Optimization-Enabled Transient Simulation for Design of Power Circuits with Multi-Modal Objective Functions

by

Keyhan Kobravi

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Optimization-Enabled Transient Simulation for Design of Power Circuits with Multi-Modal Objective Functions

BY

Keyhan Kobravi

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of

Manitoba in partial fulfillment of the requirement of the degree

MASTER OF SCIENCE

Acknowledgements

I would like to take this opportunity to thank my advisor, Prof. S. Filizadeh, for his encouragement and invaluable knowledge he provided me in the field of power systems and optimization during this thesis. This thesis could not be presented in this form without his efforts and insightful guidance.

I hope that I could have provided a good contribution in the field of power system optimization, which was initiated by Prof. S. Filizadeh and Prof. A. M. Gole at University of Manitoba Power System Division. I am also very honored to be part of the University of Manitoba power system research society. The amount of knowledge I have gained in this community is invaluable.

I would like also to thank my parents whose support played an important role in enabling me to continue my education in Canada.

Abstract

The performance of a power system is dependent on a suitable selection of system parameters. The system parameters may include element values or controller gains. Optimization-enabled electromagnetic transient simulation (OE-EMTS) has been introduced as an efficient tool for optimization of power system parameters. This is attained through a mutual interaction between a nonlinear optimization algorithm and an electromagnetic transient simulation program. The electromagnetic transient simulation program evaluates an objective function, which is representing the performance of the power system, using a set of parameters received from the external nonlinear optimization algorithm. Previously developed optimization-enabled electromagnetic transient simulation tools were only capable of finding one global or local minimum. The complexity of modern power systems is increased by incorporation of switching converters and power electronic devices, which has resulted in non-convex objective functions. These objective functions may contain more than one optimum point. Thus designers are not only interested in the global minimum but also they are interested in catching the remaining local minima, because some local minima could have other suitable features for a specific design application. The optimization of multi-modal power systems (e.g., selective harmonic elimination) has driven the need for a new nonlinear optimization algorithm capable of finding all local minima. The new algorithm developed here is an implementation of a multi-resolution search in the optimization space of a design problem, which is defined using equality or inequality constraints. The multi-resolution optimization algorithm reduces the search effort by avoiding sampling at regions containing no local minima. The optimization results of various multi-modal power systems using the multi-resolution algorithm are shown.

Table of Contents

List of Figures **vi**

List of Tables **x**

1 Introduction **1**

1.1 Power System Simulation 2

1.2 Design Using Simulation 3

1.3 Optimization-Based Design 4

1.4 Motivation 5

1.5 Problem Definition 6

1.6 Research Objectives 8

1.7 Thesis Organization 9

2 Background **11**

2.1 Introduction 11

2.2 Simulation-Based Optimization 13

2.2.1 Gradient-Based Methods 13

2.2.2 Heuristic Methods 14

2.2.3 Meta-Modeling 18

2.2.4 Direct Search Methods 18

2.2.5 Response Surface Method 19

2.2.6 Statistical Simulation-Based Optimization 20

2.2.7 Global Optimization 21

2.2.8 Multiple-Objective Optimization 22

2.2.9 Multi-Optimizer Strategy 23

2.3 Power System Simulation-Based Optimization 24

2.4 Multi-Resolution Function Reconstruction 25

2.4.1 Constraint Handling 27

2.4.2 Volume Mesh Generation 27

2.4.3 Surface Mesh Generation 28

3 Objective Function Definition and Optimization **29**

3.1 Introduction 29

3.2 Objective Function Definition 30

3.3 Objective Function Optimization 32

3.4 Constrained Optimization 33

3.5 Multi-Modal Objective Function Optimization 36

4 Multi-Resolution Optimization **39**

4.1 Introduction 39

4.2 Multi-Resolution Search Algorithm 41

4.3 Multi-Resolution Implementation for Optimization 43

4.4 Initial Mesh Generation 46

4.4.1 Mesh Enhancement Using Force Relaxation 49
4.5 Localization 55
4.5.1 Cell-Based Localization 56
4.5.2 Graph-Based Localization 59
4.6 Mesh Refinement Techniques 62
4.6.1 Center of Mass Mesh Refinement 62
4.6.2 New Mesh Refinement Technique 64
4.7 Constraint Handling 66
4.7.1 Inequality Constraints 67
4.7.2 Equality Constraints 70
4.8 Parametric Studies 74
4.8.1 Effect of Mesh Distribution 74
4.8.2 Effects of Force Relaxation 77

5 Modifications to Multi-Resolution Optimization **80**
5.1 Parallel Implementation 81
5.2 Partial Mesh Generation 83
5.2.1 Delaunay Algorithm 87
5.2.2 Convex-Hull Connectivity Test 91
5.2.3 Convex-Hull Detection Test 95
5.2.4 Partial Mesh Implementation 95

6 Design Examples **100**
6.1 Introduction 100
6.2 PSCAD Implementation 101

6.3 Example 1: Optimal Pulse-Width Modulation 102

6.4 Example 2: DC-to-DC Converter 106

6.5 Example 3: Cigre HVDC Benchmark 111

6.5.1 Defining the Objective Function 113

6.6 Example 4: HVDC Robust-Optimization 120

6.6.1 Optimization Results 121

7 Conclusions **130**

7.1 Thesis Contributions 131

7.1.1 Development of the New Nonlinear Optimization Algorithm . 131

7.1.2 Multi-Modality Phenomena in Power Systems 133

7.2 Recommendations for Future Work 133

7.2.1 Incorporation of Meta-Modeling Technique 134

7.2.2 Incorporation of a Multi-Optimizer Strategy 134

A Transformation Matrix **136**

References **138**

List of Figures

1.1 The optimization-enabled simulation tool. 5

1.2 A highly-sensitive (x_1) and a less-sensitive (x_2) local minimum. . . . 7

2.1 A simulation-based optimization problem. 12

2.2 The genetic algorithm optimization. 17

3.1 The functions $f_1(x)$ and $f_2^k(x)$ for $k = 10$. 31

3.2 The functions $f_1(x)$ and $f_2^k(x)$ for $k = \frac{5}{4}$. 32

3.3 A constrained optimization problem. 34

3.4 Constrained versus unconstrained optimization example. 35

3.5 The optimization space using an equality constraint. 37

3.6 Gradient-based optimization. 38

4.1 The multi-resolution optimization. 42

4.2 Inequality constraint. 43

4.3 Multi-resolution mesh for optimization. 45

4.4 The Delaunay triangulation and point connections. 48

4.5 The initial mesh generation. 50

4.6 Modeling of point connections by spring elements. 51

4.7 The internal and external forces on a boundary node. 51
4.8 The force-displacement characteristics of a spring element. 54
4.9 A non-uniform density function for initial mesh generation. 55
4.10 A non-uniform initial mesh generation. 56
4.11 The cell definition. 57
4.12 Localization of local minima for further mesh refinement. 58
4.13 The simplex connections in a graph-based localization method. . . . 60
4.14 Graph-based localization using connectivity graph. 61
4.15 Mesh refinement using centers of mass of simplexes. 63
4.16 Refined mesh using centers of mass method. 64
4.17 Cell-based refinement. 65
4.18 Initial added points for the new mesh refinement technique. 66
4.19 Repositioned points after force relaxation for new mesh refinement. . 67
4.20 Final generated mesh using force relaxation. 68
4.21 Initial and final meshes for the inequality constraint. 69
4.22 Surface mesh generation using voronoi filtering. 71
4.23 Initial selection of elements crossing the equality constraint. 73
4.24 Elements crossing the equality constraint after 4 level refinement. . . 74
4.25 Initial sampling of the equality constraint. 75
4.26 Final generated mesh for the equality constraint. 76
4.27 (a) Non-uniform initial mesh; (b) Uniform initial mesh. 77
4.28 Convergence rate in (a) log-log plot and (b) regular plot. 78
4.29 Effect of force relaxation for initial mesh on finding local minima. . . 79

5.1 Number of initial sampling points for a 1 and 2-dimensional problem. 81

5.2 Parallel OF evaluation in the multi-resolution algorithm. 82
5.3 Number of elements versus number of points. 84
5.4 Required connections needed for mesh refinement phase. 85
5.5 Required connections needed for localization phase. 86
5.6 Partial Mesh Generation. 87
5.7 Partial Mesh Generation for a point (p_c). 88
5.8 Convex hull of a set of random data in 2-dimensional space. 89
5.9 Delaunay triangulation as the convex hull of the transformed points. . 91
5.10 The set of hyperplane crossing x_1 and y_1. $\mathcal{L}(X)$. 92
5.11 The mapping of the points on the hyperplane $\mathcal{L}(X)$. 92
5.12 Convex-hull generation. 95
5.13 Convex-hull detection. 96
5.14 Partial Mesh Generation for a point (p_c). 97
5.15 Delaunay triangulation as the convex hull of the transformed points. . 98
5.16 Delaunay triangulation of a regular grid. 99

6.1 PSCAD and multi-resolution algorithm interfacing. 101
6.2 Three level converter. 102
6.3 A 3-level switching waveform. 103
6.4 Harmonic spectrum of the three local minima from in Table 6.1. . . 106
6.5 Schematic diagram of the dc-to-dc converter example. 107
6.6 The current controller of a dc-to-dc converter. 108
6.7 The multi-resolution optimization of dc-to-dc converter. 109
6.8 The transient response of the global minimum ($1^{st} local minimum$). . 110
6.9 The transient response of the 6^{th} local minimum. 110

6.10 Cigre HVDC benchmark model. 112

6.11 HVDC steady state control diagram. 113

6.12 Cigre HVDC benchmark controller. 114

6.13 Transient response of the system for the 1^{st} minimum in Table 6.3. . 117

6.14 Transient response of the system for the 2^{nd} minimum in Table 6.3. . 118

6.15 Transient response of the system for the 3^{rd} minimum in Table 6.3. . 118

6.16 Transient response of the system for the 4^{th} minimum in Table 6.3. . 119

6.17 Transient response of the system for the 6^{th} minimum in Table 6.3. . 119

6.18 The robust optimization-enabled simulation tool. 123

6.19 Transient response of the system for the 1^{st} local minimum in Table 6.4. 124

6.20 Transient response of the system for the 2^{nd} local minimum in Table 6.4. 125

6.21 Transient response of the system for the 3^{rd} local minimum in Table 6.4. 126

6.22 Transient response of the system for the 4^{th} local minimum in Table 6.4. 127

6.23 Transient response of the system for the 5^{th} local minimum in Table 6.4. 128

6.24 Transient response of the system for the 6^{th} local minimum in Table 6.4. 129

List of Tables

4.1 Local minima coordinates of function in (4.2). 45

4.2 Location of the local minima in (4.2). 59

4.3 Optimization result of (4.2) using graph-based localization. 61

4.4 Coordinates of the local minima of the function in (4.13). 69

4.5 Coordinates of the local minima of the function in (4.16). 73

4.6 The effect of force relaxation for derivation of local minima. 79

6.1 Selective harmonic elimination results. 105

6.2 Dc-to-dc converter optimization results. 109

6.3 Cigre HVDC benchmark optimization results. 116

6.4 HVDC robust optimization results. 122

Chapter 1

INTRODUCTION

The expansion of power networks and increase in energy demand have resulted in the deployment of modern power electronic devices in power networks. The stable behaviour of these power electronic devices in the network is very important. The issue of stability can be achieved through optimal selection of sophisticated controller gains and element values of these systems.

Optimization-enabled electromagnetic transient simulation (OE-EMTS) was used previously for the optimal design of power systems. The optimization-enabled power system simulation program achieves the optimal solution by interfacing a nonlinear optimization algorithm with an electromagnetic transient simulation program [1]. In the upcoming chapters a new nonlinear optimization algorithm is proposed for use in the optimization-enabled electromagnetic transient simulation program.

1.1 Power System Simulation

Modern power systems require extensive use of nonlinear elements and switching power electronic converters. The existence of such nonlinear devices requires the use of sufficient control systems and proper selection of element values to guarantee the stable behaviour of the network. Analytical methods find limited application in performance evaluation of such systems, because of the variety of employed devices and complexity of their modeling using analytical approaches. Simplification of these complex power systems using averaged models could ease the derivation of an analytical model suitable for optimal design, but the averaged models are neither available for such arbitrary power systems nor do they accurately model the behaviour of fast transient dynamics [2]. Such fast transients play an important role in the behaviour of a power system that uses power electronic devices. Moreover, the analytical methods often work only for linearized models, while in reality all controllers are essentially nonlinear as they contain limiters that could operate in a nonlinear mode. The optimum behaviour of a circuit is not necessarily in linear mode of operation; therefore, the analytical approach needs further modifications to be suitable for all categories of modeling including nonlinear modes of operation.

The behaviour of power electronic systems are characterized by fast switching intervals; therefore, fast transient dynamics in such switching systems cannot be ignored. Access to the actual behaviour of a system must be provided using programs, which are able to simulate these fast dynamics. The electromagnetic transient simulation program (EMTS) is a tool capable of proper simulation of switching power systems as well as other power system devices [3]. Electromagnetic transient simulation programs such as PSCAD/EMTDC [4], provide a detailed representation of complex power systems. This property makes these programs invaluable practical tools for performance evaluation of modern power systems.

1.2 Design Using Simulation

Optimization using EMTS is made possible using an objective function. The objective function is a measurement tool implemented in an electromagnetic transient simulation program which provides a metric for assessing the conformity of the system response to design objectives. The objective function can also be seen as an index quantity for assessing the quality of the system performance.

The electromagnetic transient simulation program provides the value of the objective function after running one simulation, where the proposed system elements and controller gains are substituted by the specified set of design parameters. Various kinds of objective functions could be defined for a system. The objective function measurement provides an interface between the designer specifics and the numerical optimal solution. It is the art of the designer to provide a well-defined objective function, which fits the design objectives to a numerical parameter. The designer may incorporate the behaviour of some suitable dynamical parameters into the objective function.

The performance evaluation of a power system is not necessarily evaluated using a single objective function. Sometime designers prefer to use multi-objective functions, because the behaviour of a complicated system may not be reflected by only one index. Other considerations for a well-defined objective function are convexity and smoothness. Non-convex objective functions need complicated nonlinear optimization tools to find the global minimum. Optimization of non-smooth objective functions also prohibit the usage of a wide variety of nonlinear optimization algorithms, because they may contain a large number of local minima.

1.3 Optimization-Based Design

The system optimization problem is defined by a metric for measuring the quality of a selected set of design parameters. The designer usually defines the objective function like a cost function, where it attains a smaller value for a better design parameter; therefore, the optimal design could be achieved by selecting a proper set of parameters minimizing the objective function. The designer must search all possible sets of parameter values to find the best option.

The requirement of one simulation for evaluation of an objective function for each set of input parameters and considering that the simulation for achieving a detailed transient response is time consuming, which means that one should use an intelligent algorithm to search for the best set of parameters. Conventional search methods such as sequential or random search (e.g., Monte-Carlo method) tend to become excessively intensive when the number of parameters increases. Minimization of the sets of testing parameters can be achieved through deployment of optimization algorithms. In such a strategy the actual brute force search is replaced by an optimization algorithm.

Recently, optimization-enabled electromagnetic transient simulation (OE-EMTS) has been introduced. In this tool a nonlinear optimization algorithm replaces the brute-force sequential or random search and guides the simulation toward an optimum setting of design parameters in a, hopefully, small number of simulations.

Fig. 1.1 shows the combination of the electromagnetic transient simulation and an optimization algorithm.

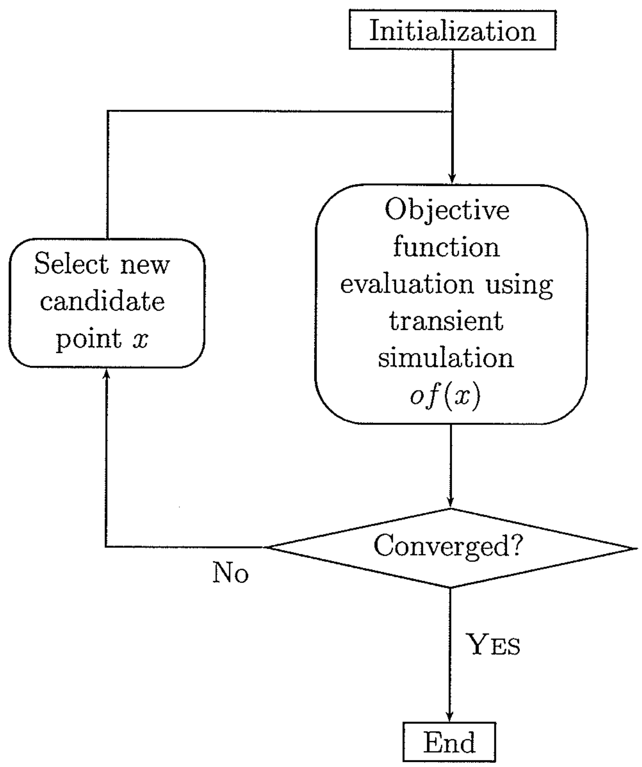


Fig. 1.1: *The optimization-enabled simulation tool.*

1.4 Motivation

The objective functions specified for complex circuits usually contain multiple local minima [5] (referred to as multi-modality). Often designers are interested in finding all possible local minima to be able to select the one, which not only minimizes the objective function but also satisfies some other designer's objectives. Optimization methods used so far in the OE-EMTS programs only converge to one local minimum depending on the initial set of parameter values provided for the nonlinear optimization algorithm. The lack of optimization methods that can detect multiple local minima (simultaneously and in an affordable manner) often requires deploying conventional, but exhaustive search methods, e.g., sequential search over a fine grid. Transient simulations are computationally intensive and time-consuming, thus often making the deployment of such approaches unaffordable.

1.5 Problem Definition

The optimization problem considered here is defined as the minimization of an objective function, which quantifies the quality of the selected set of design parameters. An optimization problem can be represented in the following general form

$$\begin{aligned}
&\text{Minimize} \quad OF(x) \\
&\text{Subject} \quad \text{to} \\
&g_i(x) = 0, i = 1, \ldots, N \\
&h_i(x) \leq 0, i = 1, \ldots, N
\end{aligned} \tag{1.1}$$

where the objective function ($OF(x)$) evaluation is done by an EMTS program and the functions $g_i(x)$ and $h_i(x)$ define the equality and inequality constraints respectively.

The objective functions defined for modern power systems tend to have multiple local minima. The designer's objective is to derive most local minima and select the best one according to its sensitivity or objective function value. Fig. 1.2 shows the situation in an optimization problem where the designer would prefer the less sensitive local minima to guarantee a more robust optimization.

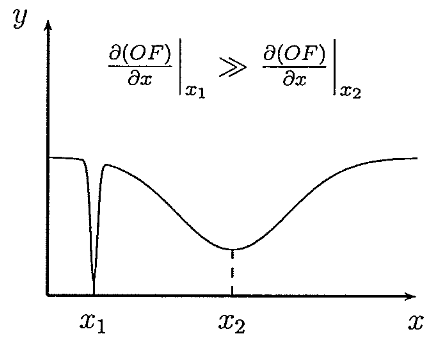


Fig. 1.2: *A highly-sensitive* (x_1) *and a less-sensitive* (x_2) *local minimum.*

The optimization of such objective functions requires deployment of specialized optimization methods that yield most local minima. Several such optimization methods are possible, including direct search method [6], tabu search [7], and grid-based search methods [8], which are capable of capturing most local minima.

The emphasis here is capturing of the local minima according to their sensitivity. This is achieved via a multi-scale search over a grid, where the coordinates of a grid node represent a set of input parameters for the optimization problem. The multi-scale search prohibits sampling in the regions with low probability of containing local minima. This approach reduces the required number of objective function evaluations. The formerly introduced grid-based optimization methods evaluate an initial regular grid and then iteratively refine the grid at locations where local minima exist. The regularity of the grid makes the handling of constraints hard. In this thesis a new approach toward grid search is developed, which uses a multi-scale irregular grid. The deployment of an irregular mesh (grid) enhances the ability of the algorithm to be more effective at solving constrained optimization problems. The new method is based on achieving a multi-scale irregular grid capable of handling implicit inequality constraints and equality constraints. The irregular grid is represented by a finite element mesh, where the nodes of the mesh are the sampling points.

The new algorithm evaluates the points at different scales on an irregular mesh of points and samples the regions that are more likely to contain local minima. The new sampling points are selected to be the nodes of a refined mesh generated inside the proposed regions. The regions and connections between the points in the irregular mesh are formed using Delaunay triangulation, because Delaunay triangulation provides the points connections that construct a suitable approximation of the objective function surface. Other kinds approaches for achieving points connections are not guaranteed to construct an approximating surface for the objective function.

The problem is to develop a method capable of achieving a multi-resolution sampling of an objective function in regions that satisfy inequality and equality constraints in an attempt to find the locations of most local minima in an affordable number of simulations.

1.6 Research Objectives

The multi-resolution algorithm is a novel optimization method that uses an adaptive mesh generation technique to find most local minima of an objective function simultaneously. The new method will enable the designer to capture multiple local minima in a reasonable number of simulations, while maintaining a high accuracy for the final solution.

The developed multi-resolution optimization method is implemented in an electromagnetic transient simulation (EMTS) program in order to enable the program to support optimization of multi-modal power systems. The new method is tested using a number of design examples ranging from optimal switching control of a dc-to-dc converter to a detailed design of control parameters for a standard High Voltage DC transmission (HVDC) system.

The simulation results of optimization are shown to justify the applicability of the new method for optimization of complex power systems with objective functions containing multiple local minima.

1.7 Thesis Organization

This thesis is divided into several chapters. The background chapter (chapter 2) describes some optimization methods for simulation-based optimization. The available methods of tessellation and Delaunay triangulation as well as the methods for surface mesh generation are introduced in this chapter, because they are used in the upcoming chapters. Chapter 3 provides details on a suitable definition of an objective function for use in an optimization-enabled transient simulation program (OE-EMTS).

The chapter on multi-resolution optimization (chapter 4) describes the proposed multi-resolution optimization algorithm for optimizing multi-modal objective functions in detail. Multi-resolution sampling is then implemented using a developed implicit mesh generation technique. The mesh generation technique is demonstrated to be able to handle a set of inequality constraints, but the handling of equality constraints is achieved using more advanced methods. A new method for handling equality constraints is then introduced.

The connection between the points of an irregular mesh is achieved using Delaunay triangulation. The Delaunay triangulation is designed to derive the simplexes connecting a set of points, while it is not specifically designed to produce the point connections. The multi-resolution search basically needs only the connections between the points in an irregular mesh. The deployment of the Delaunay algorithm for higher dimensions is restricted, because of its large storage requirement. The problem is resolved by introducing a new algorithm capable of generating the connections between the points without requiring excessive memory. The new method is introduced in the modifications chapter (chapter 5) as a partial mesh generation technique.

The application of the newly developed optimization method is demonstrated for optimal design of controller gains of a dc-to-dc converter, an Optimal Pulse Width Modulation (OPWM) firing angle generation, and controller gain optimization of a High Voltage DC transmission system (HVDC). These cases are demonstrated the chapter on design examples (chapter 6).

Chapter 2

Background

2.1 Introduction

Power system simulation tools including Electromagnetic Transient Simulation (EMTS) and Real Time Digital Simulators (RTDS) have enabled rapid performance evaluation of modern complex power systems [3]. The application of power electronic devices in power networks has resulted in complex power systems, which cannot be modeled analytically. The lack of analytical models has increased the demand for design by simulation. Recently, optimization-enabled transient simulation program (OE-EMTS) has been introduced as a design tool [9].

Design through simulation has been applied previously in many complex design applications, where analytical models of systems are not available [10]. These applications vary from computational fluid dynamics to finite element analysis of mechanical systems. These problems are categorized as Simulation-Based Optimization problems [11].

The emphasis of this thesis is on optimization of continuous variable objective functions; however optimization parameters of a power system could contain discrete variables as well as continuous variables. The design of some power electronic systems (e.g., zero current, zero voltage, and quasi-resonant converters) would require optimal selection of a set of discrete events. These systems are modeled using mixed discrete-continuous dynamical systems. The modeling, design, and optimization of mixed discrete continuous systems are studied in [12]. Reviews of available simulation-based optimization techniques for the design of purely discrete event systems are presented in [13, 14].

The concentration in this thesis is on optimal selection of a continuous set of controller parameters; therefore, the optimization problem is reduced to optimal selection of n continuous input variables of a system to minimize a single output (the objective function). Fig. 2.1 shows a general configuration of the simulation-based optimization problem used in this thesis [10].

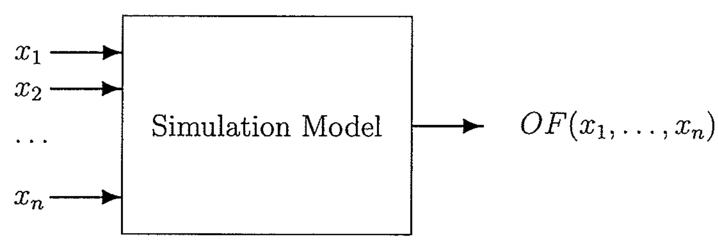


Fig. 2.1: *A simulation-based optimization problem.*

This chapter reviews some simulation-based optimization methodologies. After introducing the available simulation-based optimization algorithms the current approaches for application in power systems are discussed. The final section introduces the multi-resolution search approach as a combination of available methodologies.

2.2 Simulation-Based Optimization

Simulation-based optimization methods are aimed at optimizing a function, which cannot be evaluated explicitly but only through simulation. Commonly these experiments are expensive and time consuming; therefore, the goal in simulation-based optimization methodologies is to discover the local minima or global minimum using the least number of simulations. In this section the existing simulation-based optimization approaches and algorithms are briefly presented.

2.2.1 Gradient-Based Methods

The gradient-based optimization algorithm uses the first derivative of an objective function to estimate the shape of the function and to select a new point in the maximum descent direction. The drawbacks of gradient-based optimization techniques are the necessity of differentiability for the objective functions and a restricted ability of finding only one local minimum, which is not guaranteed to be the global minimum. However, if the objective function is convex then the gradient-based optimization is provably the best choice, because of its higher rate of convergence and low number of function evaluations.

Gradient-based methods need two pieces of information in each step to go forward: the gradient of the objective function and a step size. Various methods have been introduced for gradient estimation and calculation of a step size [15]. The common method for estimating the gradient of an objective function is finite difference derivative, where the gradient is estimated by evaluating the differences of the objective function values at some close points. This method finds a limited application for gradient estimation, because of non-smooth objective functions [16]. The other methods are infinitesimal perturbation analysis(IPA), frequency domain analysis, and likelihood ratio estimators. The IPA method is capable of estimating the gradient using a single simulation, but because of the switching nature of some power systems, this method cannot derive a good estimation for the gradient. The frequency domain method uses the power spectrum of the signal to derive the gradient [15]. The likelihood method was developed to estimate not only the gradient but also the hessian matrix of the system [15].

2.2.2 Heuristic Methods

Heuristic methods are used widely in practice, when the total number of optimization variables is large. These methods are the most useful practical tools for global optimization; however convergence to the global optimum is not guaranteed. Because of the wide applicability of these methods; three categories of heuristic optimization methodologies are presented here. These heuristic methods are also divided into point-to-point based or population based methods. The point-to-point based method uses a single new point at the next iteration (e.g., tabu search), while the population based methods generate a new set of points for the next iteration (e.g., genetic algorithms).

2.2.2.a. Tabu Search

The tabu search was originally designed to solve combinatorial optimization problems; however the demand for such heuristic strategies has resulted in the expansion of tabu search applications for continuous optimization problems.

Tabu search is a point-to-point memory-based optimization algorithm, which makes it a suitable option for simulation-based optimization. Tabu search builds a memory of available moves and a list of forbidden moves. At any stage of the algorithm if the transition $\mathbf{x} \rightarrow \mathbf{x}^*$ results in a better objective function value, the move is selected to be good and if the new objective function value is not lower then the move is added to the forbidden list. A tabu search needs four strategies of, (1) forbidding move policy; (2) freeing strategy to delete some forbidden moves; (3) short term strategy to avoid being trapped in local minima; and (4) the convergence criteria.

Tabu search has recently been extended to a globally convergent method for nonlinear simulation-based optimization cases [7]. Despite all these developments, tabu search compared to other heuristic methods like genetic algorithm and simulated annealing, is still not popular in the area of continuous optimization problems.

2.2.2.b. Simulated Annealing

Simulated annealing implements a similar approach to the metallurgical annealing process. In the annealing process the temperature of an alloy is gradually decreased. The gradual decrease of the temperature results in optimization of the alloy crystallization; therefore, the alloy achieves its maximum strength. Simulated annealing is introduced to avoid being trapped in local minima [17] using the same concept of temperature, during the optimization. A high temperature causes the sampling point to jump enough in order to prohibit being trapped in local minima; therefore, it provides a high probability of escaping local minima which are not the global minimum. Simulated annealing also prohibits early convergence to a local minimum, which makes it a suitable approach for optimization of non-smooth objective functions.

In simulated annealing the next sampling point is selected randomly near the current point. The new point is rejected or accepted by a probability function depending on the objective function differences of the two points. The probability function also depends on a global parameter of temperature (T). The global temperature (T) is gradually decreased during the simulation. In this way the algorithm avoids being trapped in a local minimum and gradually finds the global minimum. The random generation of a new point prohibits the applicability of simulated annealing for optimization of costly objective functions.

2.2.2.c. Evolutionary Algorithms

The evolutionary optimization methods are based on a natural reproduction process, where two parents produce one child and if the child possesses a better objective function value then it replaces the parents. The size of the pool at which the mates are selected could vary, based on designer's decision. Evolutionary based algorithms are widely used for optimization of objective functions with a large number of input parameters.

2.2.2.d. Genetic Algorithms

A genetic algorithm is basically an evolutionary algorithm [18], where a set of sampled points are tried to enhance their objective function values through mating and other random changes.

Similar to its counterpart, every single sampling point is represented by a chromosome, where the genes of this chromosome are the optimization parameters. During a single iteration, mutation by the members (the set of sampled points) of the population and some random changes in their genes produces a new set of members. The old members are replaced by the new members with better objective function values. The important part of the genetic algorithm is the implementation of the reproduction process and random changes in the genes. A survey of genetic and evolution algorithms is provided in [18]. Fig 2.2 shows an overall view of genetic algorithm based optimization.

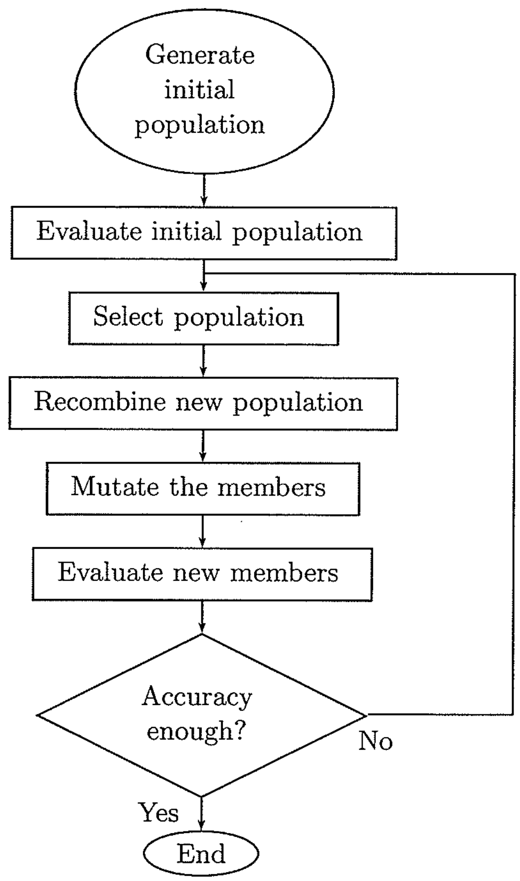


Fig. 2.2: *The genetic algorithm optimization.*

2.2.3 Meta-Modeling

Many real world simulations are not only time-consuming but also costly (e.g., crash test of a car). This way require designers to use several estimating functions instead of the real objective function. Therefore, the optimization is done using the combination of an accurate but costly objective function and an inaccurate but inexpensive one. These problems are categorized as meta-modeling simulation-based optimization problems. A comprehensive survey of meta-modeling is presented in [19]. However, the application of these methods depends on the availability of appropriate estimation procedures.

2.2.4 Direct Search Methods

Direct search methods are easy to program and some of them are reliable methods for optimization. In the past fifteen years these methods have seen a revival due to appearance of new mathematical proofs of their convergence and advancements in distributed and parallel computing. Direct search methods for optimization are best known because they do not require explicit measurement of optimization function derivatives. The direct search methods have been developed mainly to handle bounded or linearly constrained problems. Modern approaches in direct search are reviewed in [20].

