization problem is $\tilde{\mathbf{x}}^{c,(1)} = [13.3 \ 3.3 \ 3.0 \ 2.0 \ 4.0]^T$ (all values in mm) which is a subset of $\mathbf{x}^c$. The neighborhood $\tilde{X}^{(1)}$ around the central point $\tilde{\mathbf{x}}^{c,(1)}$ is selected using (4.20). For the first iteration, the initial trust radius $\delta^{(1)}$ is calculated using (4.21) and is equal to $[0.2893 \ 0.0580 \ 0.0290 \ 0.0648 \ 0.0972]^T$ (all values in mm) with $\psi^{(1)} = 3$. $\Delta^{max,(1)}$ for this example is chosen as $[0.2893 \ 0.0580 \ 0.0290 \ 0.0648 \ 0.0972]^T$ (all values in mm). Similarly, sub-optimization of the second group involves the optimization of the sub design vector $\tilde{\mathbf{x}}^{(2)}$ with the remaining design variables, i.e., the design variables in $\tilde{\mathbf{x}}^{(1)}$, fixed at the value in $\mathbf{x}^c$. The initial central point for second sub-optimization problem is $\tilde{\mathbf{x}}^{c,(2)} = [10.5 \ 3.0 \ 4.1 \ 3.5]^T$ (all values in mm) which is a subset of $\mathbf{x}^c$. The initial trust radius $\delta^{(2)}$ is calculated using (4.21) as $[0.3938 \ 0.1227 \ 0.1086 \ 0.2100]^T$ (all values in mm) with $\psi^{(2)} = 6$. $\Delta^{max,(2)}$ for this example is chosen as $[0.5250 \ 0.1635 \ 0.1447 \ 0.2800]^T$ (all values in mm).

Table 4.2 shows that the validity range of the surrogate model, i.e., $\tilde{X}^{(i)}$ with the decomposition method is much larger than the validity range of the surrogate model without decomposition [18]. The validity range expressed as trust radius is computed by trust region algorithm. This increase in the validity range of the surrogate model (trust radius as the size of deviation from the central point for each iteration) will result in larger step size for the optimization update in comparison with the optimization update obtained without decomposition [18]. Table 4.2 also shows that dividing the design vector into sub design vectors allows for an even larger trust radius for the least sensitive design variables.

Table 4.2: Comparison of the Validity Range (Expressed as Trust Radius) of the Surrogate Model with Decomposition and without Decomposition for a Four-pole Waveguide Filter

Optimization method	Optimization method without decomposition [18]	Optimization method with decomposition	
		Sub-optimization 1	Sub-optimization 2
Design variables	$\mathbf{x} = [d_{12} \ d_{23} \ h_1 \ h_2 \ h_3 \ h_{c1} \ h_{c2} \ w_p \ w_c]^T$	$\tilde{\mathbf{x}}^{(1)} = [d_{23} \ h_{c1} \ h_{c2} \ w_p \ w_c]^T$	$\tilde{\mathbf{x}}^{(2)} = [d_{12} \ h_1 \ h_2 \ h_3]^T$
Trust radius (deviation from central point) %	$[0.5 \ 0.2 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.2 \ 0.5 \ 0.5]^T$	$[2.2 \ 1.8 \ 0.9 \ 3.0 \ 3.2]^T$	$[5.0 \ 5.4 \ 3.5 \ 8.0]^T$
Surrogate model training error	1.14%	1.11 %	0.43%
Surrogate model test error	1.48%	1.32%	0.78%

In this example, the sub-optimization problems has five and four design variables in first group and second group respectively. The size of internal weighting vectors $\boldsymbol{\lambda}^{(i)}$ of the surrogate model in (4.22) varies with the sub-problem and depends on the type of regression technique. Using a simple quadratic polynomial will result in 21 $(1 + n + n(n + 1)/2)$ and 16 internal weighting variables for the first and the second groups respectively. Using DOE [66] for each sub-problem, 25 different geometrical samples are obtained. The two sub-optimizations are distributed

on multiple computers with multiple processors such that the overall workload is balanced. Further, 25 data samples in each sub-problem are again distributed in parallel on multiple processors (two level parallel approach). We use a cluster of Dell PowerEdge computers with Intel Xeon X5680 processors. Each computer has eight processor cores.

The set of 25 fine model responses are used to generate a pole-residue surrogate model valid in the region $\tilde{X}^{(i)}$. Once the internal weighting vectors are determined, the i th sub-problem is optimized using NeuroModeler-plus program [92] in the trust region $\tilde{X}^{(i)}$ using (4.23) to obtain the sub-optimization solution $\tilde{\mathbf{x}}^{\mu,(i)}$. Once, the sub-optimization solutions for each group are determined, the set $\mathbf{Z}$ similar to (4.35) is determined. For this example, two base update vectors $\mathbf{z}_{10}$ and $\mathbf{z}_{01}$ are determined using the two sub-optimization solutions $\tilde{\mathbf{x}}^{\mu,(1)}$ and $\tilde{\mathbf{x}}^{\mu,(2)}$. Since, we have two groups the crossover mechanism will lead to one crossover update vector $\mathbf{z}_{11}$. Using (4.34), the best solution $\mathbf{z}^{new}$ is calculated and is set to be the next optimization update $\mathbf{x}^{new}$ of the overall optimization process. Fig. 4.6 shows the fine model responses and value of the objective function of the EM samples in the set $\mathbf{Z}$ after the first iteration of the optimization process. Using (4.34), the best solution $\mathbf{z}^{new}$ is found as $\mathbf{z}_{11}$ which has the least value of objective function 123.33, as shown in Fig. 4.6. $\mathbf{z}^{new}$ is set to be the next optimization update $\mathbf{x}^{new}$ of the overall optimization process. The above process is repeated until one of the termination conditions in (4.38) or (4.39) is satisfied. If the termination conditions are not satisfied, the central point for the next iteration $\mathbf{x}^c$ is updated with $\mathbf{x}^{new}$, and the trust radius $\delta^{(i)}$ for the next iteration is updated with $\delta^{new,(i)}$ calculated using (4.37).

Figure 4.6: Comparison of fine model responses and the objective function values of the prospective updates in the first iteration of the proposed optimization method for a four-pole waveguide filter. (a) Fine model response and U of the base vector $\mathbf{z}_{10}$ evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$. (b) Fine model response and U of the base vector $\mathbf{z}_{01}$ evaluated using $\tilde{\mathbf{x}}^{\mu,(2)}$. (c) Fine model response and U of the crossover vector $\mathbf{z}_{11}$ evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$ and $\tilde{\mathbf{x}}^{\mu,(2)}$. As seen in the figure, $\mathbf{z}_{11}$ has the least value of objective function of 123.33 and is selected as the best solution $\mathbf{z}^{new}$ among $\mathbf{Z} = \{\mathbf{z}_{10}, \mathbf{z}_{01}, \mathbf{z}_{11}\}$ for the first iteration.

