Hybrid and Parallel-Computing Methods for Optimization of Power Systems with Electromagnetic Transient Simulators

by

Dilsha Kuranage

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, Canada

Abstract

This thesis introduces new methods for using electromagnetic transient (EMT) simulators to efficiently optimize controllers of the power electronic converters in power systems with complicated dynamic behavior. This work is motivated by several challenges that must be overcome during the design process, including high computational burden of simulating large switching systems, repetitive nature of the design cycle, the large number of variables that need to be handled, etc. These challenges are addressed in this research by combining an EMT simulator with optimization algorithms and by developing novel approaches to reduce the entire simulation time.

Two screening methods are introduced in this thesis that can identify non-influential parameters so that the number of parameters to be optimized can be reduced, thus decreasing the computational burden of the process. Moreover, multi-algorithm and parallel processing techniques are developed to achieve additional computational benefits by making the design process faster. In this research, new pathways are created to solve simulation-based design problems with a large number of parameters by amalgamating all the above approaches.

Several power system examples are simulated using PSCAD/EMTDC, and the accuracy and efficiency of the proposed methods are assessed and confirmed. The results show significant reductions in the time to design optimal systems without compromising the quality of

the optimal performance.

Copyright Notes

Part of the contents of this thesis are taken from the following article:

D. Kuranage, S. Filizadeh, and D. Muthumuni, "Improved methods for optimization of power systems with renewable generation using electromagnetic transient simulators," Electric Power Systems Research, Volume 220, July 2023, 109308.

https://doi.org/10.1016/j.epsr.2023.109308

Acknowledgements

This thesis would not have been possible without the help of many individuals and organizations.

First and foremost, I would like to express my sincere gratitude to my academic advisor Prof. Shaahin Filizadeh for his immense support, guidance, patience and encouragement throughout this research.

Besides my advisor, I would like to thank Dr. Dharshana Muthumuni for the valuable advice and suggestions given for my research.

I wish to extend my special thanks to Mr. Srimal Koruwage for the technical assistance provided throughout my work.

I would like to acknowledge the financial support from the University of Manitoba, NSERC, Manitoba Hydro International, and MITACS.

Finally, I would like to express my deepest gratitude to my husband, family and friends for their priceless support and love shown to me throughout this journey.

Dedications

To my loving husband; Buddhi.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With increasing demand for electricity and radical changes brought about by converter-based systems, renewable energy resources, and energy storage, existing power systems are becoming increasingly complex. The complexity of renewable-intensive power systems is spurred by the presence of switching converters, sophisticated control systems, and intricate dynamic behaviour. Converter-based generation schemes have shown rapid growth resulting in challenges in the design and operation of the system [1]. Converter-tied resources not only diminish the system inertia, but also release high-frequency harmonics to the system due to the high-frequency switching of the power electronic devices included in them [1]. To ensure power quality and supply reliability, converters are equipped with sophisticated control systems whose parameters must be selected carefully and after extensive studies [2]. However, their highly non-linear and discontinuous nature makes it prohibitively difficult to apply analytical methods to solve such design problems [3]. Therefore, it is necessary to have simulation-based methods and tools to model these systems in a detailed and accurate manner so that their dynamic performance can be successfully assessed, and their parameters be tuned before actual implementation [4, 5].

Electromagnetic transient (EMT) simulators are widely used for modeling converter-intensive power systems. Unlike other power system simulators (e.g., transient stability and RMS-type solvers), which ignore fast transients, EMT simulators do include high-frequency transients and provide a detailed representation of the system performance [1]- [2]. This makes them extremely suitable as a reliable evaluator of the system's performance for a given set of design parameters. A crucial task within the design process is selecting optimal parameter values for the components and controllers. Using the trial-and-error method for this purpose is not efficient - although it is often practiced - particularly when there are a large number of parameters to be optimized. Alternatively, formal optimization algorithms can be used together with EMT simulators; this approach saves significant time and effort by generating candidate parameter sets in an intelligent and automated manner using the optimization algorithm. The suitability of the generated candidate parameter sets is evaluated by the EMT simulator [5].

Even though an EMT simulator interfaced with an optimization algorithm is a powerful design tool, the existing methods are still computationally inefficient when tens or hundreds of parameters need to be optimized. In addition to the high computational complexity of EMT simulation models and the repetitive nature of the design process, inherent limitations (e.g., large populations, slow convergence, sequential execution, etc.) imposed by conventional optimization algorithms must also be overcome to improve upon the current approaches of using optimization-enabled electromagnetic transient simulation (OE-EMTS).

In this thesis, novel methods are developed to overcome the above challenges. Two screening methods are devised to reduce the dimension of the design problem by identifying the parameters that do not require optimization. Moreover, combined optimization algorithms and parallel processing techniques are explored to accelerate the design process. The efficiency and accuracy of the proposed methods are validated in optimal design of complicated controllers in power systems that represent real-world applications.

1.2 Problem Definition

When analytical methods are not sufficient to solve existing complex power system optimization problems, simulation-based design is the best solution. EMT simulators are widely used in this area since they can model power system components in a detailed and accurate manner [4]. The mainstream solution method in EMT simulators is to create equivalent circuits for system components using an integration method and then solving the circuit using nodal analysis based upon the admittance matrix of the network under consideration [6]. To solve for node voltages, the network's admittance matrix must be inverted. Due to switching actions, e.g., in power-electronic devices, the admittance matrix continually changes; therefore, inversions must be done repetitively. In addition to that, the design cycle itself is repetitive and causes an immense computational burden. Accordingly, the simulation-based design process requires novel remedies to reduce its computational burden and accelerate the process.

In regard to the optimization algorithms, genetic algorithms and nonlinear Simplex method have been widely adopted for simulation-based design of complex power systems. There exists previous work that has used OE-EMTS to optimize power system parameters, where the candidate parameter sets are generated by the optimization algorithm in an intelligent manner and the suitability of those parameter sets is evaluated by the EMT simulator [5]. In [5], OE-EMTS is used to optimize the parameter values of a voltage source converter and a dc-dc converter, while the same methodology has been used to design a HVDC controller in [7]. In both cases, Nelder-Mead's Simplex algorithm is used since it performs well for optimization of a relatively small number of parameters [5,7]. Even though the Simplex algorithm is computationally efficient, it is prone to converging into a local optimum. Moreover, in the context of optimizing power systems with power electronic devices, which tend to have a large number of parameters, the Simplex algorithm may not be a suitable choice [8]. The authors in [9] have suggested using genetic algorithms (GAs) to obtain optimal design for photovoltaic grid-connected systems, considering their efficiency and cost. There is a high likelihood of GA converging into the global optimum since it considers sets (populations) of solutions and includes advanced operators such as mutation and crossover that tend to diversify the solution set [8]. However, GAs take considerable time to complete the optimization process. For cases with a large number of parameters, it might take prohibitively long to finish the simulation. Thus, there are several inherent limitations of the optimization algorithms that should be overcome during the design process.

Therefore, in the context of using OE-EMT simulation to design complicated controllers in power systems, complexities such as high computational burden, repetitive nature of the design cycle, large number of parameters to be handled, and the inherent limitations of the optimization algorithms cause difficulties. This thesis introduces a number of novel methods to perform screening that identifies and removes non-influential variables to lower the dimension of the optimization problem, thus reducing its complexity and accelerating the design process using combined optimization algorithms and parallel-processing approaches. To demonstrate the efficacy of the proposed methods, design parameters of a HVDC controller and two different type-4 wind turbine generator controllers are optimized. All cases prove to be extremely challenging for manual parameter tuning due to the number of parameters and the complexity of the dynamic behaviour of the networks. The results confirm that the proposed approaches are effective in optimization of complicated power systems using EMT simulators.

1.3 Thesis Objectives

It is clear that the existing methods of using EMT simulation are not efficient enough to design converter-intensive power systems since they need significant computational power and time. Thus the main objective of this thesis is to develop novel methods to overcome these problems and to improve the efficiency of the design process.

There are several factors that must be considered to reduce the computational burden and simulation time of OE-EMTS design problems. One of them is the dimension of the problem at hand, i.e., the number of parameters to be optimized. In the context of converter-intensive power systems, there are a lot of parameters to be optimized. Not all of these parameters may affect the design objectives similarly. Therefore, as the first objective of this thesis, screening methods are developed to identify the non-influential parameters that do not crucially impact the design goals, and hence can be excluded without optimizing. By doing so, the dimension of the problem can be lowered while reducing its complexity. The second objective of this research is overcoming the inherent limitations of optimization algorithms by combining multiple algorithms. The expectation from this novel approach is achieving further computational benefits by integrating profitable traits of several algorithms.

The OE-EMT simulation-based design is an iterative process, which runs until the optimization algorithm finds the best parameter set that satisfies the design objectives. Therefore, the design process can be accelerated if parallel-processing techniques are adopted, which is another objective explored in this thesis. As the final objective, the novel approaches proposed in the thesis are applied to several case studies that represent real-world systems to demonstrate the efficacy of the proposed methods.

1.4 Thesis Organization

The thesis is organized in a way that it gives meticulous understanding about the novel methods found in the research and application of them to overcome the design problems.

Chapter 2 provides an overview of electromagnetic transient (EMT) simulators and optimization-enabled electromagnetic transient simulation programs. In the later part of the chapter, fundamental steps to build an objective function (OF) are explained along with the methods to modify it according to the design requirements.

Chapter 3 explains the operation of optimization algorithms and their classifications. Moreover, the principles and operations of the GA and Simplex algorithm are described in detail in this chapter.

Following Chapter 3, Chapter 4 presents a comprehensive explanation about the novel methods introduced in the thesis and the way they are implemented.

The application of proposed methods in the designing of power systems is discussed in Chapter 5 by using some real-world case studies. The results relevant to every approach are presented and compared in this chapter.

Chapter 6 concludes the thesis by presenting the conclusions and contributions of the research and provides recommendations for the future work related to the topic.

Chapter 2

Optimization-Enabled Electromagnetic Transient Simulation (OE-EMTS)

2.1 Electromagnetic Transient (EMT) Simulators

In early days of modern power systems, transient analysis was done using mathematical calculations [10]. It is complicated to use analytical methods to model existing power systems, which include a large number of switching power-electronic devices and non-linear components [2]. Even though it is possible to develop analytical representations with average-value models, they cannot fully mimic the exact response of a system including fast dynamics [2].

With the development of computer technology, electromagnetic transient simulation programs were introduced, which model the power system efficiently and give highly accurate simulation results over a wide range of frequencies [6]. There are two main approaches in

EMT simulations, namely state space-based and companion circuit-based nodal analysis approaches [6]. In the former, state equations are derived for the circuit and are solved using numerical integration methods such as trapezoidal rule [6]. This method is not popular since generation of state space equations for large systems consumes significant time and effort [6]. The popular approach is the latter, which focuses on transforming circuit elements to conductances in parallel with current sources and solving the resulting circuit using nodal analysis [6].

Since EMT programs are capable of simulating complex power systems, wide-ranging tasks such as controller tuning, protection coordination studies, power quality studies, and determination of equipment stresses can be readily carried out [5] [11]. The performance of a system can be evaluated by simulating its behavior in an EMT simulator while modifying its parameters until the expected outcome is achieved [5]. This makes the design procedure less complicated than the previously used analytical methods and helps the designer to form a deep understanding about the operation of the actual system [11]. The EMT simulator used in this thesis is PSCAD/EMTDC.

2.2 Overview of OE-EMTS

Selecting suitable values for the design parameters is a vital part of the design cycle. To find the most fitting values for a circuit, the designer must consider a large number of possible solutions and evaluate the system's performance for each solution [2]. In this case, there should be a methodology to generate trial parameter sets. The performance of the system for a given set of parameters is measured by the EMT simulator, which requires a metric formulated in the form of an objective function (OF), to determine the closeness between the actual and desired performance [5]. Low OF values typically indicate that the actual and the expected performance of the system are close, thus indicating a high-quality design.

Monte-Carlo and multiple-run methods are widely-used approaches to generate the candidate points; however, due to their unstructured search pattern, they are inefficient and have poor accuracy [5]. OE-EMTS is an advanced and efficient tool for the design of complex power systems using an EMT solver in conjunction with a (nonlinear) optimization algorithm as depicted in Fig. 2.1.

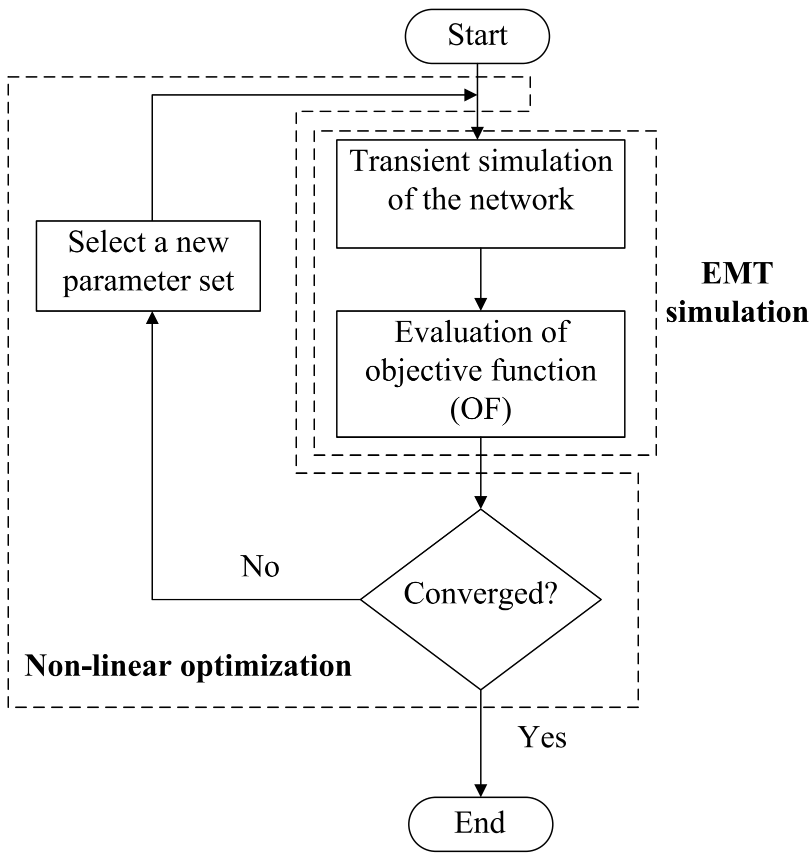

Figure 2.1: Schematic flow diagram of the OE-EMTS [12].