The lattice-based and grid-based methods are two well-known direct search optimization algorithms. These methods have been proved to be globally convergent [20]. The grid-based method searches the optimization space on a regular grid and gradually refines the grid at the locations containing local minima. The grid-based optimization algorithms have found wide application for optimization of objective functions with few optimization parameters. The interior-point method is a direct search method used for optimization and tuning of logic circuits [21].

2.2.5 Response Surface Method

The response surface method is based on sampling the objective function at a low rate and then fitting a regression model on the selected points. The regression model provides a rough approximation of the surface of the objective function; therefore, the local minima of the reconstructed function can be seen as an estimate of the local minima of the actual objective function. The sampling rate is then increased in the estimated regions. Different response surface methods have been introduced in [22]. The surface regression method can also be regarded as a global optimization method using globally convergent optimization algorithms for polynomial functions [23, 24] only if the reconstructed function (regression function) is guaranteed to represent a good approximation of the original objective function.

The drawback of the response surface method is the large number of initial sampling points. The evaluation of this set of points requires a huge initial effort before attempting to construct the estimating function. The Low-Cost Response Surface Method (LCRSM) was introduced as a methodology to reduce the number of initial sampling points [25]. The LCRSM method reduces by half the total number of simulations compared to a general response surface method.

The efficiency of the response surface method is mainly due to the iterative sampling point generation in order to avoid sampling at regions that contain local minima. Several sampling strategies are possible based on random sampling [26] or using a direct search strategy [27]. This thesis introduces a new sampling method to enhance the ability of the optimization algorithm to avoid the necessity of using regression to find an algebraic function to estimate the objective function in order to reduce the computation time.

2.2.6 Statistical Simulation-Based Optimization

The simulation model block shown in Fig. 2.1 is not necessarily a deterministic function. The result of the simulation could also be a statistical variable. The objective function in such a situation is calculated as the expected value of the simulation results. Such a situation could happen, where a non-deterministic parameter exists in the system like the consumer load consumption in a power network.

A general formulation of a statistical simulation problem [15] is presented in (2.1).

$$\begin{aligned} &\min_{\mathbf{x}\in\Theta} f(\mathbf{x}) = E[z(\mathbf{x})] \\ &\text{Subject to} : g(\mathbf{x}) = 0 \\ &\qquad\qquad h(\mathbf{x}) \leq 0 \end{aligned} \tag{2.1}$$

where the objective function and constraints are statistical functions of the deterministic input variables. The function E calculates the expected value of its input variable. A comprehensive review of practical simulation-based optimization techniques for statistical objective functions is presented in [15].

2.2.7 Global Optimization

Global optimization is a challenging task and different approaches have been introduced for addressing this issue [28, 29]. Several globally convergent methods have been introduced for linear-constraint nonlinear optimization problems, if the first order derivatives of the optimization function exist [30].

Global optimization requires a large number of sampling points to guarantee the exploration of all possible regions. These global optimization methods are required when the objective function contains many local minima and the solution is only determined by the lowest local minimum. The Cutting Angle Method (CAM) was developed for global optimization of functions containing a large number of local minima [31]. The problem of global optimization of a potential function for a molecular structure is solved by the cutting angle method [32].

Recently, some global optimization methods have been introduced, which use an initial sampling of the whole optimization space. This initial sampling is used later in a direct search method [6] or by a response surface method [22] to find the global minimum. Similar approaches for global optimization use direct methods such as interval analysis to iteratively find the regions containing local minima [33].

Rather than deterministic approaches for global optimization, some other methods have been introduced that find the global minimum with a relatively high probability. Densification and dynamical canonical descent was introduced to find the global optimum [34]. In this method a curve is built out of the objective function. The sampled curve is supposed to explore the regions containing the global minimum with high probability. If the curve pases through a region containing the global minimum, then during the next iterations the global minimum is found. This procedure, effectively, reduces the number of objective function evaluations, by reducing the optimization search space to a 1-dimensional manifold.

2.2.8 Multiple-Objective Optimization

Designers are not only interested in optimizing a single objective function but also a set of multiple objective functions to allow them to investigate the tradeoffs between various design optimization criteria (e.g., a low economical cost or better dynamical performance). Multiple objective function optimization is presented in (2.2).

$$\begin{aligned} \text{Minimize: } & F(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})] \\ \text{Subject to: } & g_i(\mathbf{x}) = 0, i = 1, \ldots, N \\ & h_i(\mathbf{x}) \leq 0, i = 1, \ldots, M \end{aligned} \tag{2.2}$$

where $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are the equality and inequality constraints respectively. The minimization of one objective function does not necessarily result in the minimization of the other objective functions. Therefore, a set of optimal points may not be found that optimizes all functions simultaneously. The easiest way to handle multiple objective function optimization is to combine the sets of objective functions into a single optimization problem with only one function. This new combined objective function could be defined as the weighted sum of all other functions. A survey of multi-objective optimization methods can be found in [35].

In some situations designers are interested in finding the set of Pareto optimal points, to investigate the tradeoffs between various sets of optimal parameters. The

Pareto optimal points are defined as the sets of points, where a single objective function is optimized, while the other objective functions are not decreased accordingly. A recently developed Pareto set identification algorithm for design problems is introduced in [36].

2.2.9 Multi-Optimizer Strategy

Various optimization algorithms have been introduced in this section. These optimization methods have found interesting applications in special fields. It is, however, the skill of the designer to choose a suitable optimization method. It should also be considered that every optimization problem mainly consists of the following two phases:

1. exploration phase, where the region containing a local minimum is observed;
2. exploitation phase, where the optimization algorithm enhances the accuracy of the estimated local minima.

Different algorithms can be applied during these two phases of optimization. Switching from one method to another might greatly enhance the rate of convergence, e.g., in the second phase the gradient based strategy might be the most useful method, while for the first phase the evolutionary algorithms might be of the best choices. The recent applications of such multi-optimizer approaches for simulation-based optimization are presented in [37].

2.3 Power System Simulation-Based Optimization

Modern power systems use a large number of controllers. These controllers are responsible for issues such as firing angle generation of an HVDC converter or exciter control of a generator. In all these applications proportional integrator controllers (PI) are commonly used, because of their robustness and ease of implementation. The problem of adjusting these PI controller gains is the challenging part of their implementation. The design of such controllers is addressed in classical control using bode plots, Nyquist diagrams, or the root locus method. These methods find limited application where the analytical model of the system is not available. This results in acquiring a nonlinear derivative-free optimization method for controller design, but the interfacing between a nonlinear optimization method and an optimal control problem needs a proper definition of an objective function. The interfacing between a nonlinear optimization algorithm and a controller design problem is addressed in [38]. The approaches introduced and implemented for the practical design of such controllers work by defining an objective function to provide a metric for assessing the quality of the designed controller. This method has gained substantial attention, because of the availability of Real Time Digital Simulators and other power system simulation software (e.g, PSCAD/EMTDC).

Simulation-based optimization has proved to be a useful tool for the design of power systems or power electronic controllers using a well-defined objective function [9]. The large number of controller parameters in a power system may reduce the applicability of simulation-based optimization approaches. Practically, most of these controller parameters are independent and they do not have serious coupled effects on system performance especially at long distances. Some approaches make the controller parameters independent by splitting them into independent clusters. The problem of interfacing between a user-defined objective function and a nonlinear optimization algorithm is addressed in [39].

The objective functions defined for power systems are commonly achieved by integration of an error signal containing numerous switchings, which results in a non-smooth objective functions [16]. The applicability of gradient-based optimization methods is restricted for non-differentiable objective functions. To address the issue of employing methods that do not require the gradient information of the objective function and also performing a global search, in the upcoming chapters a multi-resolution search strategy is proposed.

2.4 Multi-Resolution Function Reconstruction

Proposed response surface methodologies are based on sampling and estimating the objective function and then iteratively increasing the sampling rate around the local minima. This process can be seen as an iterative function reconstruction, where the refinement criteria is the closeness to the local minima. A generalized function reconstruction is addressed in [40], and the function reconstruction in cases of statistical functions is described in [41]. The function reconstruction in this thesis is achieved using multi-resolution sampling with a refinement criteria of re-sampling at the regions containing local minima.

Multi-resolution function reconstruction has been widely used for solving boundary value problems [42, 43]. In a boundary value problem the distribution of the points are intelligently selected to improve the resolution in places, where rapid changes occur in the differential equation. Similarly, the multi-resolution is achieved using a finite element mesh generation inside the boundary region of the optimization problem.

Multi-resolution function reconstruction is a combination of the response surface, direct search and gradient-based optimization methods. Initially a whole regular set of points is generated inside the region of optimization similar to grid-based search; then the surface of the objective function is reconstructed similar to response surface methods but via Delaunay triangulation [44]; and finally upon the triangulation, the gradient at every point is calculated to explore the possibility of capturing any trapped local minima in the proposed regions defined by the neighboring triangles (elements generated by Delaunay the algorithm) of each point.

The use of multi-resolution sampling avoids a large number of sampling points, while enabling the designer to investigate the whole region of optimization. This is highly dependent on the abilities of the mesh generation techniques. Various mesh generation techniques have been introduced, e.g., centroidal voronoi tessellation [45, 46] and implicit mesh generation [47]. In this thesis the implicit mesh generation is used to produce a mesh.

In the process of multi-resolution objective function sampling, the sampling points are derived as the nodes of the finite element mesh generated inside the optimization space and edges of the elements are the connections between these nodes, which represents the directions of derivative calculations as well.

2.4.1 Constraint Handling

The constraints considered in this thesis are stated explicitly by equality or inequality equations. The equality and inequality constraints are handled by restricting the mesh generation only to the regions satisfying these constraints.

Different approaches are needed to handle equality and inequality constraints, because inequality constraints only reduce the volume of the optimization space, while the equality constraints require sampling on specific manifolds satisfying them. These manifolds are necessarily at lower dimensions than the original optimization problem without equality constraints. These manifolds for example could be a line in a 2-dimensional space or a surface in a 3-dimensional space. The following sections review some available techniques addressing constraint mesh-generation.

2.4.2 Volume Mesh Generation

The primary stage of multi-resolution sampling of a function consists of sampling the objective function at nodes of an initial finite element mesh covering the entire optimization space. Such a method for mesh generation was introduced and developed in [47, 48]. This method is based on the mechanical analogy of a system of truss elements and the finite elements (simplexes) of a mesh. The algorithm works in two stages: (i) initially a set of random or regular points are generated; and (ii) the set of points are moved by the force exerted on them from the connecting bars (edges of the finite element mesh). The final mesh, after a few iterations, is supposed to be a conforming mesh in the regions satisfying all inequality constraints.

The proposed implicit mesh generation [47] is used again in the second phase of the multi-resolution sampling in order to refine the mesh in the regions containing local minima. The detailed process and algorithm is described in the upcoming chapters.

2.4.3 Surface Mesh Generation

The problem of multi-resolution sampling of a function only on a specific manifold, where equality constraints are satisfied, needs a surface mesh generation and a surface sampling technique. This new problem is defined as: (i) extracting an initial set of sampling points on the manifold satisfying the equality constraints; and (ii) connecting the sampling points by surface facets (elements) to find the local minima on the manifold.

The area of surface mesh generation has been widely investigated in mechanical design and computer graphics; but all these methods are complex, subject to some specific conditions, and cannot be generalized to higher dimensions. However, [49] proved the existence of a surface mesh for any Riemannian manifold given certain conditions; but still a generalized surface mesh generator has not been developed. Some specific generalized methods are presented only for 3-dimensional points [50], but these are not extendable to higher dimensions.

Recently, some practical methods have been introduced based on Voronoi filtering [51] and the advancing front technique [52]. Surface mesh generation using both of these methods are discussed in Chapter 4, and then a modified version of the advancing front technique is introduced as a basis for sampling and mesh generation. The algorithm is only implemented for a 2-dimensional example with one equality constraint; therefore, the optimization space will be a curve in 2-dimensional space in this example.

Chapter 3

OBJECTIVE FUNCTION DEFINITION AND OPTIMIZATION

3.1 Introduction

Design of power systems often involves selection of a number of parameters, including control systems parameters, element values, and firing angle sequences. Design problems involving optimizing controller coefficients in a linear system and some nonlinear cases can be done when an analytical model of the system is present. Recently, developments in the design of high-power semiconductor devices has further enabled incorporation of power electronic systems in power networks. The incorporation of these nonlinear and switching subsystems in various power systems has increased the complexity of power networks, thus analytical model are not readily available for these systems. The absence of analytical expressions to describe the detailed behavior of switching power devices prohibits the use of formerly developed design procedures.

Fast transient dynamics play an important role in the behaviour of switching converters. Therefore, the design of such nonlinear switching circuits (e.g., high-power static converters) requires a detailed representation of the system behaviour. Optimization-enabled electromagnetic transient simulation is designed to simulate the detailed behaviour of switching power systems and provide metrics for measuring the performance of the circuit.

3.2 Objective Function Definition

Simulation tools such as electromagnetic transient simulation (EMTS) programs [3] could provide an accurate representation of complex phenomena in power systems; thus they are being used as invaluable alternatives to analytical methods for the design and analysis of complex power systems.

Optimization-enabled electromagnetic transient simulation (OE-EMTS) is based on minimizing an objective function (OF), which is designed to provide a metric for the quantification of the consistency of the network behavior with the desired design specifications [1, 4]. The objective function is a measurement tool that provides a defined distance between the simulated behaviour and the ideal response. To clarify the definition of an objective function consider the following functions $f_1(x)$ and $f_2^k(x)$. It could be considered that the function $f_1(x)$ represents an ideal behaviour of a system while the function $f_2^k(x)$ represents the nonlinear behaviour of the corresponding simulation model of the system using the parameter k as its input.

$$f_1(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 3 \\ 2 & \text{if } 3 \leq x \leq 6 \end{cases} \tag{3.1}$$

$$f_2^k(x) = 1 + sin(11x) + \frac{e^{4\,k\,(x-3)^2}}{e^{40(x-3)^2} + 1} \tag{3.2}$$

where the parameter k should be adjusted so that the two functions resemble each other as clearly as possible. The two functions $f_1(x)$ and $f_2^k(x)$ are shown in Fig. 3.1 for $k = 10$.

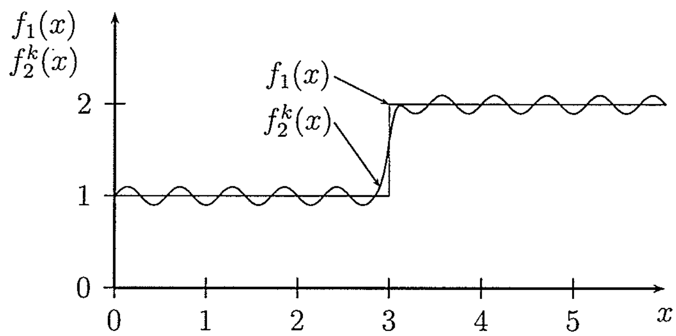


Fig. 3.1: *The functions $f_1(x)$ and $f_2^k(x)$ for $k = 10$.*

The two function are showing a similar behavior for $k = 10$, while if the parameter is selected to be $k = \frac{5}{4}$ then the the difference between the functions becomes more significant as shown in Fig. 3.2.

To measure the difference between these two functions, one can use the following

$$d(f_1, f_2) = \int_{x=0}^{x=6} \left| f_1(x) - f_2^k(x) \right| dx \tag{3.3}$$

The distance between the two functions is defined as the objective function (OF),

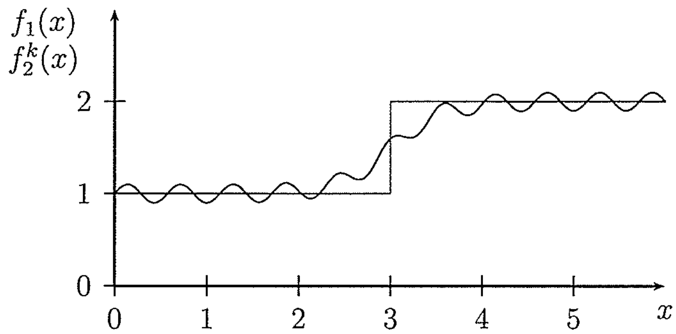


Fig. 3.2: *The functions $f_1(x)$ and $f_2^k(x)$ for $k = \frac{5}{4}$.*

which is a function of the system design parameters (e.g., controller parameters or element values).

3.3 Objective Function Optimization

Objective function evaluation using simulation is costly and time consuming because the system response must be achieved after the simulation is finalized and the distance is measured between the ideal response and the simulated result. In order to find the optimal set of parameters for the system, a search strategy must be employed to reduce the number of function evaluations before the optimum point is found. A nonlinear optimization algorithm guides the sets of test parameters towards the optimal set of parameters. This approach is often orders of magnitude faster than the conventional multiple-run search approach.

Knowledge of the objective function value for a set of system parameters requires one simulation; therefore, this kind of optimization problem is categorized as black-box function optimization. The evaluation of each set of parameters is computationally intensive. The optimization methods applicable to this type of optimization must require the least number of objective function evaluations for finding the local minima. The information provided after each simulation is only the objective function value, while achieving other information like Jacobian vector or Hessian matrix of the function at simulated points can not be achieved using a single simulation.

3.4 Constrained Optimization

Most engineering problems involve optimization, which are subject to several equality and inequality constraints. These constraints are stated either explicitly or implicitly as a function of design parameters. The implicit constraints depend on the OF value, while the explicit constraints are only stated based on the design parameters. In this thesis only explicit constraints are considered. The explicit constraints of the design parameters themselves, can be stated via explicit or implicit formulations.

The constraints are categorized as equality and inequality constraints. Different approaches are introduced for handling equality and inequality constraints. The presence of constraints essentially reduces the region of optimization to smaller volumes or smaller dimensions. Practically, any inequality constraint restricts the optimization problem to smaller volumes of the same dimension, while any equality constraint restricts the optimization problem to lower dimensional surfaces that satisfy equality constraints.

Fig. 3.3 shows an optimization problem with three inequality constraints; h_1, h_2, and h_3. These inequality constraints restrict the optimization space only to the hatched region. The equality constraint g_1 reduces the dimension of the optimization

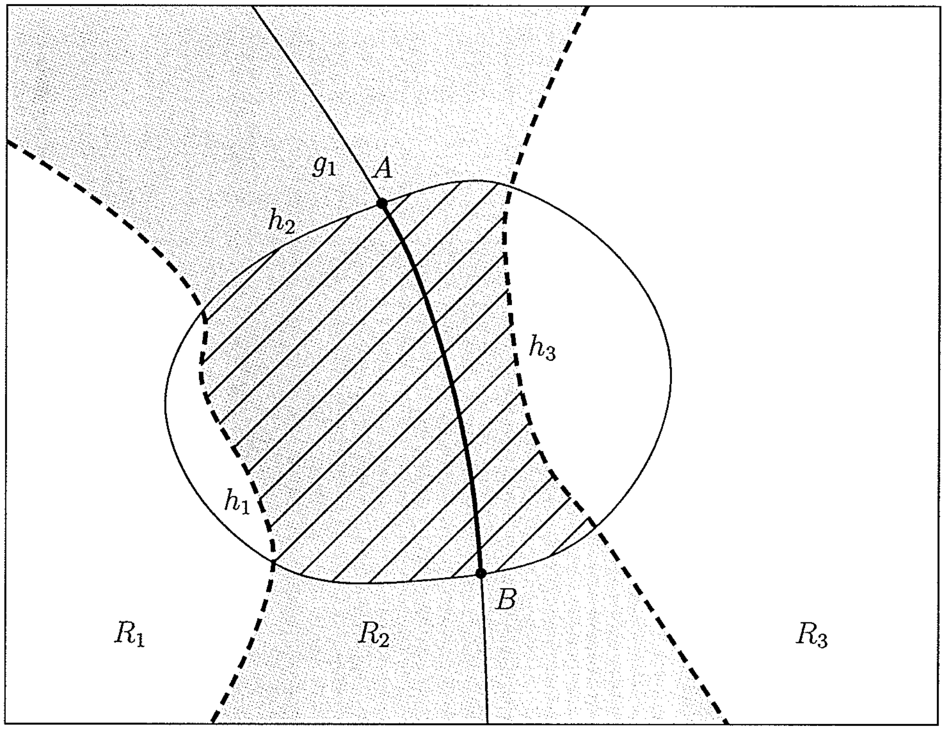


Fig. 3.3: *A constrained optimization problem.*

problem search space. The equality constraint reduces the optimization space to the curve AB, which is specified by g_1 bounded inside the hatched region. The addition of this equality constraint has reduced the optimization search space dimension from a 2-dimensional space to a 1-dimensional curve in the former 2-dimensional hatched region.

To clarify the statement above, consider a simple problem of optimization of a 1-dimensional function as introduced in (3.4).

$$f(x) = (2-x)^2 \tag{3.4}$$

The optimum point of this function is at $x = 2$; however by applying a constraint on x such that $x > 3$, the optimum point would become $x = 3$.

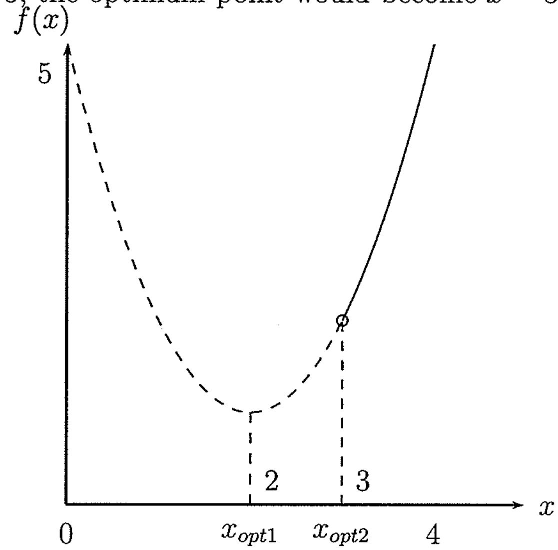


Fig. 3.4: *Constrained versus unconstrained optimization example.*

Similarly applying the equality constraint

$$(x-2)^2 = 1$$

would result in the optimization space shrinking to only two points; $x = 1$ and $x = 3$. Thus, the optimization space is reduced to a 0-dimensional space. The two optima with and without equality constraints are shown on the graph (x_{opt1} and x_{opt2}) in Fig. 3.4. Similarly in a 2-dimensional space an equality constraint would reduce the optimization space from a 2-dimensional search space to a 1-dimensional search space, but on a 2-dimensional curve.

A 2-dimensional optimization problem is stated here to clarify the concept. Consider the optimization problem in (3.5).

$$\begin{aligned}
&\text{Maximize} \quad OF(x,y) = xe^{(-x^2-y^2)} + \frac{1}{2} \\
&\text{Subject to:} \\
&-0.2 \leq x \leq 2 \\
&-0.2 \leq y \leq 2 \\
&g_1(x,y) = x^2 + y^2 - \frac{1}{4} = 0
\end{aligned} \tag{3.5}$$

The objective function is shown in Fig. 3.5. The optimum without considering the equality constraint is shown on the graph. The only constraints here are the inequality constraints. Applying the equality constraint of $g_1(x,y) = 0$ reduces the optimization space from the region of $-0.2 \leq x \leq 2$ and $-0.2 \leq y \leq 2$ to the 1-dimensional curve (g_1). The new optimum points must be located on the curve satisfying the equality constraint.

3.5 Multi-Modal Objective Function Optimization

The developed optimization-enabled electromagnetic transient simulation approach uses optimization algorithms such as simplex or gradient-based methods that start the optimization process from a given point provided by the designer. This approach is guaranteed to converge to a local minima, which is not necessarily the global minimum of the objective function.

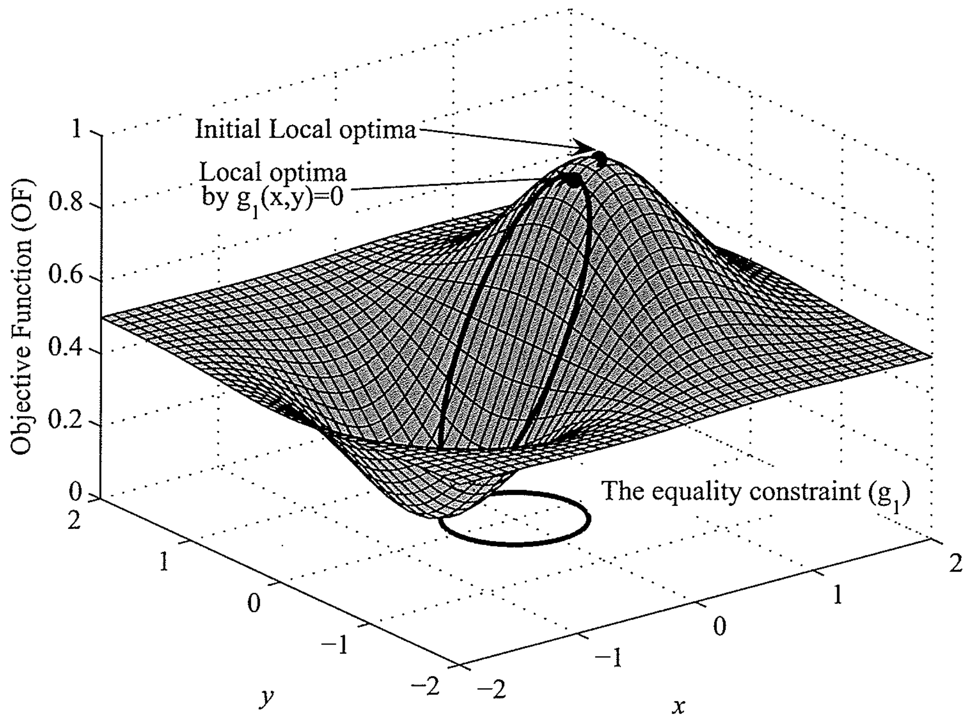


Fig. 3.5: *The optimization space using an equality constraint.*

Consider function in (3.6) whose graph is shown in Fig. 3.4.

$$f(x) = \frac{3}{10}\left((x-3)^2 - 0.9sin(11x) + 1\right) \tag{3.6}$$

The function contains 11 local minima in the range $[0, 6]$. A gradient-based optimization method could converge to different local minima for different initial values. A gradient based optimization method started from the point x_{init1} may converge to x_{opt1}, while one starting from x_{init2} may converge to the global point of x_{opt2}.

This phenomenon, which is referred to as multi-modality happens in situations,

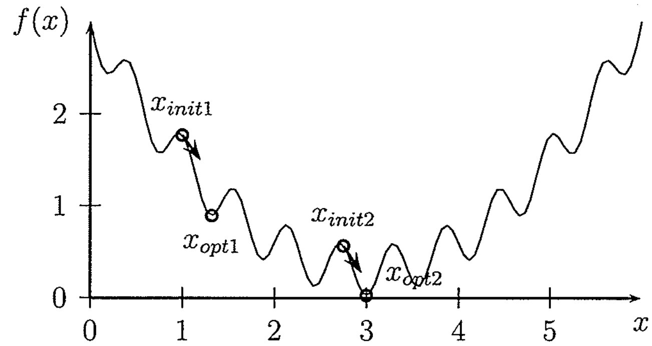


Fig. 3.6: *Gradient-based optimization.*

where a system contains more than one local minimum [5]. Optimization of functions with this multi-modality property normally requires use of exhaustive search methods such as sequential search. In this thesis a new method for global optimization of multi-modal objective functions is developed using the multi-resolution approach for reconstruction of the objective function (OF). The multi-resolution algorithm reduces the number of required simulations from a power law relationship in the sequential search to a linear relationship in the multi-resolution approach.

Chapter 4

Multi-Resolution Optimization

4.1 Introduction

This chapter provides the details of the proposed multi-resolution optimization algorithm. This method is a combination of the direct search method [6] and the response surface method [25], which have been formerly applied for global optimization of simulation-based problems. This new method performs an initial sampling of the whole optimization space [22] similar to the response surface method and then iteratively generates new sampling points via a direct search strategy, but using an irregular mesh. The deployment of the irregular mesh enhances the ability of the algorithm to handle the inequality constraints.

This new algorithm samples the whole optimization space using a multi-scale mesh in order to avoid excessive sampling at regions containing no local minima. This set of sampling points is achieved using a finite element mesh, where the sampling points are the nodes of the mesh and the connections between the points are the edges of the elements (simplexes). Upon completion of the first coarse sampling of the whole optimization space, the method finds the smaller regions that contain local minima; this process is called localization. These smaller regions are set of simplexes from a finite element mesh. In the next step a new set of points are generated to refine the mesh inside the regions where the local minima have been localized. The process of refinement and localization continues until the local minima of the function are found with specified accuracy in terms of closeness of the sampling points.

Initial sampling using a coarse mesh would fail to detect some highly sensitive local minima. These highly sensitive local or global minima are not considered for a practical design, because a sensitive global or local minimum is not guaranteed to have the same objective function in practice compared to its performance in a simulation case. This usually happens due to changes in the system conditions or other design parameter values during time or loading conditions; therefore, the less sensitive local minima are preferred.

The success of the algorithm in finding the local minima is mainly dependent on the proper implementation of the above mentioned stages of initial mesh generation, localization of local minima within the mesh, and refinement of the mesh inside the localized regions. The initial mesh generation is implemented using an existing algorithm [47, 48] based on a force relaxation method. The localization is addressed using two different approaches: graph-based and cell-based localization. The cell-based localization is described in the next chapters and is used to localize the promising areas for optimization. The refinement process is implemented either by centers-of-mass mesh refinement or by a new mesh generation technique using a force relaxation method.

4.2 Multi-Resolution Search Algorithm

Direct search methods such as grid search subdivide the search space into smaller grids and search the whole sampling regions in order to find local minima [8]. The uniform sampling of the whole optimization space requires a large number of samples. An option to decrease the large number of sampling points is to perform a coarse sampling of the whole optimization space and then, using the knowledge gained through this initial sampling, the next sampling points will be generated near local minima.

The initial sampling points, which are assumed to cover the entire optimization space, play the most important role in determining whether the algorithm misses or captures the local minima of the objective function. A dense initial sampling captures the most sensitive local minima, while a very coarse initial sampling would capture less sensitive ones. The multi-resolution optimization algorithm involves several iterative steps. Fig 4.1 shows an overview of the overall implementation of the multi-resolution algorithm.

The advantages of multi-resolution search compared to grid search originate from adaptive sampling of the objective function, as sampling is avoided in regions containing no local minima. The multi-resolution search implemented here is aimed at handling equality and inequality constraints; therefore, the search is done on an irregular grid capable of conforming to the constraints. A finite element grid is a suitable option for handling nonlinear boundaries. Finite elements have the ability to match nonlinear inequalities by covering the region with simplexes (elements) of different sizes.