Using the proposed technique, the final optimal solution $\mathbf{x}^* = [10.4005 \ 13.2362 \ 3.48163 \ 4.1764 \ 3.66747 \ 3.3129 \ 3.0066 \ 3.90751 \ 2.0134]^T$ (all values are in mm) is obtained after six iterations. Fig. 4.7(a) and Fig. 4.7(e) shows the fine model response at the initial central point (starting point) and final optimal solution respectively. For comparison purposes, we also use coarse and fine mesh space mapping optimization method [30] to optimize this filter. Fig. 4.7(c) shows the final optimal point obtained after two iterations. Further, we compare the proposed optimization method with decomposition and the optimization method without decomposition [18]. Fig. 4.7(d) shows that final solution obtained by the optimization method without decomposition [18] satisfies the design specifications. However, due to a smaller trust region of the surrogate model, the optimization method without de-

composition [18] takes 14 iterations (more iterations) to reach the optimal solution. As a further comparison, we perform direct EM optimization using HFSS internal optimization feature. For convenience we refer the HFSS internal optimization as direct EM optimization method. Pattern search and quasi-Newton methods have been used for direct EM optimization. The best result from direct EM optimization has been used for comparison with other optimization methods. Fig. 4.7(b) shows the final optimal point is obtained after 763 fine model evaluations using direct EM optimization. Table 4.4 and Fig. 4.7 shows that direct EM optimization still violates design specification after 26.48 hours of optimization with no further improvement whereas our proposed EM optimization decomposition approach takes 54.21 min to reach an optimal point.

Table 4.3: Comparison of Fine Model Evaluations Using the Optimization Method With and Without Decomposition for a Four-pole Waveguide Filter

Optimization method	Optimization without decomposition [18]	Optimization with decomposition
No. of groups L	1	2
No. of fine model evaluations per iteration	81	$(25 \times 2)^{\dagger} + 1^{\perp}$
No. of iterations	14	6
Final value of objective function	-0.09	-0.033

$^{\perp}$ - No. of fine model evaluations due to crossover mechanism in each iteration of the optimization process.

† - Executed in parallel which is equivalent to the CPU time of one fine model evaluation.

Figure 4.7: Comparison of four different optimization methods for a four-pole waveguide filter. (a) Fine model response at the starting point for all the four optimization methods. (b) Fine model response using direct EM optimization (after 763 iterations). (c) Fine model response using coarse and fine mesh SM optimization (after 2 iterations). (d) Fine model response using optimization method without decomposition method [18] (after 14 iterations). (e) Fine model response using proposed optimization with decomposition method (after 6 iterations).

Table 4.4: Comparison of CPU Time for Optimization of a Four-pole Waveguide Filter

Optimization method	Direct EM opt. using 8 cores	Coarse & fine mesh SM opt.	Opt. method without decomposition [18]	Proposed method with decomposition
Time for derivative calculations	-	-	2m 49s (1st order)	17m 22s (1st and 2nd order)
Fine model evaluation time	2m 5s	2m 5s	2m 49s	2m 49s
Evaluation time for all fine models	2.083m $\times 763$	2.083m $\times (2+1)$	2.82m $\times (14 + 1)$	2.82m $\times 6 + 2.82m \times 6^\perp$
Time for modeling and optimization	-	-	1.08s $\times 14$	25s $\times 6$
Number of coarse model evaluations	-	580	-	-
Time for coarse model evaluations	-	27s $\times 580$	-	-
Total time	26.48h*	267.2m*	60.24m + 5.64m $^\oplus$ = 65.88m	54.21m

$^\perp$ - CPU time for fine model evaluations of crossover update vectors.

* - Design specifications are not satisfied at the solution.

$^\oplus$ - CPU time for unused fine model evaluations.

The values of the objective function U for the proposed optimization with decomposition and coarse and fine mesh SM optimization are shown in Fig. 4.8(a). Using coarse and fine mesh SM method, the value of objective function decreases for two iterations and on further optimization the value of objective function doesn't improve significantly. Further, we compare the value of objective function for the proposed optimization method with decomposition and the optimization method without decomposition [18]. Fig. 4.8(b) shows that the optimization method without decomposition satisfies the design specifications after 14 iterations. The slow decrease in the value of the objective function is due to the decrease in the surrogate model range with an increase in the number of design variables. Fig. 4.8(a) and Fig. 4.8(b) shows that, the proposed optimization with decomposition converges faster (to reach the termination condition in (4.39) and to satisfy the design specifications) than the coarse and fine mesh SM optimization as well as the optimization method without decomposition [18].

(a)

(b)

Figure 4.8: (a) The values of objective function for the four-pole waveguide filter using: the proposed method with decomposition (o) and coarse and fine mesh SM method (x). (b) The values of objective function for the four-pole waveguide filter using: optimization method with decomposition (o) and optimization method without decomposition [18] (x).

4.5.2 Optimization of Iris Coupled Microwave Cavity Filter

Consider an iris coupled cavity filter [77] as shown in Fig. 4.9. r_p and r_c represent the tuning post radius and the radius of the cylinders positioned at the center of each cavity. w_a and w_b are the widths of the metal that separates the cavities as shown in Fig. 4.9. Heights h_{c1} , h_{c2} , and h_{c3} are the heights of the cylinders positioned at the center of each cavity. The required coupling bandwidths are accomplished via the iris widths w_1 , w_2 , w_3 and w_4 for pre-tuning. The design space vector $\mathbf{x} = [h_{c1} \ h_{c2} \ h_{c3} \ r_c \ r_p \ w_1 \ w_2 \ w_3 \ w_4 \ w_a \ w_b]^T$ for this example are chosen based on the sensitivity information.

Figure 4.9: Structure of an iris coupled microwave cavity filter with optimization variables $\mathbf{x} = [h_{c1} \ h_{c2} \ h_{c3} \ r_c \ r_p \ w_1 \ w_2 \ w_3 \ w_4 \ w_a \ w_b]^T$

The desired filter specifications for the above structure are defined as $|S_{11}| \leq -20$ dB in a frequency range 703-713 MHz and $|S_{21}| \leq -15$ dB at frequency range of 690-701 MHz and 715-720 MHz. $\mathbf{x}^c = [42.8 \ 50.3 \ 50.4 \ 35.0 \ 6.0 \ 115.3 \ 51.3 \ 46.9 \ 51.0 \ 6.0 \ 12.0]^T$