The role of the optimization algorithm is to generate new candidate values for the design parameters in an intelligent manner. The EMT simulator runs repetitively with new values assigned to it in each run, and after every simulation run the evaluated OF is given to the optimization algorithm, which uses it to judiciously generate new parameter sets that lead to lower OF value. This process ends after obtaining a parameter set that gives a minimum OF value that satisfies the design requirements. Due to the strategic search abilities of a non-linear optimization algorithms, the simulation-centered optimization process takes comparatively fewer iterations than manual trail-and-error [5].

2.3 Development of Objective Functions for Simulation-Based Optimization

OE-EMTS requires an OF, which includes all the design objectives, to evaluate the performance of a given system [7]. The OF is a measure of closeness between the actual output and the desired output [12]. Therefore, the lower the OF value, the more fitting the design. To further understand OFs, an example is presented here assuming that the functions in (2.1) and (2.2) describe the actual and expected responses of a system, respectively. The design objective is to make Y_1(t,a) and $Y_2(t)$ closer as possible.

$$Y_1(t,a) = 1 + ae^{-at}\sin(8at) \tag{2.1}$$

$$Y_2(t) = 1 \tag{2.2}$$

The actual response of the system depends on parameter a whose value changes the performance of the system as illustrated in Figs. 2.2 and 2.3. To achieve the goal of the design, the error between the two functions must be minimized. Hence, an OF is defined as the integral of the absolute difference between the two functions as in (2.3), where $[0,T]$ is the

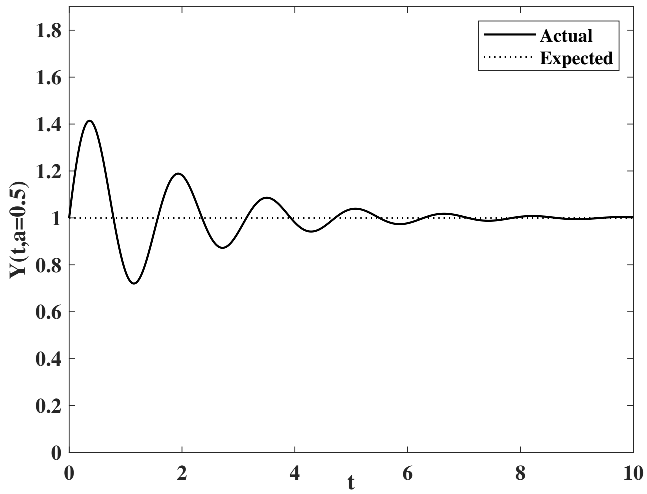

Figure 2.2: $Y_1(t, a)$ function variation when $a = 0.5$.

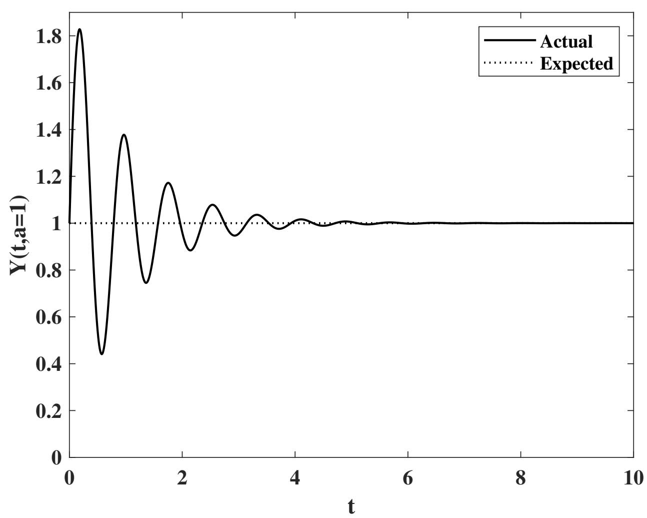

Figure 2.3: $Y_1(t, a)$ function variation when $a = 1$.

time period over which the OF is calculated.

$$OF(a) = \int_0^T |Y_1(t, a) - Y_2(t)| \, \mathrm{d}t \tag{2.3}$$

The goal of the optimization algorithm is to select a such that the OF is minimized. In simulation-based optimization, the OF is evaluated by the EMT simulator. The EMT simulator does not require an explicit expression of the system performance, which is often hard to achieve due to the complexity of the design problem. A number of other arrangements for the OF in this design problem are shown in Fig. 2.4. The designer can select a suitable OF formulation according to the design problem and modify it to include all design objectives. Moreover, there can be situations where the OF must include different weighting

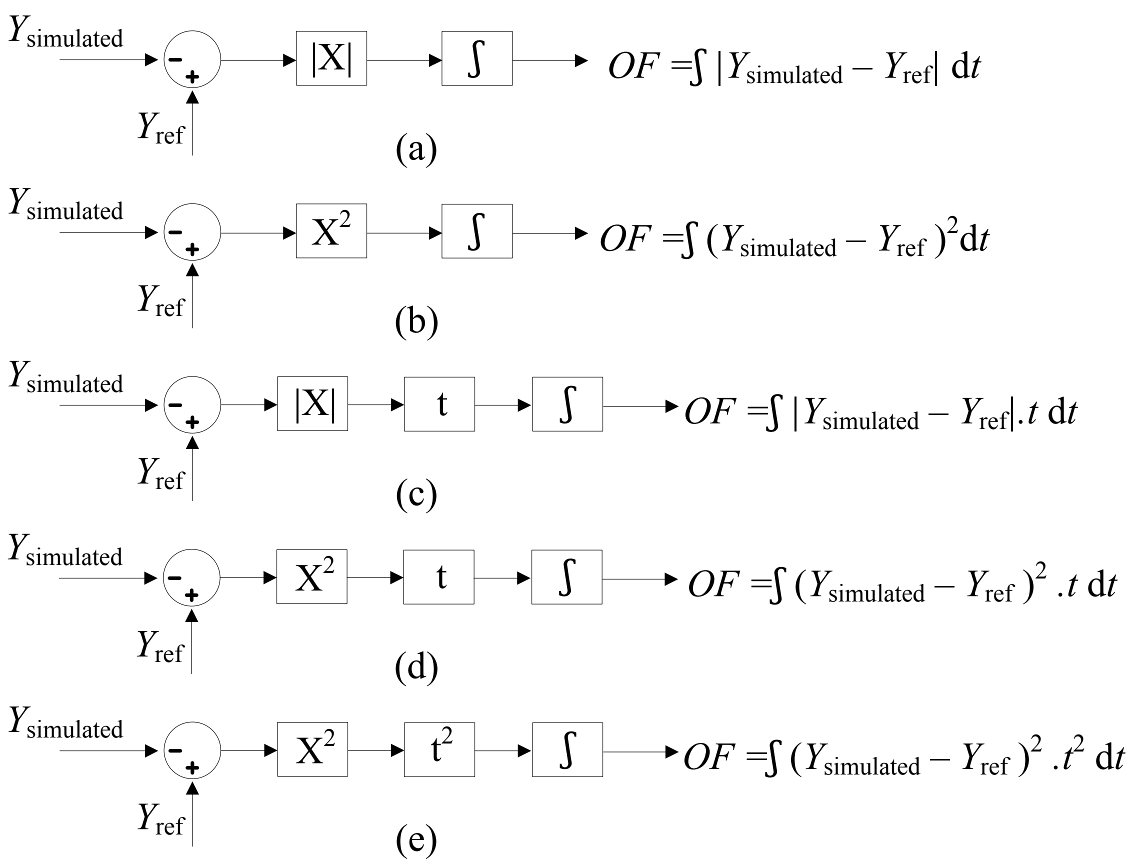

Figure 2.4: Block diagrams of alternative OF formulations: (a) integral absolute error (IAE), (b) integral square error (ISE), (c) integral time absolute error (ITAE), (d) integral time square error (ITSE), (e) integral square time error (ISTE) [13].

factors. For example, if the OF is defined as an addition of several sub-OFs as in (2.4c), the numerical value of each sub-OF should remain in the same range [5] in order to avoid an overly restrictive focus on one sub-OF.

$$of_1 = \int_0^T |Y_1 - Y_{1,\mathrm{ref}}| \, \mathrm{d}t \tag{2.4a}$$

$$of_2 = \int_0^T |Y_2 - Y_{2,\mathrm{ref}}| \, \mathrm{d}t \tag{2.4b}$$

$$OF = of_1 + of_2 \tag{2.4c}$$

If of_1 is significantly larger than of_2 or vice versa, the sub-OFs should be assigned suitable weighting factors as in (2.5) or else the optimization process will concentrate only on the sub-OF with the larger value.

$$OF = K_1 \times of_1 + K_2 \times of_2 \tag{2.5}$$

In addition to this, weightings can also be assigned to different time intervals as shown in (2.6), if the time intervals require different levels of attention [7]. Higher weightings can be applied to more important time periods while applying smaller weightings to insignificant ones.

$$OF = C_1 \int_0^{T_1} |Y_1 - Y_{1,\mathrm{ref}}| \, \mathrm{d}t \; + C_2 \int_{T_1}^{T_2} |Y_1 - Y_{1,\mathrm{ref}}| \, \mathrm{d}t \; + C_3 \int_{T_2}^{T} |Y_1 - Y_{1,\mathrm{ref}}| \, \mathrm{d}t \tag{2.6}$$

In this way, the designer can adjust the OF according to the design requirements. Since the optimization algorithm determines the suitability of candidate parameter sets considering the OF value, selecting a proper OF is an important part of the simulation-based design.

When there are multiple objectives to be satisfied, the OF should reflect all of them so that the optimization process gives compromised solution between the all objectives.

This chapter presented a review of EMT simulators and the basic operation of OE-EMTS. The structure of OE-EMTS was discussed along with the advantages of it over existing approaches. Finally, an overview of developing OFs was provided, which described fundamental steps to build an OF. The modifications that can be done to the OF were also discussed.

Chapter 3

Optimization Algorithms

3.1 Overview of Optimization Algorithms

Optimization may be defined as a process of improving an initial concept with the knowledge gained by making amendments to it [14]. Generally, in the field of engineering, optimization is often used to minimize the cost or to maximize the performance of a system [15]. Fig. 3.1 shows the basic engineering optimization procedure.

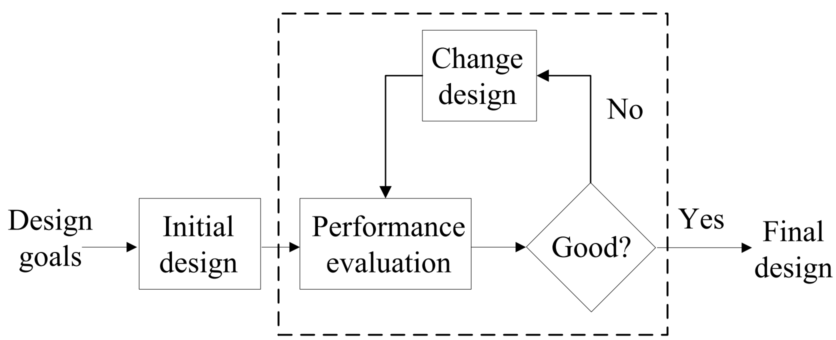

Figure 3.1: Block diagrams of the optimization process [16].

When there are several parameters to optimize, executing this process manually requires significant time and effort; therefore, optimization algorithms were introduced in order to automate the process within the dashed box shown in Fig. 3.1 [16]. The role of the optimization algorithm is generating new candidate parameters and processing of the performance evaluation function (OF) of the system [5]. Optimization algorithms run repetitively while improving the design until they reach the allowed deviation between the actual and expected design [16]. When compared with the brute-force optimization methods (e.g., trial-and-error), formal optimization algorithms have methodical and logical ways of leading the search that accelerate the design process while avoiding human error [16].

There are several ways to categorise optimization algorithms. One way is classifying them as gradient-based and gradient-free methods. Steepest descent and Gauss-Newton methods are gradient-based since they utilize derivative information in the algorithm [15]. Gradient-based methods require an explicit mathematical expression of the OF to calculate derivatives; hence there is a limitation on their use in complex optimization problems whose performance cannot be encapsulated into a mathematical function [5]. Gradient-free methods, such as Nelder-Mead's downhill Simplex method, only use OF evaluations and are thus more desirable than gradient-based methods in the context of simulation-based design [5,15].

Another way of classifying optimization algorithms is as deterministic or stochastic. Deterministic algorithms always lead the search using the same order without any randomness; hence they end up with the same final solution for the same initial point [15, 17]. The Simplex algorithm is an example of deterministic algorithms. On the other hand, stochastic algorithms, such as genetic algorithms, use randomness in their search strategy; thus they may give different final solutions or at least undergo a different path to the same final solution [15]. In this thesis, genetic algorithms and Nelder-Mead's Simplex algorithm are used for optimization and they are interfaced with PSCAD/EMTDC simulator.

3.2 Nelder-Mead's Simplex Algorithm

Nelder-Mead's Simplex algorithm is a non-gradient-based optimization method that only requires function evaluations [14]. To solve an n-dimensional optimization problem, a simplex with $n + 1$ vertices is used and value of the objective function is calculated at the point corresponding to every vertex. Then the vertex with the worst objective function value (highest value if the problem is a minimization problem or vice versa) is replaced by another point, which is found through the algorithm operators [18].