Fig. 4.2 shows a finite element mesh generated to satisfy the inequality constraints

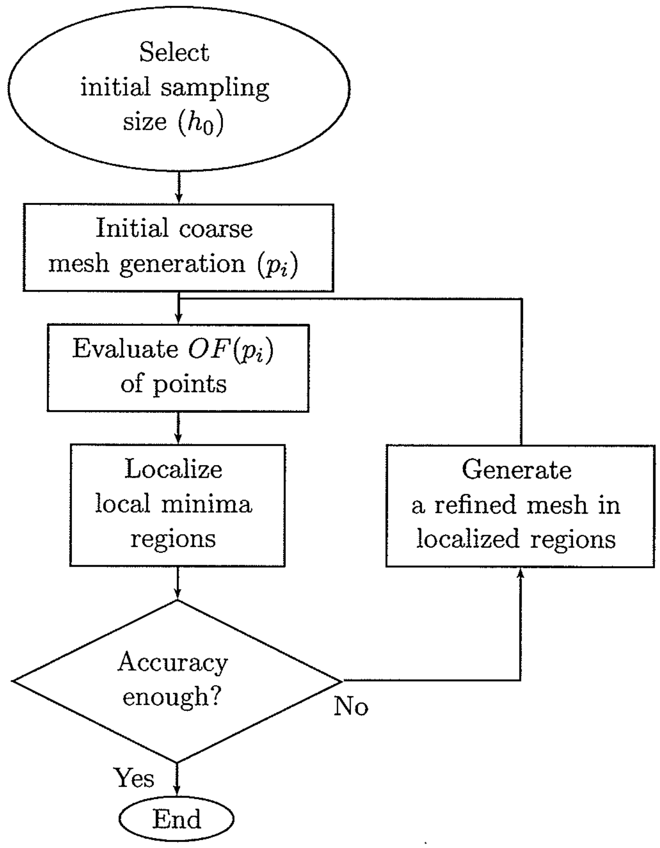


Fig. 4.1: *The multi-resolution optimization.*

in (4.1).

$$
\begin{aligned}
-6 \leq x \leq 6 \\
-6 \leq y \leq 6 \\
16 - x^2 - y^2 \leq 0
\end{aligned}
\tag{4.1}
$$

The handling of the same nonlinear constraints using a regular grid needs more sampling points in order to have a conforming mesh at the boundaries, because a

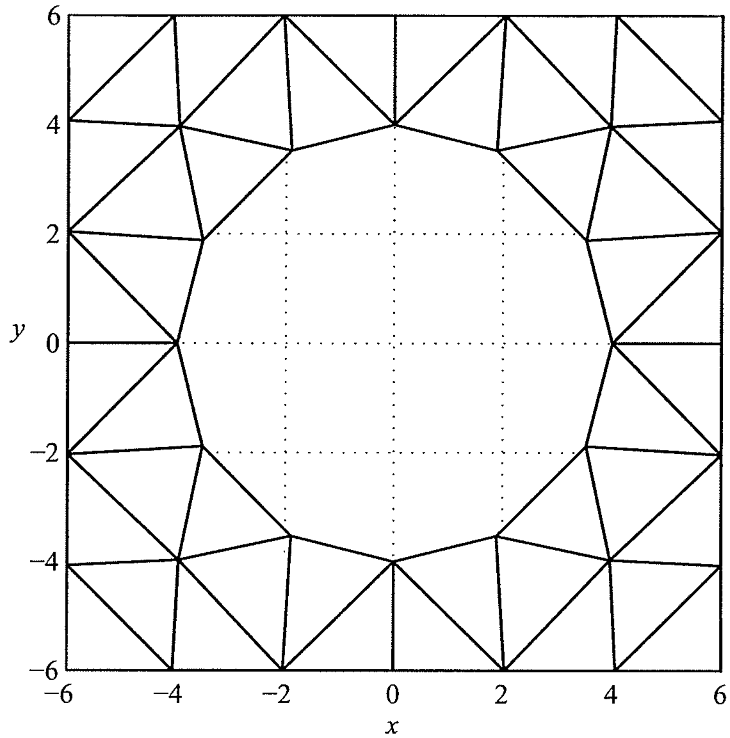


Fig. 4.2: *Inequality constraint.*

finite element mesh has more freedom on positioning the points of a mesh in order to be located on the boundaries to avoid extra sampling.

4.3 Multi-Resolution Implementation for Optimization

The multi-resolution representation of a function has been used previously in solution of boundary value problems [42] in order to reduce the computation effort in achieving the final solution. The idea is to use adaptive sampling to avoid excessive computation in regions where the variation of the function is low and consequently increase the sampling rate in regions with more variations [43]. The criteria of over-sampling and under-sampling are determined via an estimation of the function variation, which could be determined analytically or by a rough initial solution. The criteria of over-sampling in multi-resolution algorithm is determined by the closeness to the local minima and the initial estimate is attained by evaluation of nodes of an initial finite element mesh inside the optimization space. The coordinates of the finite element mesh nodes represent the sampling points and the elements of the mesh (simplexes) are used later to derive connections between the points in the sampling space.

The multi-scale mesh provides a tool for multi-resolution sampling of the optimization space. The multi-scale mesh is generated through an iterative processes of sampling and refining, where only the regions close enough to the estimated local minima are refined. These local minima regions are detected using the localization procedure. The new points are generated as a new set of nodes for the finite element mesh. The coordinate of these new points represent the new sets of samplings of the optimization space.

The following objective function is selected as a standard example used throughout this chapter to represent the implementation of different stages of the multi-resolution optimization algorithm [53].

$$\begin{aligned}
&f(x,y) = (x^2 + y - 11)^2 + (y^2 + x - 7)^2 \\
&-6 \leq x \leq 6 \\
&-6 \leq y \leq 6
\end{aligned} \tag{4.2}$$

The coordinates of local minima of function in (4.2) are tabulated in Table 4.1. These were produced using a gradient-based solver with four different initial sampling point (e.g., Mathcad).

Figs. 4.3 (a) and (b) show contour plots of function in (4.2) and the corresponding generated multi-scale mesh using a multi-scale sampling method which will be described later in the thesis. The generated mesh does not have a uniform element size. The element sizes vary according to the closeness to the local minima. Using this approach the local minima could be easily found without any need for a high sampling rate over all the optimization space.

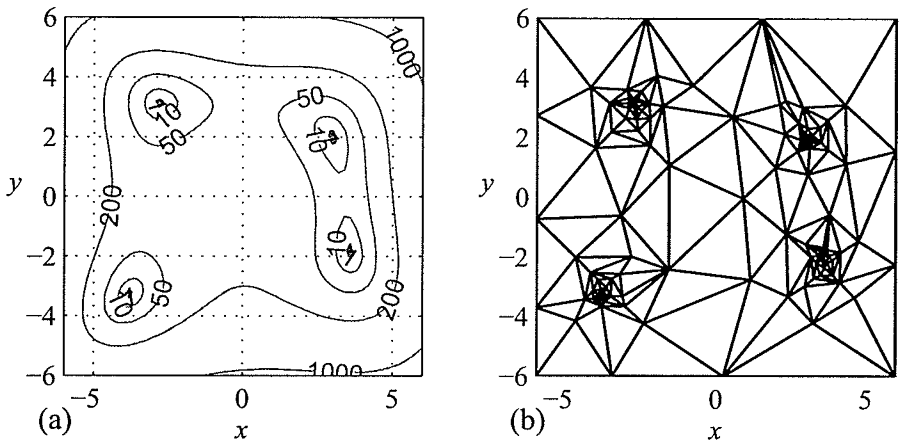


Fig. 4.3: *Multi-resolution mesh for optimization.*

Table 4.1: *Local minima coordinates of function in (4.2).*

Coordinates of the local minima, $f(x,y) = 0$ for all four points			
$(-3.78, -3.28)$	$(3.58, -1.85)$	$(3.0, 2.0)$	$(-2.8, 3.13)$

The procedure of multi-scale mesh generation adopted for function optimization is an iterative process consisting of the following three major parts:

1. Initial mesh generation;

2. Localization of local minima;

3. Refinement of the mesh in the localized regions.

The following sections describe in detail these different parts and some approaches for their implementation.

4.4 Initial Mesh Generation

The ability of the multi-scale optimization method in finding local minima is dependent on the initial mesh covering the whole region of optimization. There is always a tradeoff between the probability of capturing local minima and the cast of many evaluating initial sampling points.

The initial mesh generation must satisfy the following conditions:

1. The sampling points must be only in the optimization space;

2. The whole optimization space must be covered by finite elements;

3. The algorithm must have the ability to distribute the initial points upon a designer's a-priori knowledge of the potential location of local minima.

The problem of mesh generation within the implicit boundary conditions was introduced in [47, 48]. In mesh generation method the boundary conditions are specified by n implicit inequalities as

$$h_i(p_j) \leq 0, i = 1, \ldots, n \text{ and } j = 1, \ldots, N \tag{4.3}$$

where p_j is a point belonging to the initial mesh, N is the total number of initial mesh points, and n is the number of inequality constraints. The inequality constraints $h_i(p_j)$ are satisfied by ensuring that the generated sampling points (p_j for $j = 1, \ldots N$) fall in regions with the property of $h_i(p) \leq 0$ for $i = 1, \ldots, n$. The proposed implicit mesh generation is based on an analogy between mechanical truss elements and the connected nodes of a finite element mesh, where the sampling points are modeled as truss joints and the truss elements are the connections between these joints. The joint connections are generated using Delaunay triangulation algorithm. The Delaunay triangulation algorithm connects the nodes with elements (simplexes). The sampling point connections are derived as the edges of these finite elements (simplexes).

The Delaunay algorithm provides the set of triangles T_i, $i = 1, \ldots, m$ connecting the points. Fig. 4.4 shows four sampling point p_1, p_2, p_3, and p_4. The set of Delaunay triangulations for these points are

$$T = \{T_1, T_2\}$$

$$T_1 = \{p_1, p_3, p_4\}$$

$$T_2 = \{p_1, p_2, p_3\}$$

The two points p_i and p_j are connected if the following condition is satisfied.

$$\exists T \mid \{p_i, p_j\} \in T \Rightarrow p_i \text{ and } p_j \text{ are connected}$$

Fig. 4.4 shows a set of four points and their corresponding connections generated using the two triangles derived by the Delaunay triangulation.

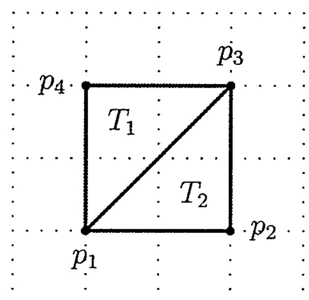


Fig. 4.4: *The Delaunay triangulation and point connections.*

Different approaches for implicit mesh generations have been proposed; in this thesis mesh generation based on the force relaxation method [47] is chosen to provide the sampling points all over the optimization space. This method is selected due to its ease of implementation for handling any nonlinear inequality constraint. The force relaxation is also used for mesh refinement.

The overall algorithm of mesh generation using force relaxation method is listed below:

1. Initial regular or random sampling points are selected inside the optimization space;

2. The points outside the boundary conditions are removed;

3. The new points are moved in the direction of the total force exerted on it until the appropriate distribution of points is achieved.

Random or regular sets of points do not necessarily provide a conforming sampling of the optimization space; therefore, these new points must be relocated in order to (i) avoid excessively close initial sampling; and (ii) to meet the inequality constraints. The relocation of the points is done using the analogy between a set of finite elements and a network of truss elements. The truss elements exert forces on their nodes until the network becomes stable and the total forces at all nodes become zero. The nodes then would represent a conforming distribution of the points. Fig. ?? shows the process of mesh generation. The force relaxation method is described in the next section.

The total points of the mesh are represented by

$$P = [p_1, \ldots, p_N] \tag{4.4}$$

4.4.1 Mesh Enhancement Using Force Relaxation

The mesh generation method used in this thesis is based on the implicit mesh generation method using the force relaxation method [47]. This mesh generation algorithm starts with a regularly or randomly spaced mesh of points and then moves the points in an iterative approach to confine the search only to the optimization space. The force relaxation method is employed in order to satisfy the following two conditions: (i) placing the nodes (sampling points) to cover the whole optimization space; and (ii) to enable the designer to redistribute the initial sampling points by prior knowledge (this could be specified as a density function).

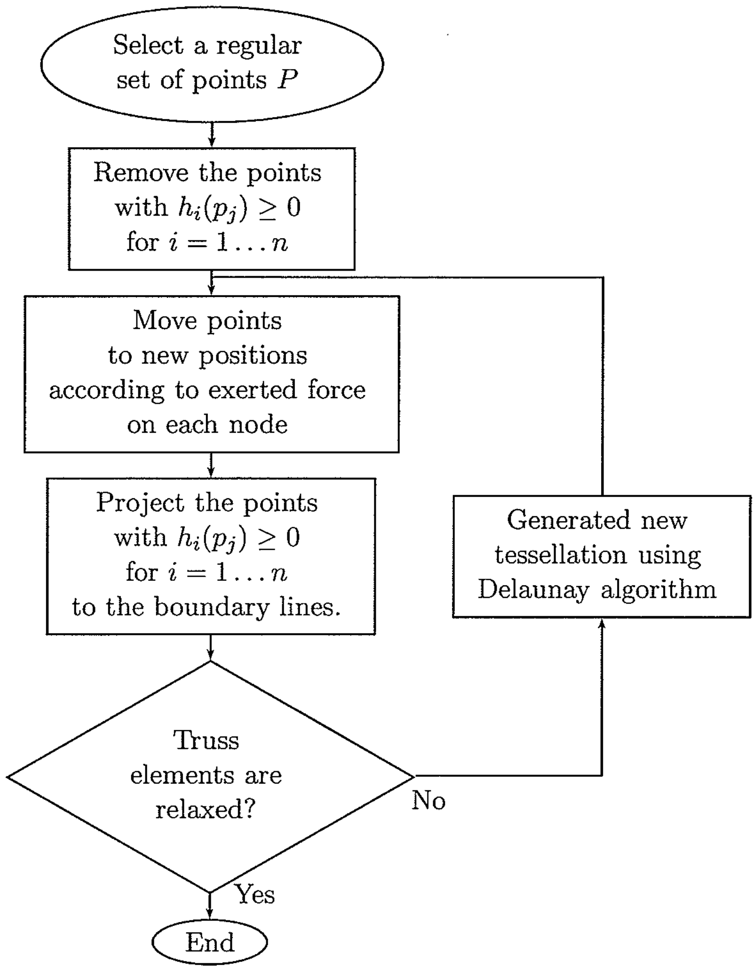


Fig. 4.5: *The initial mesh generation.*

In force relaxation method the points are modeled as nodes of a mechanical truss structure and the connections between the points are modeled as spring elements. Fig. 4.6 shows an example of sampling points and their connections as the spring elements.

The forces exerted on the nodes are from the following two sources

1. Forces exerted from other nodes;

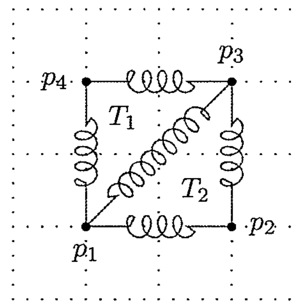


Fig. 4.6: *Modeling of point connections by spring elements.*

2. Forces exerted from boundaries.

Fig. 4.7 shows a node on the boundary and the forces exerted on the node from the neighboring elements and the boundary. The force from the boundary to the node is considered to ensure that the sampling points remain inside the specified boundary conditions.

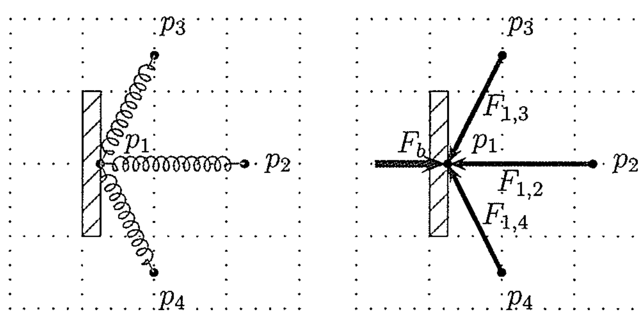


Fig. 4.7: *The internal and external forces on a boundary node.*

The initial mesh is generated when all the nodes become stationary or, equivalently, the internal and external forces on all nodes become zero. The nonlinear equation in (4.5) specifies a condition, which must be satisfied by the mesh points in order to guarantee a conforming distribution of initial sampling points.

$$F_t(P) = \left[\vec{F}_{p_1}, \ldots, \vec{F}_{p_N}\right] = 0 \tag{4.5}$$

where P is the collection of all points ($P = [p_1, \ldots, p_N]$), $\vec{F}_{p_j}$ is the total force exerted on the j^{th} node, p_j is the point representing the j^{th} node, F_t is the collection of all these forces in a matrix format, and N is the total number of points. The columns of this matrix are the forces exerted on each node. The system of equations in (4.5) is a nonlinear set of equations and could not be solved explicitly; therefore, an iterative approach is used to find a solution to the equation. The points are moved according to the differential equation in (4.6).

$$\frac{d\vec{P}}{dt} = \vec{F}(P) \tag{4.6}$$

The differential equation is solved iteratively using euler method.

$$\delta\vec{P} = \vec{F}(P).\delta t \tag{4.7}$$

where δt is assumed to be the time step for solution of the differential equation.

The points movement stop when $d\vec{P}$ becomes zeros, which means also that the net force has become zero and consequently (4.5) is satisfied. The mechanical analogy between the truss structure and the nodes distribution inside the optimization space must be achieved by modeling of the node links in the finite element mesh by spring elements. The force-displacement relation of the nodes links (springs) is specified by a the following nonlinear characteristics.

$$f(l) = \begin{cases} k(l - l_0) \text{ if } l \leq l_0 \\ 0 \text{ elsewhere} \end{cases} \tag{4.8}$$

where the l_0 is the expected length of the spring and force-displacement factor is defined as

$$k = \frac{l + l_0}{2l_0} \tag{4.9}$$

where l is the current length of the spring element and l_0 is the expected length of the element connecting two points. The link sizes are assumed to reach approximately to their corresponding expected length when the truss structure becomes stable. The expected length is determined using a density function. The expected length is calculated as in.

$$l_0(i,j) = h_d(\frac{p_i + p_j}{2}) \left(\frac{\sum\limits_{\forall k,m|\exists l_{k,m}} \|l_{k,m}\|^2}{\sum\limits_{\forall k,m|\exists l_{k,m}, p_k, p_m \in l_{k,m}} h_d^2(\frac{p_k + p_m}{2})} \right)^{\frac{1}{2}} \tag{4.10}$$

where $l_0(i,j)$ is the expected length of the link connecting node i to node j, $l_{k,m}$ specifies the connecting link of node k to node m, if such a connection exists, the operator $\|l_{k,m}\|$ returns the length of the the link connecting node k to node m, and h_d is the density function specifying a distribution for the size of elements in the mesh. The summation is held only on the connected points in the mesh, for example if node i is connected to node j then the $l_{i,j}$ exists and represents the vector connecting these two points. In steady state, the points distribution will resemble the density function h_d.

Equs.(4.8) and (4.9) represent a spring with the following characteristics

1. The spring exerts repulsive force for $l < l_0$.

2. The attractive force is zero for $l \geq l_0$.

A spring at the three different lengthes of: (i) $l \leq l_0$; (ii) $l = l_0$; and (iii) $l \geq l_0$ is shown in Fig. 4.8. The repulsive forces ensure that the final points do not come too close to each other.

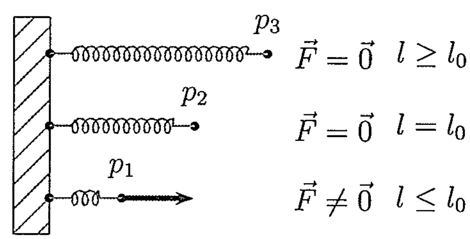


Fig. 4.8: *The force-displacement characteristics of a spring element.*

The distribution of the nodes can be altered using different density functions. Fig. 4.9 shows a simple density function.

The final mesh generated is expected to have a dense sampling in regions with lower density function, because the connection of the points are expected to be lower due a less expanding force between the connected points. Fig. 4.10 shows the final generated mesh using the specified density function, which clearly has higher density around the point $(2, 2)$.

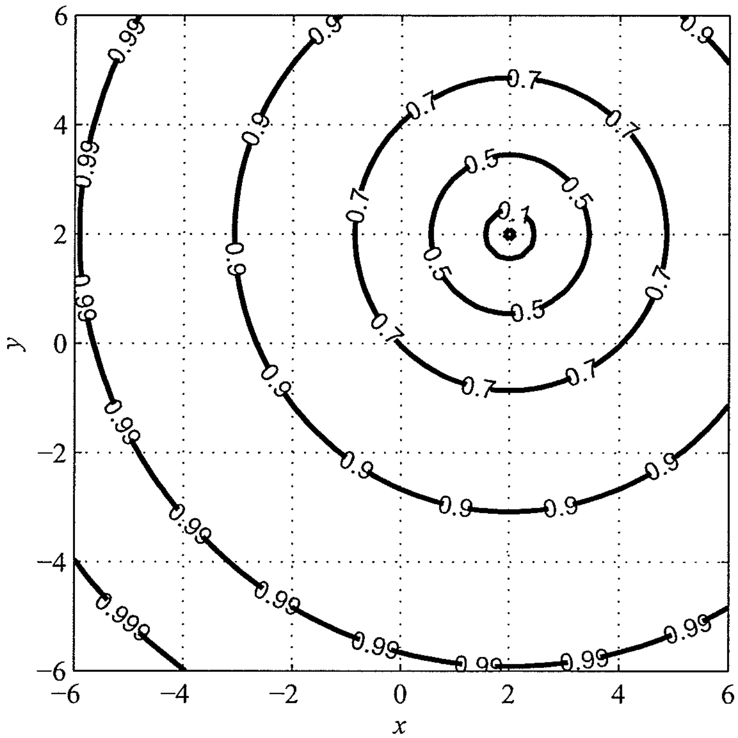


Fig. 4.9: *A non-uniform density function for initial mesh generation.*

4.5 Localization

The multi-scale sampling of the objective function is aimed at increasing the sampling rate in regions where local minima exist. The finite element mesh subdivides the space into smaller regions. These smaller regions are the elements of the mesh. The local minima are located within some of these elements (simplexes). The process of localization is performed in order to find the set of simplexes that contain local minima. In this thesis two techniques are introduced for the localization of local minima within the elements (simplexes) of a mesh. The first proposed method compares each point's objective function with the objective functions of its nearest neighbor

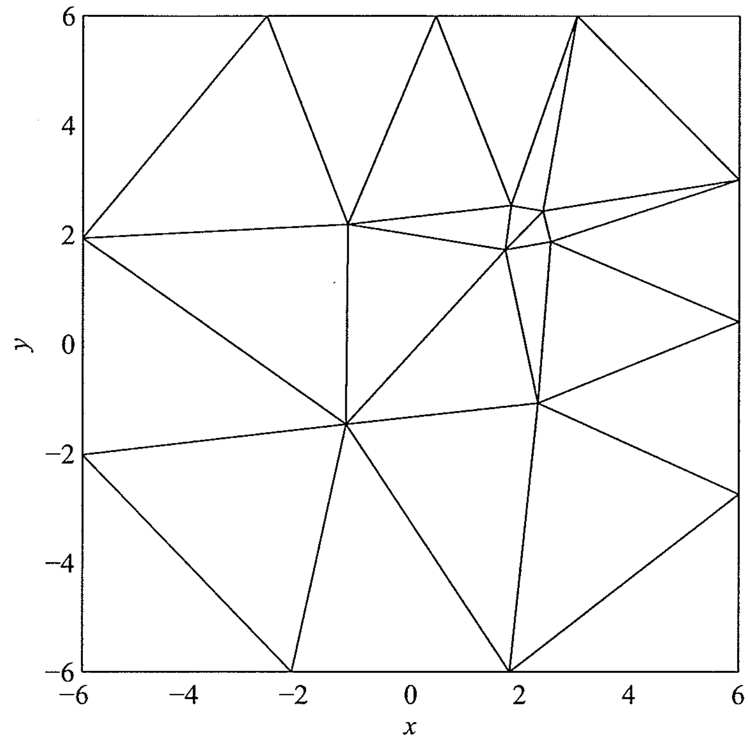


Fig. 4.10: *A non-uniform initial mesh generation.*

points. This method is referred to here as the cell-based method. The second proposed method is a simplex-based method which builds a directional graph, connecting the simplexes inside the mesh as nodes of the graph. The nodes (elements) inside the graph that contribute to a loop are the set of simplexes (elements) that contain local minima. This method is referred to here as the graph-based method.

4.5.1 Cell-Based Localization

The cell-based method is a tool for localizing the elements, where local minima exist. The cell-based method is guaranteed to find at least one region containing a local minimum, because there is always one point with the least objective function value in a set of evaluated points. The set of nearest neighborhood points of a single point inside a finite element mesh is defined as the point's cell and the point is called the center of the cell. These points and the cell-center point share at least one element (simplex). The collection of these points forms a cell and the cell-center point is located inside of them. Fig. 4.11 shows the cell of the point p_c, which is formed by its six surrounding points.

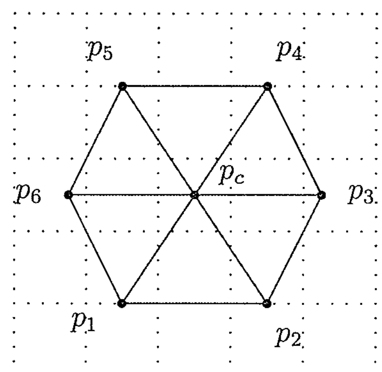


Fig. 4.11: *The cell definition.*

The points $p_1, \ldots, p_6$ form the cell of the point p_c. This cell could contain a local minimum if the condition in (4.11) is satisfied.

$$OF(p_c) \leq OF(p_i) \text{ for } i = 1, \ldots, n_{p_c} \tag{4.11}$$

where n_{p_c} is the number of neighbor points around p_c and $OF(p_i)$ represents the objective function value specified by the set of coordinates of p_i. Every cell satisfying this condition is selected for further refinement, because there is possibility of existing a point with zero first order derivatives near this selected region. Every point with zero second derivatives is considered to be a local minimum.

Fig. 4.12 shows the selection of a suitable set of simplexes to be refined in the next iteration for optimization of the problem specified in (4.2). The cell-based method localizes some simplexes for further refinement. The selected simplexes are labeled with dots. These elements are refined in the subsequent iterations using a mesh refinement technique.

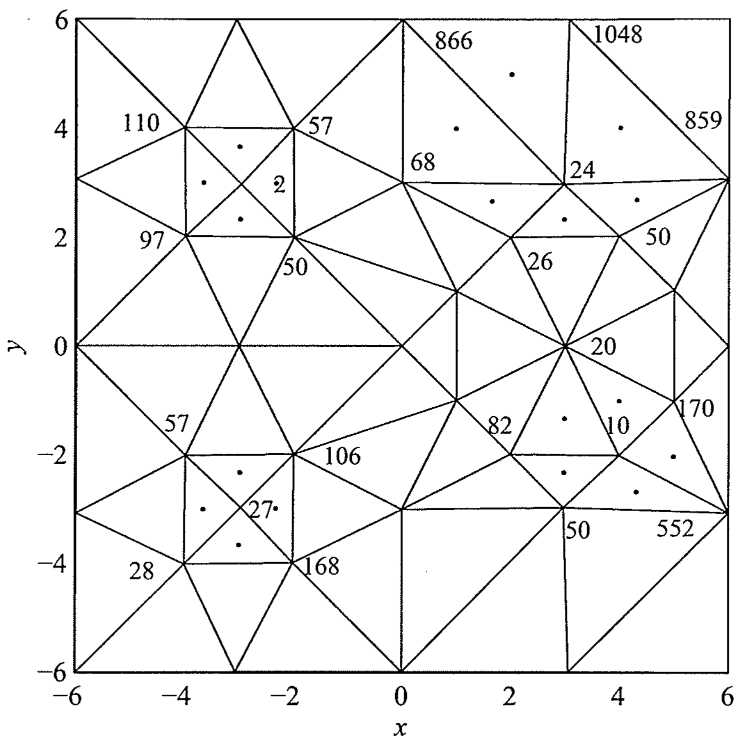


Fig. 4.12: *Localization of local minima for further mesh refinement.*

The roots of the function in (4.2) are found by multi-resolution optimization using cell-based localization. Table 4.2 shows the results of optimization. Comparison of the results of optimization using the multi-resolution algorithm and the solutions derived by a numerical solver in Table 4.1 shows the ability of the cell-based method to effectively find the location of local minima within a mesh.

Table 4.2: *Location of the local minima in (4.2).*

Coordinates of the local minima, $f(x, y) = 0$ for all four points			
$(-3.78, -3.28)$	$(3.58, -1.85)$	$(3.0, 2.0)$	$(-2.8, 3.13)$

4.5.2 Graph-Based Localization

A new method for localization is now introduced using a combination of graph search and Nelder's Simplex optimization method. The method is referred in this thesis as graph-based simplex method of localization. The graph-based simplex method is an extension to simplex optimization method, except that, unlike the simplex method, the new simplex is selected from the neighboring elements of the current simplex in the finite element mesh. This method connects every simplex to another simplex in a directional graph, where the nodes of the graph represent the elements (simplexes) of the mesh.

Every simplex in n-dimensional space consists of n+1 nodes and n+1 neighboring elements. Fig. 4.13 shows a simplex and its neighboring elements. A simplex a is connected to simplex b if the facet connecting the two simplexes corresponds to the node in simplex a with the highest objective function value. Fig. 4.13 shows the element (simplex) T_1 and three neighboring elements (simplexes); T_2, T_3, and T_4. The objective function of each node of the element T_1 is evaluated. The element T_1 is connected to the element T_3, which is opposite to the node p_1. The node p_1 is selected, because it has the highest objective function value of all other nodes of the element T_1.

$$OF(p_1) \geq OF(p_2) \text{ and } OF(p_1) \geq OF(p_3)$$

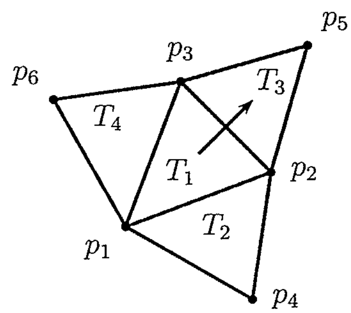


Fig. 4.13: *The simplex connections in a graph-based localization method.*

The graph-based localization method connects the simplexes of the mesh and creates a connectivity graph (see Fig. 4.14). In this new graph the elements of the mesh are the nodes of the graph and the connections are made between these nodes (simplexes). The set of connections between these elements makes the connectivity graph. The resulting graph will be a tree ending in a loop. These loops in the connectivity graph contains the simplexes that contain the local minima. These are identified in Fig. 4.14 . The number of independent graphs would vary according to the number of discovered local minima.

The simplexes that contribute to a loop in the connectivity graph are refined in the subsequent iterations to increase the resolution of the local minima. Fig. 4.14 shows the connections between the simplexes (nodes of the connectivity graph) for the objective function in (4.2) and four identified loops of a, b, c, and d are identified on the graph by dotted ellipses. The simplexes selected for more refinement are identified using small circles. The center of mass of the localized simplexes are added at the next iterations to the existing mesh in order to increase the resolution of the local minima within the mesh.

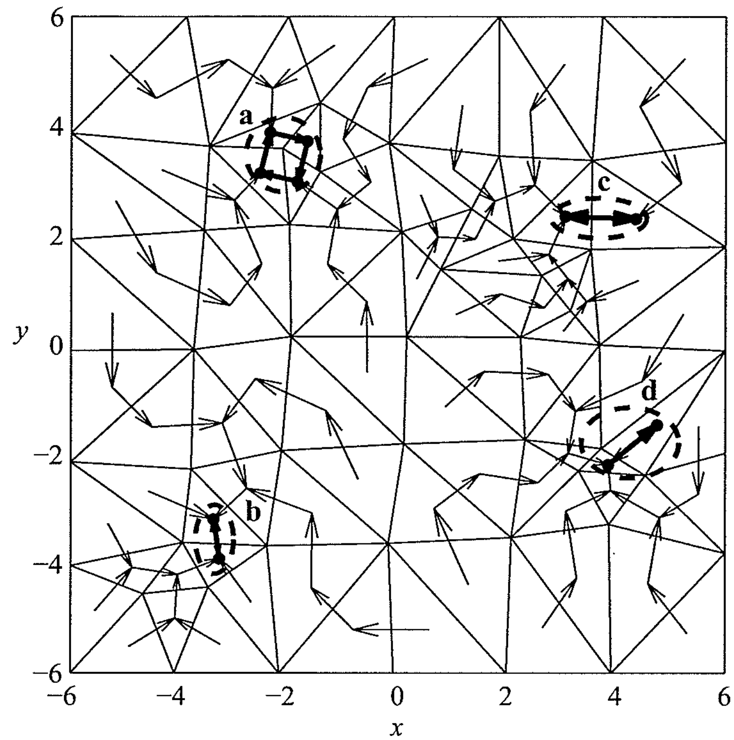


Fig. 4.14: *Graph-based localization using connectivity graph.*

Table 4.3 shows the roots found using graph-based localization method. The comparison of these local minima and the roots found using numerical solution shows the ability of the new graph-based localization to select the appropriate simplexes for further refinement. The graph-based method has the ability of finding the locations of local minima, but it needs more iterations to reach the same accuracy of as the cell-based method. The graph-based method also needs more computation for recognizing the loops in the constructed graph, because the loops are not all of the same size.

Table 4.3: *Optimization result of (4.2) using graph-based localization.*

Coordinates of the local minima, $f(x, y) = 0$ for all four points			
$(-3.94, -3.62)$	$(3.64, -1.85)$	$(3.0, 1.93)$	$(-2.79, 3.12)$

The graph-based localization method has shown to have more abilities to find the local minima in case that the initial sampling is done using a coarse mesh.

4.6 Mesh Refinement Techniques

The resolution of the local minima locations are increased in each iteration of the algorithm using mesh refinement. In this thesis two methods for mesh refinement are presented. The first method is based on refinement of a selected set of simplexes by adding their centers of mass to the previous mesh and the second method of mesh refinement generates a new mesh near the localized regions, where local minima are estimated to exist, using the force relaxation method. The former needs less computation, while the latter has the ability of controlling the number of sampling points. The center-of-mass mesh refinement technique finds limited application in higher dimension, because the number of simplexes sharing one node increases rapidly, while the latter technique controls the number of sampling points to a limited number.