(all values are in mm) is selected as the initial central point (starting point) [77] for the optimization process. The vector $\mathbf{x}$ has eleven design variables ($n = 11$). Using the decomposition method discussed earlier in section 4.3, matrix $\mathbf{C}$ is given as

$$\mathbf{C} = \begin{pmatrix} - & 101 & 134 & 207 & 15 & 21 & 13 & 7 & 5 & \underbrace{76} & 7 \\ - & - & 271 & \underline{407} & 21 & 46 & 43 & 37 & 20 & 120 & 15 \\ - & - & - & \underline{1721} & 53 & 54 & 48 & 40 & 30 & 206 & 35 \\ - & - & - & - & 9 & 13 & 17 & 23 & 14 & 53 & 12 \\ - & - & - & - & - & \underbrace{116} & \underbrace{78} & 26 & 19 & \underbrace{286} & 28 \\ - & - & - & - & - & - & 1 & 1 & 1 & 4 & 1 \\ - & - & - & - & - & - & - & 2 & 1 & 11 & 2 \\ - & - & - & - & - & - & - & - & 3 & 12 & 3 \\ - & - & - & - & - & - & - & - & - & 38 & 6 \\ - & - & - & - & - & - & - & - & - & - & 12 \\ - & - & - & - & - & - & - & - & - & - & - \end{pmatrix}$$

Using the proposed decomposition technique, the initial number of groups (L_{ini}) for this example is chosen to be three ($2 \leq L_{ini} < \sqrt{n}$). d_{min} and d_{max} are calculated using (4.12) as three and five respectively based on L_{ini} . In the $\mathbf{C}$ matrix, the strongest value of eccentricity is found as 1721 using (4.14). Therefore, the fourth design variable r_c and the third design variable h_{c3} are grouped together to form $J_1 = \{3, 4\}$ and $\tilde{\mathbf{x}}^{(1)} = [h_{c3} \ r_c]^T$. The process is repeated and the group J_1 and sub design vector $\tilde{\mathbf{x}}^{(1)}$ are concatenated with the other design variables until the number of elements in J_1 is greater than or equal to three ($d_1 \geq d_{min}$). For this example,

when the condition ($d_1 \geq d_{min}$) is satisfied, the group J_1 contains $\{3, 4, 2\}$. The next strongest element 286 corresponds to the fifth design variable and the ninth design variable. The sixth and the ninth design variable does not have a strong interaction with respect to any element in J_1 . Therefore, group J_1 has only $\{3, 4, 2\}$ design variables which is shown by underline (-) in the matrix $\mathbf{C}$. Update the $\mathbf{C}$ matrix by replacing the entire rows and columns whose indices belong to J_i with zeros. Increment the group number and repeat the process until the termination condition in (4.17) is satisfied. If the termination condition is satisfied, we form the last group and stop the decomposition process. For this example, the second group J_2 is given by $\{5, 10, 6, 7, 1\}$ (shown as underbrace in matrix $\mathbf{C}$) and the last group comprises of $J_3 = \{8, 9, 11\}$ which is found using (4.19). The total number of groups in this example is $L = L_{ini} = 3$. Using the decomposition algorithm, the sub design vector $\tilde{\mathbf{x}}^{(1)} = [h_{c2} \ h_{c3} \ r_c]^T$ represents the first group, the sub design vector $\tilde{\mathbf{x}}^{(2)} = [h_{c1} \ r_p \ w_1 \ w_2 \ w_a]^T$ represents the second group and the sub design vector $\tilde{\mathbf{x}}^{(3)} = [w_3 \ w_4 \ w_b]^T$ represents the third group. The vector $\mathbf{z}$ is given as $[(\tilde{\mathbf{x}}^{(1)})^T \ (\tilde{\mathbf{x}}^{(2)})^T \ (\tilde{\mathbf{x}}^{(3)})^T]^T$.

Sub-optimization of the first group involves the optimization of the sub design vector $\tilde{\mathbf{x}}^{(1)}$ with the remaining design variables, i.e., the design variables in $\tilde{\mathbf{x}}^{(2)}$ and $\tilde{\mathbf{x}}^{(3)}$, fixed at a value in $\mathbf{x}^c$. The initial central point for the first sub-optimization problem is $\tilde{\mathbf{x}}^{c,(1)} = [50.3 \ 50.4 \ 35.0]^T$, (all values in mm) which is a subset of $\mathbf{x}^c$. The initial trust radius $\delta^{(1)}$ for the first sub-optimization problem is calculated using (4.21) and is equal to $[0.2515 \ 0.121 \ 0.147]^T$ (all values in mm) with $\psi^{(1)} = 0.5$. $\Delta^{max,(1)}$ for this example is chosen as $[0.2515 \ 0.121 \ 0.147]^T$ (all

values in mm). Similarly, the initial central point for the second sub-optimization problem is $\tilde{\mathbf{x}}^{c,(2)} = [42.8 \ 6.0 \ 115.3 \ 50.3 \ 6.0]^T$ (all values in mm) which is a subset of $\mathbf{x}^c$. The initial trust radius $\delta^{(2)}$ is calculated using (4.21) and is equal to $[0.4708 \ 0.12 \ 2.306 \ 1.006 \ 0.102]^T$ (all values in mm) with $\psi^{(2)} = 2$. $\Delta^{max,(2)}$ for this example is chosen as $[0.709 \ 0.18 \ 3.459 \ 1.509 \ 0.1542]^T$ (all values in mm). The initial central point for the third sub-optimization problem is $\tilde{\mathbf{x}}^{c,(3)} = [46.9 \ 51.0 \ 12.0]^T$ (all values in mm) which is a subset of $\mathbf{x}^c$. The initial trust radius $\delta^{(3)}$ calculated using (4.21) is $[0.7504 \ 1.02 \ 0.24]^T$ (all values in mm) with $\psi^{(3)} = 2$. $\Delta^{max,(3)}$ for this example is chosen as $[1.1256 \ 1.53 \ 0.36]^T$ (all values in mm).

Table 4.5 clearly shows that the validity range of the surrogate model, i.e., $\tilde{X}^{(i)}$ with the decomposition method is much larger than the validity range of the surrogate model without decomposition [18]. The validity range expressed as trust radius is computed by trust region algorithm. This increase in the validity range of the surrogate model (trust radius as the size of deviation from the central point for each iteration) will result in larger step size for the optimization update in comparison with the optimization update obtained without decomposition.