For a better understanding of the Simplex method, minimization of an n-variable function, $F(X)$ is described here. At the beginning of the algorithm, only one point (X_1) is given by the user. To calculate the other n points, (3.1) is used [14].

$$X_{i+1} = X_1 + s \cdot p_{i+1} \tag{3.1}$$

where s is a scaling factor and p_{i+1} is the unit vector for $i = 1, 2, \cdots, n$. After finding all the points (vertices), objective function values for all point are calculated. Let $X_1, X_2, \cdots, X_n, X_{n+1}$ be the $(n + 1)$ vertices of the initial simplex after arranging them in order from best performing to worst. Therefore, X_1 and X_{n+1} are the best and worst points that give the minimum and maximum objective function values, respectively. Thus, the order of the objective function values is as shown in (3.2).

$$F(X_1) < F(X_2) < \cdots < F(X_n) < F(X_{n+1}) \tag{3.2}$$

$\bar{X}$ is the centroid of all the points except the worst point (X_{n+1})(see (3.3)). The objective of the algorithm is to replace X_{n+1} with a better point, i.e., one with a better objective function value.

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.3}$$

The first operation used in the algorithm is reflection, which reflects the worst point through the centroid (see (3.4)).

$$X_{\mathrm{R}} = \bar{X} + \alpha(\bar{X} - X_{n+1}) \tag{3.4}$$

where α is a positive constant called the reflection coefficient and is generally set to 1 [16]. If $F(X_{\mathrm{R}})$ is between $F(X_1)$ and $F(X_n)$, then X_{n+1} is replaced by X_{R}. Then the algorithm moves to the next iteration and starts with a new simplex [18, 19].

If $F(X_{\mathrm{R}})$ is smaller than $F(X_1)$, it indicates a potentially very promising direction and a new point in the same direction is investigated through the expansion operation as in (3.5).

$$X_{\mathrm{E}} = \bar{X} + \beta(X_{\mathrm{R}} - \bar{X}) \tag{3.5}$$

where β is the expansion coefficient, which is generally set to 2 [16]. If $F(X_{\mathrm{E}})$ is less than $F(X_{\mathrm{R}})$, then X_{n+1} is replaced by X_{E}. If $F(X_{\mathrm{E}})$ is greater than $F(X_{\mathrm{R}})$, then X_{n+1} is replaced by X_{R} and the iteration is terminated [19].

In case $F(X_{\mathrm{R}})$ is larger than or equal to $F(X_n)$, the reflected point must not replace the worst point. Hence a contraction operation is performed between $\bar{X}$ and $\min(X_{n+1}, X_{\mathrm{R}})$. When $F(X_n) \leq F(X_{\mathrm{R}}) < F(X_{n+1})$, outside contraction is performed as in (3.6) [19].

$$X_{\mathrm{C1}} = \bar{X} + \gamma_1(\bar{X} - X_{n+1}) \tag{3.6}$$

When $F(X_{\mathrm{R}}) \geq F(X_{n+1})$, inside contraction is performed as in (3.7) [19].

$$X_{\mathrm{C2}} = \bar{X} - \gamma_2(\bar{X} - X_{n+1}) \tag{3.7}$$

where γ_1 and γ_2 are contraction coefficients and are typically set to 0.5 [16]. If $F(X_{\mathrm{C1}}) \leq F(X_{\mathrm{R}})$, X_{n+1} is replaced with X_{C1} and the algorithm moves to the next iteration. In case $F(X_{\mathrm{C2}}) < F(X_{n+1})$, X_{n+1} is replaced with X_{C2} and the algorithm moves to the next iteration.

$F(X_{\mathrm{C1}}) > F(X_{\mathrm{R}})$ or $F(X_{\mathrm{C2}}) \geq F(X_{n+1})$ means contraction operation has failed to find a better point. Therefore the algorithm performs a shrink operation by moving all the points towards the best point, i.e., (X_1). Here all the vertices except X_1 are changed as in (3.8).

$$Z_i = X_1 + \delta(X_i - X_1) \tag{3.8}$$

where $i = 2, 3, \cdots, n, n+1$, and δ is normally set to 0.5. Then $X_1, Z_2, \cdots, Z_n, Z_{n+1}$ are the new vertices (not in order) for the next iteration. The operations in the Simplex algorithm and its flowchart are shown in Figs. 3.2 and 3.3, respectively.

The convergence of the algorithm is checked by comparing the standard deviation of the sample of function values to the tolerance provided by the user. The Simplex algorithm has a number of advantages and disadvantages. The main advantages are that it is computationally efficient [8] and does not require derivative calculations [20]. However, when the number of variables increases, the iterations of the algorithm grow rapidly; it also tends to converge towards a local optimum [8].

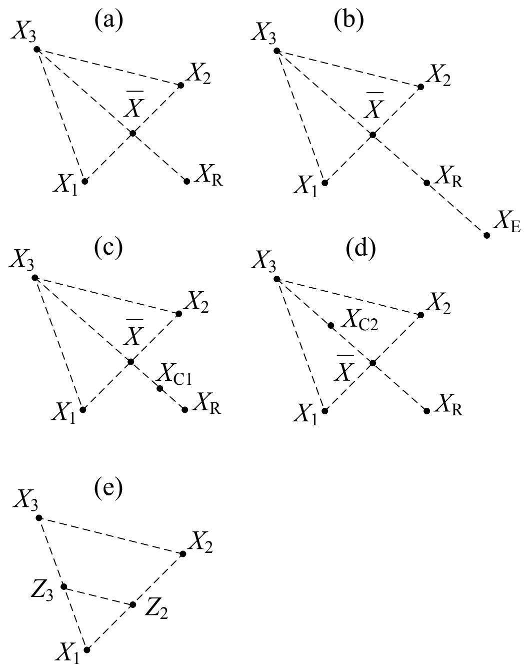

Figure 3.2: The Nelder-Mead's Simplex operations for a two-dimension problem (a) reflection, (b) expansion, (c) outside contraction, (d) inside contraction, and (e) shrink. (X_1, X_2 and X_3 are the vertices of the initial simplex) [19]

3.3 Real-Valued Genetic Algorithms

Genetic algorithms (GAs) are a member of evolutionary computing methods; they are developed based upon principles from biological evolution where the fittest individuals transfer their genes to the next generation [16]. GAs start with an initial population that includes randomly selected candidate points. The candidate points are usually called chromosomes. After evaluating the suitability of each chromosome through evaluating its corresponding OF value, the fittest chromosomes will survive and produce the next generation [8]. The chromosomes can either be expressed as a binary string or real values; thus GAs can be used to optimize both continuous and discrete variables including binary variables [5]. In this thesis, only continuous variables are optimized and hence the continuous GA (real-valued GA) is

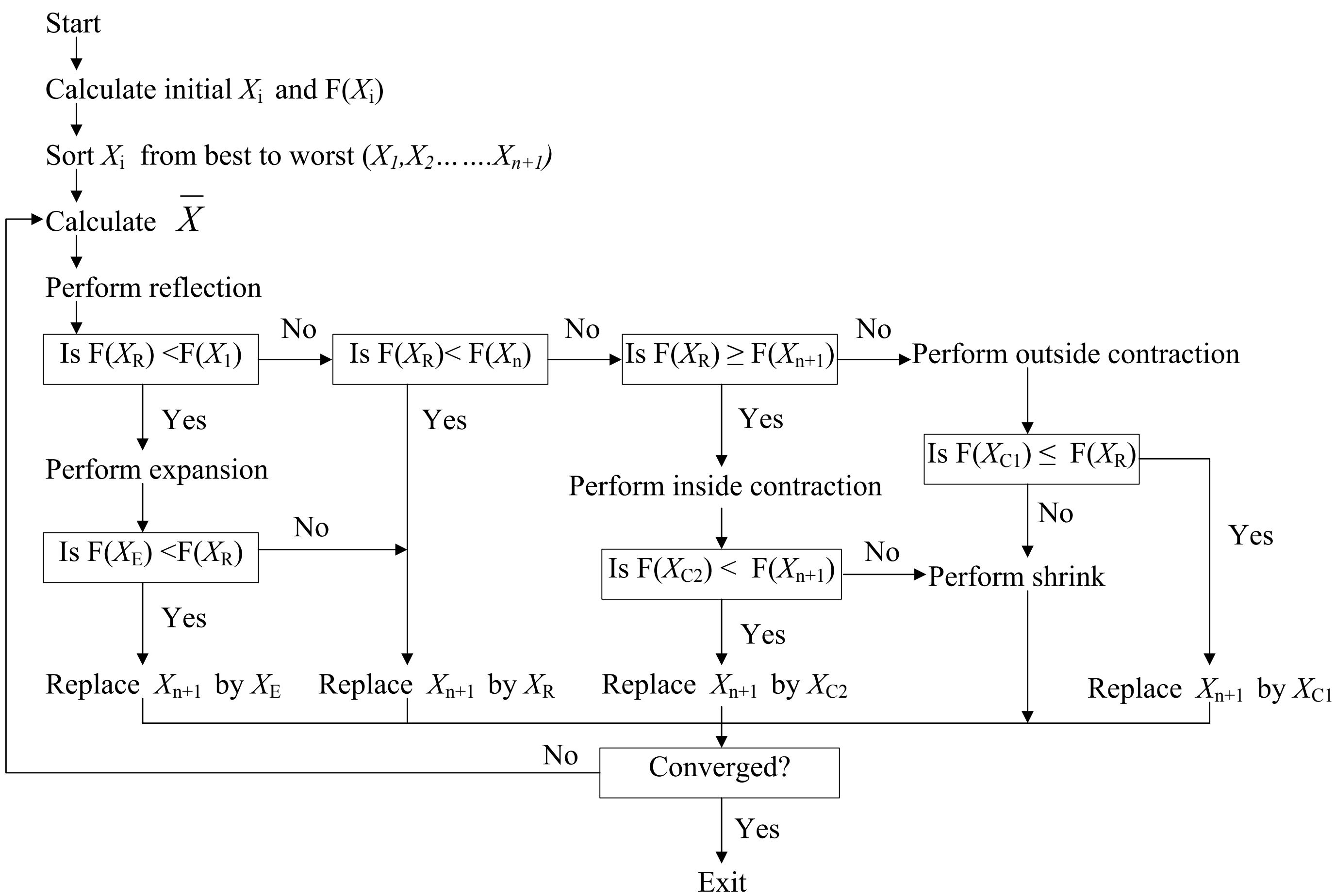

Figure 3.3: The flowchart of the Nelder-Mead's Simplex algorithm [16, 18].

used. The continuous GA algorithm, whose flowchart is shown in Fig. 3.4, is explained in this section using a minimization problem with N variables. The further details about the algorithm can be found in [14].

3.3.1 Initial Population

At the beginning of the process, the user assigns an upper limit ($X_{\mathrm{U},k}$) and a lower limit ($X_{\mathrm{L},k}$) for each variable and selects the number of candidate parameter sets (chromosomes)

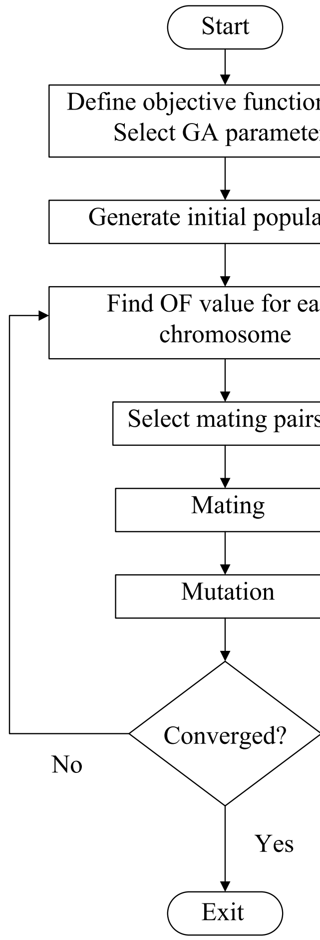

Figure 3.4: Flowchart of the continuous (real-valued) GA. [14]

in the initial (N_I), surviving (N_S), and mating (N_M) populations ($X_{\mathrm{U},k}$ and $X_{\mathrm{L},k}$ denote the upper limit and lower limit of k^{th} variable respectively). When there are N variables to optimize, one chromosome is a $(1 \times N)$ array that includes N random real values (see (3.9)) and the initial population has N_I such chromosomes (see (3.10)). One row of the initial population matrix represents one chromosome.

$$\mathrm{Chromosome}_j = [X'_{j,1} \quad X'_{j,2} \quad X'_{j,3} \quad \cdots \quad X'_{j,N}] \tag{3.9}$$

Therefore, the initial population matrix is as follows:

$$I_{\text{pop}} = \begin{bmatrix} X'_{1,1} & X'_{1,2} & X'_{1,3} & . & . & . & X'_{1,N} \\ X'_{2,1} & X'_{2,2} & X'_{2,3} & . & . & . & X'_{2,N} \\ . & . & . & . & . & . & . \\ X'_{N_{\text{I}},1} & X'_{N_{\text{I}},2} & X'_{N_{\text{I}},3} & . & . & . & X'_{N_{\text{I}},N} \end{bmatrix} \tag{3.10}$$

Generally, the numbers in this $N_{\text{I}} \times N$ initial population matrix are generated using a random number generator, which produces normalized values between 0 and 1 that must then be mapped to within their limits using (3.11).

$$X_{j,k} = (X_{\text{U},k} - X_{\text{L},k})X'_{j,k} + X_{\text{L},k} \tag{3.11}$$

where $k = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, N_{\text{I}}$; $X_{\text{U},k}$ and $X_{\text{L},k}$ denote upper limit and lower limit of k^{th} variable respectively.

3.3.2 Natural Selection

Natural selection takes place in order to discard the chromosomes with large OF values, i.e., low-performing ones in a minimization problem. Therefore, after the OF values for all chromosome are calculated, they are ranked from the lowest OF value to the highest OF value. Then the top N_{S} chromosomes (chromosomes that have the lowest OF values) are selected as the surviving population to the next generation. N_{S} can be less than or equal to N_{I} and for the remainder of the process N_{S} is kept constant.