4.6.1 Center of Mass Mesh Refinement

Center-of-mass mesh refinement works by adding the centers of mass of the simplexes belonging to the localized cells to the mesh. Fig. 4.15 shows the centers of mass of the localized cells. These points are added to the existing mesh as new refinement points. The points are then evaluated and a new approximation for the objective function is reconstructed. The new mesh is generated again using the Delaunay triangulation. Fig. 4.16 shows the refined mesh after one iteration of localization and mesh refinement achieved by adding points corresponding to simplexes centers of mass.

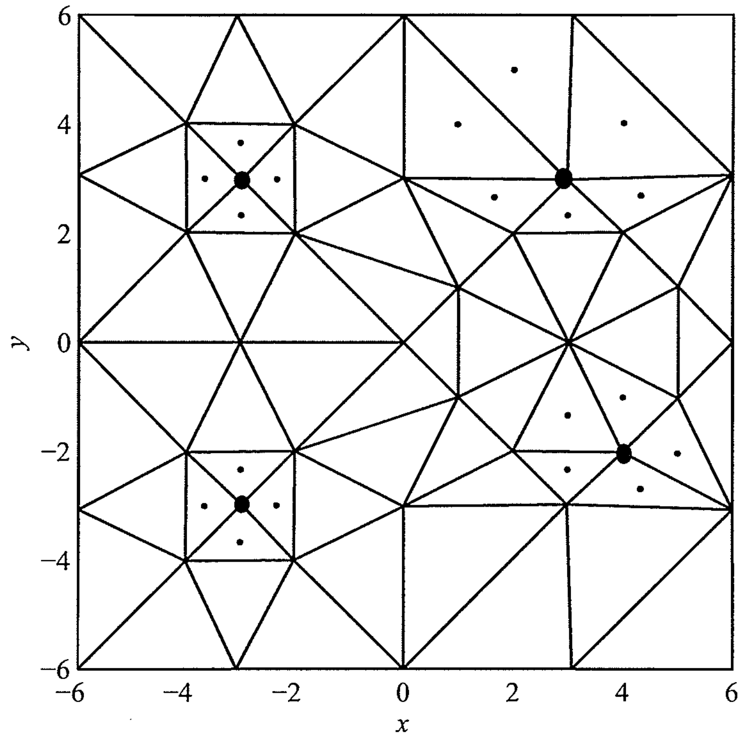


Fig. 4.15: *Mesh refinement using centers of mass of simplexes.*

The center-of-mass mesh refinement lacks the ability of restricting the number of generated points for mesh refinement. The number of elements contributing to one cell is typically small in a 2-dimensional mesh, but this phenomena is not the same for higher dimensions. The center-of-mass mesh refinement technique is avoided in the next sections due to its lack of efficiency at higher dimensions.

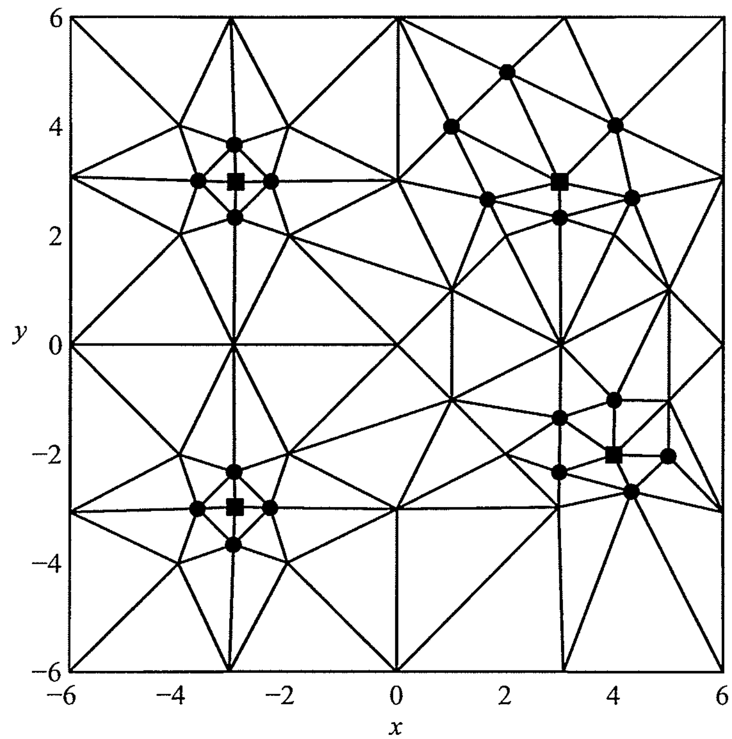


Fig. 4.16: *Refined mesh using centers of mass method.*

4.6.2 New Mesh Refinement Technique

This new method is introduced based on force relaxation method for mesh refinement. This method adds new points near the cell-center point, which is selected using the cell-based localization method for further refinement. The new points are added as the corners of a cube with a size equal to the medium length of the edges connecting the cell-center point to its surrounding points. Fig. 4.17 shows a selected cell and the added points around the center of the cell. These new points are added on the corners of a cube with the size determined by l_{avg,p_c} (see Fig. 4.17).

The force relaxation problem is solved in a number of iterations until the new

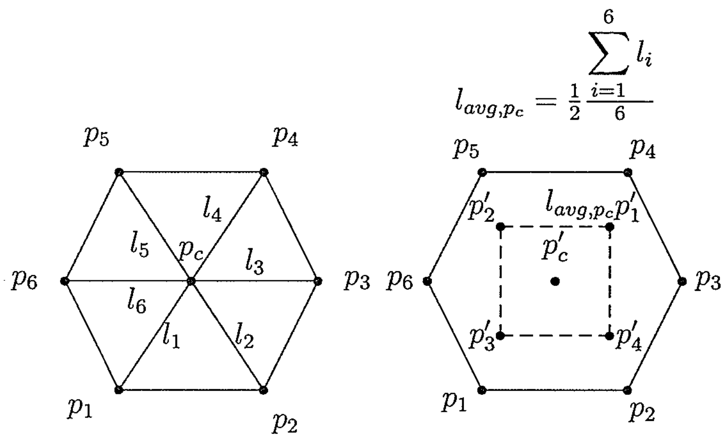


Fig. 4.17: *Cell-based refinement.*

points are repositioned. The previous points of the mesh are held constant during these iterations. The density function of the mesh distribution is modified in order to enable a refined mesh generation around the selected cell by reducing the density function near the localized cells using a gaussian-basis functions:

$$h_{d,new}(\mathbf{x}) = h_d(\mathbf{x}) \prod_{i=1}^{n} \left(1 - \alpha e^{-\frac{(\mathbf{x} - p_{c,i})^2}{2\, l^2_{avg,p_{c,i}}}} \right) \tag{4.12}$$

where $p_{c,i}$ is the center of the i^{th} localized cell, $l_{avg,p_{c,i}}$ is the average size of the cell (as defined in Fig. 4.17), and α is the reduction of the density function, which is selected to be 0.9 (this was selected empirically using experimental optimizations). The gaussian-basis function is selected here because of its ability to reduce the density function at a specific location and its ease of computation during mesh generation. Fig. 4.18 shows the initial added points and Fig. 4.19 shows the repositioned points after the force relaxation is solved in a few iterations. Fig. 4.20 shows the final generated mesh using the proposed new mesh refinement technique. This final mesh represents different resolutions in different areas of the objective function. The sampling rate is increased in regions containing local minima, while the regions containing no local minima are abandoned.

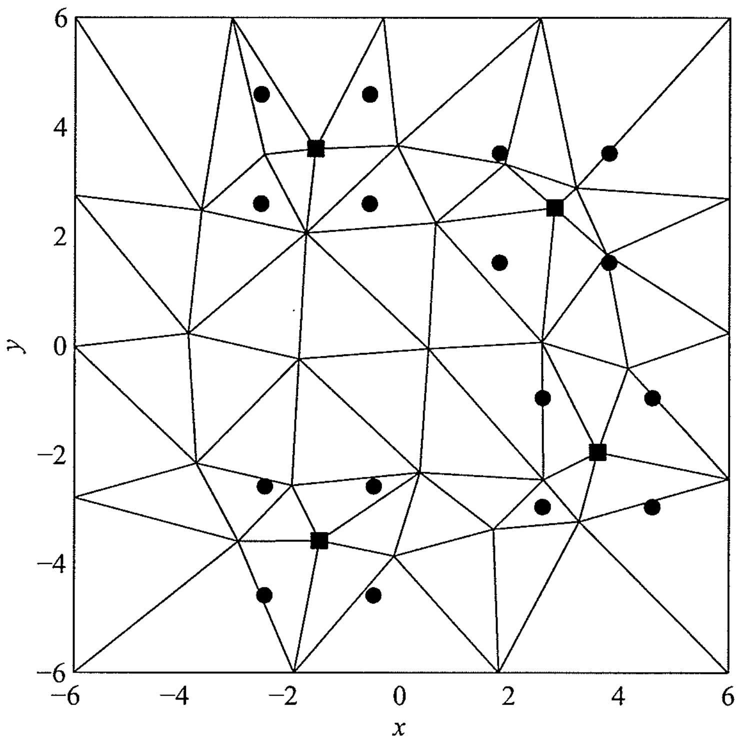


Fig. 4.18: *Initial added points for the new mesh refinement technique.*

4.7 Constraint Handling

The constraints are either specified by inequality constraints, equality constraints, or both. The inequality constraints are handled by restricting the sampling and mesh generation to only be inside the regions specified by the inequality boundaries. The

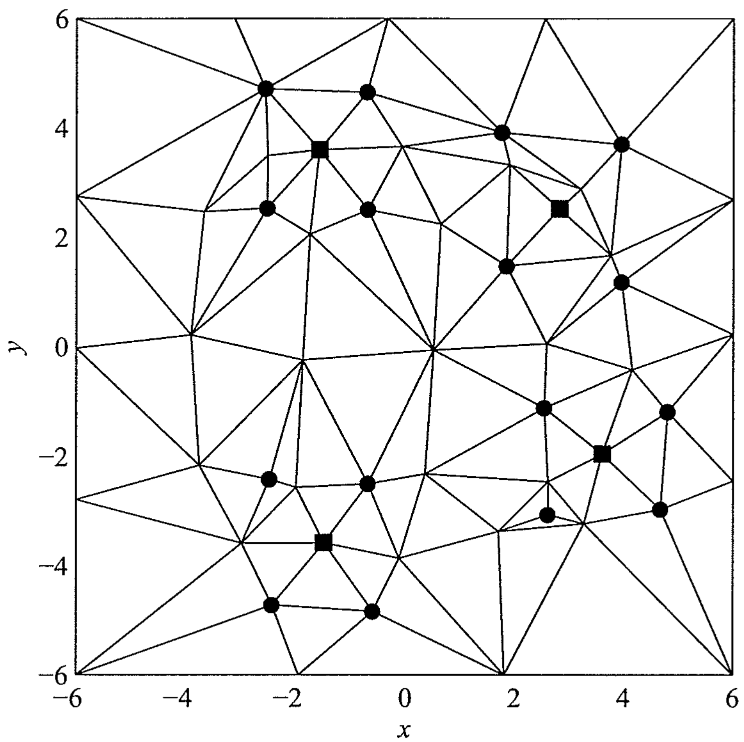


Fig. 4.19: *Repositioned points after force relaxation for new mesh refinement.*

equality constraints are handled by generating a mesh on the set of points satisfying the equality constraints, which is necessarily at a lower dimension than the original problem.

4.7.1 Inequality Constraints

Implicit mesh generation technique provides an efficient tool for restricting the sampling space only to the regions satisfying the inequality constraints [47]. The inequality constraints are handled by exerting forces on nodes which are at the boundaries of the inequality constraints. These forces are modeled by returning back the

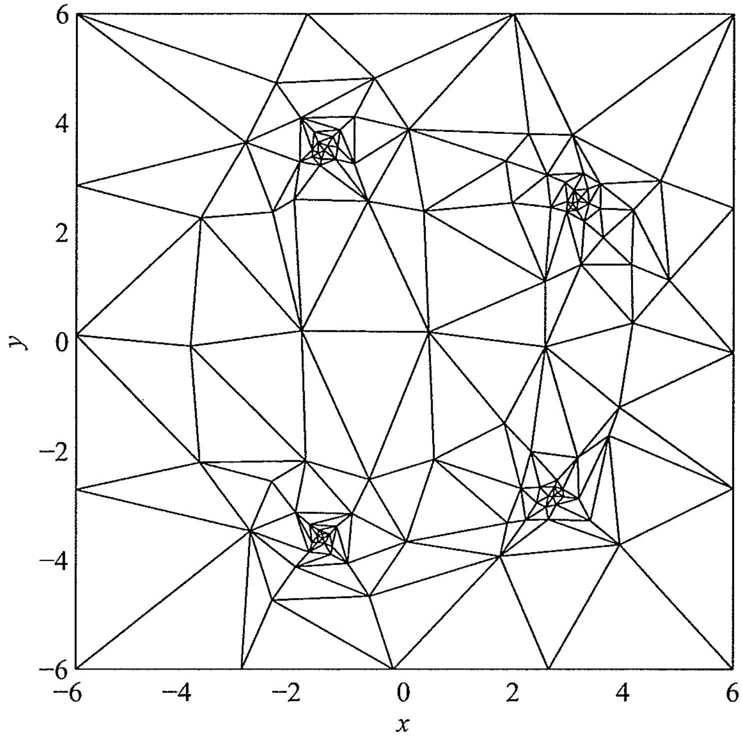


Fig. 4.20: *Final generated mesh using force relaxation.*

points which have crossed the inequality constraints to the boundary defined by these constraints using a gradient based solver. As an example, consider the optimization problem with three inequality constraints specified in (4.13).

$$
\begin{aligned}
& f(x,y) = (x^2 + y^2 - 16)^2 + (y^2 + x - 10)^2 \\
& -6 \leq x \leq 6 \\
& -6 \leq y \leq 6 \\
& 49 - (x-6)^2 - (y-6)^2 \leq 0
\end{aligned}
\tag{4.13}
$$

Figs. 4.21 (a) and (b) show the initial and final mesh generated for multi-resolution optimization of the objective function in (4.13).

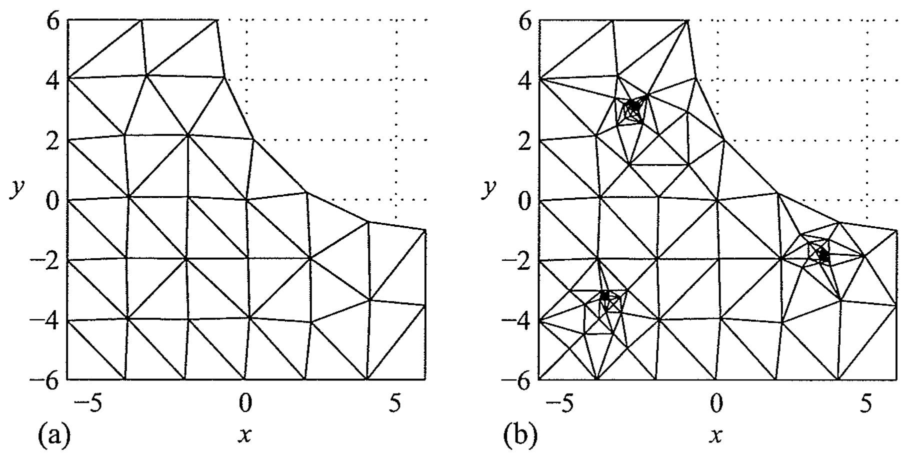


Fig. 4.21: *Initial and final meshes for the inequality constraint.*

The results of optimization is shown in Table. 4.4.

Table 4.4: *Coordinates of the local minima of the function in (4.13).*

Coordinates of the local minima, $f(x, y) = 0$ for all four points		
$(-3.78, -3.28)$	$(3.59, -1.82)$	$(-2.8, 3.12)$

The set of local minima presented in Table. 4.4 shows the ability of implicit mesh generation to handle inequality constraints. One local minima, which was located in the abandoned region is not derived and was automatically eliminated from the set of solutions.

4.7.2 Equality Constraints

Equality constraints are handled by restricting the sampling points to be on the surfaces satisfying the equality constraints. The equality constraints define new surfaces in the optimization space; for example, an optimization problem in 2-dimensional space with an equality constraint will be reduced to optimization only on a curve specified by the equality condition. In this thesis two methods are introduced to handle the equality constraints by generating a mesh on the surfaces or curves that satisfy these conditions.

Voronoi Filtering

Voronoi filtering [51] is introduced for generating the surface mesh of scattered data in 3-dimensions, but it is not guaranteed to work for every kind of point distribution on a surface. This method could be generalizable to higher dimensions using algorithms capable of deriving voronoi points for these higher dimensions.

The Voronoi points of a set of initial points are the corners of the collection of separate regions. Each region corresponds only to one of the initial points. The points located in every one of these regions, which corresponds only to one of initial set of points, are closer to the selected initial point than to any other point of the initial set of points. Voronoi filtering adds the voronoi points of the original set of points and forms a new set of points; then the Delaunay triangulation is generated for the combined set of voronoi points and the original set of points. The set of elements from the Delaunay triangulation are filtered by accepting elements containing only one point from the voronoi set of points. The surface elements are then derived from the remaining elements by removing the voronoi points. Fig. 4.22 shows scattered data on the surface of a unit sphere and the resulting surface elements covering the sphere.

The voronoi-based surface mesh generator only provides the surface elements,

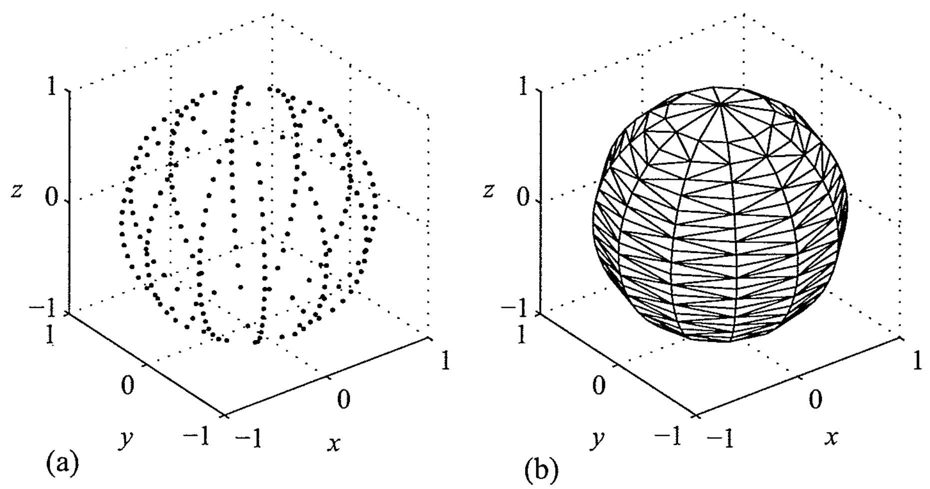


Fig. 4.22: *Surface mesh generation using voronoi filtering.*

while it cannot provide a useful tool for deriving a set of sampling points of the surface defined by the equality constraint. The generation of surface elements is an essential part of localization procedure.

Advancing Front

The advancing front technique is widely used to construct surface meshes. In this approach an arbitrary mesh is generated inside the region where the equality constraint is defined. The set of elements, which are crossing the equality constraint are found and refined by adding their centers of mass to the existing points, for example consider the following equality and inequality constraints.

$$f(x, y) = (x^2 + y - 11)^2 + (y^2 + x - 7)^2 \tag{4.14}$$

$$-6 \leq x \leq 6 \text{ and } -6 \leq y \leq 6 \tag{4.15}$$

$$8y^2 + x^2 = 16 \tag{4.16}$$

Fig. 4.23 shows the initial mesh generated inside the equality constraints and the selected elements, which are crossed by the equality constraint. The selected elements are colored in gray. These elements are selected based on having nodes with objective function values of different signs. Fig. 4.24 shows the crossed elements after further refinements.

From the selected elements (the elements crossed by the equality constraints) one node with the least distance to the equality constraint is selected. That node is then projected to the equality constraint using a gradient-based solver. The gradient-based solver receive this point and solves the nonlinear equation corresponding to the equality constraint. This results in projection of the initial node to the manifold defined by the equality constraint. These new points are the initial samplings of the optimization space. Fig 4.25 shows the initial samplings of the optimization space, defined by the combination of the inequality and equality constraints.

The initial sampling points are evaluated and the regions possessing local minima are detected using the cell-based method. The refinement process is done by selecting the mass centers of the simplexes belonging to the selected cells (e.g., one cell here consists of one point and its two connected simplexes). The optimization space here is a curve and the simplexes are the elements connecting these sampling points. This

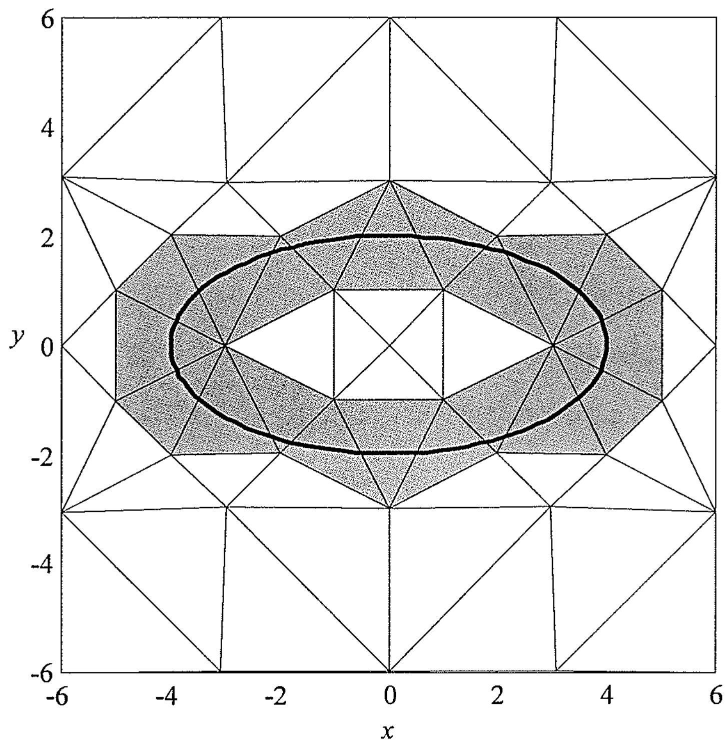


Fig. 4.23: *Initial selection of elements crossing the equality constraint.*

process continues until the local minima are found with enough accuracy. Fig 4.28 shows the final mesh and the final detected local minima. The result of optimization is shown in Table 4.5.

Table 4.5: *Coordinates of the local minima of the function in (4.16).*

Coordinates of the local minima, $f(x, y) = 0$ for all four points			
$(-3.03, -1.31)$	$(3.43, -1.03)$	$(3.08, 1.27)$	$(-2.44, 1.58)$

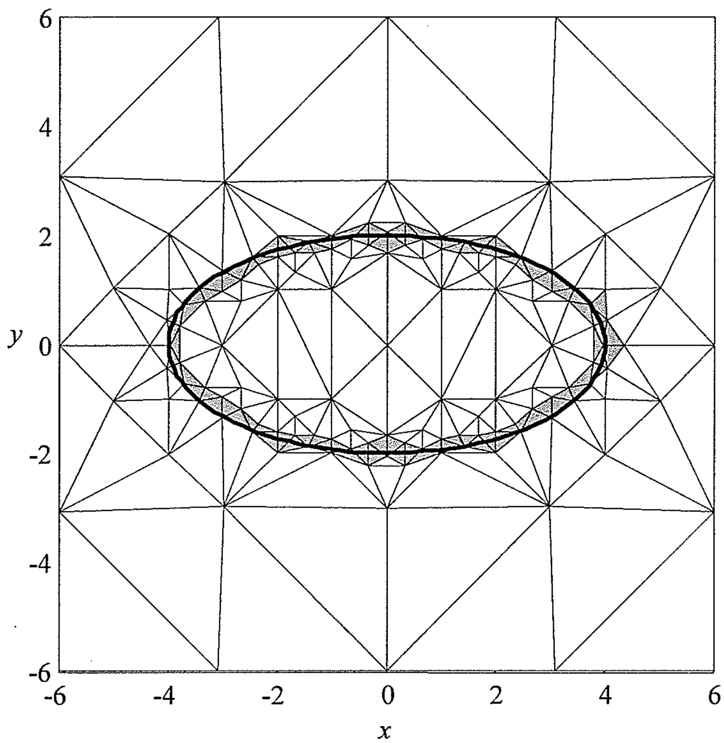


Fig. 4.24: *Elements crossing the equality constraint after 4 level refinement.*

4.8 Parametric Studies

In this section a parametric study is done to investigate the effect of initial non-uniform mesh distribution and force relaxation on rate of convergence to local minima.

4.8.1 Effect of Mesh Distribution

The initial mesh distribution could be specified using the force relaxation method. The effect of the initial mesh size is investigated on the rate of convergence of solutions

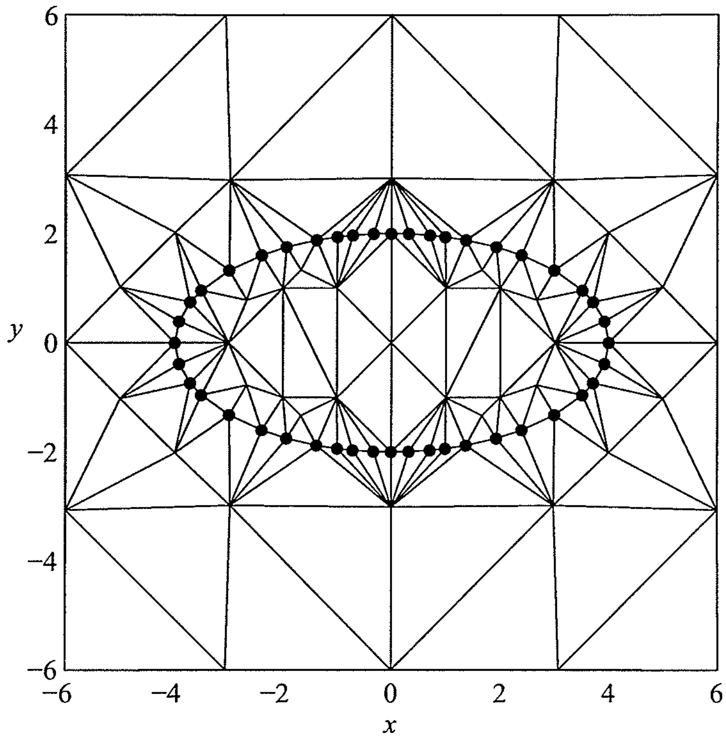


Fig. 4.25: *Initial sampling of the equality constraint.*

to local minima. The function in (4.2) was selected to be investigated. Fig. 4.27 (a) and (b) show a non-uniform initial mesh and a uniform initial mesh generated for multi-resolution optimization of the function in (4.2). The square root of the objective function itself is used here as the density function to determine the mesh distribution.

Fig. 4.28 shows the rate of convergence for the objective function in terms of the number of iterations. The rate of convergence for the non-uniform mesh is higher than the uniform mesh.

This example shows that the designer could specify an estimated objective func-

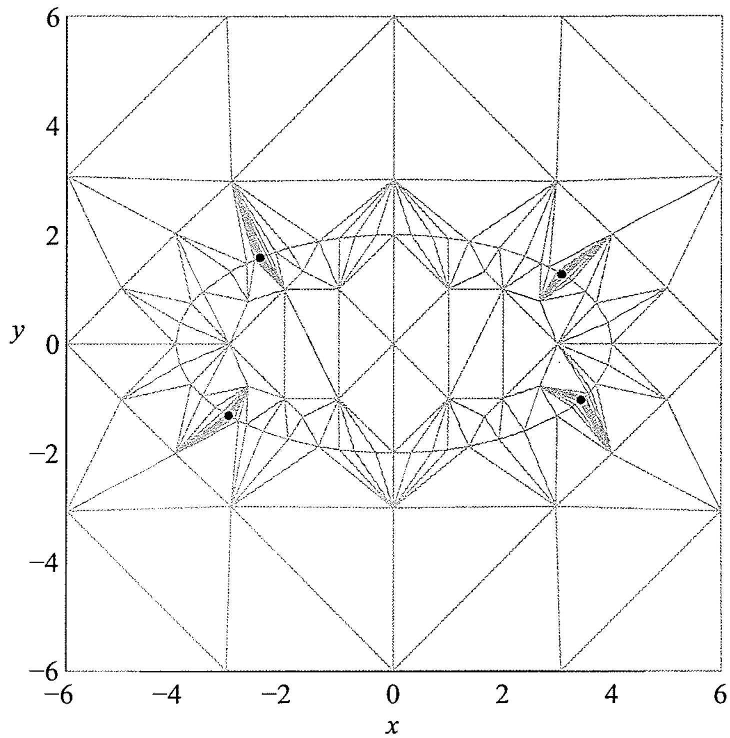


Fig. 4.26: *Final generated mesh for the equality constraint.*

tion prior to initial sampling point generation in order to enhance the rate of convergence to local minima. The rate of convergence is increased because more initial points are generated inside the regions containing local minima. This estimated objective function could be used by force relaxation method to redistribute the initial mesh nodes in order to increase the sampling rate in regions with greater possibility of containing more local minima. This is very useful in the case of meta-modeling optimization, where different estimates of the objective functions may be available at low cost [19].

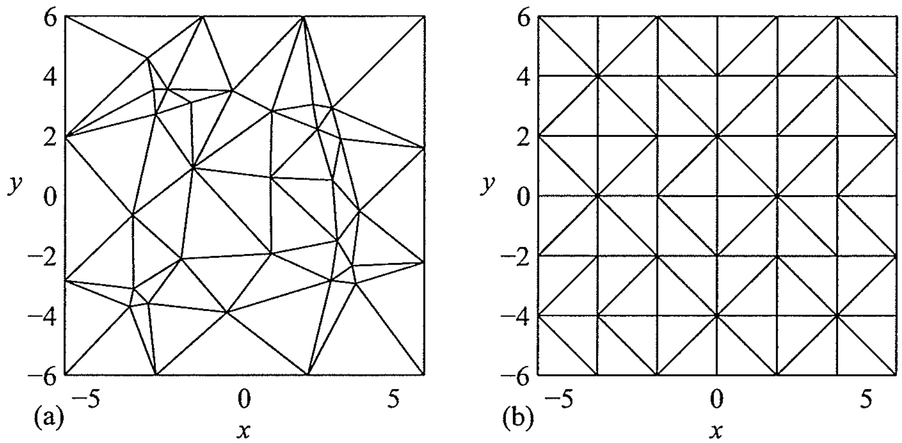


Fig. 4.27: *(a) Non-uniform initial mesh; (b) Uniform initial mesh.*

4.8.2 Effects of Force Relaxation

The number of discovered local minima is related to the initial sampling size of the objective function. A coarse initial sampling may miss some local minima, while a fine initial sampling might find all local minima. However, the number of initial sampling could not be very large for time consuming simulation cases; therefore, the force relaxation is used to redistribute the points in the optimization space to maximize the amount of information gathered from the initial sampling. The effect of force relaxation was investigated on the number of found local minima.

Fig. 4.29 (a) and (b) show the final generated mesh for optimization of (4.2) with an initial mesh size of $h_0 = 4$. The initial generated mesh has converged to all four local minima, while the initial generated mesh without force relaxation has only found two local minima.

Table 4.6 shows results for different instances of the multi-resolution optimization method implementation with and without force relaxation. As can be seen, by using

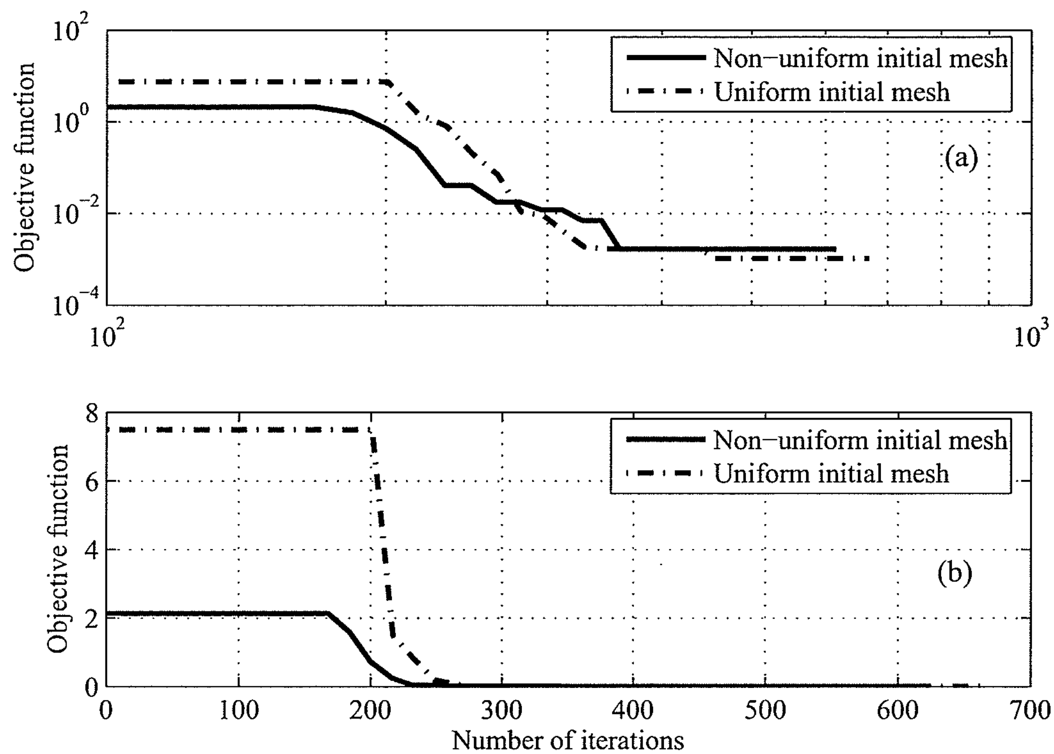


Fig. 4.28: *Convergence rate in (a) log-log plot and (b) regular plot.*

force relaxation, an initial mesh of 14 points could find all local minima, while without force relaxation only two were found. These results show that the force relaxation method using a proper density function for initial mesh, enables the algorithm of maximum usage of initial sampling points.