In this example, the sub-optimization problems has three, five and three design variables in first, second and third groups respectively. The size of internal weighting vector $\boldsymbol{\lambda}^{(i)}$ of the surrogate model in (4.22) is different for different sub-problems. Using a simple quadratic polynomial will result in 16, 21 and 16 internal weighting variables for the first, second and third groups respectively. Using DOE for each sub-problem, the number of EM samples varies with the sub-problem for constructing an accurate surrogate model. In this example, first group uses 16 geometrical samples

Table 4.5: Comparison of the Validity Range (Expressed as Trust Radius) of the Surrogate Model with Decomposition and without Decomposition for an Iris Coupled Cavity Filter

Optimization method	Optimization method without decomposition [18]	Optimization method with decomposition		
		Sub-optimization 1	Sub-optimization 2	Sub-optimization 3
Design variables	$\mathbf{x} = [h_{c1} \ h_{c2} \ h_{c3} \ r_c \ r_p \ w_1 \ w_2 \ w_3 \ w_4 \ w_a \ w_b]^T$	$\tilde{\mathbf{x}}^{(1)} = [h_{c2} \ h_{c3} \ r_c]^T$	$\tilde{\mathbf{x}}^{(2)} = [h_{c1} \ r_p \ w_1 \ w_2 \ w_a]^T$	$\tilde{\mathbf{x}}^{(3)} = [w_3 \ w_4 \ w_b]^T$
Trust radius (deviation from central point) %	[0.05 0.05 0.025 0.05 0.1 0.1 0.1 0.1 0.1 0.1 0.1] ^T	[0.5 0.24 0.42] ^T	[1.1 2.0 2.0 2.0 1.7] ^T	[1.6 2.0 2.0] ^T
Surrogate model training error	3.56%	3.2274%	3.2278%	3.4281%
Surrogate model test error	4.5247%	3.954%	3.7451%	3.814%

(i.e., $N_s^{(1)} = 16$), second group uses 25 geometrical samples (i.e., $N_s^{(2)} = 25$), and the third group uses 16 geometrical samples (i.e., $N_s^{(3)} = 16$) to generate multiple fine model responses on multiple computers with multiple processors such that the overall workload is balanced.

Pole-residue surrogate models are constructed in the region $\tilde{X}^{(i)}$ simultaneously for the three sub-problems. The three sub-problems are optimized in parallel using NeuroModeler-plus program [92] in the trust regions $\tilde{X}^{(1)}$, $\tilde{X}^{(2)}$, and $\tilde{X}^{(3)}$ using (4.23) to obtain the sub-optimization solutions $\tilde{\mathbf{x}}^{\mu,(1)}$, $\tilde{\mathbf{x}}^{\mu,(2)}$, and $\tilde{\mathbf{x}}^{\mu,(3)}$. Once, the sub-optimization solutions for each group are determined using (4.23) and (4.27), the set $\mathbf{Z}$ in (4.35) is determined. For this example, three base update vectors $\mathbf{z}_{100}$, $\mathbf{z}_{010}$ and $\mathbf{z}_{001}$ are determined using the three sub-optimization solutions $\tilde{\mathbf{x}}^{\mu,(1)}$, $\tilde{\mathbf{x}}^{\mu,(2)}$ and $\tilde{\mathbf{x}}^{\mu,(3)}$. Since we have three groups, the crossover mechanism will lead to four crossover update vectors $\mathbf{z}_{011}$, $\mathbf{z}_{101}$, $\mathbf{z}_{110}$, and $\mathbf{z}_{111}$. Fig. 4.10 shows the fine model responses evaluated using the central point $\mathbf{z}_{000}$ and its prospective updates $\mathbf{z}_{100}$, $\mathbf{z}_{010}$, $\mathbf{z}_{001}$, $\mathbf{z}_{011}$, $\mathbf{z}_{101}$, $\mathbf{z}_{110}$, and $\mathbf{z}_{111}$ in the first iteration. Using (4.34), the best solution $\mathbf{z}^{new}$ is calculated as $\mathbf{z}_{111}$ with the least value of objective function 103.45 as shown in Fig. 4.10. $\mathbf{z}^{new}$ is set to be the next optimization update $\mathbf{x}^{new}$ of the overall optimization process. The above process is repeated until one of the termination conditions in (4.38) or (4.39) is satisfied. If the termination conditions are not satisfied, the central point for the next iteration $\mathbf{x}^c$ is updated with $\mathbf{x}^{new}$, and the trust radius $\delta^{(i)}$ for the next iteration is updated with $\delta^{new,(i)}$ (calculated in (4.37)).

Figure 4.10: Comparison of fine model responses and the objective function values of the central point and at the prospective updates in the first iteration of the optimization method for an iris coupled cavity filter. (a) Fine model response and U of the central point z_{000} for the first iteration. (b) Fine model response and U of the base vector z_{100} evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$. (c) Fine model response and U of the base vector z_{010} evaluated using $\tilde{\mathbf{x}}^{\mu,(2)}$. (d) Fine model response and U of the base vector z_{001} evaluated using $\tilde{\mathbf{x}}^{\mu,(3)}$. (e) Fine model response and U of the crossover vector z_{011} evaluated using $\tilde{\mathbf{x}}^{\mu,(2)}$ and $\tilde{\mathbf{x}}^{\mu,(3)}$. (f) Fine model response and U of the crossover vector z_{101} evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$ and $\tilde{\mathbf{x}}^{\mu,(3)}$. (g) Fine model response and U of the crossover vector z_{110} evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$ and $\tilde{\mathbf{x}}^{\mu,(2)}$. (h) Fine model response and U of the crossover vector z_{111} evaluated using $\tilde{\mathbf{x}}^{\mu,(1)}$, $\tilde{\mathbf{x}}^{\mu,(2)}$ and $\tilde{\mathbf{x}}^{\mu,(3)}$. As seen in the figure, z_{111} has the least value of objective function of 103.45 and is selected as the best solution z^{new} among $\mathbf{Z} = \{z_{100}, z_{010}, z_{001}, z_{011}, z_{101}, z_{110}, z_{111}\}$ for the first iteration.

Using the proposed technique, the final optimal solution $\mathbf{x}^* = [42.7341 \ 50.2747 \ 50.3851 \ 35.0571 \ 5.8707 \ 115.332 \ 50.2582 \ 44.532 \ 48.1664 \ 6.0479 \ 12.2067]^T$ (all values are in mm) is obtained after six iterations. Fig. 4.11(a) and Fig. 4.11(e) shows the fine model response at the initial central point (starting point) and final optimal solution respectively. For comparison purposes, we also use coarse and fine mesh space mapping optimization method [30] to optimize this filter. Fig. 4.11(c) shows the final optimal point obtained after five iterations. Further, we compare the EM optimization method with decomposition and the EM optimization method without decomposition [18]. Fig. 4.11(d) shows that the optimization method without decomposition does not satisfy the design specifications after 13 iterations. As a further comparison, we perform direct EM optimization using HFSS internal optimization feature. Pattern search and quasi-Newton have been used for direct EM optimization. The best result from direct EM optimization has been used for comparison with other optimization methods. Fig. 4.11(b) shows the final optimal solution obtained after 638 fine model evaluations using direct EM optimization. Table 4.7 and Fig. 4.11 shows that direct EM optimization, coarse and fine mesh SM method and the optimization method without decomposition [18] are trapped in a local minima and therefore cannot converge further to the optimal solution whereas our proposed EM optimization with decomposition approach avoids the local minima trap and takes 200.54 min to reach an optimal point.