3.3.3 Selection

Even though the top N_{S} chromosomes survive to the next generation, only the best N_{M} chromosomes from this pool are selected for mating. The rest of the chromosomes in surviving population are replaced by the offspring produced by the top N_{M} after mating. Two chromosomes from the mating pool are paired to produce two offspring and the mating process continues until (N_{S}-N_{M}) offspring are produced. Before moving to the mating process, the row numbers of the parent chromosomes to be paired should be selected from the mating pool. There are several selection methods for this purpose :

- Top-to-bottom pairing: In this method, pairing starts with the top two chromosomes and chromosomes are sequentially paired from top to bottom of the ranked list until (N_{S}-N_{M}) offspring are produced. The row numbers of the paired chromosomes are $(1, 2)$, $(3, 4)$, $(5, 6)$, etc.

- Random pairing: This approach selects the row numbers of parents randomly by generating (N_{S}-N_{M}) random numbers between 1 and N_{M}. Then the chromosomes corresponding to the rows indicated in the list are paired.

- Weighted random pairing: Here the chromosomes in the mating population are assigned probabilities that are inversely proportional to their OF values. Hence, the chromosome that has the lowest OF value has the highest probability of mating. A random number decides the chromosome that is to be a parent. There are two techniques of implementing this approach:

  Rank weighting: Since the mating pool is already ranked according to their OF values, the probability is assigned using that rank (n) as in (3.12). The rank of the top chromosome is 1 and the chromosome which has the highest OF value in the mating pool has the rank of N_{M}.

$$P_n = \frac{N_{\mathrm{M}} - n + 1}{\sum_{n=1}^{N_{\mathrm{M}}} n} \tag{3.12}$$

where $n = 1, 2, \cdots, N_{\mathrm{M}}$. Then the cumulative probability for each chromosome is calculated using (3.13), which is used to select the parent chromosomes. Cumulative probabilities always have values less than or equal 1.

$$CP_n = \sum_{i=1}^{n} P_n \tag{3.13}$$

Following this, a random number between 0 and 1 is generated and compared with the cumulative probabilities of the list from top to bottom. The first chromosome that has a cumulative probability higher than the random number is selected as a parent. Random numbers are generated until (N_{S}-N_{M}) parents are selected and then they are paired sequentially.

Cost weighting: This approach assigns probabilities to the chromosomes in the mating pool according to their OF values. First, a normalized cost is calculated as in (3.14).

$$C_n = OF_n - OF_{N_{\mathrm{M}+1}} \tag{3.14}$$

where $OF_{N_{\mathrm{M}+1}}$ is the lowest cost of the eliminated chromosomes and n is the rank of the chromosomes. C_n always has a negative value. The individual probabilities are assigned to the chromosomes as in (3.15).

$$P_n = \left| \frac{C_n}{\Sigma_{i=1}^{N_M} C_{\mathrm{i}}} \right| \tag{3.15}$$

After calculating the cumulative probability using (3.13), the same procedure as in rank weighting is followed.

- Tournament: In this method, a small subset including two or three chromosomes is selected from the mating pool by generating random row numbers from 1 to N_{M}. The chromosome that has the lowest OF value in the selected subset is chosen as a parent. This process is continued until (N_{S}-N_{M}) parents are selected. Unlike previous methods, this approach does not require a sorted population.

3.3.4 Mating

After selecting suitable pairs for mating, crossover operation is used to produce new offspring. To explain the crossover process, two parent chromosomes are considered as in (3.16).

$$\text{parent}_1 = [X_{m,1} \quad X_{m,2} \quad X_{m,3} \quad \cdots \quad X_{m,y-1} \quad X_{m,y} \quad X_{m,y+1} \quad \cdots \quad X_{m,N}] \tag{3.16a}$$

$$\text{parent}_2 = [X_{f,1} \quad X_{f,2} \quad X_{f,3} \quad \cdots \quad X_{f,y-1} \quad X_{f,y} \quad X_{f,y+1} \quad \cdots \quad X_{f,N}] \tag{3.16b}$$

The simplest form of crossover is swapping the variable values between the parents with respect to one or two randomly selected crossover points. Swapping variables does not introduce new variables to the process. It only passes the randomly generated initial variables to the next generation with different combinations. Therefore, blending methods are introduced to generate new variables in the offspring by combining crossover point variables of the parent chromosomes. Using multiple crossover points is also possible; however, single point crossover is discussed here as it is simpler. Let y be the selected crossover point, which is between 1 and N, and β be a randomly selected value between 0 and 1. Then $X_{m,y}$ and

$X_{f,y}$ are combined to obtain new variables to be in the offspring as in (3.17).

$$X_{m,\text{new}} = X_{m,y} - \beta(X_{m,y} - X_{f,y}) \tag{3.17a}$$

$$X_{f,\text{new}} = X_{f,y} + \beta(X_{m,y} - X_{f,y}) \tag{3.17b}$$

The produced offspring after crossover are as in 3.18.

$$\text{offspring}_1 = [X_{m,1} \quad X_{m,2} \quad X_{m,3} \quad \cdots \quad X_{m,y-1} \quad X_{m,\text{new}} \quad X_{f,y+1} \quad ...... \quad X_{f,N}] \tag{3.18a}$$

$$\text{offspring}_1 = [X_{f,1} \quad X_{f,2} \quad X_{f,3} \quad \cdots \quad X_{f,y-1} \quad X_{f,\text{new}} \quad X_{m,y+1} \quad ...... \quad X_{m,N}] \tag{3.18b}$$

A total of (N_S-N_M) offspring are produced during the mating process and now the next generation consists of N_M chromosomes.

3.3.5 Mutation

To prevent premature convergence, another operator called mutation is adopted, which randomly changes randomly selected parameters [14]. A mutation rate, which determines the number of variables that are mutated, should be given by the user. If the mutation rate is m_r, the number of mutations is found as in (3.19) [14].

$$\text{Number of mutations} = m_\text{r} \times N_\text{S} \times N \tag{3.19}$$

where $N_\text{S} \times N$ gives the total number of variables in the surviving population. After finding the total number of mutations, random numbers between 1 to N_S are generated to find the row numbers while random numbers between 1 to N are generated to find column numbers.

The variable values that are in the selected locations are replaced by a random number. This happens until the total number of mutations are achieved.

In case if the user wants to keep the best solutions of the generation unchanged, they should not be affected by the mutation. For that, there is another population introduced in the algorithm called the elite population (N_{E}). This includes the top N_{E} chromosomes of the surviving population. When there is an elite population, row numbers for the mutation are selected from $N_{\mathrm{E}} + 1$ to N_{S} and the number of mutations are also changed as in (3.20).

$$\text{Number of mutations} = m_{\mathrm{r}} \times (N_{\mathrm{S}} - N_{\mathrm{E}}) \times N \tag{3.20}$$

After mutation is over, OF values of the chromosomes of the new generation are evaluated. This process repeats until the desirable result is achieved.

In this thesis, GA is used since it does not require derivative calculations, works well with large numbers of variables, is well-suited for parallel programming, and does not depend on a single initial point. In some cases, other optimization algorithms may find the solution faster than a GA as GAs have low convergence efficiency. However, a GA is suitable for solving many real-world optimization problems with a large number of variables.

This chapter provided a brief introduction to a number of derivative-free optimization algorithms. The principles and operations used in the Simplex algorithm and GA were described in detail together with the advantages and disadvantages of these algorithms.

Chapter 4

Novel Approaches to Reduce Simulation Time and Computational Burden

This chapter presents the novel methods explored in this research to improve the existing methods of using OE-EMTS for optimization of complex power systems. Screening methods, which identify and remove non-influential variables, are studied to reduce the dimension of the design problem; hybridized optimization algorithms and parallel processing techniques are also explored to enhance the computational efficiency of the design cycle by reducing unnecessary, time-consuming EMT simulations.

4.1 Screening Methods

When it comes to converter-intensive power systems, there are often many parameters that need to be assigned optimally. Not all of these parameters crucially affect the design objectives. Therefore, two screening methods are developed in this thesis to identify such parameters and they are excluded from the optimization process. This will facilitate the design process as reduction of optimizing parameters reduces the complexity of the design problem and hence the computational burden of its solution.

4.1.1 Initial Screening

Initial screening is done before the main loop of optimization. In this screening method, the initial value of each variable is changed by applying positive and negative increments and for each increment a simulation run is conducted to evaluate whether the increment has a significant impact on the OF value. The designer can decide the value of increment(s) according to the design problem.

The process of perturbing the original optimization variables entails calculation of the OF for each perturbed variable. The relative variations of the OF values are analyzed to separate the influential and non-influential variables. Variables that do not significantly affect the OF are excluded from the optimization process and they are assigned their original values.

As the number of variables is now reduced, smaller populations may be used if the optimization is done using a population-based optimization algorithm such as GA. For algorithms like Simplex, the number of iterations likely reduces with a smaller number of parameters. Therefore, this screening will definitely reduce the simulation time and computational burden. While this method proves successful in many cases, its effectiveness depends on the initial values of the optimization variables. For a highly nonlinear system, if the initial multi-dimensional point is far from the optimum, this method may discard variables that may indeed be influential. Therefore, this method for initial screening must be used with limited liberty. Selection of the initial values for the parameters to be optimized is also a crucial task. The general expectation from the initial values is to produce a response that is stable even though it may feature poor dynamic performance. Improvement of the response is left to the simulation-based optimization.

The initial parameter values for the cases in this thesis are selected using a few rounds of trial-and-error while utilizing basic insight about controller gains, e.g., that higher proportional gains generally tend to accelerate the response, but may lead to instability at large enough values, and that smaller integral time-constant values may settle the response faster, but may cause oscillations.

4.1.2 Run-Time Screening

The second screening method can only be used in population-based optimization algorithms such as GA, which run for several successive generations. The first run of the new generation gives the best parameter set found by the algorithm until that point. It should be noted that since there is an elite population in the algorithm, the best solution set is not affected by the mutation process. These parameter values may contain information about the variables that require further optimization (influential variables). Thus in the proposed run-time screening method, the best solution sets found in the first two or three generations are analyzed and conclusions are drawn accordingly. If the value of a parameter does not change noticeably in the first few generations, it can be argued that the parameter has already converged to its optimal interval and does not need further optimization. For example, if the values of a parameter for the first three generations are 3.21, 3.43 and 2.98, then the variation of the maximum value from the minimum value is around 15%, which may be considered as small. Hence there is a high possibility that they are in the optimal region. On the contrary, if a parameter has values of 1.56, 3.41, and 0.57 in the first three generations, the maximum value deviates 498% from the minimum value, which does not indicate convergence. The parameters that vary considerably must be optimized further until they converge into a small interval.

In this approach, the number of generations that GA should be launched for screening is changed depending on the design problem. If the case gives an acceptable solution after 8-10 generations of GA, then analysis of 2-3 generations' data would be sufficient for run-time screening. For the cases that give final optimal results after a significant number of generations of GA, this screening process may require data from more generations. Thus it should be decided according to the design problem. Alternatively, this process can be automated by updating the GA code to check the OF value of the best parameter set after every generation. If the OF value of the parameter set is less than a certain value selected by the designer, then the best solution sets up to that generation can be used for run-time screening process.

Moreover, these results may reveal further insight about the range of the parameters values. If the designer has assigned large search intervals to the variables, they can be reduced so that smaller populations can be used, which leads to more computational benefits. After deciding the influential parameters, the optimization process starts again with a smaller number of parameters with new boundaries and smaller populations. The number of generations that the algorithm runs can also be reduced due to the reduced dimension of the optimization problem. This will significantly reduce the simulation time and computational burden.

4.2 Inclusion of Parallel-Processing Techniques

In this thesis, a GA is used since it performs well for complex optimization problems. However, due to its slow convergence, it consumes a lot of time to find the optimal solution. GAs can readily benefit from parallel-processing techniques [21, 22]. Due to the independence of the iterations in the GA on one another, it is possible to adopt parallelism in the algorithm.

Several research contributions have been made in this area to improve the efficiency of GAs. In [22] improvements are achieved by simultaneously performing genetic operations for two populations called searching population and elite population. The best solutions found in the searching population are given to the elite population and the worst solutions generated in the elite population are given to the searching population. These two populations evolve in parallel, thus convergence happens faster. [23] has used sub-population-type paralleled GA, which evaluates several sub-populations in parallel instead of one population while sharing information among them at prescribed time intervals. In all of these methods, parallelism is added to the GA using separate sub-populations that run in parallel while sharing information; this is somewhat complex and the examples used in previous work have explicit formulations for OFs, which is different from the approach used in this thesis. In [24], the OF is calculated through the simulation for reactive power optimization; however, the adopted parallel GA method is almost the same as previously described ones, which use sub-populations. In the simulation-based optimization approach discussed in this thesis, calculation of the OF value causes the highest computational burden and far exceeds those of basic GA operations of selection, crossover, and mutation. Therefore, this thesis adopts a specific parallel GA implementation, which focuses on parallelizing the iterations (i.e., OF calculations) in a single large population instead of among small sub-populations.

As described in Chapter 3, Section 3.3, the sequential GA (normal GA) starts with the user-defined values for the parameter boundaries and the number of chromosomes in the initial, surviving, and mating populations. The algorithm then generates random number sets for the initial population considering parameter boundaries. In a sequential GA, the algorithm releases only one chromosome at a time and the EMT simulator runs sequentially with different parameter values assigned to it in each run and gives the respective OF value back to the optimization algorithm. After evaluating the first generation, the chromosomes that have the lowest OF values will be selected for the next generation as the surviving population. The best solution sets from that surviving population are chosen as the mating pool to generate new offspring. This is done by using the crossover operator where the two parent chromosomes exchange their parameter values with respect to one or more randomly selected crossover points. The remaining surviving population after selecting the mating pool is replaced with the offspring. To prevent premature convergence, mutation is adopted [14], which randomly changes randomly selected parameters. The new generation will then be evaluated using EMT simulations. This continues for several generations until the algorithm converges into an optimal solution. The genaral procedure followed by the proposed parallel GA is the same as in sequential GA. The major difference between the two methods is that the OF values of chromosomes in sequential GA are evaluated one at a time in a sequential manner, but the parallel GA evaluates OFs of several chromosomes simultaneously; hence it lowers the simulation time considerably.