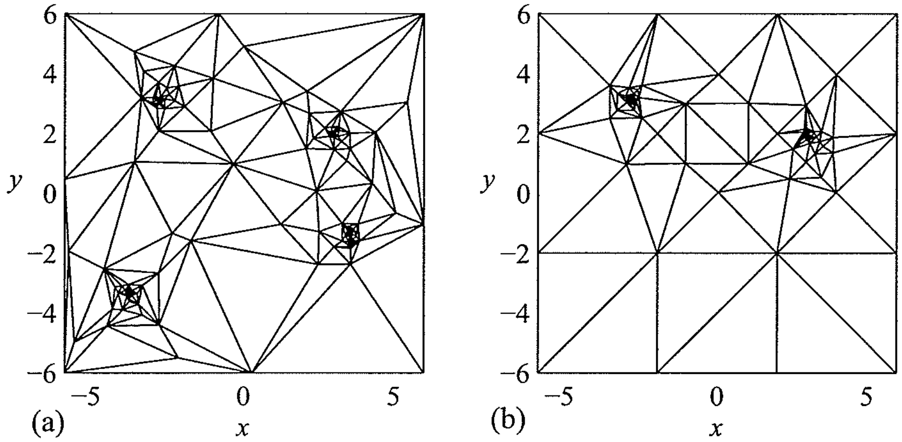


Fig. 4.29: *Effect of force relaxation for initial mesh on finding local minima.*

Table 4.6: *The effect of force relaxation for derivation of local minima.*

Initial mesh size	Initial number of samples	Number of local minima: with force relaxation	Number of local minima: without force relaxation
1	175	4	4
2	46	4	4
3	23	4	3
4	14	4	2
4.5	12	2	2

Chapter 5

Modifications to Multi-Resolution Optimization

The extension of the multi-resolution optimization algorithm to higher dimensions is subject to the following obstacles: (i) the total number of sampling points for the initial mesh increases as n^d, where n is the total number of grid divisions in each direction and d is the dimension of the optimization problem; and (ii) the Delaunay triangulation becomes highly memory demanding at higher dimensions, because the number of simplexes generated by the Delaunay triangulation increases exponentially for higher dimensions than 4.

The first obstacle could be be solved using parallel computation and the second can be solved by introducing a partial mesh generation technique that only generates the connections of required points in each iteration and avoids using Delaunay triangulation.

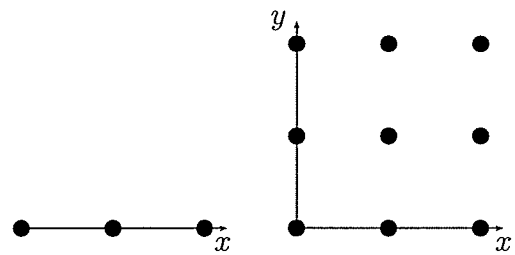


Fig. 5.1: *Number of initial sampling points for a 1 and 2-dimensional problem.*

5.1 Parallel Implementation

The proposed multi-resolution optimization algorithm performs an initial sampling of the entire optimization space. Considering a simple example of such a sampling only by three points in each direction, the number of initial sampling points for a 1-dimensional case is 3, for 2-dimensional case is 3^2 and for a three dimensional case is 3^3. Fig. 5.1 shows the initial sampling points on a regular grid for a 1-dimensional and a 2-dimensional case. Considering a simulation-based optimization of 6 variables with a simulation time of 2 minutes and an initial sampling rate at 6 division in each direction, the initial mesh evaluation would need 65 days. This makes the algorithm inefficient for optimization using simulations at higher dimensions.

Despite the large number of initial samplings, the algorithm is suitably parallelizable, because in each iteration the new sets of points can be evaluated by a distributed network of machines. Once the set of points (sets of parameters) have been evaluated the multi-resolution algorithm generates a new set of points. Fig. 6.1 shows a parallel implementation strategy for the multi-resolution algorithm. Here, in every iteration, the set of sampling points ($[p_1, \ldots, p_n]$) are evaluated using n simulation agents one per machine; therefore, the computation time is reduced to the number of iterations performed by the multi-resolution algorithm assuming there are n available processors.

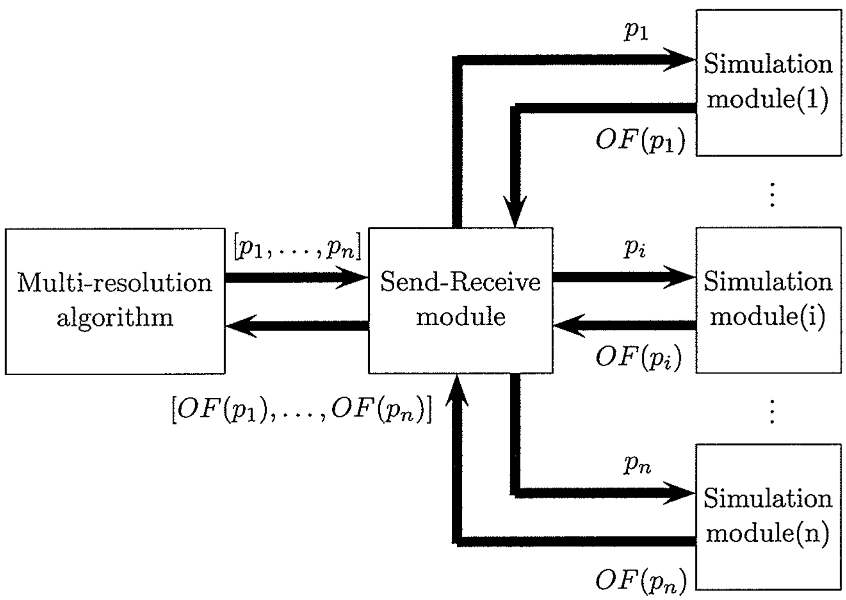


Fig. 5.2: *Parallel OF evaluation in the multi-resolution algorithm.*

5.2 Partial Mesh Generation

Delaunay triangulation is used in each step of the multi-resolution algorithm to generate the connections between the points of the mesh. These connections are used for: (i) initial mesh generation; (ii) localization of local minima; and (iii) mesh refinement. The algorithm derives the point connections (links of the nodes) using the set of elements generated by the Delaunay triangulation algorithm. As the number of points is increased in a mesh, the number of elements generated by Delaunay triangulation increases as well. Fig. 5.3 shows the number of elements versus the number of points for a mesh for different dimensions. As is shown in the figure, the number of elements at higher dimensions will require an excessive amount of memory to be stored so they can be used later to derive the point connections. However, at lower dimensions than 5 the number of elements is almost linear versus the number of points; therefore for optimization problems at these dimensions the partial mesh generation is not preferred. The maximum number of point connections is $\frac{N(N-1)}{2}$ (where N is the total number of points in a mesh), which is independent of the dimension of the points; therefore, a new method is required to generate these connections without using the Delaunay algorithm.

After the generation of the initial mesh and localization of the local minima, the generation of all point connections is not necessary. In the next mesh refinement stages only the point connections for the newly added points are required (i.e., the points added to the refined mesh). Fig 5.4 shows the connections of newly added points for a mesh refinement phase of the multi-resolution algorithm. The required point connections are shown with dotted lines.

Similarly in the next localization phases, only the connections of the former lo-

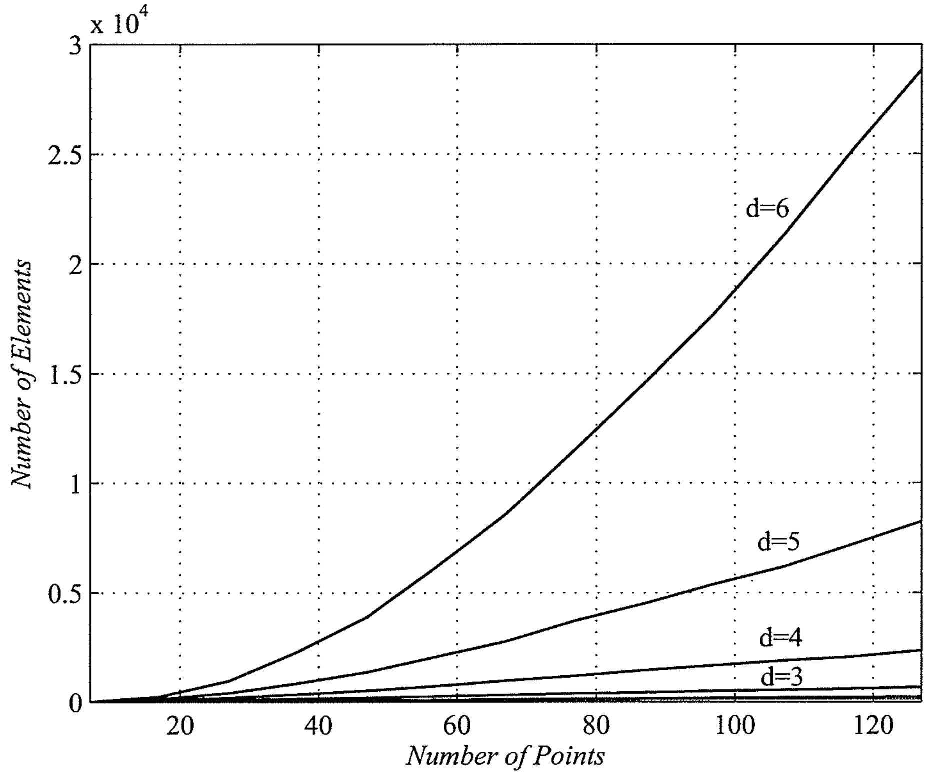


Fig. 5.3: *Number of elements versus number of points.*

calized cells points and the new sampling points are required. Fig 5.5 shows the required connections for the next localization phase of the multi-resolution algorithm with dotted lines.

The Delaunay algorithm should be replaced by another algorithm capable of generating a partial mesh. This partial mesh should only contain the connections of the required points, which will be used in the next iterations of multi-resolution algorithm. The partial mesh generation introduced in this section selects a point and then checks its connectivity to other points of the mesh. All the points of the mesh need not be tested. A selected set of points with a distance less than a constant number are checked. Fig. 5.6 shows a selected point p_c and the other points of the

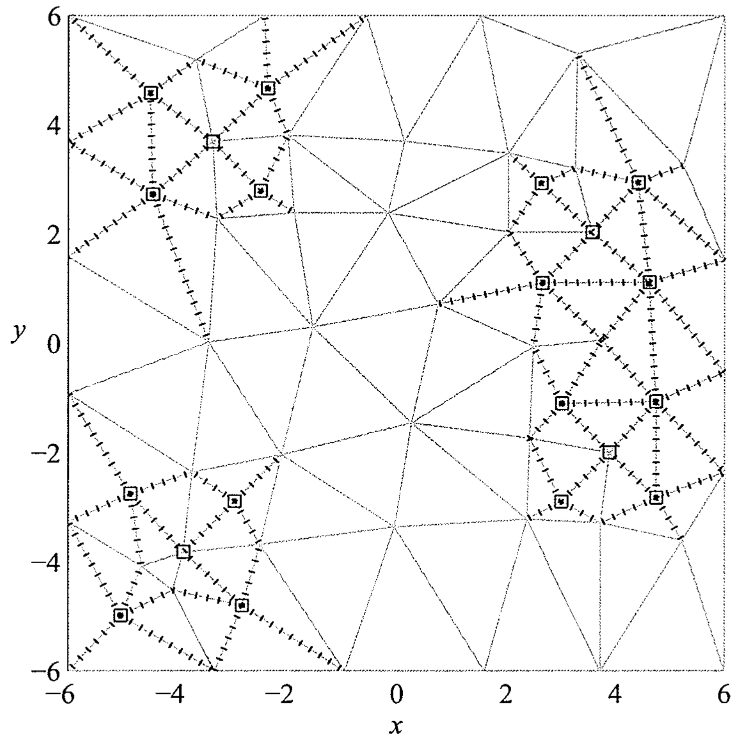


Fig. 5.4: *Required connections needed for mesh refinement phase.*

mesh. The points with a distance lower than r_c are selected. The connectivity of the set of points $p_1 \dots p_{12}$ to p_c are tested; for example, the connectivity test of p_4 to p_c is accepted, but the connectivity of p_{10} to p_c ($l_{c,10}$) is rejected (see Fig. 5.6). The constant r_c must be specified in a way that the points of the cell defined by p_c are selected. The constant distance r_c, here, is selected to be twice the initial mesh size distribution h_0 (see Fig. 4.1); however, for a refined mesh the r_c could be set equal to twice the selected cell-size (see Fig. 4.17).

Fig. 5.7 shows the algorithm for partial mesh generation for a point p_c. The set of connected points to p_c is returned in P_c^{cell}.

The algorithm tests the connectivity of a set of points. The connectivity test veri-

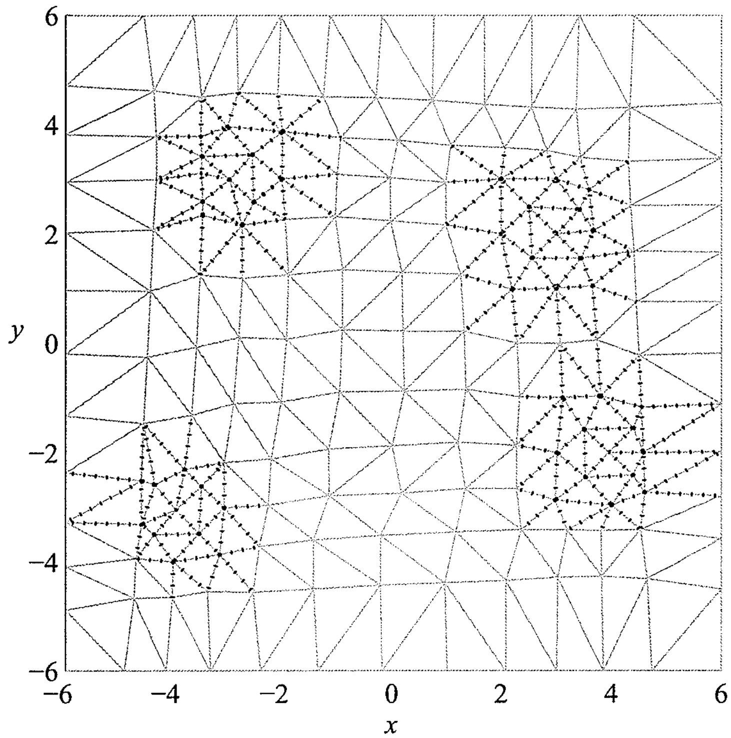


Fig. 5.5: *Required connections needed for localization phase.*

fies the connection of the two points p_c and p_k (see Fig. 5.7) in a Delaunay triangulated mesh. The test is performed using the same procedure of Delaunay triangulation but without generating the elements (triangles). The next section describes the Delaunay algorithm using a convex hull generation of a set of points. Using the Delaunay algorithm, the connectivity test of two points p_c and p_k (see Fig. 5.7) is transformed to a problem of connectivity test of two points on a convex hull of the transformed set of selected points $p_{selected} = [p_1^s, \ldots, p_m^s]$ (see Fig. 5.6). This algorithm will be capable of generating the connections of one point p_c independently of the tessellation (triangulation) of all sets of points of a mesh.

Partial mesh generation only provides the connections between the points, while

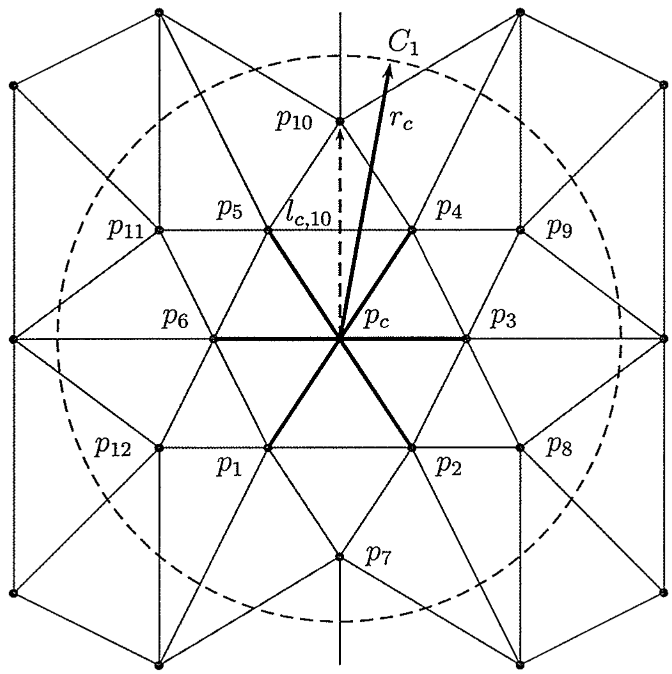


Fig. 5.6: *Partial Mesh Generation.*

it does not provide the elements of the finite element mesh. This phenomena restricts the applicability the graph-based localization and center of mass mesh refinement techniques, because they need the set of elements covering the optimization space.

5.2.1 Delaunay Algorithm

The nearest neighbor points of a selected point x_1 are the points connected to x_1 in the Delaunay triangulation. The Delaunay triangulation is generated using convex hull generation. The smallest convex boundary enclosing a group of points is called their convex hull; for example, in 2-dimensions, the convex hull is formed by stretching a band around the points so that all of the points lie inside the band. Fig. 5.8 shows the convex hull for a set of randomly scattered points in a 2-dimensional space. A

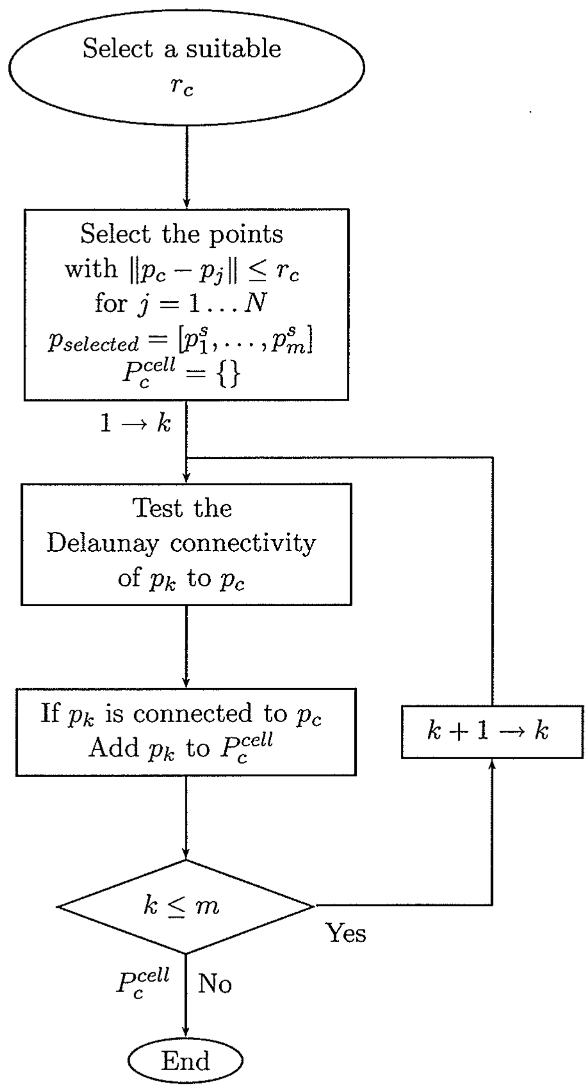


Fig. 5.7: *Partial Mesh Generation for a point (p_c).*

convex hull is represented with a set of facets connecting the points. These facets here are the point connections. Fig. 5.8 shows two points p_1 and p_2 that belong to the convex-hull of a set of scattered points. The two points are connected by the facet (element) e_1. The elements for a 2-dimensional convex-hull are 1-dimensional facets.

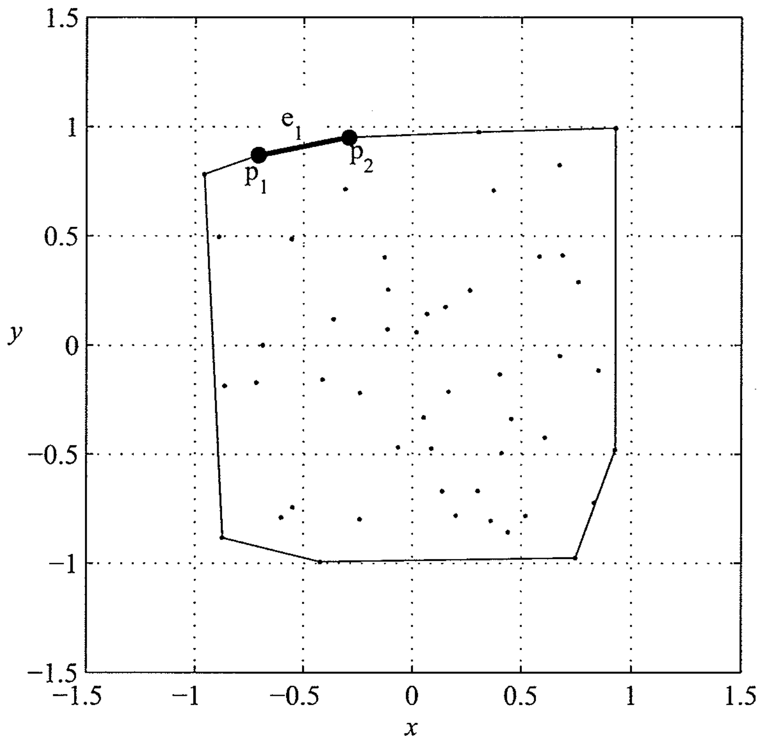


Fig. 5.8: *Convex hull of a set of random data in 2-dimensional space.*

A Delaunay triangulation in R^n can be computed from a convex hull in R^{n+1} [44]. The Delaunay triangulation of a set of points is the convex hull of the lifted points on a paraboloid. The set of facets covering this convex hull are the set of elements of Delaunay triangulation. To generate the Delaunay triangulation, the points are lifted to a paraboloid in a higher dimension using the transformation in (5.1).

$$p_1 = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \rightarrow p_1' = \begin{bmatrix} x_1 \\ \vdots \\ x_d \\ \sum_{i=1}^{i=d} x_i^2 \end{bmatrix} \tag{5.1}$$

The facets of the convex hull of these points are the elements of the Delaunay triangulation. Fig. 5.9 shows a set of randomly scattered points in $[-1, 1]$ and the corresponding transformed points on a paraboloid. The transformed points are depicted with '+'. The convex hull of the transformed points is also shown on the graph. The lines connecting the points on the paraboloid are the 1-dimensional point-connections generated using the Delaunay algorithm. These connections are the Delaunay elements generated for a 1-dimensional case. Fig. 5.9 shows two points, p_1 and p_2 and their set of transformed points on a hyperbola in 2-dimensional space. The transformed points are shown as p_1' and p_2'. The two points p_1' and p_2' are connected on the convex-hull of the paraboloid by the element e_1'; therefore, the points p_1 and p_2 must be connected on 1-dimensional data by the element e_1.

The partial mesh generation is supposed to generate all the connections of a point (p_c) in the mesh. This is achieved by first transforming the points to a paraboloid using (5.1) and then finding the points, which are connected to point p_c on the convex hull; therefore, the connectivity of two points in a mesh, provided by the Delaunay algorithm, is replaced by connectivity of the two points on a convex-hull generated by transforming the points on a paraboloid using (5.1). The next section describes a method to test the connectivity of two points on the convex hull of a set of points.

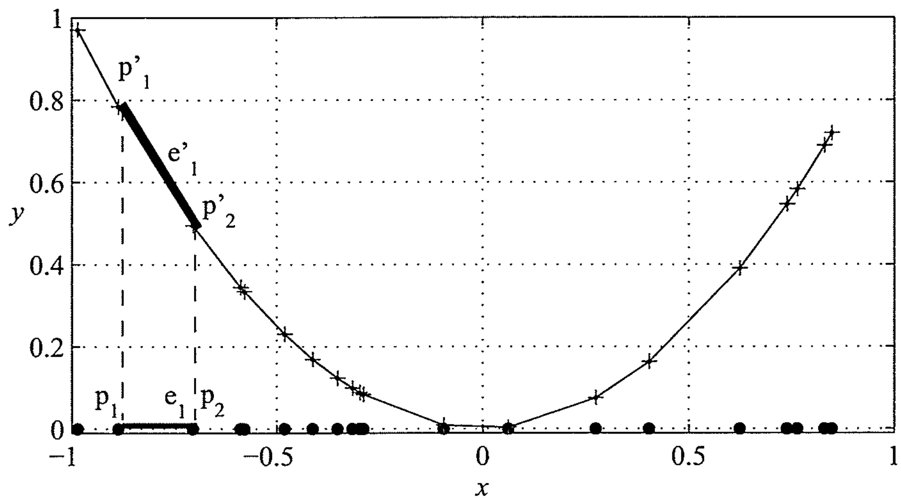


Fig. 5.9: *Delaunay triangulation as the convex hull of the transformed points.*

The proposed algorithm is described for a 3-dimensional convex hull; however, it is extendable to higher dimensions.

5.2.2 Convex-Hull Connectivity Test

In this section a new method is introduced to verify the connectivity of an arbitrary point x_1 to another point y_1 on a convex hull of a set of points P. Two points x_1 and y_1 are connected on the convex-hull of a set of points (P) if there exist a hyperplane that connects x_1 to y_1 while the other points of the set stand only on one side of that plane. The set of all hyperplanes that connect x_1 and y_1 could be represented by a set of lines on another hyperplane at x_1 and perpendicular to the vector $\vec{u}$ that connects the point x_1 to y_1. Fig. 5.10 shows two hyperplanes $\mathcal{L}_1(X)$ and $\mathcal{L}_2(X)$ that cross the points x_1 and y_1. The two hyperplanes could be represented by the lines l_1 and l_2 on the hyperplane $\mathcal{L}(X)$. The problem of existence of a hyperplane that crosses the two points x_1 and y_1 while all other points of the convex hull stands only on one side of it can be reduced to a problem of the existence of a line on the hyperplane $\mathcal{L}(X)$ that crosses x_1 while all other projected points along the vector $\vec{u}$ on the hyperplane $\mathcal{L}(X)$ stands on one side of that line.

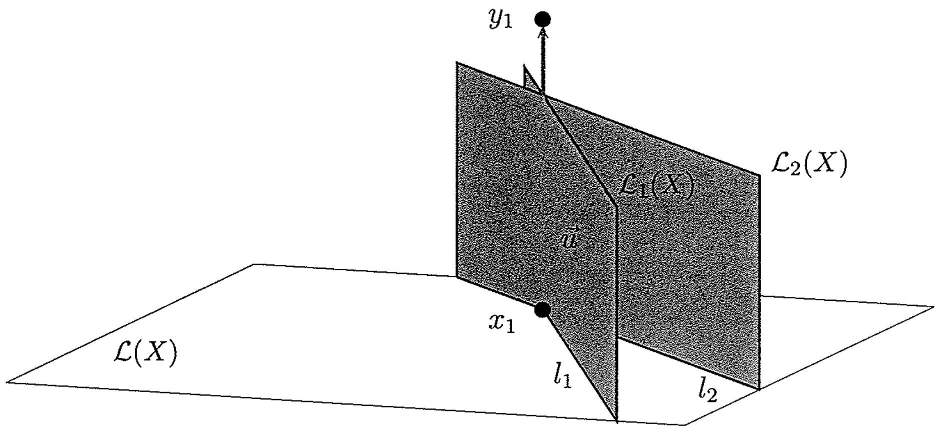


Fig. 5.10: *The set of hyperplane crossing x_1 and y_1. $\mathcal{L}(X)$.*

Fig. 5.11 shows the mapping of two points z_1 and z_2 along the vector $\vec{u}$ to the hyperplane $\mathcal{L}(X)$.

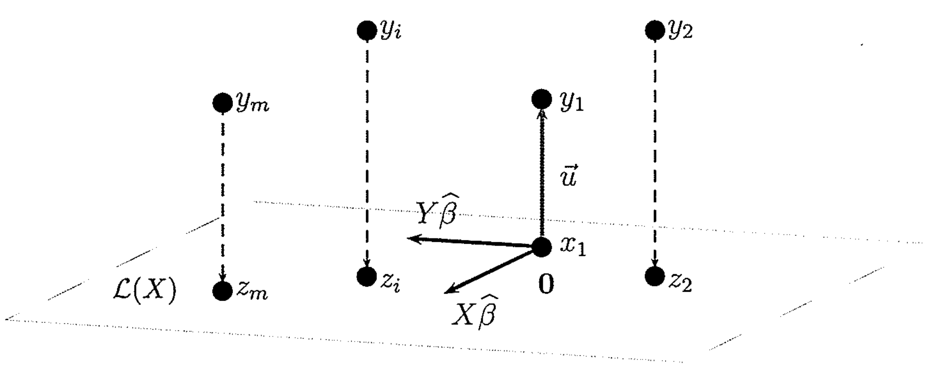


Fig. 5.11: *The mapping of the points on the hyperplane $\mathcal{L}(X)$.*

The projected points on the plane $\mathcal{L}(X)$ can be specified in a new coordinate system represented by vectors $X\widehat{\beta}$ and $Y\widehat{\beta}$. The points x_1 and y_1 are transformed to the origin in the new coordinate system.

Using such a transformation, the existence of a line on the hyperplane $\mathcal{L}(X)$ that crosses x_1 while all other projected points $(z_1 \ldots z_m)$ stand on one side of it, can be stated as, x_1 belongs to the convex-hull of the set of transformed points $z_1 \ldots z_m$ on the hyperplane $\mathcal{L}(X)$. To check that x_1 belongs to the convex-hull of the points $z_1 \ldots z_m$, all the points are represented in the β coordinate system. The transformation of the points (z_i) to the new coordinate system of β on the hyperplane $\mathcal{L}(X)$ is done using an affine transformation

$$y_{\widehat{\beta}} = A^{-1} \otimes [y_i - x_1] \tag{5.2}$$

$$z_i = y_{\widehat{\beta}}(2, \ldots, d) \tag{5.3}$$

where $\otimes$ is the matrix product, d is the dimension of the points on the convex hull, $y_{\widehat{\beta}}$ depicts the coordinates of point y in the new coordinate system of $\widehat{\beta}$, and A is a matrix constructed as follows to transform the coordinates of points into the new coordinate system $(\widehat{\beta})$. The first element in the new coordinate system is the height from the hyperplane $\mathcal{L}(X)$; therefore, it is removed from the coordinates of $y_{\widehat{\beta}}$ in order to represent the new points only on the new coordinate system standing on the hyperplane $\mathcal{L}(X)$.

$$A = [\vec{u}, X\widehat{\beta}, Y\widehat{\beta}] \tag{5.4}$$

z_i is the projected point of y_i with the coordinates stated in the coordinate system of β.

The vectors $X\widehat{\beta}$ and $Y\widehat{\beta}$ that define the new coordinate on the plane $\mathcal{L}(X)$ can be chosen arbitrarily; however, they must be perpendicular to each other and also to the vector $\vec{u}$. The construction of such a set of basis vectors is described in Appendix A. After representing the points in the new coordinate system, the origin is tested to see if it belongs to the convex hull of the set of points $z_1 \ldots z_m$ on the hyperplane $\mathcal{L}(X)$. If the origin belongs to this convex-hull then the two points x_1 and y_1 are connected.