Table 4.6: Comparison of Fine Model Evaluations Using the Optimization Method With and Without Decomposition for an Iris Coupled Cavity Filter

Optimization method	Optimization without decomposition [18]	Optimization with decomposition
No. of groups L	1	3
No. of fine model evaluations per iteration	98	$(25 \times 2 + 16 \times 1)^\dagger + 4^\perp$
No. of iterations	13	6
Final value of objective function	45.37	0.83

$^\perp$ - No. of fine model evaluations due to crossover mechanism in each iteration of the optimization process.

† - Executed in parallel which is equivalent to the CPU time of one fine model evaluation.

Figure 4.11: Comparison of four different optimization methods for an iris coupled cavity filter. (a) Fine model response at the starting point for all the four optimization methods. (b) Fine model response using direct EM optimization (after 638 iterations). (c) Fine model response using coarse and fine mesh SM optimization (after 5 iterations). (d) Fine model response using optimization method without decomposition method [18] (after 13 iterations). (e) Fine model response using proposed optimization with decomposition method (after 6 iterations).

Table 4.7: Comparison of CPU Time for Optimization of an Iris Coupled Cavity Filter

Optimization method	Direct EM opt. using 8 cores	Coarse & fine mesh SM opt.	Opt. method without decomposition [18]	Proposed method with decomposition
Time for derivative calculations	-	-	13m 16s (1st order)	29m 28s (1st & 2nd order)
Fine model evaluation time	11m 32s	11m 32s	13m 16s	13m 16s
Evaluation time for all fine models	11.53m $\times$ 638	11.53m $\times$ (5 + 1)	13.27 $\times$ (13 + 1)	13.27 $\times$ 6 + 13.27m $\times$ 6 [⊥]
Time for modeling and optimization	-	-	3.1m $\times$ 13	2m $\times$ 6
Number of coarse model evaluations	-	1165	-	-
Time for coarse model evaluations	-	5.083m $\times$ 1165	-	-
Total time	122.6h*	98.69h*	227.38m + 5 $\times$ 16.47m [⊕] = 309.73m*	200.54m

[⊥] - CPU time for fine model evaluations of crossover update vector.

* - Design specifications are not satisfied at the solution.

[⊕] - CPU time for unused fine model evaluations.

The values of the objective function U for the proposed optimization with decomposition and coarse and fine mesh SM optimization are shown in Fig. 4.12(a). Further, we compare the value of objective function for the EM optimization method with decomposition and the EM optimization method without decomposition. Fig. 4.12(b) shows that the value of objective function decreases slowly using an optimization method without decomposition [18]. As shown in Table 4.7, Fig. 4.11(d) and Fig. 4.12(b), the final solution obtained using optimization method without decomposition [18] is trapped in a local minima because of smaller validity range of the surrogate model. Fig. 4.11(e) and Fig. 4.12 shows that, the proposed optimization with decomposition converges faster than the coarse and fine mesh SM optimization as well as the optimization method without decomposition [18].

(a)

(b)

Figure 4.12: (a) The values of objective function for the iris coupled microwave cavity filter using: the proposed method with decomposition (o) and coarse and fine mesh SM method (x). (b) The values of objective function for the iris coupled microwave cavity filter using: optimization method with decomposition (o) and optimization method without decomposition [18] (x).

4.6 Conclusion

An EM optimization decomposition approach has been proposed to address the challenges in EM optimization with many design variables. The proposed method decomposes a single large EM optimization problem into multiple smaller sub-optimizations to improve the optimization efficiency. The sub-optimizations including construction of a surrogate model and optimization of each sub-problem are performed in parallel. The proposed decomposition approach combined with parallel computations significantly reduces the CPU time required for EM optimization. Further, the proposed EM optimization decomposition approach has a better chance of avoiding local minima traps and converges to the optimal solution by incorporating crossover mechanism into the algorithm.

Chapter 5

Conclusions and Future Research

5.1 Conclusions

In this thesis, a new gradient based EM optimization technique using parallel computational approach is presented. An EM optimization decomposition approach has also been proposed to address the challenges in EM optimization with many design variables.

In the first part of the thesis, the proposed EM optimization technique achieves speed up in the optimization process even if a coarse model is not available. Speed up in optimization has been achieved by using large and effective optimization update in each iteration which resulted in fewer optimization iterations. In the proposed technique, we deliberately increase the number of fine model evaluations without increasing the computation time by using a parallel computational approach. This method has high parallel efficiency using fast simulation feature and interpolating simulation feature in available EM simulation tools.

Second part is a further advancement of the proposed EM optimization technique

by considering a new challenge in EM optimization. The proposed EM optimization decomposition approach addresses the issue of scalability to EM optimization with many design variables. The proposed method decomposes a single large EM optimization problem into multiple smaller sub-optimizations to improve the optimization efficiency. The sub-optimizations including construction of a surrogate model and optimization of each sub-problem are performed in parallel. The proposed decomposition approach combined with parallel computations significantly reduces the CPU time required for EM optimization.

In both parts of our work, comparisons have been made with both the direct EM optimization (in-built internal optimization feature in the EM simulator) and the recently developed coarse and fine mesh space mapping technique [30]. Comparisons show the proposed EM optimization technique using parallel computational approach achieves a significant speed up and higher parallel efficiency in the optimization process. The speed up due to parallel computations of the proposed optimization doesn't replace the speed up achieved by using parallel computations in the existing EM simulators. The speed up achieved using proposed method is an additional speed up. It is also shown that the proposed EM optimization technique with large and effective optimization updates has a better chance of avoiding local minima traps than classical quasi-Newton which uses smaller optimization updates. In the proposed EM optimization decomposition approach described in chapter 4, the step size of the optimization update is much larger than the step size of the optimization update without decomposition approach. A large step size for the sub-optimization results in fewer iterations of the overall optimization process. Us-

ing conventional gradient based optimization methods, the increase in the volume of the search space will result in the increased likelihood of stopping at a undesired local minima. Therefore, by decomposing the large EM optimization problem into smaller sub-optimizations and by incorporating crossover mechanism into the algorithm the proposed approach has a better chance of avoiding local minima traps and converges to the optimal solution. The optimal solution obtained is verified by computing the fine model response using the EM simulator.

5.2 Future Research

Some of the future directions that can be taken to continue the work that has been initiated in this thesis towards addressing EM optimization challenges are stated as follows

- When a large number of design variables are used for EM optimization, the proposed EM optimization decomposition will lead to large number of groups. As the number of groups increases the interaction between groups becomes more important. Therefore, it is essential to explore the improvement in the optimization efficiency of the proposed decomposition technique with much large number of EM design variables (greater than 20).
- To investigate higher order regression techniques to handle more design variables with much larger neighborhood so that a close match between the fine model response and surrogate model is achieved. There are two pertinent issues to be addressed with the increase in the number of design variables and

with an increase in the neighborhood. Firstly, the increase in the neighborhood (larger trust region) increases the number of data samples required to accurately match fine model response and surrogate model. Second, the increase in the number of design variables increases the number of data samples required to accurately match fine model response and surrogate model. Therefore, to handle more design variables in a much large neighborhood, a large number of unknowns (so that the system of equations is over-determined) are required. This may be achieved by using higher order regression techniques.