To implement the parallel GA, PSCAD/EMTDC's inbuilt concept of a simulation set is used, which makes the basic pathway for parallel computing in the simulator. Within a simulation set, there can be several simulation cases. All the simulation cases included in the simulation set are launched in parallel using all processor cores available [11]. Therefore, in the proposed parallel GA, parallel processing happens within the PSCAD/EMTDC simulation set where it is externally controlled by the GA, coded in a Python script. The communication between the Python script and the PSCAD cases is maintained by the PSCAD Automation Library as shown in Fig. 4.1. All the examples of parallel processing in the the-

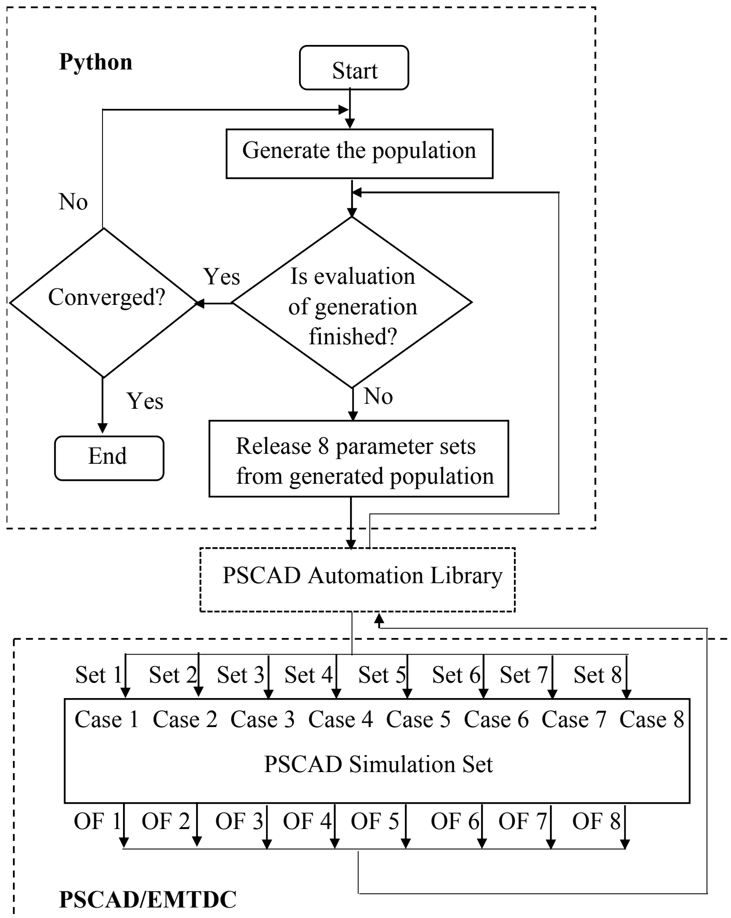

Figure 4.1: Schematic diagram of the parallel genetic algorithm.

sis are done using eight simulation cases within one simulation set. Those eight simulation cases are copies of the same file; however, the parameter values in them are assigned independently by the parallel GA. After generating the first generation, the Python script sends eight chromosomes at a time to PSCAD through the Automation Library. Those eight chromosomes are assigned to the eight cases in the simulation set. Unlike in the sequential GA, eight parameter sets are evaluated concurrently in the parallel GA. After the EMT solver runs all eight cases in parallel, the OF values for every case are sent back to the Python GA script. Only the evaluation of the OF value is done in PSCAD. After determining OF values corresponding to all the chromosomes in the population, processing the OF values and production of new generation are done in Python. Then this process happens repetitively until the required number of generations are completed.

This novel method is more suitable for simulation-based optimization and can be easily implemented. It speeds up the optimization process significantly. The efficiency and accuracy of the proposed method is confirmed using several examples in the upcoming chapters.

4.3 Hybrid GA-Simplex Optimization Algorithm

The main advantage of the GA is that it is more likely to converge into global optimum, since it does not focus on one point; rather several points scattered in the regions specified by the designer are explored. However, because of this the algorithm is computationally inefficient and has a low convergence rate. To overcome these issues, which become particularly pronounced in simulation-based optimization, this thesis proposes a novel method by combining the GA with the nonlinear Simplex algorithm, which has a powerful local search ability [20].

By using proper values for the population numbers in GA, an acceptable solution (not the best one) can be obtained after a small number of populations (e.g., within the first one or two), which reflects the global optimum area. Thus in the proposed hybrid algorithm, the GA solver is run first to identify the area wherein the global optimal exists, after which the search will continue in that area with the Simplex algorithm that has much better convergence properties.

The quality of the Simplex results depends on the initial parameter values assigned to it, which come from the GA. The possibility of having a good initial point for Simplex algorithm increases with the number of generations that the GA is calculated. However, if the GA is launched for more generations, simulations consume more time; hence there should be a compromise between the time and quality of the output, which should be decided by the designer according to the design problem. The exemplar cases shown in the next chapters demonstrate that hybridization leads to significant reduction in computation time and complexity.

Chapter 5

Application of Proposed Methods in Design of Power Systems

This chapter presents how the proposed methods are effectively used in the optimization of complex power systems with power electronic converters. The results obtained from each method are compared and the efficiency of the proposed methods is confirmed at the end.

5.1 Case Study 1

5.1.1 System and Controller Configuration

The first example is a 2 MW type-4 wind turbine generator connected to the grid as shown in Fig. 5.1. The control system for the case is shown in 5.2 and 5.3. Even though this system does not have a complicated dynamic behavior, it is selected here as the first example in order to explain the proposed methods of the thesis. Reactive power exchange in the wind power plant is maintained at zero and active power is changed as in (5.1).

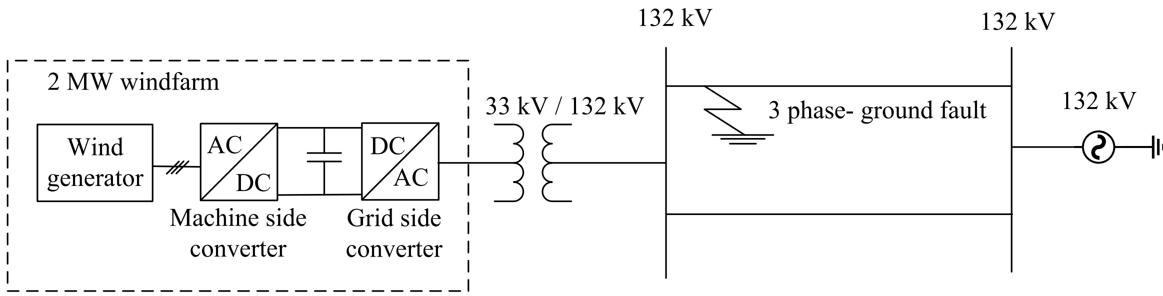

Figure 5.1: Schematic diagram of the system in case study 1.

$$P = \begin{cases} 0.25\,\text{pu} & t \leq 15\,\text{s} \\ 0.4\,\text{pu} & t > 15\,\text{s} \end{cases} \tag{5.1}$$

A three-phase-to-ground fault is applied at $t = 5.5$ s and cleared after 0.2 s. In this

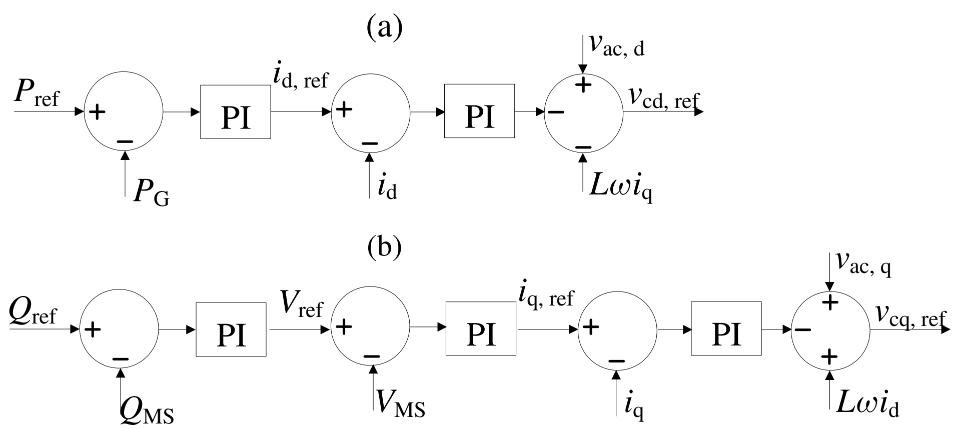

Figure 5.2: Block diagram of the machine-side controller (a) active power controller, (b) reactive power controller.

example all the PI control parameters, including the inner control loops, are optimized. The controllers and their parameters are shown in Table 5.1. The initial OF value of the system is 0.721. The objective of the controller is to control the active and reactive power output of the wind power plant properly. Therefore, the addition of integral square errors of active

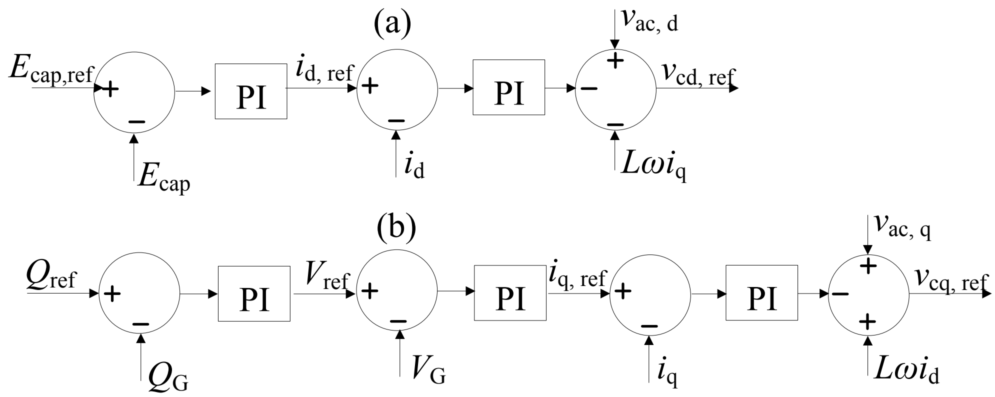

Figure 5.3: Block diagram of the grid-side controller (a) capacitor voltage controller, (b) reactive power controller.

and reactive power curves is used as the OF as in (5.2).

$$OF = \int_0^T ((P - P_{\text{ref}})^2 + (Q - Q_{\text{ref}})^2)\ \mathrm{d}t \tag{5.2}$$

where $[0, T]$ is the simulation time period.

5.1.2 Screening of the Optimization Variables

In this example the initial screening method is used; thus it is explained before presenting the optimization results. There are 18 parameters to optimize in this example. With the expectation of reducing the number of optimizing variables, screening method I (initial screening) is used with ±3% and ±10% changes applied to the initial parameter values in separate runs. After observing the obtained OF values in each run, which are shown in Fig. 5.4, parameters $K_{\text{pq\_M}}$, $T_{\text{iq\_M}}$, $T_{\text{id\_M}}$, $K_{\text{p\_AC}}$, $T_{\text{iq\_G}}$ are identified as non-influential due to their small impact on the OF. Parameter numbers in Fig. 5.4 correspond to those in Table 5.1. It should be noted that this depends on the selected initial values for the parameters. Different

Table 5.1: Controller Parameters for Optimization in Case Study 1

Machine side converter			Grid side converter		
Controller	Number	Parameter	Controller	Number	Parameter
Active power	1	$K_{\mathrm{p\_P\_M}}$	DC voltage	9	$K_{\mathrm{p\_Edc}}$
	2	$T_{\mathrm{i\_P\_M}}$		10	$T_{\mathrm{i\_Edc}}$
I_{q} current	3	$K_{\mathrm{pq\_M}}$	Reactive power	11	K_{pQ}
	4	$T_{\mathrm{iq\_M}}$		12	T_{iQ}
I_{d} current	5	$K_{\mathrm{pd\_M}}$	AC voltage	13	$K_{\mathrm{p\_Vac}}$
	6	$T_{\mathrm{id\_M}}$		14	$T_{\mathrm{i\_Vac}}$
AC voltage	7	$K_{\mathrm{p\_AC}}$	I_{d} current	15	$K_{\mathrm{pd\_G}}$
	8	$T_{\mathrm{i\_AC}}$		16	$T_{\mathrm{id\_G}}$
			I_{q} current	17	$K_{\mathrm{pq\_G}}$
				18	$T_{\mathrm{iq\_G}}$

initial parameter values may lead to different influential and non-influential parameters.

5.1.3 Optimization of Parameters

Even though initial screening identified the influential parameters, to demonstrate the efficiency of the other methods proposed and for comparison purposes, all 18 parameters are optimized in the first three sub-sections of this section and optimization of the influential parameters is shown in the last sub-section.