As an example in a 2-dimensional space consider a simple set of points and their corresponding convex hull as is shown in Fig. 5.12 (a). Two points x_1 and y_1 are shown on the graph. l is a line passing through x_1 perpendicular to the vector $\vec{u}$, which is connecting x_1 to y_1. The other points are also mapped along this vector to the line l. Two points of x_1 and y_1 are mapped to a similar point on the line, which is again depicted by x_1. The mapped point of two points x_1 and y_1 belongs to the convex hull of the set of transformed points on line l; therefore, the two points x_1 and y_1 are connected. The same procedure is shown for another two points of x_1 and y_1 in Fig. 5.12 (b). The two points are again mapped along their connecting vector to the line l, which is passing through x_1 perpendicular to the vector connecting the two points x_1 and y_1. The two points x_1 and y_1 are again transformed to a single point x_1. The point x_1 does not belong to the convex hull of the transformed points on line l; therefore, the two points x_1 and y_1 are not connected on this convex hull. The convex hull of a set of points on a line are the two ending points.

This procedure for testing the connections between the points on a convex hull can be generalized to higher dimensions using higher dimensional hyperplanes for projection of the points.

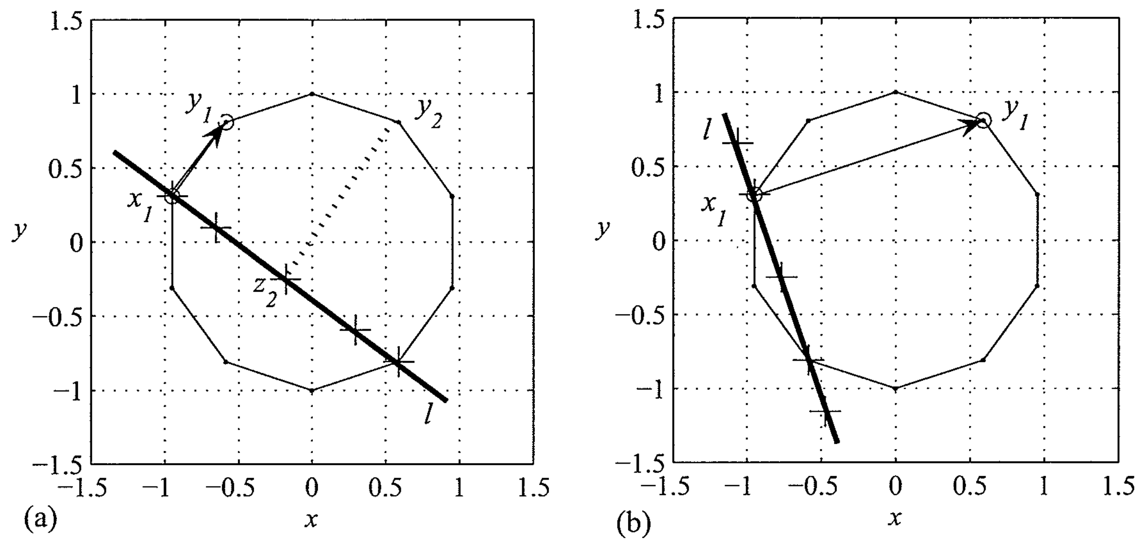


Fig. 5.12: *Convex-hull generation.*

5.2.3 Convex-Hull Detection Test

In this section a simple algorithm is introduced to check that a point x_1 belongs to the convex hull of a set of points $z_1 \ldots z_m$. The algorithm first finds two points z_a and z_b from the set of points $(z_1, \ldots, z_m)$ that make the biggest angle with the point x_1. Fig. 5.13 shows two such points, z_a and z_b, so that their angle $(\alpha_{a,b})$ is the biggest among all angles of the other pair of points that are connected to x_1. If all the remaining points stay on one side of the region defined by the two lines l_a and l_b then the point x_1 belongs to the convex-hull of the set of points $z_1 \ldots z_m$.

5.2.4 Partial Mesh Implementation

Using the algorithms introduced in the last sections the problem of partial mesh generation is implemented by:

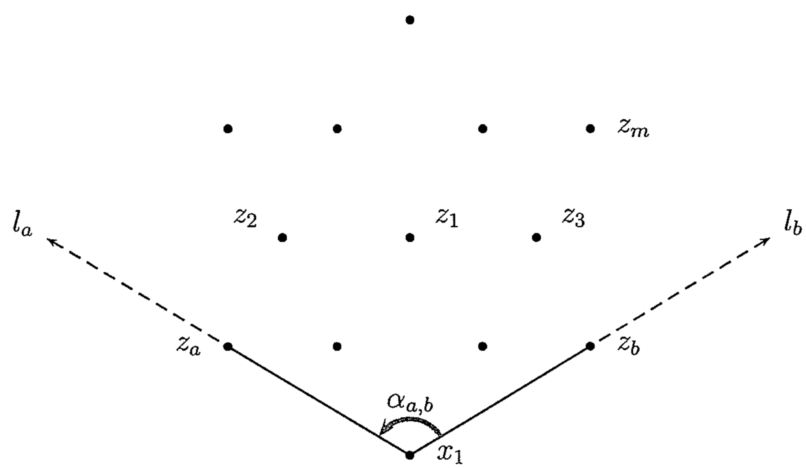


Fig. 5.13: *Convex-hull detection.*

1. Selecting the required points for partial mesh generation (p_c).

2. Detecting the connections of the selected points.

where each of these steps are done in each phase of mesh refinement and localization of the multi-resolution algorithm. The required points in the mesh refinement phase are the newly added points and the required points in the localization phase are the formerly detected cell points (cell-center plus surrounding cell points) plus the newly added points. The algorithm for determining the connections of a selected point p_c is shown in Fig. 5.14.

This algorithm is tested for a regular set of points in 2-dimensional space. These points are selected regularly on a 5×5 grid. Fig. 5.15 shows the initial points and the transformed points on a paraboloid using (5.1). The connections then are formed on the paraboloid. These connections correspond to connections of points in a 2-dimensional Delaunay triangulation.

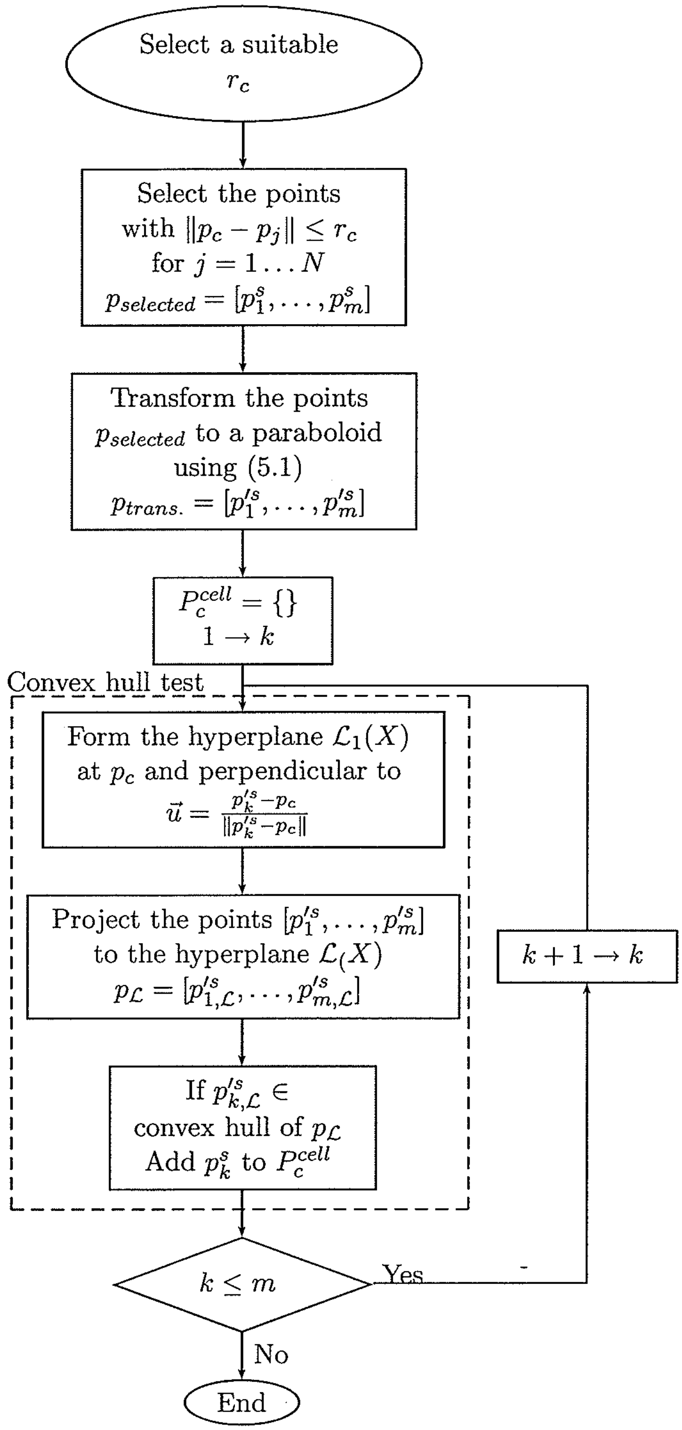


Fig. 5.14: *Partial Mesh Generation for a point (p_c).*

Fig. 5.16 shows the same paraboloid, but from a top view. It shows that the

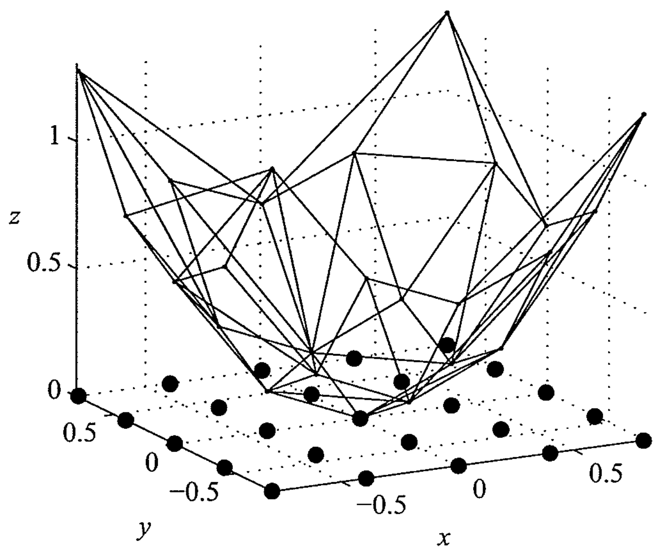


Fig. 5.15: *Delaunay triangulation as the convex hull of the transformed points.*

connections on the paraboloid are the Delaunay triangulation of the 2-dimensional data.

Using the proposed algorithm, the generation of a partial mesh is done without any need of Delaunay triangulation. Unlike the Delaunay triangulation this algorithm only determines the connections of the required points; therefore, it has decreased the amount of memory required and computation time needed for Delaunay triangulation.

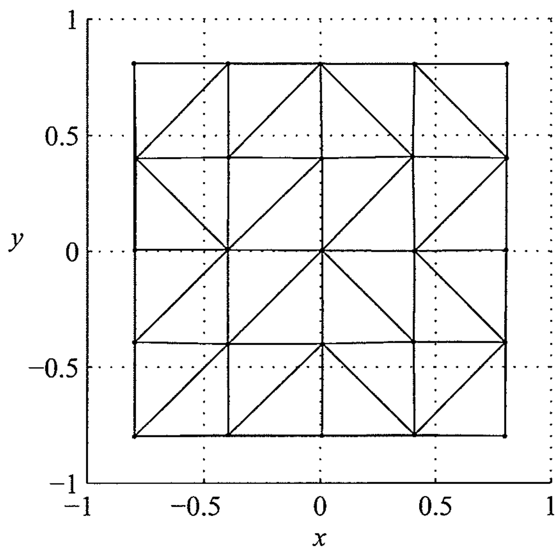


Fig. 5.16: *Delaunay triangulation of a regular grid.*

Chapter 6

Design Examples

6.1 Introduction

The multi-resolution optimization method is implemented in this thesis using PSCAD as a simulation agent that evaluates a single set of design parameters and returns the corresponding objective function value. The interfacing between PSCAD and the nonlinear optimization algorithm implemented in Matlab is shown in Fig. 6.1.

The proposed algorithm is tested through four examples. The first example is an optimal switching problem, which has a known analytical form. There are two solutions, which are derived using the multi-resolution algorithm. This example shows the ability of the proposed method to capture all local minima for a problem with known solutions. The last three examples are cases that have no analytical solutions. These examples cover a wide range of applications including a power electronic converter, a standard HVDC example, and a robust optimization case. The robust optimization is done using a multi-objective optimization method.

6.2 PSCAD Implementation

PSCAD has the ability to communicate with Matlab under the windows operating system. PSCAD receives the sets of testing design parameters from Matlab and substitutes them as the system parameters in the simulation case. The objective function value will be available after the completion of simulation. The objective function value corresponding to the received set of parameters is sent back to Matlab. The optimization algorithm will either send a new set of design parameters or finish the optimization in the case of reaching local minima with an enough accuracy. Fig. 6.1 shows a detailed diagram for interaction of the multi-resolution algorithm in Matlab and PSCAD.

The multi-resolution algorithm generates a set of points during each iteration. This new set of points is passed to a send-receive module, which is responsible for sending the sets of design parameters to PSCAD and receiving the evaluations. The set of results is then passed to the multi-resolution algorithm.

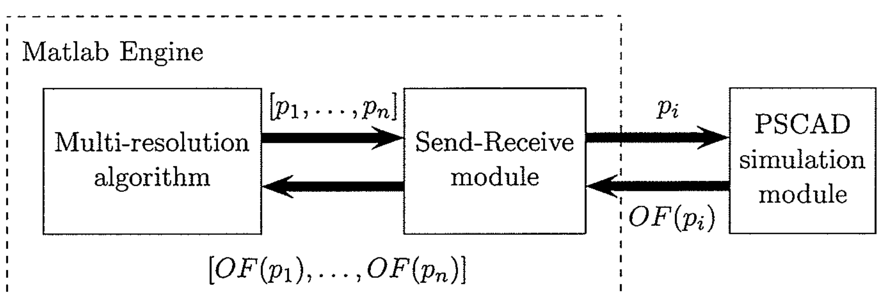


Fig. 6.1: *PSCAD and multi-resolution algorithm interfacing.*

6.3 Example 1: Optimal Pulse-Width Modulation

Derivation of firing angles of an optimal pulse-width modulation using selective harmonic elimination is done, as the first design example, for a 3-level converter. Fig. 6.2 shows an implementation of a 3-level converter. Each converter leg is capable of generating three levels of voltage, E, $-E$, and 0. The pulse-width modulation technique selects the appropriate voltage level in each leg at appropriate sequences of time in order to produce the desired output voltage at the synchronous frequency. This is achieved using several switchings during one period of synchronous frequency.

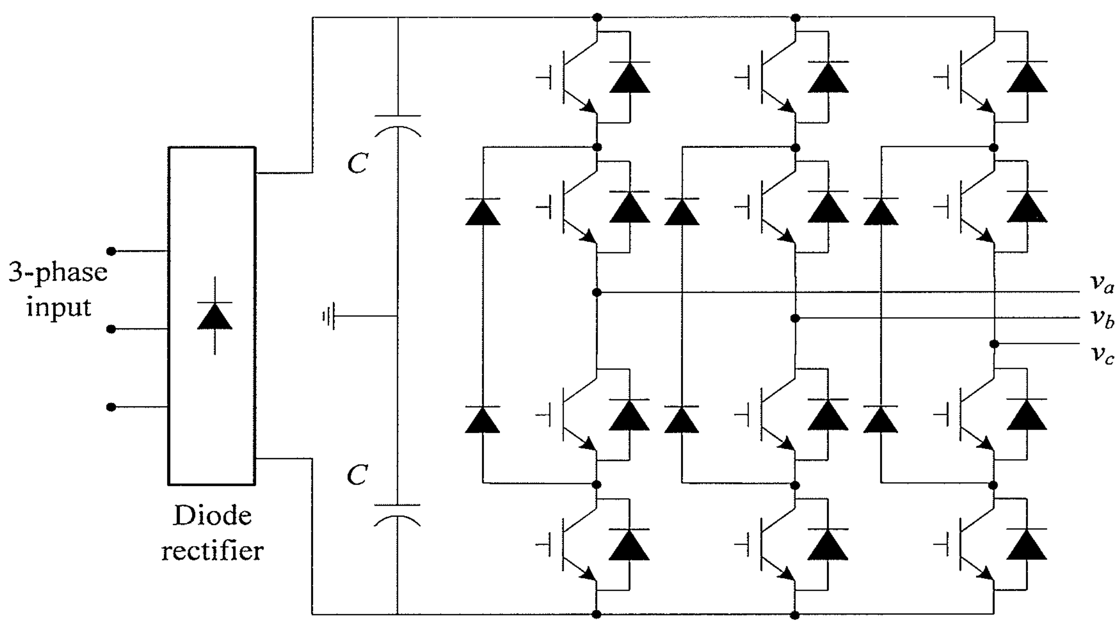


Fig. 6.2: *Three level converter.*

The switchings cause undesirable harmonics and losses. The switching losses are proportional to the number of switchings that occur during one period of the synchronous frequency. The selective harmonic elimination scheme was introduced to keep the number of switchings, occurring during one cycle, low and also to eliminate some selected harmonics. This method is also referred to as optimal pulse-width modulation (OPWM). In the selective harmonic elimination scheme a limited number of switchings happen at pre-selected instances of time (angles). An OPWM waveform is shown in Fig. 6.3. Only three switching angles can be selected arbitrarily, because the other switching instances could be derived using quarter-cycle symmetry.

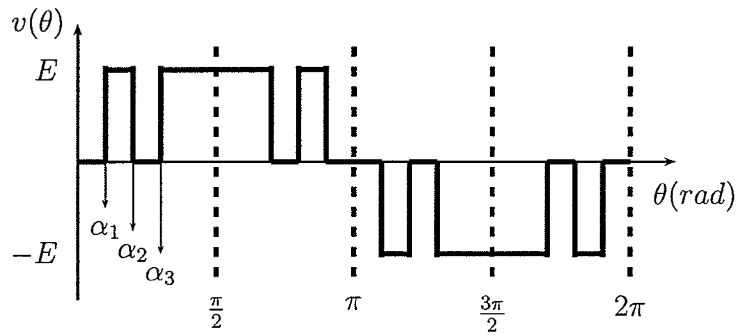


Fig. 6.3: *A 3-level switching waveform.*

The input voltage to the converter is either provided by a constant dc link or a rectified voltage supplied from a rectifier. Assuming a constant dc input voltage the output harmonic magnitudes are calculated as in (6.1). If the dc link voltage is not constant this formulation is no longer valid, but still provides a good approximation [54].

$$V_k = \frac{4E}{k\pi}\left|\sum_{i=1}^{i=N}(-1)^i cos(k\alpha_i)\right| \tag{6.1}$$

where $2E$ is the total dc link voltage and N is the total number of switchings in the first quarter-cycle. Using these N switching angles per quarter-cycle, the fundamental component can be adjusted and $N-1$ arbitrary harmonic components can be eliminated or minimized. The quarter-cycle symmetry results in elimination of even harmonics; therefore, the odd harmonics are selected to be eliminated. The triple-n harmonics are not considered because they are automatically eliminated in the line voltages. Practically $3n \pm 1$ harmonics are tried to be removed from the output voltage. The solution is achieved by solving N nonlinear equations of the form (6.1). Here, for example, the fundamental synchronous component is selected to be 10kV and the 5^{th} and 7^{th} harmonics are set to zero. This is achieved by solving the following equations

$$\begin{aligned} v_1 &= \frac{4E}{\pi}\left|\sum_{i=1}^{i=3}(-1)^i cos(k\alpha_i)\right| = 10 \\ v_5 &= \frac{4E}{5\pi}\left|\sum_{i=1}^{i=3}(-1)^i cos(k\alpha_i)\right| = 0 \\ v_7 &= \frac{4E}{7\pi}\left|\sum_{i=1}^{i=3}(-1)^i cos(k\alpha_i)\right| = 0 \end{aligned} \tag{6.2}$$

The solution of this system of equations is not unique [5]. The set of solutions have been proved to be two zero points. The optimization problem is defined to minimize the following nonlinear equation with three inequality constraints.

$$\begin{aligned} &\text{Minimize}\, OF(\alpha_1, \alpha_2, \alpha_3) = (v_1 - v_{1ref})^2 + (v_5)^2 + (v_7)^2 \\ &\text{Subject to: } 0 \leq \alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \frac{\pi}{2} \end{aligned} \tag{6.3}$$

where v_{1ref} is the desired fundamental component and v_5 and v_7 are the 5^{th} and 7^{th} harmonic components respectively. The objective function defined in (6.3) attains its absolute minimum at the zeros of the nonlinear set of equations in (6.2).

Table 6.1 shows the optimization results of the objective function in (6.3) with a variable dc-side voltage. Analytical solutions of the idealized problem are also shown separately. The idealized problem has two distinct zeroes. The idealized problem is solved assuming a constant dc input voltage, while the input voltage is provided through a six-pulse diode rectifier. The six-pulse diode rectifier provides an almost constant input voltage with some extra harmonics, which are not considered in the ideal case. Two of the local minima (Table 6.1. (a) and (b)) are very close to solutions of the idealized problem, indicating that practically, the solutions for the non-ideal case, are not far from the ideal solutions. This happens because of small dc-voltage variations resulting from the six-pulse diode rectifier [54]. The third local minimum (c) has a larger objective function value, which indicates the third local minima is not a solution of the proposed selective harmonic elimination problem.

Table 6.1: *Selective harmonic elimination results.*

Idealized Problem			
	α_1	α_2	α_3
Solution 1	22.42^o	37.76^o	46.45^o
Solution 2	13.76^o	73.58^o	82.31^o
Optimization Results			
Local Minima	α_1	α_2	α_3
(a) $OF = 0.0066$	21.81^o	37.26^o	45.29^o
(b) $OF = 0.086$	14.32^o	76.73^o	82.94^o
(c) $OF = 0.357$	4.51^o	15.70^o	29.26^o

Fig. 6.4 shows the harmonic spectrum of the output voltage for each of these three sets of local minima. The spectrum shows that the fundamental component is matched with the desired value for the first two local minima and the 5^{th} and 7^{th} harmonics are essentially eliminated using the first two local minima. Both the (a) and (b) local minima are solutions of the non-idealized problem. However comparison

of the harmonic spectrum of the first and second local minima reveals that the first local minima is superior to the second local minima, because of a lower 3^{rd} harmonic content. In practice it is desirable to have a lower third harmonic content as it will not cause overly disadvantageous situations under unbalance conditions.

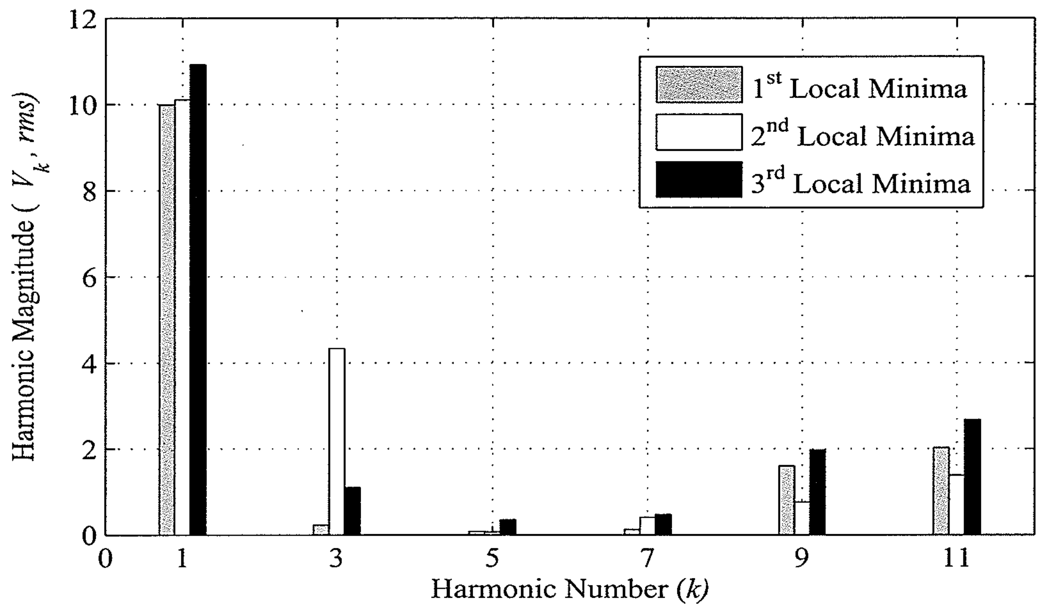


Fig. 6.4: *Harmonic spectrum of the three local minima from in Table 6.1.*

The third local minimum is not a good choice for selective harmonic elimination, because of higher harmonic content and an unmatched fundamental frequency magnitude.

6.4 Example 2: DC-to-DC Converter

A dc-to-dc converter is designed in this section with the ability of adjusting its output dc voltage. A schematic diagram of this dc-to-dc converter is shown in Fig. 6.5.

The converter is fed from a diode bridge rectifier, which provides an almost constant dc voltage.

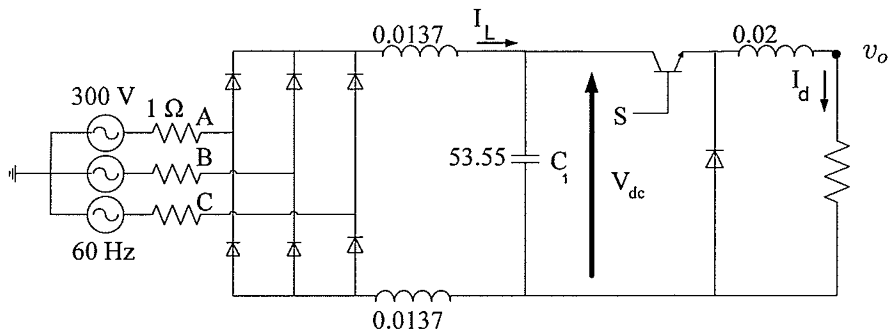


Fig. 6.5: *Schematic diagram of the dc-to-dc converter example.*

The average output voltage of the dc-to-dc converter is

$$v_o = Dv_{in} \tag{6.4}$$

where D is the duty cycle ratio of the switchings, v_o is the output voltage of the rectifier, and v_{in} is the input voltage.

By controlling the switching duty cycle (D), the output voltage of the converter can be controlled. The output voltage controls the output current; therefore, a closed control loop can be designed to regulate the output current. A PI (Proportional-Integrator) controller is designed to adjust D dynamically in order to enhance the quality of transient response. Fig. 6.6 shows controller designed for adjusting the output current to a reference current (I_{ref}).

The transient behaviour of the circuit depends on the PI controller gain (K) and integration time constant (T). The design problem is to find an optimum set of values for K and T so that the two currents I_{out} and I_{ref} become as similar as possible.

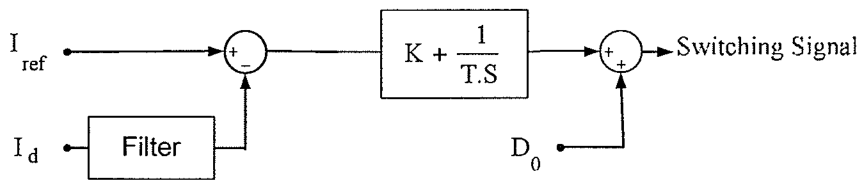


Fig. 6.6: *The current controller of a dc-to-dc converter.*

The objective function defined in (6.5) is used to measure the difference between the given reference current and the simulated transient response.

$$
\begin{aligned}
& OF(K,T) = \int_{x=0}^{x=T_f} \left(1 - \frac{I_d(t)}{I_{ref}}\right)^2 dt \\
& \text{Subject to:} \qquad\qquad (6.5) \\
& 0 \leq T \leq 0.1 \text{ and } 0 \leq K \leq 5
\end{aligned}
$$

where I_d is the load current, I_{ref} is the reference current, and T_f is the total simulation time.

The defined objective function will vanish to zero when the load current (I_d), completely tracks the reference current (I_{ref}); however, because of the switchings and inductive load, the load current cannot exactly follow the reference current and the objective function will not converge to zero; however an optimum design still can be achieved.

The simulation case with the proposed objective function in (6.5) was optimized using the multi-resolution optimization method. The generated multi-resolution mesh for the optimization of this system is shown in Fig. 6.7. The multi-resolution mesh has converged to six local minima; however very close local minima are rejected.

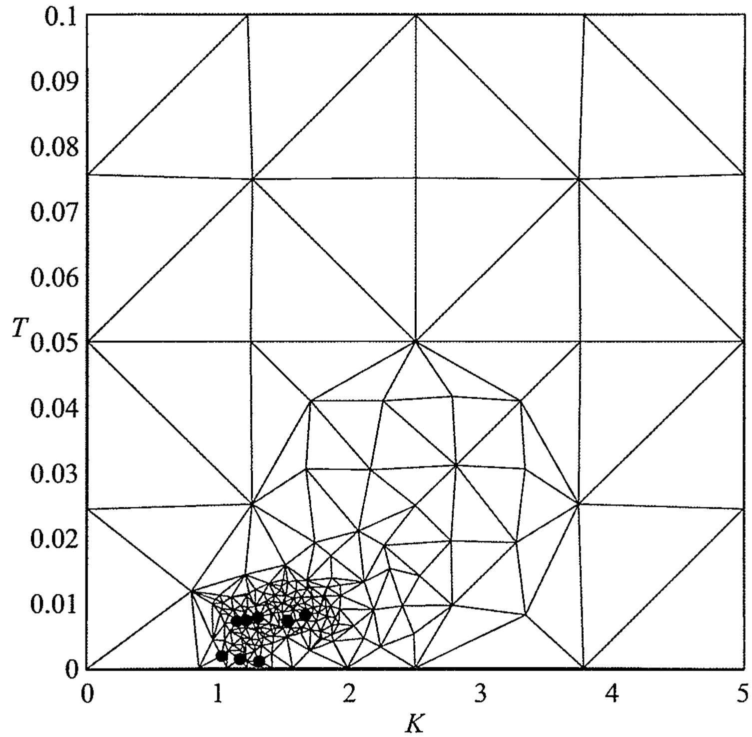


Fig. 6.7: *The multi-resolution optimization of dc-to-dc converter.*

Table 6.2: *Dc-to-dc converter optimization results.*

Local Minima	Control System Parameters Optimization Results		
	K	T	$OF(K,T)$
(1)	1.406	0.0088	0.4091
(2)	1.242	0.0082	0.4092
(3)	1.764	0.0091	0.4244
(4)	1.414	0.0020	0.4244
(5)	1.269	0.0024	0.4249
(6)	1.130	0.0028	0.4262
Total number of simulation runs: 198			
$0 \leq T \leq 0.1$			$0 \leq K \leq 5$

Table 6.2 shows the optimization results for the dc-to-dc converter design example.

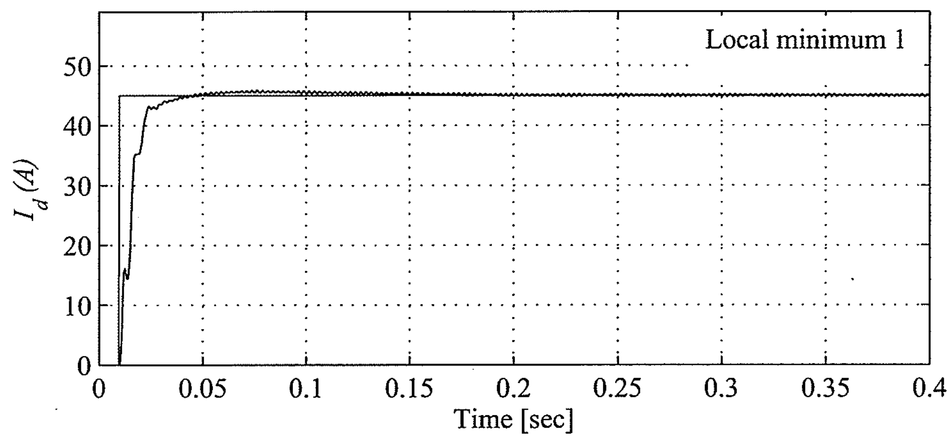


Fig. 6.8: *The transient response of the global minimum (1^{st}localminimum).*

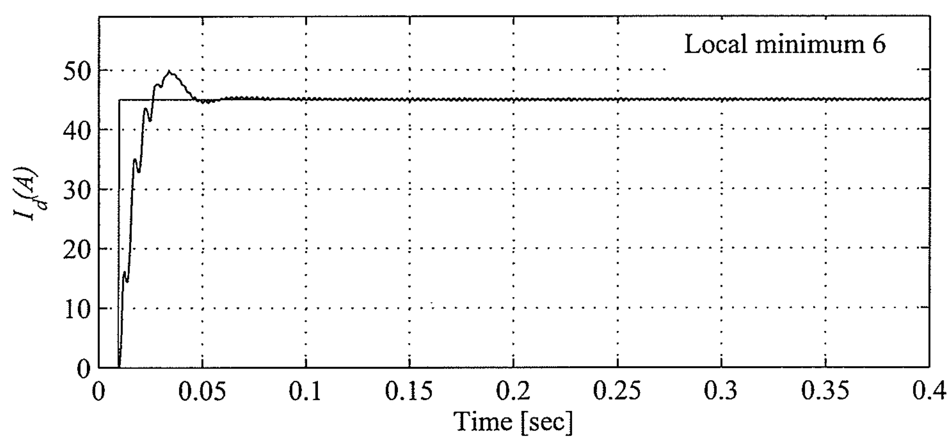


Fig. 6.9: *The transient response of the 6^{th} local minimum.*

Fig. 6.8 shows the transient response of the global minimum (1^{st} local minimum) and Fig. 6.9 shows the transient response for the local minimum with worst performance. The current follows the step input for both of the settings with a good approximation during steady state and also during rise time using the two sets of global and 6^{th} local minima solutions.