- The proposed EM optimization decomposition approach moves the design solution close to the optimal solution quickly. However, as the design solution moves closer to the optimal solution, the speed of convergence decreases. As a further work, we can explore the possibility of using co-ordinate rotation to reduce the interactions between the design variables.
- To investigate the use of the proposed optimization technique that uses surrogate model for a yield-driven optimization of EM structures. Surrogate model response of the EM structure under consideration is more accurate as the range of the trust region decreases. Therefore, the surrogate model can be used to estimate the yield at the design center rather than simulating the EM structure thousands of times (using Monte Carlo (MC) analysis), which is prohibitive. Now, the nominal central point (design solution) can be optimized for a better yield.
- To expand the proposed parallel optimization approach for active circuit de-

sign which typically require computationally expensive non-linear circuit simulation within optimization loops.

References

- [1] M. B. Steer, J. W. Bandler, and C. M. Snowden, “Computer-aided design of RF and microwave circuits and systems,” *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 3, pp. 996-1005, Mar. 2002.
- [2] H. Wu and A. Cangellaris, “A finite-element domain-decomposition methodology for electromagnetic modeling of multilayer high-speed interconnects,” *IEEE Trans. Adv. Packag.*, vol. 31, no. 2, pp. 339-350, May 2008.
- [3] Y. Liu and J. Yuan, “A finite element domain decomposition combined with algebraic multigrid method for large-scale electromagnetic field computation,” *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 655-658, Apr. 2006.
- [4] M. Xue and J. Jin, “A hybrid conformal/nonconformal domain decomposition method for multi-region electromagnetic modeling,” *IEEE Trans. Antennas Propag.*, vol. 62, no. 4, pp. 2009-2021, Apr. 2014.
- [5] G. Scotti, P. Tommasino, and A. Trifiletti, “MMIC yield optimization by design centering and off-chip controllers,” *IET Proc. Circuits, Devices and Syst.*, vol. 152, no. 1, pp. 54-60, Feb. 2005.

- [6] S. Koziel and J. W. Bandler, "Rapid yield estimation and optimization of microwave structures exploiting feature-based statistical analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 1, pp. 107-114, Jan. 2015.
- [7] J. S. Ochoa and A. C. Cangellaris, "Random-space dimensionality reduction for expedient yield estimation of passive microwave structures," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 12, pp. 4313-4321, Dec. 2013.
- [8] R. Biernacki, S. Chen, G. Estep, J. Rousset, and J. Sifri, "Statistical analysis and yield optimization in practical RF and microwave designs," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Montreal, QC, Canada, Jun. 2012, pp. 1-3.
- [9] Y. Cao, J. J. Xu, V. K. Devabhaktuni, R. T. Ding, and Q. J. Zhang, "An adjoint dynamic neural network technique for exact sensitivities in nonlinear transient modeling and high-speed interconnect design," in *IEEE MTT-S Int. Microw. Symp. Dig.*, PA, Philadelphia, Jun. 2003, pp. 165-168.
- [10] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifier using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025-1033, Mar. 2004.
- [11] M. Isaksson and D. W. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422-3428, Nov. 2005.

- [12] B. O'Brien, J. Dooley, and T. J. Brazil, "RF power amplifier behavioral modeling using a globally recurrent neural network," in *IEEE MTT-S Int. Microw. Symp. Dig.*, San Francisco, CA, Jun. 2006, pp. 1089-1092.
- [13] D. Schreurs, J. Wood, N. Tufillaro, L. Barford, and D. E. Root, "Construction of behavioral models for microwave devices from time domain large signal measurements to speed up high-level design simulations," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 13, no. 1, pp. 54-61, Jan. 2003.
- [14] H. Sharma and Q. J. Zhang, "Automated time domain modeling of linear and nonlinear microwave circuits using recurrent neural networks," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 18, no. 3, pp. 195-208, May 2008.
- [15] S. A. Sadrossadat, Y. Cao, and Q. J. Zhang, "Parametric modeling of microwave passive components using sensitivity-analysis-based adjoint neural-network technique," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 5, pp. 1733-1747, May 2013.
- [16] S. Koziel, S. Ogustov, Q. S. Cheng, and J. W. Bandler, "Rapid electromagnetic-based microwave design optimization exploiting shape-preserving response prediction and adjoint sensitivities," *IET Trans. Antennas Propag.*, vol. 8, no. 10, pp. 775-781, Feb. 2014.
- [17] M. Ghassemi, M. Bakr, and N. Sangary, "Antenna design exploiting adjoint sensitivity-based geometry evolution," *IET Trans. Antennas Propag.*, vol. 7, no. 4, pp. 268-276, Jan. 2013.

- [18] V. M. R. Gongal-Reddy, S. Zhang, C. Zhang, and Q. J. Zhang, "Parallel computational approach to gradient based EM optimization of passive microwave circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 44-59, Jan. 2016.
- [19] V. M. R. Gongal-Reddy, F. Feng, C. Zhang, S. Zhang, and Q. J. Zhang, "Parallel decomposition approach to gradient based EM optimization," *IEEE Trans. Microw. Theory Techn.*, Feb. 2016 (submitted).
- [20] J. W. Bandler, and S. H. Chen, "Circuit optimization: the state of the art," *IEEE Trans. Microw. Theory Techn.*, vol. 36, no. 2, pp. 424-443, Feb. 1988.
- [21] J. W. Bandler, W. Kellermann, and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design," *IEEE Trans. Microw. Theory Techn.*, vol. 33, no. 12, pp. 1519-1530, Dec. 1985.
- [22] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard, "Space mapping: The state of the art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 337-361, Jan. 2004.
- [23] J. W. Bandler, R. M. Biernacki, S. H. Chen, R. H. Hemmers, and K. Madsen, "Electromagnetic optimization exploiting aggressive space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 43, no. 12, pp. 2874-2882, Dec. 1995.
- [24] J. W. Bandler, Q. S. Cheng, N. K. Nikolova, and M. A. Ismail, "Implicit space mapping optimization exploiting preassigned parameters," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 378-385, Jan. 2004.