5.1.3.1 Optimization Using Sequential GA

The sequential GA is launched for six generations using initial and surviving populations of 240 and 160, respectively. The parameter limits used in the GA and their optimized values are shown in the 2$^{\mathrm{nd}}$ and 4$^{\mathrm{th}}$ columns of Table 5.2, respectively. The insight gained during

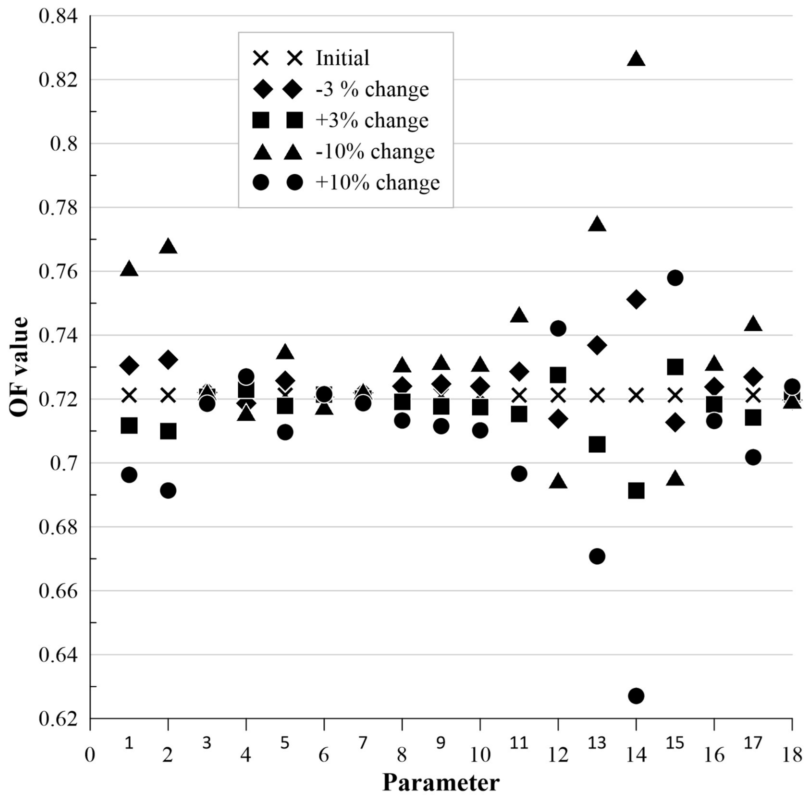

Figure 5.4: Distribution of OF values with parameter perturbations in case study 1.

the selection of initial values using trial-and-error also informs the designer of suitable, albeit approximate, parameter ranges. Such insight is used in selecting the ranges for the examples in this thesis. In general, assigning larger limits does not affect the final solution since the GA is a global optimization algorithm; however, larger limits often require larger initial and surviving populations as the algorithm has to search a larger space. Conversely, narrow limits may adversely impact the solution by excluding the optimal area. Active power variations before and after optimization are shown in Fig. 5.5(a) and Fig. 5.5(b), respectively. The design takes 25.47 h to complete, which shows the need for improved methods.

Table 5.2: Initial and Optimized Values for Case Study 1

Parameter	Initial		Sequential GA	Hybrid GA-Simplex		Parallel GA	Parallel GA with screening
	Limits	Values		GA	Simplex		
$K_{\text{p\_P\_M}}$	(0,5)	1	4.99	4.98	5.042	3.73	4.83
$T_{\text{i\_P\_M}}$	(0,1)	0.01	0.073	0.25	0.0097	0.081	0.011
$K_{\text{pq\_M}}$	(0,5)	1	4.83	4.83	4.92	3.95	1
$T_{\text{iq\_M}}$	(0,1)	0.01	0.85	0.747	0.914	0.33	0.01
$K_{\text{pd\_M}}$	(0,5)	1	1.836	1.836	2.01	3.66	2.15
$T_{\text{id\_M}}$	(0,1)	0.01	0.48	0.48	0.66	0.065	0.01
$K_{\text{p\_AC}}$	(0,5)	1	0.368	0.368	0.52	0.837	1
$T_{\text{i\_AC}}$	(0,1)	0.01	0.0206	0.081	0.21	1.14	0.167
$K_{\text{p\_Edc}}$	(0,5)	0.5	3.075	1.735	2.01	4.03	3.395
$T_{\text{i\_Edc}}$	(0,1)	0.01	0.342	0.34	0.53	0.86	0.318
K_{pQ}	(0,5)	0.5	1.089	1.85	2.24	4.29	4.78
T_{iQ}	(0,1)	0.01	0.151	0.13	0.021	0.041	0.016
$K_{\text{p\_Vac}}$	(0,5)	0.5	4.5	4.50	4.49	4.75	1.068
$T_{\text{i\_Vac}}$	(0,1)	0.01	0.796	0.95	1.11	0.935	0.681
$K_{\text{pd\_G}}$	(0,5)	0.5	2.227	2.23	2.54	2.502	1.285
$T_{\text{id\_G}}$	(0,1)	0.05	0.067	0.66	0.74	0.152	0.337
$K_{\text{pq\_G}}$	(0,5)	0.5	0.0805	0.081	0.233	0.677	3.056
$T_{\text{iq\_G}}$	(0,1)	0.05	0.664	0.97	1.321	0.247	0.05
OF value		0.721	0.011	0.0124	0.0085	0.0091	0.0089

5.1.3.2 Optimization Using Hybrid GA-Simplex Algorithm

In this part, the sequential GA is launched for two generations and then the optimization is continued using Simplex with the best solution given by the GA. The optimization results after the GA and after the Simplex algorithm are shown in the 5$^{\text{th}}$ and 6$^{\text{th}}$ columns of Table 5.2 . Fig. 5.5(c) and Fig. 5.5(d) show the dynamics of the active power in the system for optimized values obtained from intermediate GA and simplex algorithm respectively. The

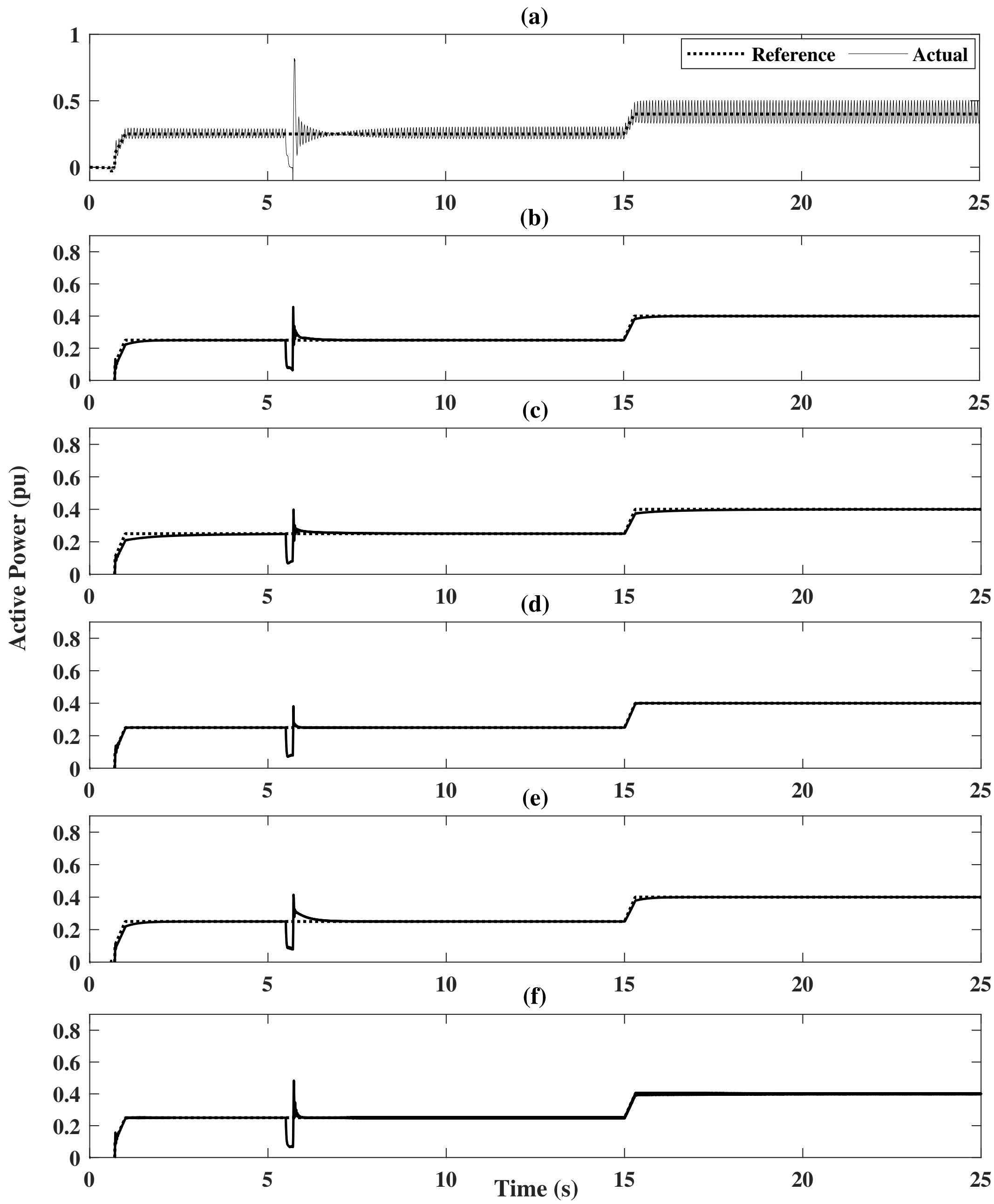

Figure 5.5: Active power output with (a) initial values, (b) sequential GA optimized values, (c) intermediate GA values, (d) final Simplex optimized values, (e) parallel GA optimized values (f) parallel GA optimization of 13 influential parameters.

time taken by this approach is 11.08 h, which is considerably lower than before.

5.1.3.3 Optimization Using Parallel GA

In this case, eight parameter sets are evaluated simultaneously. Optimal values obtained are shown in the 7^{th} column of Table 5.2 and Fig. 5.5(e) shows the dynamics of the active power output. The simulation time is markedly reduced to 9.8 h using this method.

5.1.3.4 Optimization of Influential Parameters

In this part, only the parameters identified as influential are optimized using the parallel GA. The parameters excluded from the optimization are assigned their initial values. Since the dimension of the problem is reduced from 18 to 13, initial and surviving populations are reduced to 160 and 120, respectively, and the parallel GA is launched for six generations. Optimized results are shown in the 8^{th} column of Table 5.2 , and the optimal active power output is shown in Fig. 5.5 (f). This design takes merely 7.4 h, which is a significant reduction of time.

Time and population details comparisons for this case are shown in Table 5.3 and Table 5.4 respectively. In this case, both the hybrid algorithm and parallel GA give significant improvements. Time taken by the hybrid algorithm can be further reduced by using the parallel GA. The results show that the time taken by sequential GA is be reduced by nearly 16 h using the methods proposed in this thesis. Furthermore, this example demonstrates that the initial screening method is effective in reducing the simulation time without adversely affecting the quality of the final optimal design.

Table 5.3: Time Comparisons for Case Study 1

Method	Time (h)
Sequential GA	25.47 h
Hybrid GA-Simplex	11.08 h
Parallel GA	9.8 h
Parallel GA + screening	7.4 h

Table 5.4: Comparison of Population Details and Number of Simulation Runs for Case Study 1

	Sequential GA	Hybrid GA-Simplex		Parallel GA	Parallel GA with screening
		GA	Simplex		
Initial population	240	240	-	240	160
Surviving population	160	160	-	160	120
Generations	6	2	-	6	6
Simulation runs	1041	401	115	1048	768

5.2 Case Study 2

5.2.1 System and Controller Configuration

The second test system is a 125 MW (5 MW×25) type-4 wind power generation plant connected to the grid as shown in Fig. 5.6. Even though the wind farm controllers of this case are the same as in case study 1, the system is complicated and it has complex dynamic behavior, which is hard to optimize manually since the system is weak with low short circuit ratio (SCR). Therefore, it is selected as the second example to convey the efficacy of the proposed methods. During normal operation, the short-circuit MVA (SCMVA) at the point of interconnection (POI) is 165 MVA indicating a weak system. A three-phase-to-ground fault is applied at $t = 5$ s and is cleared after 0.2 s by disconnecting the faulted line, which drops the SCMVA to 78 MVA, thus making the system even weaker and unstable for the initial controller parameter values shown in the 2nd column of Table 5.5. Therefore, the

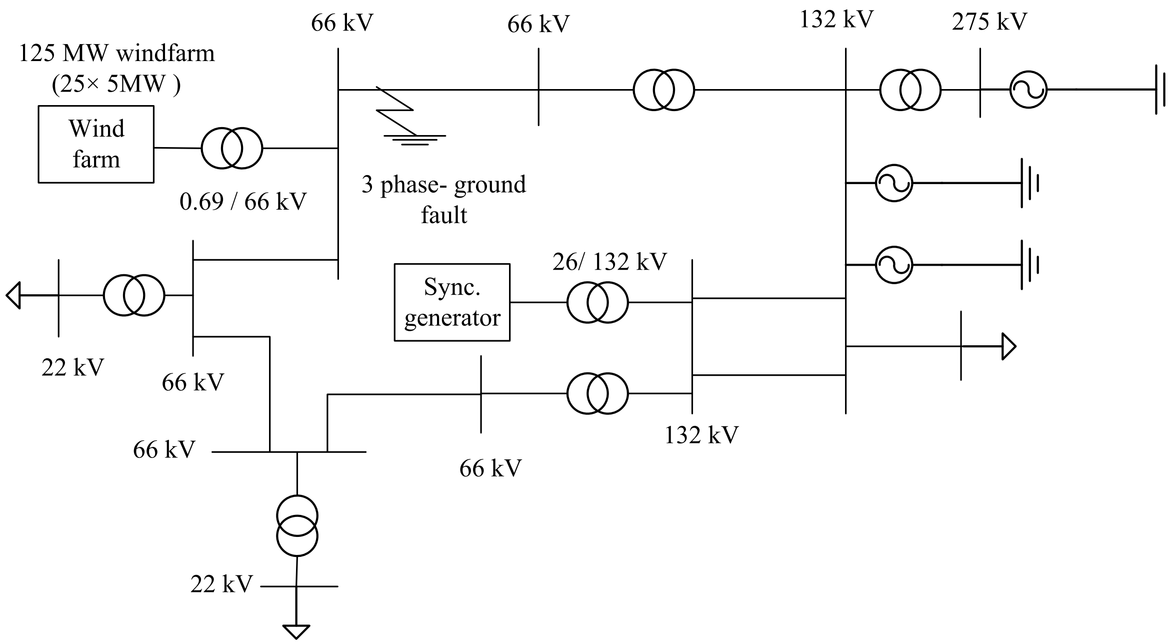

Figure 5.6: Schematic diagram of the system for case study 2.

objective is to optimize the wind power plant controller parameters to obtain stable operation before and after the fault.