Note that the corresponding objective function values for the six local minima are very close and also that the performance of the first and sixth local minima show satisfactorily results. This phenomena is attributed to the non-smooth behaviour of the objective function, because the objective function is a result of integration of a switching signal with a high frequency content over time. The optimization of such a non-smooth objective function can be treated as optimization of a noisy objective function. The multi-resolution optimization algorithm is robust to noise because of an initial coarse sampling; therefore, at initial stages of localization of the local minima, the noise would not play a major role. However, as the algorithm tries to increase the accuracy of local minima estimation, a noisy function may cause detection of an excessive number of local minima. The actual local minima here are detected as 3 separate clusters of local minima; however, due to flatness of the objective function in the optimized region the location of the local minima are sensitive to the uncertainty in the calculation of the objective function. The 1^{st} local minimum and the 6^{th} local minimum correspond to two different clusters that show different dynamical behaviours, which shows the multi-modality of the objective function in this case.

6.5 Example 3: Cigre HVDC Benchmark

In this subsection a standard example of an HVDC system is optimized. A closed form expression for the dynamical behaviour of this HVDC system is unattainable; therefore, the system can only be optimized using a simulation-based approach. Multi-resolution optimization is used for the optimization of this complicated power system. The multi-resolution optimization algorithm captures more than one local minimum, which shows the multi-modality feature of the proposed HVDC system.

The Cigre HVDC benchmark is a detailed model of a realistic mono-polar, 12-pulse, back-to-back configuration of an HVDC system [55]. The schematic diagram of the HVDC sending and receiving ends are shown in Fig. 6.10. The system is rated at $2\,kA$ and $500\,kV$.

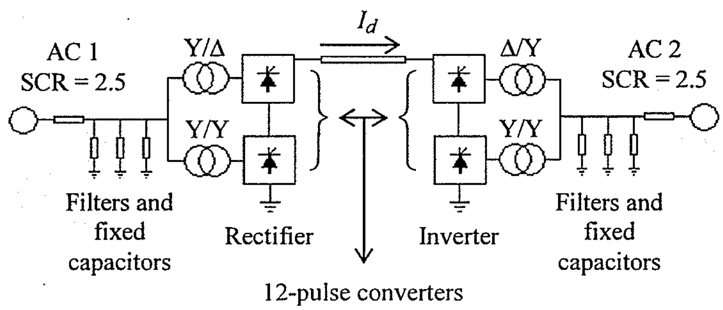


Fig. 6.10: *Cigre HVDC benchmark model.*

The control strategies for the rectifier and inverter are shown in Fig. 6.11. HVDC systems are responsible to transfer a specific amount of power; therefore, the control system must guarantee that a constant amount of power is transferred. This is achieved by controlling the current and voltage of the inverter and rectifier, respectively. The inverter works at constant extinction angle control to avoid commutation failure. The extinction angle control provides an almost constant voltage; therefore, the current control must be achieved by adjusting the firing angle of the rectifier side.

The operating point of the system is determined by the crossing point of the steady state $V_d - I_d$ characteristics of the rectifier and inverter. The constant current of the rectifier side is provided by firing angle control using a PI controller. Similarly the constant gamma (constant extinction angle) control of the inverter side is maintained by adjusting the firing angle of the inverter side, which is again provided via another PI controller. The implemented control diagram for firing angle generation

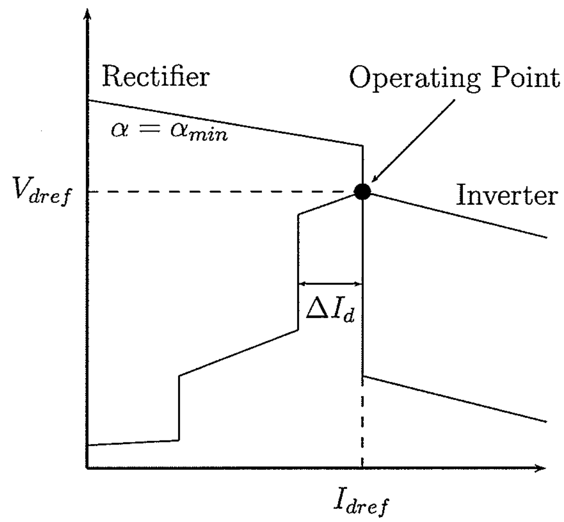


Fig. 6.11: *HVDC steady state control diagram.*

for the rectifier and inverter sides are shown in Fig. 6.12. The design problem in this case is the proper selection of parameters of the two PI controllers in order to obtain the desired transient performance.

6.5.1 Defining the Objective Function

The objective function must be defined as an index penalizing any deviation or failure in rectifier or inverter operation from the specified objectives. The variable of the system, which is sensitive to abnormal behaviour of both sides is the dc link current. The dc link current could be regarded as a factor describing how well the HVDC system behaves. Almost any failure in the system directly alters the magnitude of dc current; therefore, the objective function here is defined as the integration of the difference between the current order and dc-link current over time. The objective

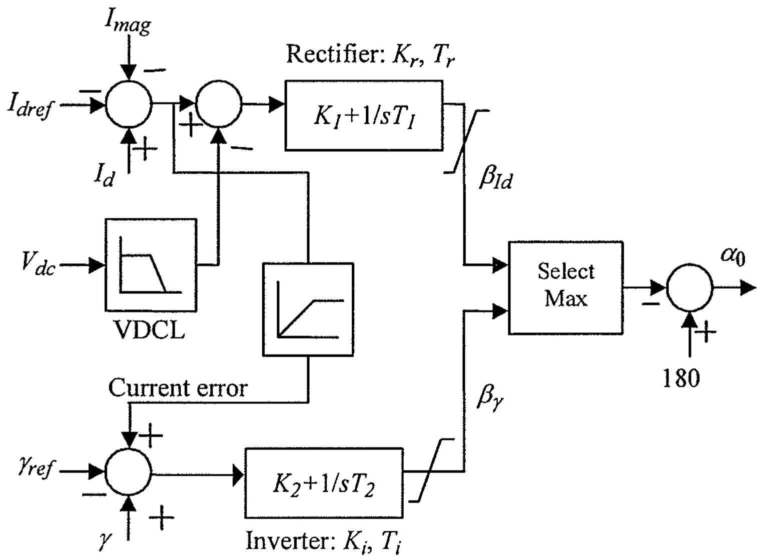


Fig. 6.12: *Cigre HVDC benchmark controller.*

function is then defined as shown in (6.6).

$$OF(K_r, T_r, K_i, T_i) = \int_{t=T_0}^{t=T_f} \left(1 - \frac{i_d}{i_{d,order}}\right)^2 dt \tag{6.6}$$

where i_d and $i_{d,order}$ are the dc current and current order values, respectively. The duration of integration is specified by T_0 and T_f. The simulation time for this case is 6s. The objective function will become zero, when the dc current exactly follows the given current order. In practice the dc current cannot exactly follow the current order; therefore, the solution is not the zero of the objective function, but a local minimum. The local minima of the system were found using the multi-resolution algorithm. Some local minima including the global minimum were found to have a good match between the desired performance and the actual performance.

The controllers are not only responsible for performing well during steady state, but they are also designed to maintain the stability of the system during disturbances. These disturbances could be internal like the change of the real power flow or external like a voltage drop at the inverter or rectifier sides. Therefore, during the calculation of the objective function, some faults are imposed on the system. This objective function evaluates the ability of the controllers for establishing a new stable operating point or bringing the system into the previous stable point; any disturbance either changes the stability point or alters the trajectory of the dynamical system.

To evaluate the quality of the designed controller the following sets of disturbances were imposed on the system:

1. $t = 0.04sec$, the magnitude of the inverter ac network voltage was reduced by 5%;

2. $t = 0.08sec$, the magnitude of the inverter ac network voltage was restored to the initial voltage;

3. $t = 1.0sec$, the current order was reduced by 50%;

4. $t = 1.5sec$, the current order was restored to its initial value.

The first set of disturbances (reduction of the ac network voltage) is a common consequence of faults occurring at the ac networks connected to the HVDC system. This test was performed on the inverter side, because of higher vulnerability to commutation failure upon the reduction of the voltage magnitude at the inverter. The last two sets of disturbances were done to evaluate the ability of the system to control the dc current and, consequently, the power flow of the system. These two changes are among the most common disturbances challenging the stable operation of an HVDC system.

Optimization of the OF in (6.6) using the proposed multi-resolution algorithm was done and resulted in six different local minima. The optimization results and their corresponding objective function values are listed in Table 6.3. Noting that a conventional sequential multiple-run approach to acquire 1% accuracy needs a grid with 50 divisions in each coordinate. The evaluation of such a grid needs $50^4 = 6,250,000$ number of simulations, while the multi-resolution algorithm found six local minima with only 3948 simulation runs. This shows the multi-resolution optimization algorithm is 3.2 orders of magnitude faster than conventional sequential search in this case.

Table 6.3: *Cigre HVDC benchmark optimization results.*

Local Minima	Control system parameters				OF
	Rectifier		Inverter		
	K_p	T_i	K_p	T_i	
(1)	2.144	0.0085	0.110	0.0876	1.655
(2)	3.068	0.0136	0.309	0.0737	1.698
(3)	3.030	0.0760	0.501	0.0765	1.765
(4)	2.546	0.0920	1.547	0.0660	2.405
(5)	1.260	0.10	5.0	0.0467	36.71
(6)	0.932	0.0994	4.596	0.0903	38.68
Total number of simulation runs: 3948					
Search interval for K_p : $[0.1, 5]$			Search interval for T_i : $[0.001, 0.1]$		

The first 4 local minima result in good system performance, while the other two local minima (with distinctively higher OF values) lead to poor performance. Note also that a conventional optimization algorithm using a gradient-based method only converges to a single local minimum based on the initial point for optimization; therefore it cannot guarantee the convergence to the global minimum.

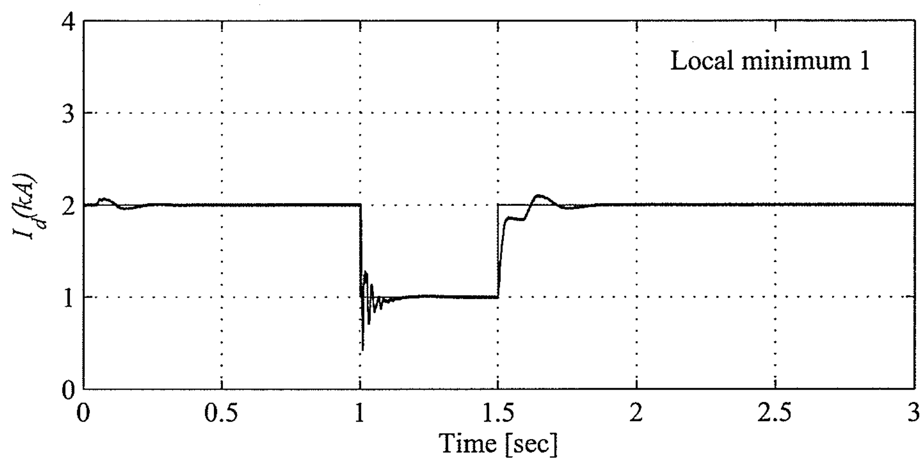


Fig. 6.13: *Transient response of the system for the 1^{st} minimum in Table 6.3.*

The transient responses of the dc current compared to the current order for the first four local minima are shown in Figs. 6.13 to 6.16. The HVDC dc-link current follows the changes of current order after the 1^{st} and 2^{nd} disturbances and also during the steady state run using the first three local minima settings. The ac voltage disturbance during [0.04s,0.08s] resulted in a small increase in the dc current, because of the voltage reduction at the inverter side.

The simulation result for the 4^{th} local minimum shows small oscillations from the desired value, but they are still acceptable. These sustained oscillations resulted in an objective function value of 2.405, which is almost 50% more than the best local minimum.

The transient response of the system for the last local minimum shows repetitive commutation failures. The 6^{th} local minimum has failed to restore a stable behaviour after the imposed disturbances (see Fig. 6.17).

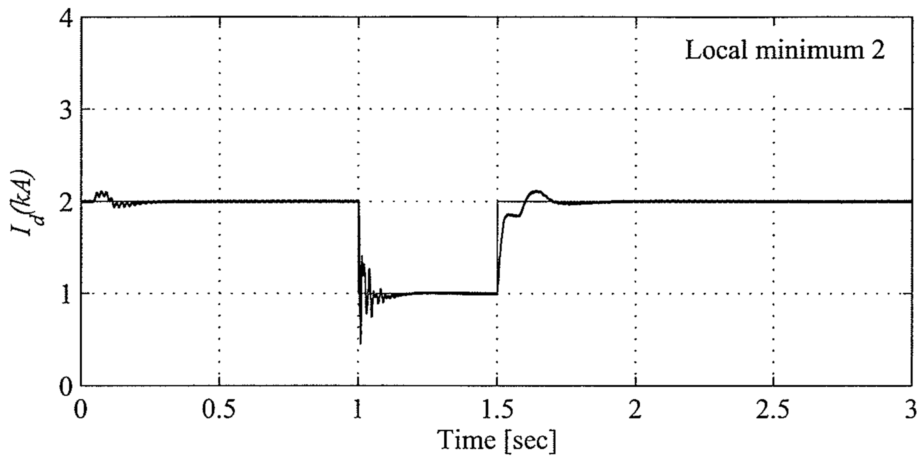


Fig. 6.14: *Transient response of the system for the* 2^{nd} *minimum in Table 6.3.*

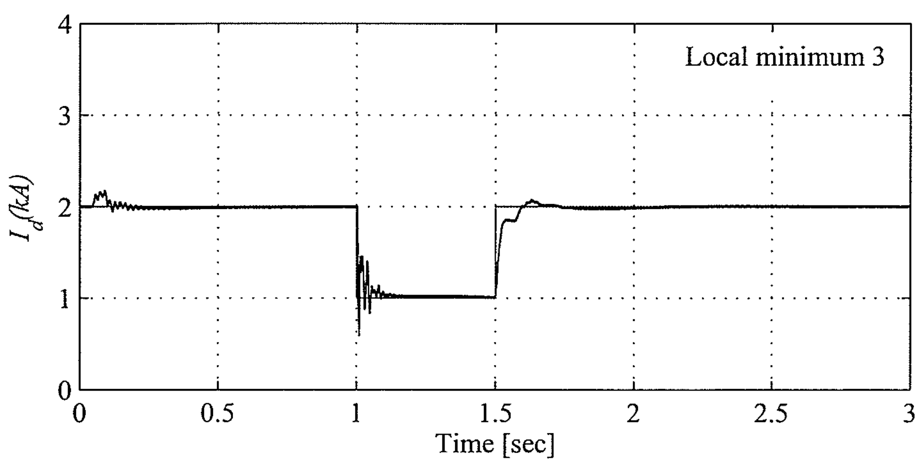


Fig. 6.15: *Transient response of the system for the* 3^{rd} *minimum in Table 6.3.*

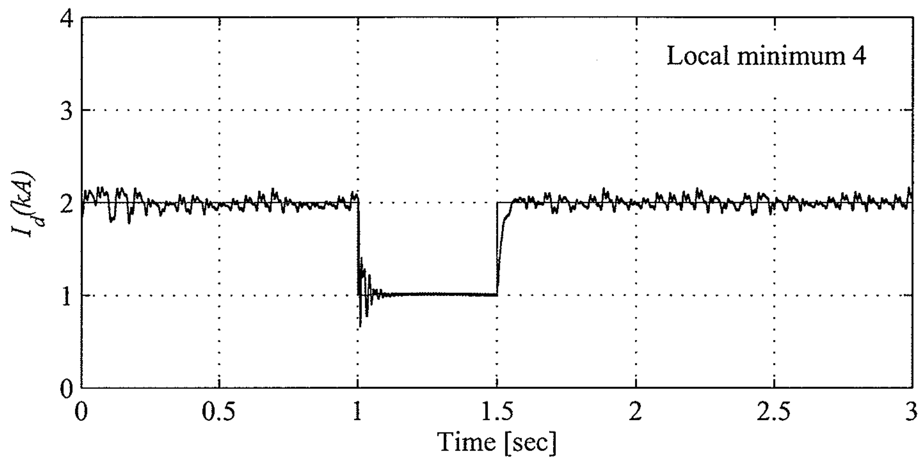


Fig. 6.16: *Transient response of the system for the* 4^{th} *minimum in Table 6.3.*

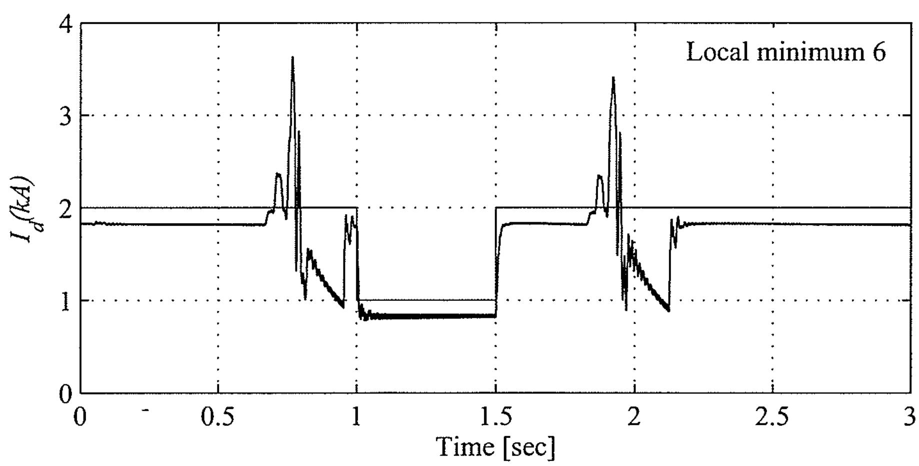


Fig. 6.17: *Transient response of the system for the* 6^{th} *minimum in Table 6.3.*

6.6 Example 4: HVDC Robust-Optimization

The rectifier and inverter parts of an HVDC system are connected by a transmission line to an ac network. The ac network is modeled using a voltage source and an impedance. The impedance is not constant during operation of a real ac network. The network impedance as well as the amount of reactive power provided by the filters, capacitive loads or synchronous compensators at the connecting node of ac network and HVDC converters play an important role in the performance of the system during any disturbance imposed on the system. The impedance of the network could be changed based on different loading conditions and changes of transmission lines. The amount of reactive power injection could also be changed due to changes in the number of synchronous compensators or capacitive loads at the ac bus. These effects are described using an index called short circuit ratio (SCR). A weak system is supposed to have an SCR factor of lower than 2.5 and a strong system is supposed to have an SCR factor of more than 4. The controller designed for the last HVDC example used a system with an SCR of 2.5. The designed controller is not guaranteed to work well for a system with a lower SCR. A weak system with a lower SCR is more susceptible to a commutation failure in comparison to a strong system. An approach to find a suitable optimum design for both weak and strong system is solved by using robust optimization approach. In robust optimization the objective function is defined to be the summation of a set of objective functions for a strong, medium and weak system [56]. Fig. 6.18 shows the robust optimization approach used for optimal design of a controller, which works for a weak system as well as a medium and a strong system.

The imposed disturbances on the system were modified to be

1. $t = 1.0sec$, the current order is reduced by 20%;

2. $t = 1.8sec$, the current order is restored to its initial value.

The objective function in this example was defined to be the summation of the objective functions of the systems with $SCR = 2$, $SCR = 3$, and $SCR = 5$.

$$OF(K_r, T_r, K_i, T_i) = of_{SCR=2} + of_{SCR=3} + of_{SCR=5} \tag{6.7}$$

where the $of_{SCR=2}$, $of_{SCR=3}$, and $of_{SCR=5}$ are defined using (6.6) for each system with the specified SCR value. The duration of objective function evaluation was from $T_0 = 0.8s$ to $T_f = 3$. The SCR value in each case was changed by assigning different ac network impedances for each system at the inverter ac bus.

6.6.1 Optimization Results

The optimal design using the multi-resolution optimization is shown in Table. 6.4. Six local minima were found for the robust optimization of the Cigre HVDC benchmark. The transient response of all local minima are depicted for all configurations of the system in Figs. 6.19 to 6.24.

All local minima except the 4^{th} local minimum have shown satisfactory results. This is because of the selection of the objective function evaluation interval which is from $T_0 = 0.8s$ to $T_f = 3s$. The 4^{th} local minimum has bad performance at start up; however, after startup the behaviour of the 4^{th} local minima is satisfactory; therefore, it is selected as a local minimum.

The proposed robust optimization implementation using the summation of the objective function values is shown to be capable of capturing local minima with good

Table 6.4: *HVDC robust optimization results.*

Local Minima	Control system parameters				Objective function value			
	Rectifier		Inverter					
	K_{pR}	T_{iR}	K_{pI}	T_{iI}	SCR=2	SCR=3	SCR=5	OF
(1)	1.9587	0.0081	0.6592	0.0816	1.00	0.77	0.65	2.57
(2)	1.4616	0.0029	0.6534	0.0528	1.18	0.84	0.77	2.80
(3)	1.9843	0.0313	0.6639	0.0685	1.43	0.76	0.76	2.95
(4)	2.0802	0.0676	0.6913	0.0771	1.53	0.89	0.86	3.28
(5)	14.446	0.0186	0.2723	0.0489	1.56	1.34	1.21	4.11
(6)	12.743	0.0078	0.2792	0.0873	1.68	1.54	1.16	4.35
Total number of simulation runs: 5730								
Search interval for $K_p : [0.01, 20]$					Search interval for $T_i : [0.001, 0.1]$			

performance for all systems with different SCRs. The different transient responses for each local minimum and each SCR value represent good performance; however, designers could choose one of these six local minima based on their other transient performance such as start up behaviour or overshoot at each disturbance.

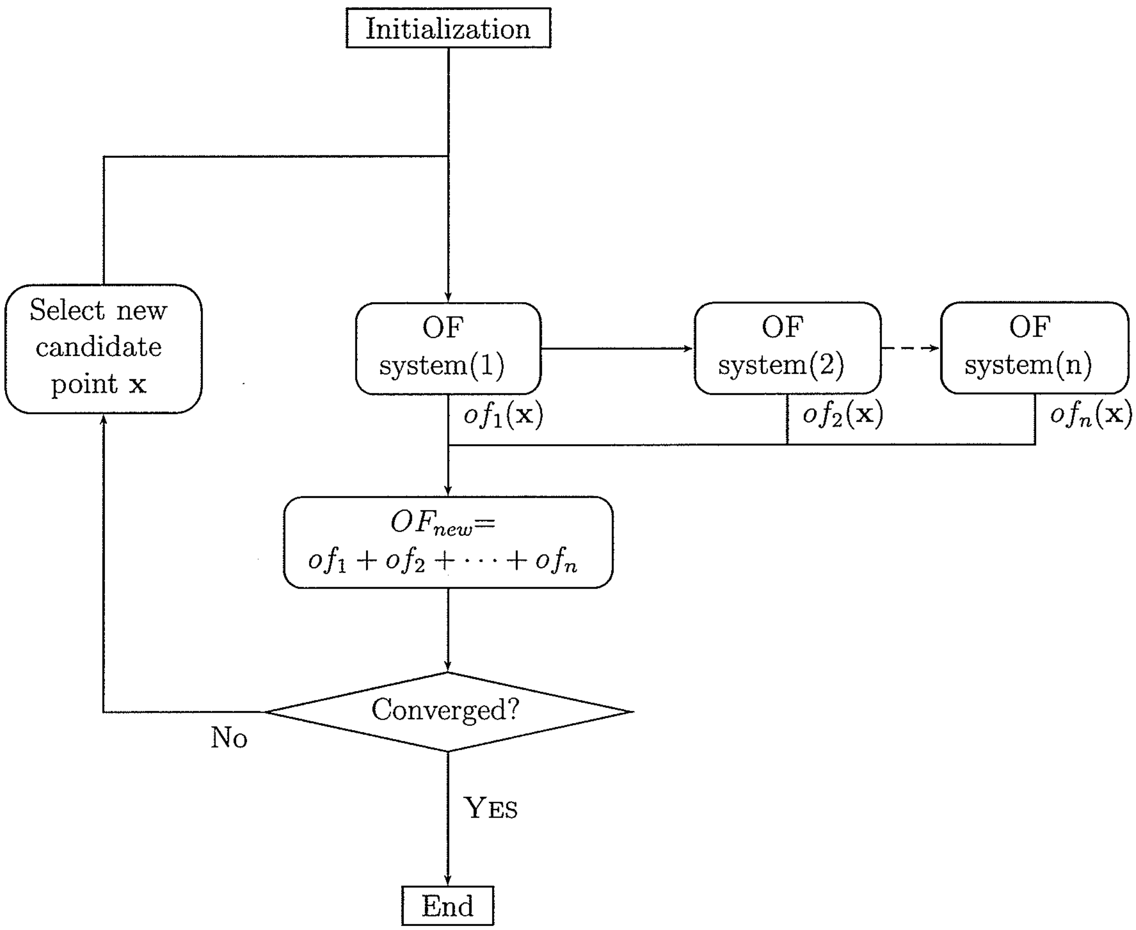
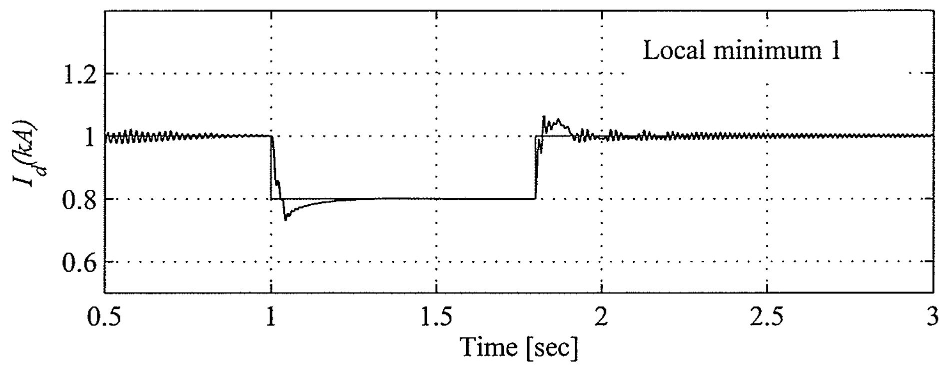
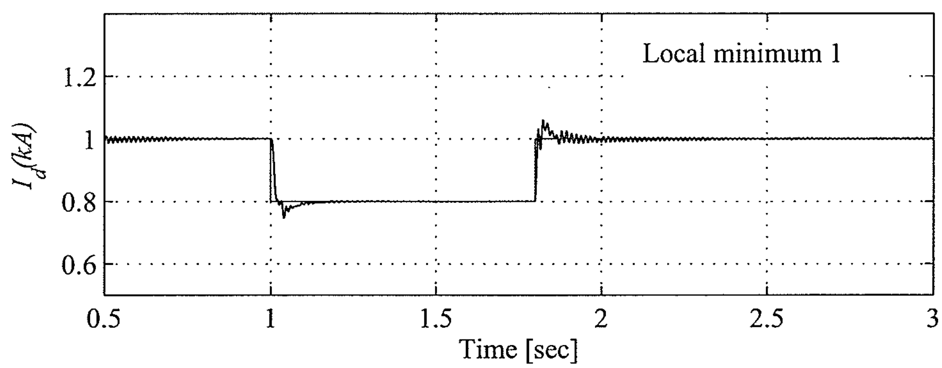


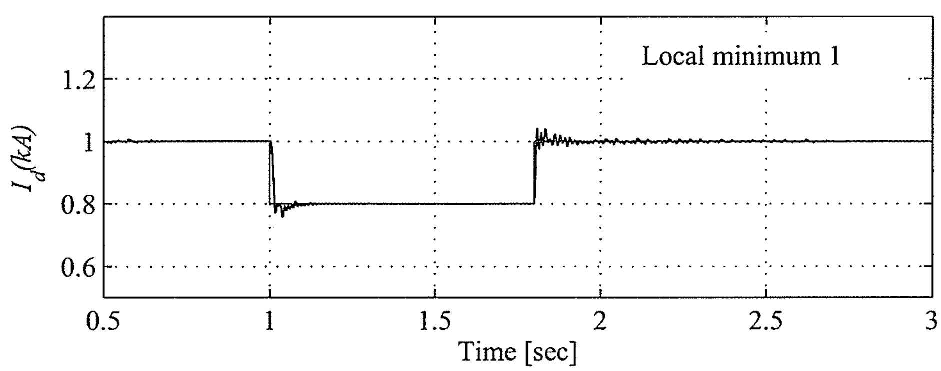
Fig. 6.18: *The robust optimization-enabled simulation tool.*



(a) SCR=2

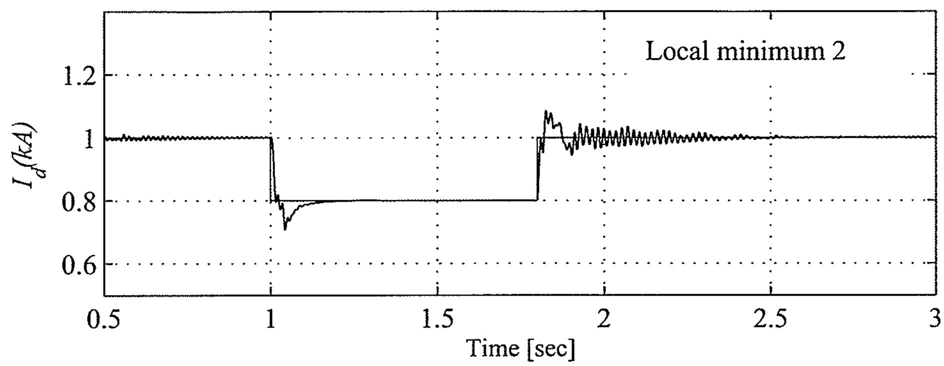


(b) SCR=3

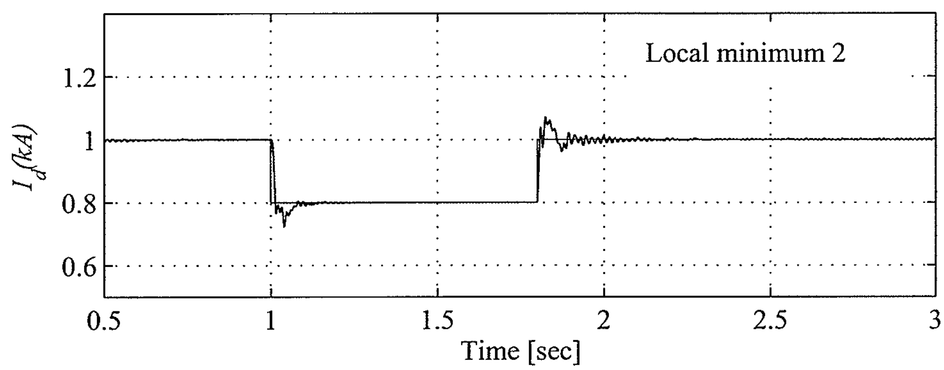


(c) SCR=5

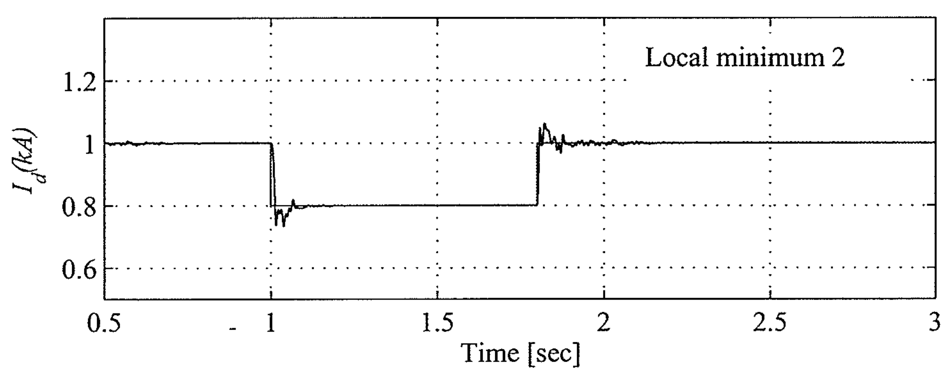
Fig. 6.19: *Transient response of the system for the* 1^{st} *local minimum in Table 6.4.*