- [25] J. W. Bandler, M. A. Ismail, and J. E. Rayas-Sanchez, "Expanded space-mapping EM-based design framework exploiting preassigned parameters," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 12, pp. 1833-1838, Dec. 2002.
- [26] J. W. Bandler, Q. S. Cheng, D. H. Gebre-Mariam, K. Madsen, F. Pedersen, and J. Sondergaard, "EM-based surrogate modeling and design exploiting implicit, frequency and output space mappings," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Philadelphia, PA, Jun. 2003.
- [27] J. W. Bandler, D. M. Hailu, K. Madsen, and F. Pedersen, "A space mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 11, pp. 2593-2600, Nov. 2004.
- [28] J. Meng, S. Koziel, J. W. Bandler, M. H. Bakr, and Q. S. Cheng, "Tuning space mapping: A novel technique for engineering design optimization," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Atlanta, Georgia, Jun. 2008, pp. 991-994.
- [29] S. Koziel and J. W. Bandler, "Space-mapping with adaptive response correction for microwave design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 2, pp. 478-486, Feb. 2009.
- [30] C. Zhang, Feng Feng, and Q. J. Zhang, "EM optimization using coarse and fine mesh space mapping," in *Proc. Asia-Pacific Microw. Conf.*, Seoul, Korea, Dec. 2013, pp. 824-826.

- [31] S. Koziel, Q. S. Cheng, and J. W. Bandler, "Space mapping," *IEEE Microw. Mag.*, vol. 9, no. 6, pp. 105-122, Dec. 2008.
- [32] N. M. Alexandrov, J. E. Dennis Jr., R. M. Lewis, and V. Torczon, "A trust-region framework for managing the use of approximation models in optimizations," *Struct. Optim.*, vol. 15, no. 1, pp. 16-23, Feb. 1998.
- [33] N. M. Alexandrov and R. M. Lewis, "An overview of first-order model management for engineering optimization," *Optim. Eng.*, vol. 2, no. 4, pp. 413-430, Dec. 2001.
- [34] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman, "Approximation and model management in aerodynamic optimization with variable-fidelity models," *J. of Aircraft*, vol. 38, no. 6, pp. 1093-1101, Dec. 2001.
- [35] R. Ben Ayed, J. Gong, S. Brisset, F. Gillon, and P. Brochet, "Three-level output space mapping strategy for electromagnetic design optimization," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 671-674, Feb. 2012.
- [36] S. Koziel, J. W. Bandler, and Q. S. Cheng, "Constrained parameter extraction for microwave design optimization using implicit space mapping," *IET Microw., Antennas Prop.*, vol. 5, no. 10, pp. 1156-1163, Jul. 2011.
- [37] S. Koziel, S. Ogurtsov, J. W. Bandler, and Q. S. Cheng, "Reliable space-mapping optimization integrated with EM-based adjoint sensitivities," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 10, pp. 3493-3502, Oct. 2013.

- [38] S. Koziel, Q. S. Cheng, and J. W. Bandler, "Fast EM modeling exploiting shape-preserving response prediction and space mapping," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 3, pp. 399-407, Mar. 2014.
- [39] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design – From theory to practice," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1339-1350, Apr. 2003.
- [40] J. E. Rayas-Sanchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 420-435, Jan. 2004.
- [41] V. K. Devabhaktuni, M. C. E. Yagoub, Y. Fang, J. J. Xu, and Q. J. Zhang, "Neural networks for microwave modeling: Model development issues and non-linear modeling techniques," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 11, pp. 4-21, Jan. 2001.
- [42] V. Rizzoli, A. Costanzo, D. Masotti, A. Lipparini, and F. Mastri, "Computer-aided optimization of non-linear microwave circuits with the aid of electromagnetic simulation," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 1, pp. 362-377, Jan. 2004.
- [43] M. Joodaki and G. Kompa, "A systematic approach to a reliable neural model for pHEMT using different numbers of training data," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Seattle, WA, Jun. 2002, pp. 1105-1108.

- [44] P. Burrascano and M. Mongiardo, "A review of artificial neural networks applications in microwave CAD," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 9, pp. 158-174, May 1999.
- [45] P. M. Watson, K. C. Gupta, and R. L. Mahajan, "Development of knowledge based artificial neural networks models for microwave components," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Baltimore, MD, Jun. 1998, pp. 912.
- [46] F. Wang, and Q. J. Zhang, "Knowledge based neuromodels for microwave design," *IEEE Trans. Microwave Theory Techn.*, vol. 45, pp. 2333-2343, Dec. 1997.
- [47] M. H. Bakr, J. W. Bandler, M. A. Ismail, J. E. Rayas-Snchez, and Q. J. Zhang, "Neural space-mapping optimization for EM-based design," *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2307-2315, Dec. 2000.
- [48] L. Zhang, J. Xu, M. C. E. Yagoub, R. Ding, and Q. J. Zhang, "Efficient analytical formulation and sensitivity analysis of neuro-space mapping for nonlinear microwave device modeling," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 9, pp. 2752-2767, Sep. 2005.
- [49] D. Gorissen, L. Zhang, Q. J. Zhang, and T. Dhaene, "Evolutionary neuro-space mapping technique for modeling of nonlinear microwave devices," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 2, pp. 213-229, Feb. 2011.
- [50] H. El-Rewani, and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*, Hoboken, NJ, John Wiley & Sons, 2005.

- [51] *MPI: A Message-Passing Interface Standard*, version. 2.2, Message Passing Interface Forum, sep. 2009.
- [52] *OpenMP Application Interface*, version. 3.1, OpenMP Architecture Review Board, Jul. 2011.
- [53] P. Pacheco, *An Introduction to Parallel Programming*, San Francisco, CA: Morgan Kaufmann, 2011.
- [54] J. Zhang, K. Ma, F. Feng, and Q. J. Zhang, "Parallel gradient-based local search accelerating particle swarm optimization for training microwave neural network models," *IEEE MTT-S Int. Microw. Symp. Dig.*, Phoenix, Arizona, USA, May 2015.
- [55] L. Zhang, Y. Cao, S. Wan, H. Kabir, and Q. J. Zhang, "Parallel automatic model generation technique for microwave modeling," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Honolulu, Hawaii, Jun. 2007, pp. 103-106.
- [56] S. Koziel, and J. W. Bandler, "Space mapping with distributed fine model evaluation for optimization of microwave structures and devices," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Atlanta, GA, USA, Jun. 2008, pp. 1377-1380.
- [57] S. Koziel, and J. W. Bandler, "Distributed fine model evaluation for rapid space-mapping optimization of microwave structures," *IET Microw. Antennas Propag.*, vol. 3, no. 5, pp. 798-807, Aug. 2009.