The short circuit ratio (SCR) at point of interconnection (POI) reduces further after the fault, thus the voltage at POI becomes more undesirable. It is determined that satisfactory performance is obtained if the gains and time-constants of proportional-integral (PI) controllers are tuned to maintain 5 MW output from the wind farm and to avoid overvoltages that are greater than selected overvoltage value at POI even after the fault. Therefore, an objective function is formed by adding the integral square error (ISE) of the active power and the integral of overvoltage at the POI (see (5.3)); minimization of this objective function

Table 5.5: Initial and Optimized Values for Case study 2

Parameter	Initial		Sequential GA	Hybrid GA-Simplex		Parallel GA	Parallel GA with screening	
	Values	Limits		GA	Simplex		New limits	Results
$K_{\text{p\_Edc}}$	4	(0,7)	6.39	5.434	5.749	6.505		6.935
$T_{\text{i\_Edc}}$	0.02	(0,2)	1.72	1.975	2.049	1.388	(0,2)	0.874
$K_{\text{p\_Q}}$	1	(0,7)	0.346	2.631	2.734	5.97	(0, 6.5)	1.342
$T_{\text{i\_Q}}$	0.2	(0,2)	1.314	0.849	0.922	0.534	(0,2)	0.559
$K_{\text{p\_Vac}}$	4	(0,7)	1.48	1.527	1.584	1.203	(0,2.5)	2.087
$T_{\text{i\_Vac}}$	0.05	(0,2)	0.977	0.0714	0.0818	1.015		1.906
$K_{\text{p\_P}}$	2	(0,7)	0.772	0.535	0.625	0.237	(0,2.5)	0.531
$T_{\text{i\_P}}$	0.05	(0,2)	0.133	0.906	0.926	0.046	(0,1)	0.081
OF value	211.654		16.407	25.58	22.49	16.87		17.023

yields optimal parameter values for the controllers.

$$OF = K(t) \int_{t_0}^{T} (P - P_{\text{ref}})^2 \, dt \ + \int_{t_1}^{t_2} |V_{\text{over}} - V_{\text{ref}}| \, dt \tag{5.3}$$

where

$$K(t) = \begin{cases} k_1 & t_0 < t \leq T_1 \\ k_2 & T_1 < t \leq T \end{cases} \tag{5.4}$$

In (5.3) and (5.4), $[t_0, T]$, $[t_0, T_1]$ and $[t_1, t_2]$ denote the entire OF evaluation time period, the time period when transients occur, and the time period when overvoltages occur, respectively. It should be noted that this OF calculates the error during both transient and steady state conditions; thus the algorithm returns parameters that give improved transient and steady state response. In this thesis, $k_1 = k_2 = 1$ is used, which places a balanced focus on both transient and steady state intervals. If $k_1 > k_2$, the OF places a heavier penalty on the deviations during the transient period; therefore, if the designer wants to place more emphasis on the transient period of the response, a larger weighting factor may be assigned to the time period when transients occur.

In practice the inner loop controllers are expected to act rapidly, thus leaving the most significant dynamics to the external loop parameters. Hence in this example, the parameters of the capacitor voltage controller ($K_{\mathrm{p\_Edc}}$, $T_{\mathrm{i\_Edc}}$), grid-side reactive power controller ($K_{\mathrm{p\_Q}}$, $T_{\mathrm{i\_Q}}$), grid-side rms voltage controller ($K_{\mathrm{p\_Vac}}$, $T_{\mathrm{i\_Vac}}$), and active power controller ($K_{\mathrm{p\_P}}$, $T_{\mathrm{i\_P}}$) are considered for optimization.

5.2.2 Optimization of Parameters

First, the system parameters are optimized with a sequential GA and the application of novel methods to the optimization is described later. From the suggested screening methods, run-time screening is used in this example. Thus it is explained in the final part of this section. The results from each method are compared to demonstrate their efficacy.

5.2.2.1 Optimization Using a Sequential GA

Before moving to the improved methods introduced in this thesis, the case is optimized with a sequential GA. With the initial and surviving populations of 104 and 48, respectively, the GA solver is launched for 15 generations, with results shown in the 4$^{\text{th}}$ column of Table 5.5. The parameter limits used in this case are shown in the 3$^{\text{rd}}$ column of the same table. Parameter limits in this example are selected in the same manner as in the previous example. Fig. 5.7(a) and Fig. 5.7(b) show the initial and optimal rms voltage waveforms, respectively.

Even though the optimized controllers produce markedly better results, the time taken

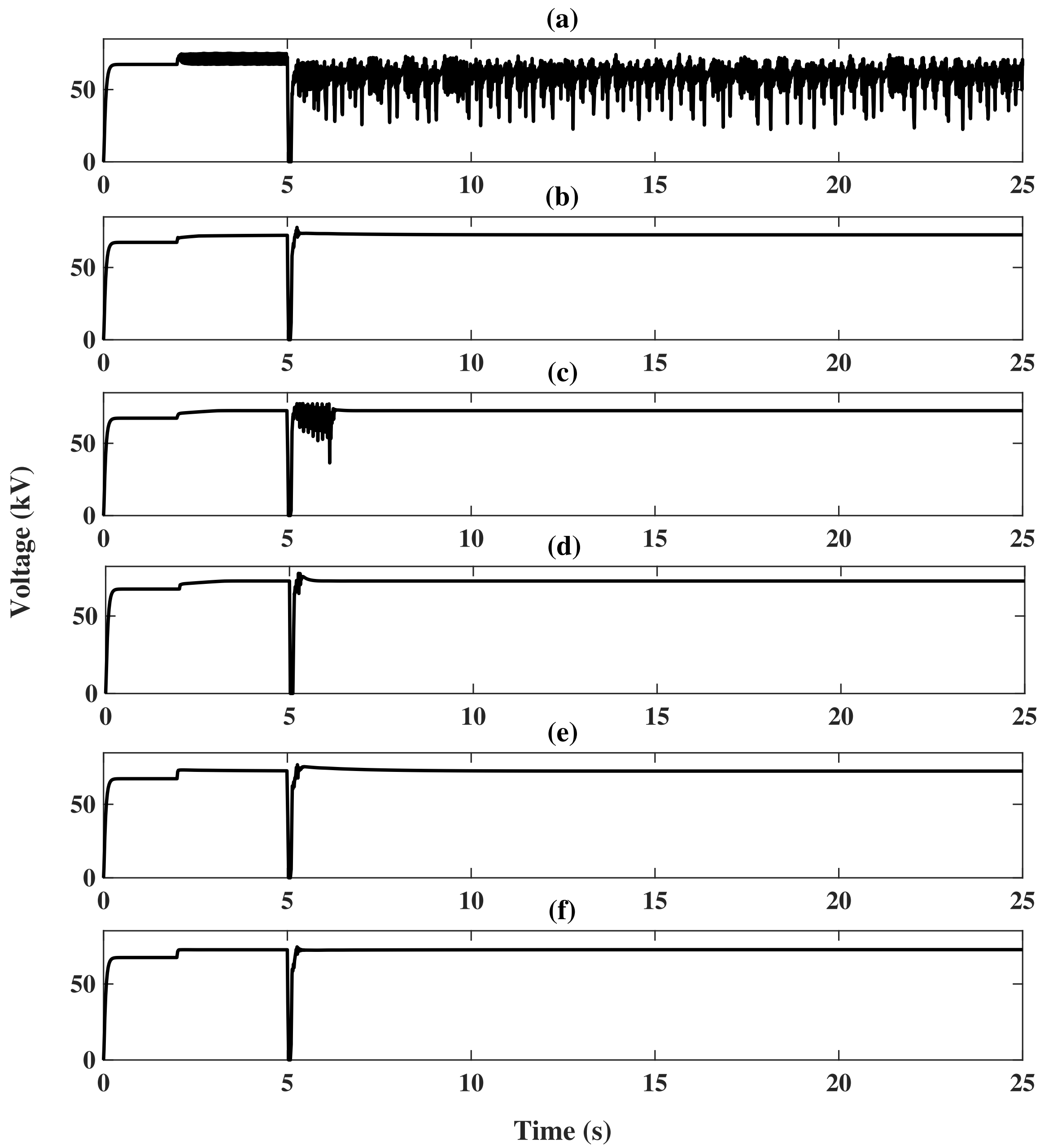

Figure 5.7: POI voltage for (a) initial values, (b) sequential GA optimized values, (c) intermediate GA values, (d) final Simplex optimized values, (e) parallel GA optimized values (f) optimized values after run-time screening.

by the algorithm to complete the task is 43.48 h, which is significant. To reduce this time, the proposed hybrid algorithm and parallel GA are used as described in the next sub-sections. It should be noted that the steady state voltage at POI after the fault is higher than 66 kV due to insufficient reactive power compensation in the design, which is not included in optimization.

5.2.2.2 Optimization Using Hybrid GA-Simplex Algorithm

Here, the same example is optimized with the new Hybrid GA-Simplex algorithm described in Chapter 4, section 4.3. First the sequential GA is run for two generations with the same populations and parameter boundaries as before. The best values obtained after the second generation are used as the initial values for the Simplex algorithm. The results obtained using this method are shown in the 5^{th} and 6^{th} columns of Table 5.5. The waveforms of rms voltage at POI for intermediate GA values and Simplex optimized values are shown in Fig. 5.7(c) and Fig. 5.7(d) respectively.

Although the solution found by this method is slightly less optimal than the one found after 15 generations of sequential GA, it is still an acceptable solution, which gives better OF value than initial values in 21.31 h, which is almost half of the time consumed by the sequential GA.

5.2.2.3 Optimization Using Parallel GA

In this solution, eight copies of the same case are run in parallel in a PSCAD/EMTDC simulation set, with different parameter values assigned to them using the GA coded in a Python script. When the example case is optimized using parallel GA with the same population values and parameter limits, the simulation takes only 13.11 h which is almost

3.3 times faster than the sequential GA and gives a desirable output. Optimization results are shown in 7$^{\text{th}}$ column of Table 5.5 and Fig. 5.7(e) illustrates the waveforms of the rms voltage at POI.

5.2.3 Screening of Optimization Variables

The run-time screening method is applied in this example to reduce the simulation time further. The parallel GA solver is run for three generations with the same limits and population values as before. The best solutions after each generation (Table 5.6) are examined, which reveals that $T_{\text{i\_Edc}}$, $K_{\text{p\_Q}}$, $T_{\text{i\_Q}}$, $K_{\text{p\_Vac}}$, $K_{\text{p\_P}}$ and $T_{\text{i\_P}}$ require further optimization since they vary considerably, while $K_{\text{p\_Edc}}$, $T_{\text{i\_Vac}}$ show markedly lower variations. For $K_{\text{p\_Edc}}$ and $T_{\text{i\_Vac}}$, variation of the maximum value from the minimum value is between 25-30% and for the other parameters that variation is significantly high.

Table 5.6: Run-Time Screening for Case Study 2

Parameter	Values after each generation			Parameter	Values after each generation		
	1$^{\text{st}}$	2$^{\text{nd}}$	3$^{\text{rd}}$		1$^{\text{st}}$	2$^{\text{nd}}$	3$^{\text{rd}}$
$K_{\text{p\_Edc}}$	6.045	5.516	6.935	$K_{\text{p\_Vac}}$	2.243	1.366	1.788
$T_{\text{i\_Edc}}$	0.888	1.277	1.853	$T_{\text{i\_Vac}}$	1.906	1.497	1.906
$K_{\text{p\_Q}}$	2.389	6.045	4.006	$K_{\text{p\_P}}$	1.826	0.269	1.826
$T_{\text{i\_Q}}$	1.566	0.379	0.843	$T_{\text{i\_P}}$	0.483	0.778	0.483

Hence, the best values obtained from the GA up to this point are assigned to $K_{\text{p\_Edc}}$ and $T_{\text{i\_Vac}}$, while the remaining parameters are optimized further. With the knowledge acquired from screening, the search limits may also be reduced, thus smaller populations can be used. The remaining six parameters are optimized after running the parallel GA for five additional generations with initial and surviving populations of 72 and 24, respectively. The new limits and optimization results are shown in the 8^{th} and 9^{th} columns of Table 5.5. The rms voltage at POI with optimized values is shown in Fig. 5.7(f). The total time taken for this process is 9.74 h and the results are as satisfactory as before, which confirms the efficiency of the proposed run-time screening method.

Optimization time and population details comparisons for the four methods discussed are shown in Table 5.7 and Table 5.8, respectively. Hybrid GA-Simplex approximately consumes 50% of the time consumed by the sequential GA while the parallel GA approximately consumes 30% of it. The time taken by the hybrid GA-Simplex method can be further reduced by using the parallel GA instead of the sequential GA for the initial part of the method. The arallel GA together with run-time screening shows incredibly good results by completing the simulation by spending only 22% of the time taken by the sequential GA. Thus, the results confirm that a considerable amount of time is saved by using the enhanced methods proposed in this thesis while obtaining high-quality optimal results.

Table 5.7: Time Comparison of Optimization Methods for Case Study 2

Method	Time (h)
Sequential GA	43.48
Hybrid GA-Simplex	21.31
Parallel GA	13.11
Parallel GA + screening	9.74

Table 5.8: Comparison of Population Details and Number of Simulation Runs for Case Study 2

	Sequential GA	Hybrid GA-Simplex		Parallel GA	Parallel GA with screening	
		GA	Simplex		GA for screening	GA after screening
Initial population	104	104	-	104	104	72
Surviving population	48	48	-	48	48	24
Generations	10	2	-	10	3	5
Simulation runs	537	153	95	544	208	176

5.3 Case Study 3

5.3.1 System and Controller Configuration

This example further confirms the effectiveness of the novel methods suggested in this thesis for optimizing sophisticated power systems with EMT simulators. The system considered here is a 200 MW back-to-back HVDC scheme shown in Fig. 5.8, which has a strong effective short circuit ratio (ESCR) of 4.62 at the rectifier side and a weak ESCR of 1.9 at the inverter side. During the disturbances, there can be commutation failure and performance of the system can be undesirable and at the same time it is hard to manually optimize the converter's controllers; thus it is selected as the third example.