(a) SCR=2



(b) SCR=3



(c) SCR=5

Fig. 6.20: *Transient response of the system for the* 2^{nd} *local minimum in Table 6.4.*

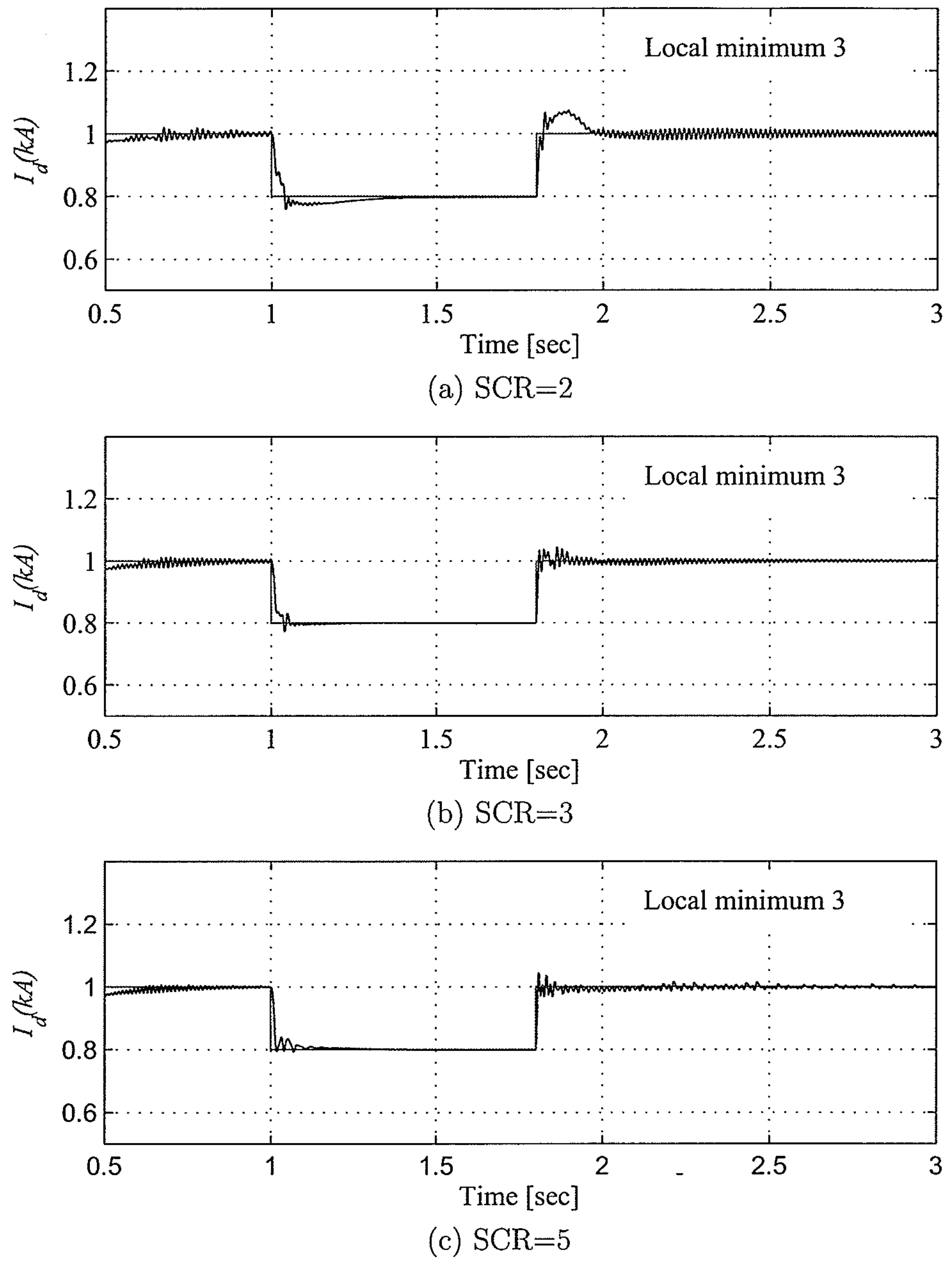
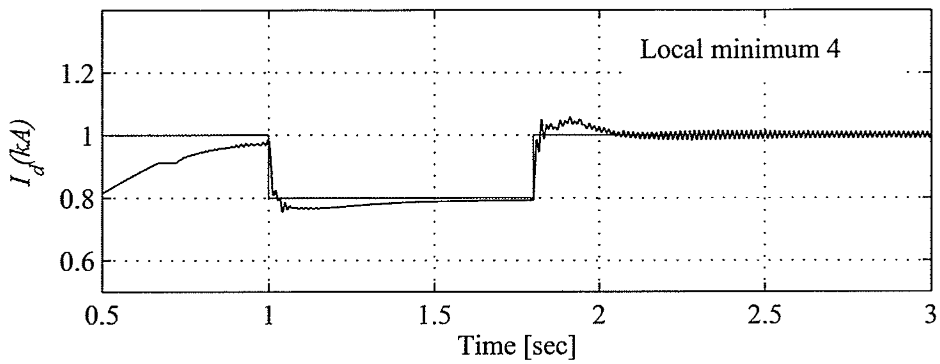
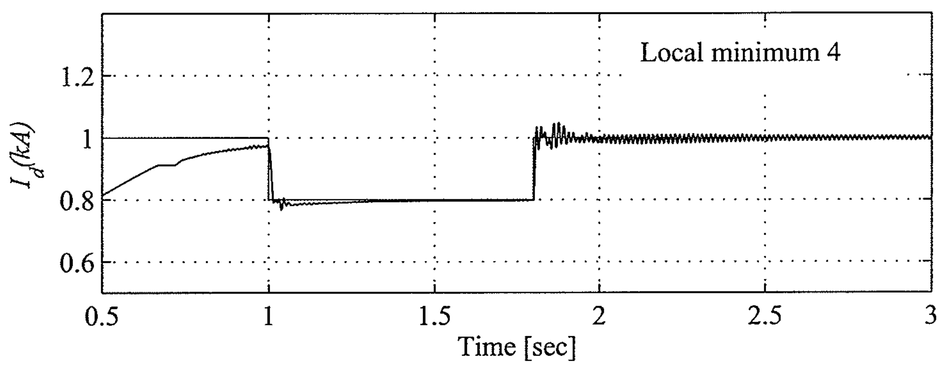


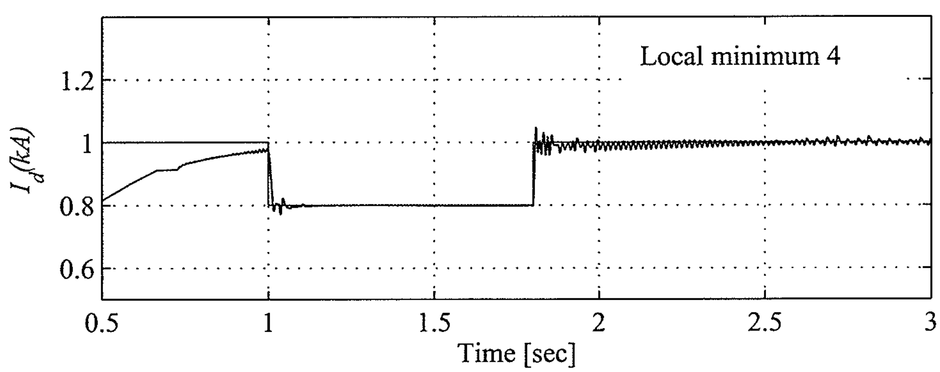
Fig. 6.21: *Transient response of the system for the 3^{rd} local minimum in Table 6.4.*



(a) SCR=2

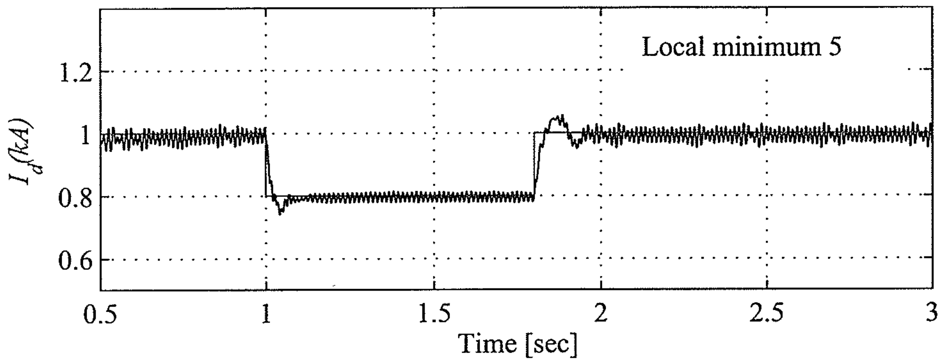


(b) SCR=3

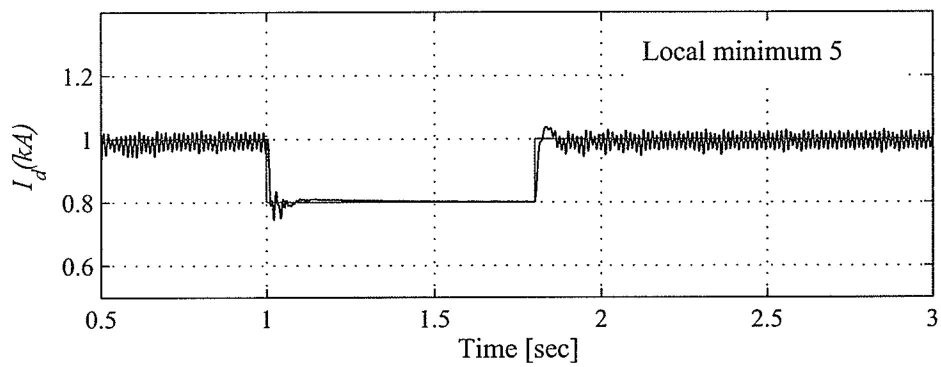


(c) SCR=5

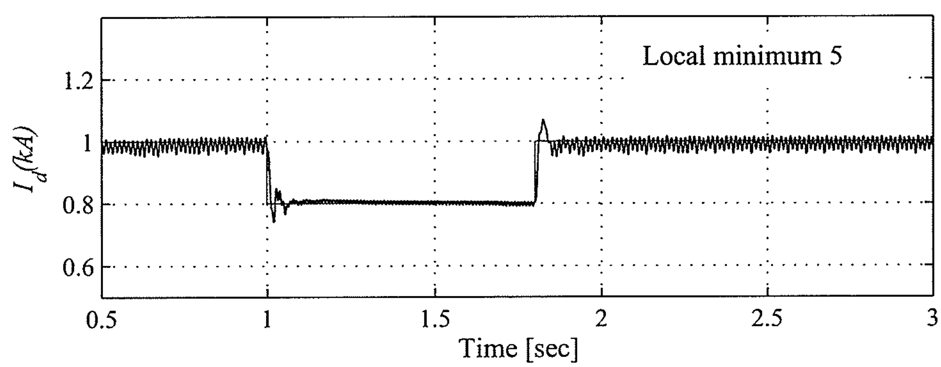
Fig. 6.22: *Transient response of the system for the* 4^{th} *local minimum in Table 6.4.*



(a) SCR=2



(b) SCR=3



(c) SCR=5

Fig. 6.23: *Transient response of the system for the* 5^{th} *local minimum in Table 6.4.*

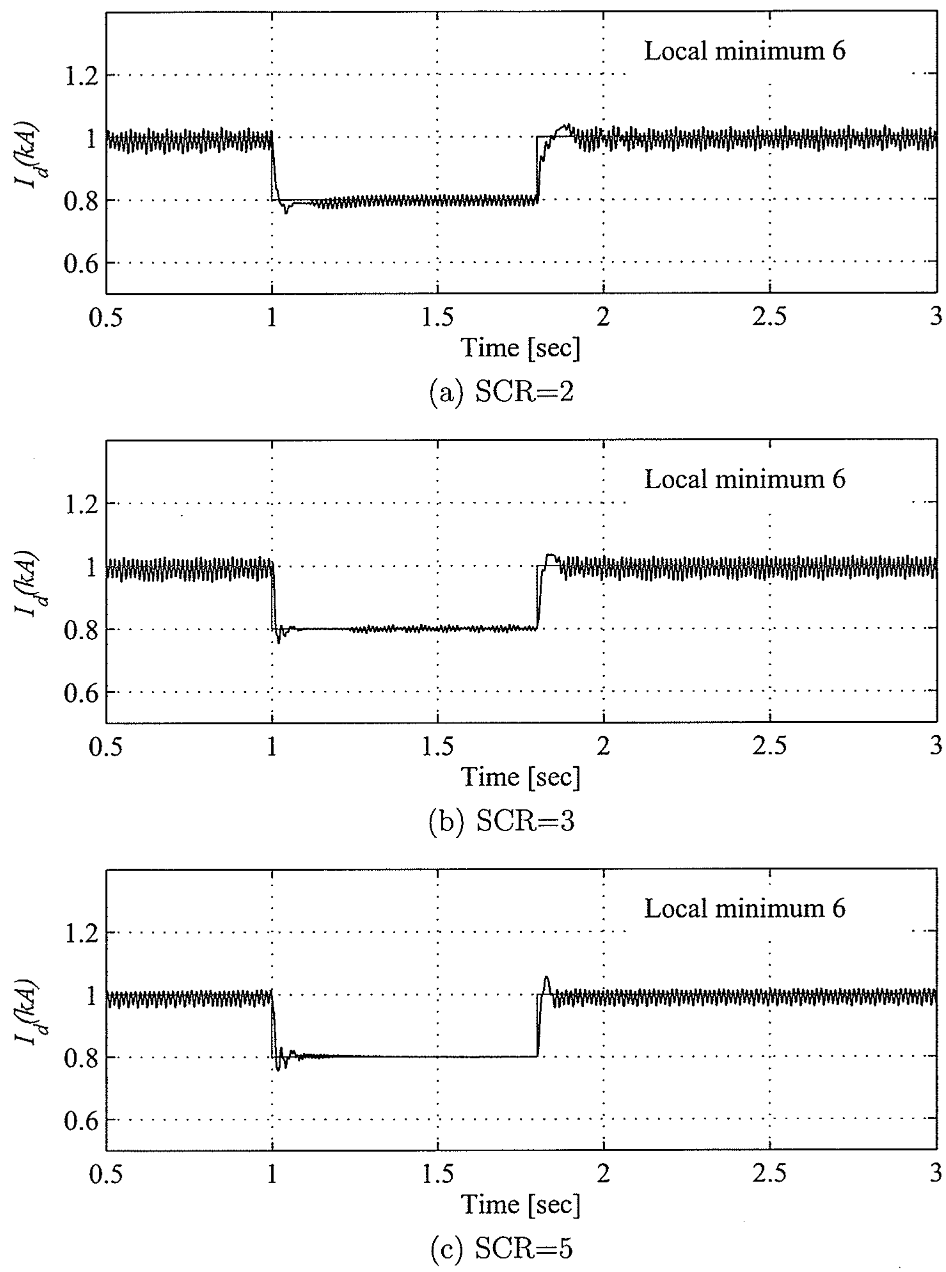


Fig. 6.24: *Transient response of the system for the 6^{th} local minimum in Table 6.4.*

Chapter 7

Conclusions

Optimization-enabled transient simulation was introduced previously as an efficient tool for optimal design of complex power systems. This design tool is a combination of a nonlinear optimization algorithm and an electromagnetic transient simulation program. The interfacing between the simulation and nonlinear optimization algorithm is provided by a well-defined objective function.

The objective functions selected may not always have only one local optimum. The existence of multiple solutions for an optimization case was shown analytically for an optimal pulse-width modulation previously, while due to lack of efficient methods capable of capturing all local minima of a simulation-based objective function, the phenomena of multi-modality was not investigated for simulation-based optimization cases. In this thesis a new method was introduced capable of capturing most local minima in an affordable number of simulations using a multi-resolution function reconstruction of the objective function.

7.1 Thesis Contributions

The contributions of this thesis are: (i) the development of a new optimization algorithm capable of capturing most local optima of an objective function; and (ii) investigation of the multi-modality phenomena in some power systems.

7.1.1 Development of the New Nonlinear Optimization Algorithm

A new nonlinear optimization algorithm was introduced in this thesis based on a combination of the response surface method and direct search method. This new algorithm samples the whole optimization space and then iteratively generates new sampling points at locations where local minima exist. The sampling of the whole optimization space enables the algorithm to discover most local minima. The algorithm performs a multi-resolution search to estimate the objective function. In order to make the proposed algorithm capable of solving a general constrained simulation-based optimization problem the following steps are done:

1. The sampling of the function is done by a finite element mesh, where the coordinates of the finite element nodes are the sets of parameters. The finite element mesh covers the optimization space with a multi-scale mesh. The sampling at nodes of the multi-scale mesh has enabled the algorithm to avoid excessive objective function evaluations in unnecessary regions. The multi-scale mesh provides a multi-resolution reconstruction of the objective function with the criteria of over-sampling at regions where local minima exist, thus enhancing the search process for finding local minima by orders of magnitude.

2. The distinguishing feature of using a finite element mesh for multi-resolution sampling is the ease of handling of equality and inequality constraints. The equality constraints are proposed to be handled by surface mesh generation techniques and the inequality constraints are handled effectively using, formerly introduced, implicit mesh generators.

3. The force relaxation method is incorporated in multi-resolution algorithm to enhance the initial mesh generation and subsequent mesh refinements. The force relaxation method has enabled the algorithm to generate an initial mesh with a desired initial sampling point distribution; thereby enabling the designer to incorporate a priori knowledge of the objective function in order to enhance the efficiency of the optimization process. Force relaxation was also shown to be useful for enhancing the efficiency of initial sampling. The optimization cases, which have used force relaxation for the initial mesh generation, have been able to capture more local minima comparing to cases that have not used force relaxation for initial mesh generation.

4. The drawbacks of every response surface or direct search optimization method is an excessive amount of initial sampling. The algorithm has properties that should allow it to be effectively parallelized to help address this phenomena.

5. The multi-resolution algorithm is dependent upon the point-connections generation using Delaunay triangulation. As the number of sampling points increases the Delaunay triangulation can not be used effectively, because of excessive memory usage requirement. This problem has been addressed using a partial mesh generation technique.

The incorporation of all these tools enables the multi-resolution optimization algorithm to effectively find most local minima of an objective function. This method was tested on various problems with known and unknown local optima and has shown satisfactory results for optimization of multi-modal objective functions.

7.1.2 Multi-Modality Phenomena in Power Systems

Multi-modality refers to optimization problems that contain more than one optimum; for example, optimal pulse-width modulation was shown to have more than one solution. The problem of multiple local minima was not previously investigated for power system optimization cases, where an analytical solution does not exist. The phenomena of multi-modality in power systems was investigated in this thesis for a problem of optimal selection of controller gains of a standard HVDC system. The algorithm has found more than one local minima for the studied system, which shows the existence of multi-modality phenomena in other power systems that incorporate switching converters.

7.2 Recommendations for Future Work

The main obstacles of multi-resolution optimization are: (i) large number of sampling point evaluations at the initial mesh; and (ii) a low rate of convergence to the local minima at higher resolutions. A potential solution to the first obstacle was discussed using parallel computation in Chapter 5. Other methods are also available to overcome these two obstacles including meta-modeling and multi-disciplinary optimization algorithms.

7.2.1 Incorporation of Meta-Modeling Technique

Meta-modeling could be implemented in situations where different estimating functions of the actual objective function exist. The drawback of these estimations is the uncertainty of their estimated value. However, uncertainty of objective functions is not important when the sampling points are far apart and the uncertainty is less than the objective function variation at these far points. The initial sampling points of a multi-resolution optimization are commonly at far distances; therefore, evaluations of these initial sampling points is not required to be done using a highly accurate measurement of the objective function. Incorporation of an approximation for the evaluations of the initial sampling points, with some uncertainty, will not alter the result of localization phase of multi-resolution algorithm during early iterations. Therefore, at early stages of the multi-resolution algorithm, objective function evaluations could be done using some approximation methods. The benefits of this approach is that these approximations could be done using some low cost estimating functions; for example, an analytical solution for the average case model of a power system. The incorporation of a low cost objective function will enhance the multi-resolution optimization algorithm during the early stages, where a large number of initial sampling point evaluations are required.

7.2.2 Incorporation of a Multi-Optimizer Strategy

The multi-resolution optimization algorithm has shown great ability to separate regions contributing to each local minima during early iterations (the exploration phase). during subsequent iterations the algorithm only increases the accuracy of the estimated local minima (the exploitation phase). The multi-resolution algorithm is very effective during the exploration phase, while the efficiency of the algorithm during the exploitation phase is often worse than conventional nonlinear optimization algorithms such as simplex or gradient-based methods. These properties makes the multi-resolution optimization algorithm a suitable option for a multi-disciplinary optimization algorithm, where the exploration phase is done using multi-resolution optimization and the exploitation phase is done by switching to other methods such as simplex or gradient-based solver. The multi-resolution algorithm would then be responsible to optimize the objective function until the regions of local minima are clearly separated and then the simplex or gradient-based method would use each estimated local minima as an initial guess to find an accurate location of each optimal point.

Appendix A

Transformation Matrix

The required transformation matrix for mapping the points along the vector connecting two points of x_1 and y_1 to a surface at x_1 and perpendicular to this vector can be stated as the construction of a matrix, where the first column of the matrix must be a vector parallel to $\mathbf{x}_1 - \mathbf{y}_1$ and the other columns must be perpendicular to each other and also to the first column. The following algorithm is introduced to produce such a transformation matrix. The first column is the vector $\frac{(\mathbf{x}_1-\mathbf{y}_1)}{\|(\mathbf{x}_1-\mathbf{y}_1)\|}$ and the remaining columns are derived using an iterative approach to guarantee that each column is perpendicular to the other columns.

$a_1 = \frac{(\mathbf{x}_1-\mathbf{y}_1)}{\|(\mathbf{x}_1-\mathbf{y}_1)\|}$

for $i = 2 \rightarrow d$

$a_{i,d\times 1} = [-a_1(1), 0, \ldots, 0]^T$

for $j = 2 \rightarrow i-1$

$a_i(j) = -\frac{a_i^T \otimes a_j}{a_j(j)}$

end

$a_i(i) = -\frac{a_i^T \otimes a_0}{a_0(i)}$

$a_i = \frac{a_i}{\sqrt{a_i^T \otimes a_i}}$

end

$A = [a_1, ..., a_d]$

where a_i is the i^{th} column of the transformation matrix (A), $\otimes$ indicates matrix product, $a_i(j)$ is the j^{th} element of the column vector a_i, and d is the dimension of the transformation. The iterative approach defined above guarantee, that the columns of the transformation matrix will be perpendicular to each other, while the first column is parallel to $(\mathbf{x}_1 - \mathbf{y}_1)$; therefore, the affine transformation introduced here of the matrix A satisfies the requirements specified in chapter 5.

References

[1] A. M. Gole, S. Filizadeh, R. W. Menzies, and P. L. Wilson, "Optimization-enabled electromagnetic transient simulation," *IEEE Trans. Power Delivery*, vol. 20, no. 1, pp. 512-518, January 2005.

[2] C. Osauskas and A. Wood, "Small-signal dynamic modeling of HVDC systems," *IEEE Trans. Power Delivery*, vol. 18, no. 1, pp. 220-225, January 2003.

[3] H. W. Dommel, "Digital computer solution of electromagnetic transients in single and multiphase networks," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-88, no. 4, pp. 388-399, April 1969.

[4] Manitoba HVDC Research Center, EMTDC User's Guide, Winnipeg, MB, Canada, 2005.

[5] V. G. Agelidis, A. Balouktsis, I. Balouktsis, and C. Cossar, "Multiple sets of solutions for harmonic elimination PWM bipolar waveforms: Analysis and experimental verification," *IEEE Trans. Power Electron.*, vol. 21, no. 2, pp. 415-421, March 2006.

[6] M. Bjorkman and K. Holmstrom, "Global optimization using the DIRECT algorithm in Matlab," *Advanced Modeling and Optimization*, vol. 1, no. 2, pp. 17-37, 1999.

[7] A. R. Hedar and M. Fukushima, "Tabu Search directed by direct search methods for nonlinear global optimization," *European Journal of Operational Research*, vol. 170, no. 2, pp. 329-349, April 2006.

[8] K. B. Ensor and P. W. Glynn, "Stochastic optimization via grid search," *Amer. Math. Soc.*, vol. 33, pp. 89-100, 1997.

[9] S. Filizadeh, "Optimization-enabled electromagnetic transient simulation," Ph.D. Dissertation, University of Manitoba, Winnipeg, Canada, 2004.

[10] Y. Carlson and A. Maria "Simulation optimization: methods and applications," in *Proc. 1997 Winter Simulation Conference*, pp. 118-126.

[11] S. Andradottir, "A review of simulation optimization technique," in *Proc. 1998 Winter Simulation Conference*, pp. 151-158.

[12] P. I. Barton and C. K. Lee, "Modelling, Simulation, sensitivity analysis, and optimization of hybrid systems," *ACM Transanction on Modeling and Computer Simulation*, vol. 12, no. 4, pp. 256-289, October 2002.

[13] M. Huang and J. S. Arora, "Optimized design with discrete variables: some numerical experiments," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 1, pp. 165-188, 1997.

[14] J. R. Swisher, P. D. Hyden, S. H. Jacobson, and L. W. Schruben, "A survey of simulation optimization techniques and procedures," in *Proc. 2000 Winter Simulation Conference*, pp. 119-127.

[15] F. Azadivar, "Simulation optimization methodologies," in *Proc. 1999 Winter Simulation Conference*, pp. 93-100.

[16] S.L. Woodruff, "Complexity in power systems and consequences for real-time computing," *Power Systems Conference and Exposition*, vol. 3, pp. 1770-1775, 10^{th} October 2004.

[17] P. J. M. Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, Norwell, MA, USA: Kluwer Academic Publishers, 1987.

[18] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3^{rd} ed., New York: Springer, 1996.

[19] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design operation," *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370-380, 2007.

[20] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: new perpective on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385-482, 2005.

[21] A. Wachter, Application of an interior point method for large-scale nonlinear programming to circuit tuning. Presented at January 2002 IMA workshop on Optimization in Simulation-Based Models. [online on 27^{th} Aug. 2007]. Available: http://www.ima.umn.edu/talks/workshops/1-9-16.2003/waechter/IMA02.pdf

[22] L. Wang, S. Shan, and G. G. Wang, "Mode-Pursuing sampling method for global optimization on expensive black-box functions," *Engineering Optimization*, vol. 36, no. 4, pp. 416-438, August 2004.

[23] P. Shen and C. Wang, "Global optimization for sum of linear ratios problem with coefficients," *Applied Mathematics and Computation*, vol. 176, no. 1, pp. 219-229, May 2006.

[24] P. A. Parrilo and B. Sturmfels, "Minimizing Polynomial Functions," *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 60, pp. 83-99, 2003.

[25] T. T. Allen and L. Yu, "Low-Cost response surface methods from simulation optimization," *Quality and Reliability Engineering International*, vol. 18, no. 1, pp. 5-17, 2002.

[26] A. A. Zhigljavsky, *Theory of Global Random Search*, Dordrecht, Boston, London: Kluwer Academic Publishers, 1991.

[27] Ladkhdar Chiter, "A new Sampling method in the DIRECT algorithm," *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 297-306, 2006.

[28] A. A. Torn and A. Zilinskas, *Global Optimization, Lecture Notes in Computer Science 350*, Nw York: Springer, 1989.

[29] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications, Lecture Notes in Computer Science*, New York: Springer, 1987.

[30] M. P. Friedlander and M. A. Saunders, "A Globally convergent linearly constraint Lagrangian method for nonlinear optimization," *SIAM Journal of Optimization*, vol. 15, no. 3, 2005, pp. 863-897.

[31] M. Andramonov and A. Rubinov, "Cutting angle method in global optimization," *Applied Mathematics Letters*, vol.12, no.3, pp. 95-100, 1999.

[32] G. Beliakov and K. F. Lim, "Challenges of continuous global optimization in molecular structure prediction," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1198-1213, 2007.

[33] E. R. Hansen, *Global Optimization Using Interval Analysis*, New York: Marcel Dekker, 1992.

[34] K. Bousson and S. D. Correia, "Optimization algorithm based on densification and dynamic canonical descent," *Journal of Computational and Applied Mathematics*, vol. 191, no. 2, pp. 269-279, July 2006.

[35] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural Multidisciplinary Optimization*, vo.l 26, no. 6, pp. 369-395, April 2004.

[36] S. Shan and G. G. Wang, "An efficient pareto set identification approach for multiobjective optimization on black-box functions," *Journal of Mechanical Design*, vol. 127, no. 5, pp. 866-874, September 2005.

[37] R. O. Bowden and J. D. Hall," Simulation optimization research and development," in *Proc. 1998 Winter Simulation Conference*, pp. 1693-1697.

[38] P. Apkarian and D. Noll, "Controller design via non-smooth multidirectional search," *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 1923-1949, 2006.

[39] M. Heinkenschloss and L. N. Vicente, "An interface between optimization and application for the numerical solution of optimal control problems," *ACM Transactions on Mathematical Software*, vol. 25, vo. 2, pp. 157-190, June 1999.

[40] S. Smale and D. Zhou, "Shanon sampling and function reconstruction from point values," *Bulletin of the American Mathematical Society*, vol. 41, no. 3, pp. 279-305, April 2004.

[41] J. Huang and Y. Chen, "A regularization method for the function reconstruction from apprpximate average fluxes," *Institute of Physics Publishing, Inverse Problems*, vol. 21, no. 5, pp. 1667-1684, October 2005.

[42] J. C. Xu and W. C. Shann, "Galerkin-wavelet methods for two point boundary value problems," *Numer. Math.*, vol. 63, pp. 123-142, 1992.

[43] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz, "Adaptive re-meshing for compressible flow computations," *J. Comput. Phys.*, vol. 72, no. 2, pp. 449-466, 1987.

[44] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469-483, December 1996.

[45] Q. Du, V. Faber ,and M. Gunzburger, "Centroidal Voronoi tessellations: applications and algorithm," *SIAM Review*, vol. 41, no. 4, pp. 637-676, 1999.

[46] Q. Du and D. Wang, " Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations," *International Journal for Numerical Methods in Engineering* , no. 56, pp. 1355-1373, 2003.

[47] P. O. Persson, "Mesh generation for implicit geometries," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, 2005.

[48] P. O. Persson and G. Strang, "A Simple mesh generator in Matlab," *SIAM Review*, vol. 46, no. 2, pp. 329-345, 2004.

[49] G. Leibon and D. Letscher, " Delaunay triangulation and Voronoi diagrams for Riemannian manifolds," in *Proc. June 2000* 16^{th} *Symp. Computational Geometry, ACM*, pp. 341-349.

[50] J. Boissoinnat and S. Oudot, "Provably good sampling and meshing of surfaces," *European Symposium on Geometry Processing*, vol. 67, no. 5, pp. 405-451, September 2005.

[51] N. Amenta and M. Bern, "Surface reconstruction by Voronoi filtering," in *Proc. 1998 fourteenth annual symposium on Computational geometry*, pp. 39-48.

[52] R. Lohner, "A parallel advancing front grid generation scheme," in *Proc. October 1999* 8^{th} *International Meshing Roundtable, South Lake Tahoe, CA, U.S.A.*, pp. 67-74.

[53] A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, *Engineering Optimization*, 2^{nd} ed., 2006.

[54] S. Filizadeh and A. M. Gole, "Harmonic Performance Analysis of an OPWM-Controlled STATCOM in Network Applications," *IEEE Trans. Power Delivery*, vol. 20, no. 2, pp. 1001-1008, April 2005.

[55] M. Sczechtman, T. Wess, and C. V. Thio, "First benchmark model for HVDC control studies," *Electra*, no. 135, pp. 55-73, April 1991.

[56] A. M. Gole, S. Filizadeh, P. L. Wilson, "Inclusion of robustness into design using optimization-enabled transient simulation," *IEEE Trans. Power Delivery.*, vol. 20, no. 3, pp. 1991-1997, July 2005.