- [58] Feng Feng, C. Zhang, V.-M.-R.- Gongal-Reddy, Q. J. Zhang, and J. Ma, "Parallel space-mapping approach to EM optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 5, pp. 1135-1148, May 2014.
- [59] J. M. Johnson and V. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 39, no. 4, pp. 7-21, Aug 1997.
- [60] G. Crevecoeur, P. Sergeant, L. Dupre, and R. Van de Walle, "A two-level genetic algorithm for electromagnetic optimization," *IEEE Trans. Magn.*, vol. 46, no. 7, pp. 2585-2595, Jul. 2010.
- [61] C. M. Coleman, E. J. Rothwell, and J. E. Ross, "Investigation of simulated annealing, ant-colony optimization, and genetic algorithms for self-structuring antennas," *IEEE Trans. Antennas Propag.*, vol. 52, no. 4, pp. 1007-1014, Apr. 2004.
- [62] S. S. Rao, *Engineering Optimization*, NJ, John Wiley & Sons, 1996.
- [63] S. Koziel, *Simulation-Driven Design Optimization and Modeling for Microwave Engineering*, Imperial College Press, 2013.
- [64] V. Gongal-Reddy, S. Zhang, Y. Cao, and Q. J. Zhang, "Efficient design optimization of microwave circuits using parallel computational methods," in *Proc. 7th Eur. Microw. Integr. Circuits Conf.*, Amsterdam, NL, Oct. 2012, pp. 254-257.

- [65] S. Koziel, J. W. Bandler, and Q. S. Cheng, “Robust trust-region space-mapping algorithms for microwave design optimization,” *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 8, pp. 2166-2174, Aug. 2010.
- [66] P. M. Watson, and K. C. Gupta, “EM-ANN models for microstrip vias and interconnects in dataset circuits,” *IEEE Trans. Microw. Theory Techn.*, vol. 44, no. 12, pp. 2495-2503, Dec. 1996.
- [67] N. K. Nikolova, J. Zhu, D. Li, M. H. Bakr, and J. W. Bandler, “Sensitivity analysis of network parameters with electromagnetic frequency-domain simulators,” *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 2, pp. 670-681, Feb. 2006.
- [68] Y. Cao, G. Wang, and Q. J. Zhang, “A new training approach for parametric modeling of microwave passive components using combined neural networks and transfer functions,” *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 11, pp. 2727-2742, Nov. 2009.
- [69] B. Gustavsen, and A. Semlyen, “Rational approximation of frequency domain responses by vector fitting,” *IEEE Trans. Power Delivery*, vol. 14, no. 3, pp. 1052-1061, Jul. 1999..
- [70] B. Gustavsen, “Improving the pole relocating properties of vector fitting,” *IEEE Trans. Power Delivery*, vol. 21, no. 3, pp. 1587-1592, Jul. 1999.
- [71] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, “Macromodeling of multiport Systems using a fast implementation of the vector fitting method,”

- IEEE Microw. and Wireless Components Lett.*, vol. 18, no. 6, pp. 383-385, Jun 2008.
- [72] Z. Kang, "An analytical pascal matrix transform for s-to-z domain transfer functions," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 597-600, Oct. 2006.
- [73] J. V. Morro, P. Soto, H. Esteban, V. E. Boria, C. Bachiller, M. Taroncher, S. Cogollas, and B. Gimeno, "Fast automated design of waveguide filter using aggressive spacemapping with a new segmentation strategy and a hybrid optimization algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 4, pp. 1130-1142, Apr. 2005.
- [74] Y. F. Ruan, Y. X. Guao, and X. Q. Shi, "Double annular-ring dielectric resonator antenna for ultra-wideband application," *IEEE Microw. and Opt. Techn. Lett.*, vol. 49, no. 2, pp. 362-366, Feb. 2007.
- [75] M. Lapierre, Y. Antar, A. Ittipiboon, and A. Petosa, "Ultra wideband monopole/dielectric resonator antenna," *IEEE Microw. and Wireless Components Lett.*, vol. 15, no. 1, pp. 7-9, Jan. 2005.
- [76] C. Ozzaim, "Monopole antenna loaded by a stepped-radius dielectric ring resonator for ultrawide bandwidth," *IEEE Antennas and Wireless Propag. Lett.*, vol. 10, pp. 843-845, Aug. 2011.
- [77] C. Zhang, F. Feng, V. M. R. Gongal-Reddy, Q. J. Zhang, and J. W. Bandler, "Cognition-driven formulation of space mapping for equal-ripple optimization

- of microwave filters,” *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 7, pp. 2154-2165, Jul. 2015.
- [78] L. O. Chua and L. K. Chen, “Diakoptic and generalized hybrid analysis,” *IEEE Trans. Circuits Syst.*, vol. 23, no. 12, pp. 694-705, Dec. 1976.
- [79] F. Wu, “Solution of large-scale networks by tearing,” *IEEE Trans. Circuits Syst.*, vol. 23, no. 12, pp. 706-713, Dec. 1976.
- [80] S. Liao, H. Kabir, Y. Cao, J. Xu, Q. J. Zhang, and J. Ma, “Neural-network modeling for 3-D substructures based on spatial EM-field coupling in finite-element method,” *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 1, pp. 21-38, Jan. 2011.
- [81] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, “High-dimensional neural-network technique and applications to microwave filter modeling,” *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 1, pp. 145-156, Jan. 2010.
- [82] Z. Lu, X. An, and W. Hong, “A fast domain decomposition method for solving three-dimensional large-scale electromagnetic problems,” *IET Microw. Antennas Propag.*, vol. 56, no. 8, pp. 2200-2210, Aug. 2008.
- [83] Y. Cao, S. Reitzinger, and Q. J. Zhang, “Simple and efficient high-dimensional parametric modeling for microwave cavity filters using modular neural network,” *IEEE Microw. and Wireless Components Lett.*, vol. 21, no. 5, pp. 258-260, May 2011.

- [84] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225-239, Jun. 2004.
- [85] J. W. Bandler and Q. J. Zhang, "An automatic decomposition approach to optimization of large microwave systems," *IEEE Trans. Microw. Theory Techn.*, vol. 35, no. 12, pp. 1231-1239, Dec. 1987.
- [86] Y. J. Wei, Q. J. Zhang, and M. S. Nakhla, "Multilevel optimization of high-speed VLSI interconnect networks by decomposition," *IEEE Trans. Microw. Theory Techn.*, vol. 42, no. 9, pp. 1638-1650, Sep. 1994.
- [87] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative co-evolution," *Information sciences*, vol. 178, pp. 2985-2999, Aug. 2008.
- [88] M. N. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Proc. IEEE Congress Evol. Comput.*, Barcelona, Spain, Jul. 2010, pp. 1762-1769.
- [89] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378-393, Jun. 2014.
- [90] N. K. Georgieva, S. Glavic, M. H. Bakr, and J. W. Bandler, "Feasible adjoint sensitivity technique for EM design optimization," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 12, pp. 2751-2758, Dec. 2002.

- [91] F. Feng, C. Zhang, J. Ma, and Q. J. Zhang, “Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions,” *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60-77, Jan. 2016.
- [92] Q. J. Zhang, *NeuroModeler plus*, Dept. of Electronics, Carleton University, Ottawa, Canada, 2005.
- [93] *HFSS. Ver. 15*, ANSYS Corporation, Canonsburg, PA, 2013.