The rated voltage of the dc system is 83.3 kV. Control strategies of the system are the same as in [7] and are shown in Figs. 5.9, 5.10, 5.11 and 5.12. Master controller generates the reference current by dividing the required power reference by the actual dc-voltage. That current reference is again compared with the reference current generated by the voltage-dependent current limit (VDCL), which is present to reduce the current order if a low

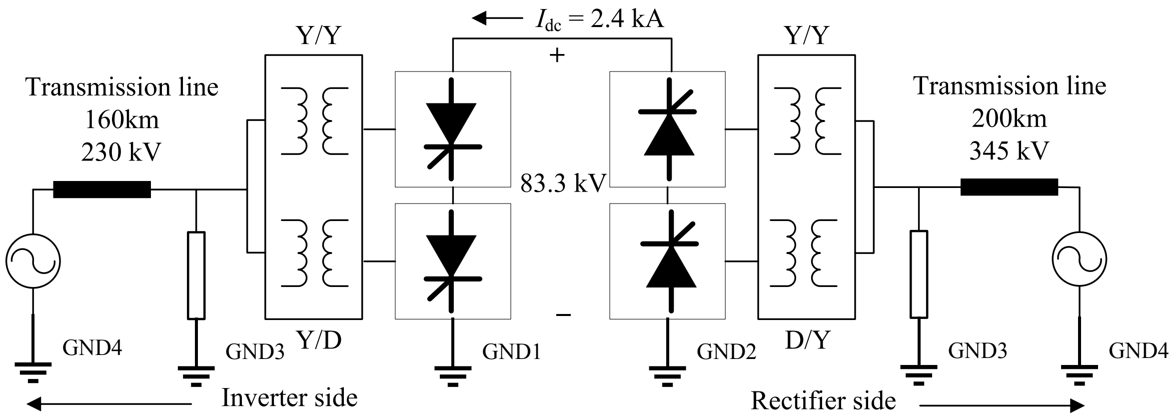

Figure 5.8: Schematic diagram of the HVDC system in case study 3.

dc-voltage occurs and to achieve smooth fault recovery [7]. There is a rate limiter to limit the rate of change of the current in the dc side. The minimum current selected goes through the rate limiter and generates the current reference to the converter. There are three control loops to process current, voltage, and extinction angle errors in the rectifier and inverter controllers. The minimum error between the three is used for generating firing angles. To check the system's performance, disturbances are applied as follows for the inverter side with lower ESCR.

(1) [4.1 s- 4.6 s] – The phase angle of the network is reduced by 15 degrees and restored;

(2) [5.1 s- 5.6 s] - AC voltage magnitude is reduced by 7% and restored;

(3) [6.1 s- 6.6 s] – Power reference is reduced by 50% and restored;

(4) [8 s- 8.05 s] – Three-phase-to-ground fault is applied at the inverter terminal.

The gain and time constant parameters of the PI controllers in the rectifier side (G_{r}, T_{r}) and inverter side (G_{i}, T_{i}) along with the parameters of VDCLs in the rectifier ($X_{\mathrm{h\_r}}$, IR_{r},DR_{r}) and inverter ($X_{\mathrm{h\_i}}$, IR_{i}, DR_{i}) side are optimized to achieve desirable performance in the system even when the disturbances occur. X_{h} is the upper break point of VDCL.

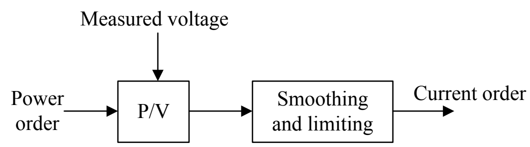

Figure 5.9: Master control. [7]

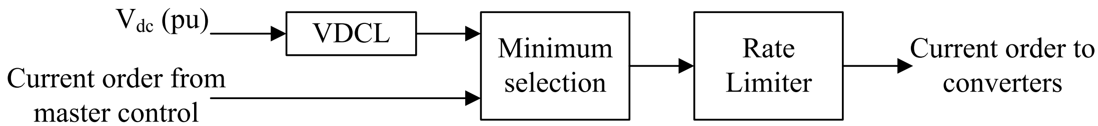

Figure 5.10: Voltage-dependent current limit (VDCL) structure [7].

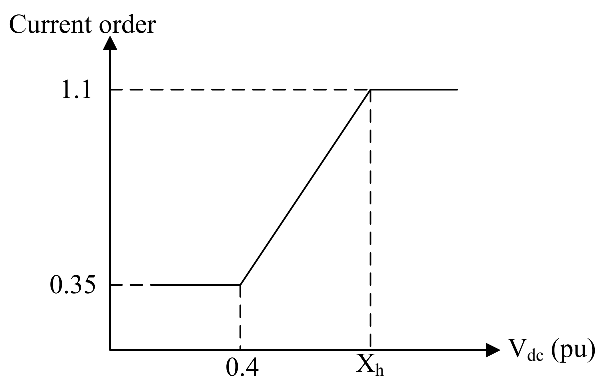

Figure 5.11: VDCL characteristic [7].

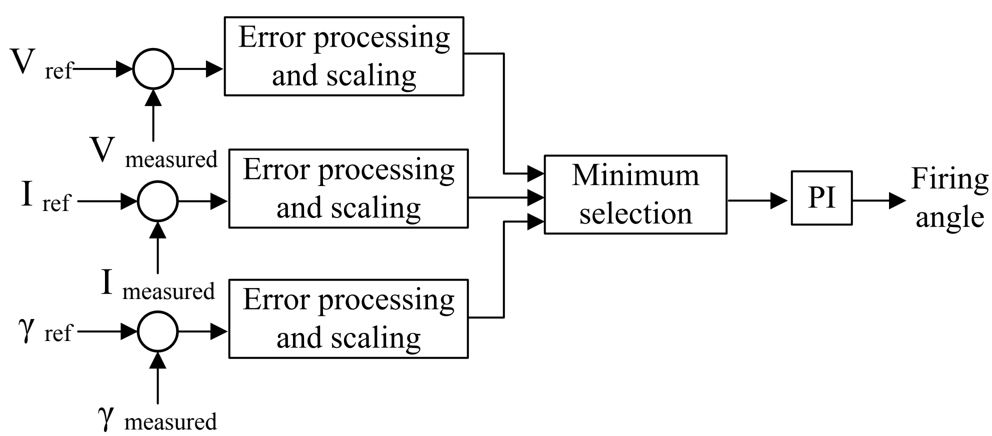

Figure 5.12: Valve group control loops [7].

IR and DR are the increasing rate and decreasing rate of the rate limiter respectively. DC current (I_{dc}) in the HVDC system is a good measurement to evaluate the performance of the system. Therefore, the time square error (ISE) of the DC current is used as the OF in this case as follows:

$$OF = \int_0^T (I_{\text{dc}} - I_{\text{ref}})^2 \, \mathrm{d}t \tag{5.5}$$

5.3.2 Screening of Optimization Variables

In this case, the first screening method is applied to identify the parameters that do not require optimization at the selected point. The initial parameter values are shown in the 3$^{\text{rd}}$ column of Table 5.9. According to Method I discussed in Chapter 4, section 4.1.1, the starting parameter values are changed by ±3% and ±10% in the separate runs and the distribution of the resulting OF values are shown in Fig. 5.13. The parameter numbers in Fig. 5.13 are from Table 5.9. The OF value results show that it does not vary significantly even though the values of DR_{i}, IR_{r} and DR_{r} are changed by ±10%. This indicates that these parameters can be left without optimizing. By doing so, the size of the optimization problem can be reduced from 10 to 7. Although it is a small reduction, considerable computer efficiency can be achieved while using GA for optimization as the population values can be reduced due to the lower number of parameters to be optimized.

5.3.3 Optimization of Parameters

In this case all the parameters are optimized first for the comparison purposes and optimization of influential parameters is done in the final part.

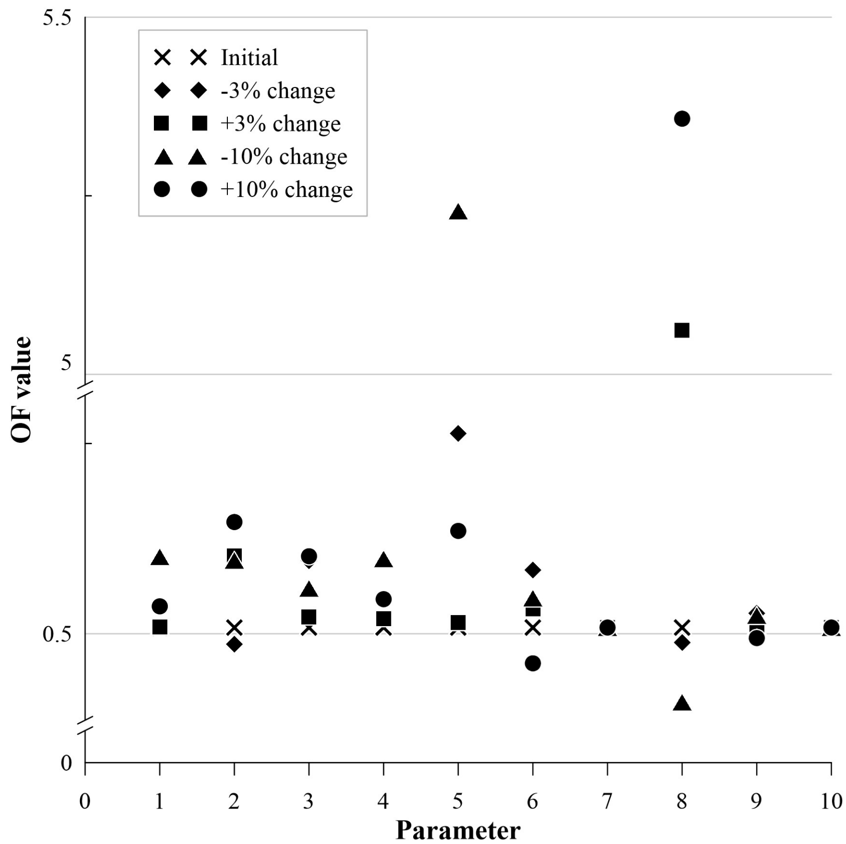

Figure 5.13: Distribution of OF values with parameter perturbations in case study 3.

5.3.3.1 Optimization Using Sequential GA

Using the initial population of 504 and the surviving population of 400, the sequential GA is run for 15 generations to optimize all the parameters in the system. The parameter limits and final optimized results are in the 4th and 5th columns of Table 5.9 and the waveforms of DC current obtained for initial and optimized parameters are shown in Fig. 5.14(a) and Fig. 5.14(b) respectively .

According to the results, after tuning the controller parameters, the system performs well during disturbances and faults. However, it takes 20.35 h for the simulations; thus the

Table 5.9: Initial values and Optimized Values for Subsections 1, 2 for Case Study 3

Parameter Number	Parameter	Initial values	Limits	Sequential GA	Hybrid GA-Simplex	
					GA	Simplex
1	G_{r}	1	(0,2)	1.364	0.961	1.157
2	T_{r}	0.04	(0,0.1)	0.004	0.015	0.0048
3	G_{i}	1.5	(0,2)	0.439	0.523	0.617
4	T_{i}	0.008	(0,0.1)	0.023	0.075	0.027
5	$X_{\mathrm{h\_i}}$	0.94	(0.7,1)	0.783	0.81	0.813
6	IR_{i}	2.5	(2,5)	2.586	3.44	3.15
7	DR_{i}	100	(50,120)	63.29	55.16	54.36
8	$X_{\mathrm{h\_r}}$	0.94	(0.7,1)	0.751	0.701	0.702
9	IR_{r}	2.5	(2,5)	4.79	3.15	3.39
10	DR_{r}	100	(50,120)	106.3	73.9	70.8
	OF value	0.508		0.115	0.272	0.14

new approaches proposed in the thesis are used to improve the computational efficiency of the solution.

5.3.3.2 Optimization Using Hybrid GA-Simplex Algorithm

In this section, sequential GA with the same population values and parameter limits is launched only for 2 generations and the results acquired after that are used as the initial values to the Simplex algorithm. The 6^{th} and 7^{th} columns of Table 5.9 illustrate the optimization results and waveforms obtained with intermediate GA values and the final Simplex optimized values are shown in Fig. 5.14(c) and Fig. 5.14(d), respectively .

Even though the solution found here is not good enough when it is compared with the optimal solution found after 15 generations of GA, the hybrid algorithm has found

Table 5.10: Initial values and Optimized Values for Subsections 3,4 and 5 for Case Study 3

Parameter	Initial values	Limits	Parallel GA	Hybrid parallel GA-Simplex		Parallel GA with screening
				Parallel GA	Simplex	
G_{r}	1	(0,2)	1.247	1.789	1.446	1.275
T_{r}	0.04	(0,0.1)	0.006	0.0097	0.0075	0.0037
G_{i}	1.5	(0,2)	0.56	0.384	0.517	0.655
T_{i}	0.008	(0,0.1)	0.026	0.026	0.02	0.040
$X_{\mathrm{h\_i}}$	0.94	(0.7,1)	0.837	0.865	0.779	0.755
IR_{i}	2.5	(2,5)	2.41	4.66	4.78	2.096
DR_{i}	100	(50,120)	91.2	118.6	119.5	100
$X_{\mathrm{h\_r}}$	0.94	(0.7,1)	0.764	0.782	0.869	0.753
IR_{r}	2.5	(2,5)	4.74	4.86	5.05	2.5
DR_{r}	100	(50,120)	61.1	80.5	77.9	100
OF value	0.508		0.119	0.2	0.158	0.159

an acceptable solution, which gives a better OF value than the initial values while only consuming 3.13 h time. Nevertheless, having a good initial point always directs the Simplex algorithm to obtain better solutions. Therefore, it depends on the parameter set generated by the GA.

5.3.3.3 Optimization Using Parallel GA

Here, the parallel GA is used to decrease the simulation time while running the simulation using GA for 15 generations. The optimum solution found is presented in the 4$^{\text{th}}$ column of Table 5.10. The DC current waveforms are shown in Fig. 5.15(b). The results are almost the same as the results obtained using the sequential GA and this method only takes 13.37

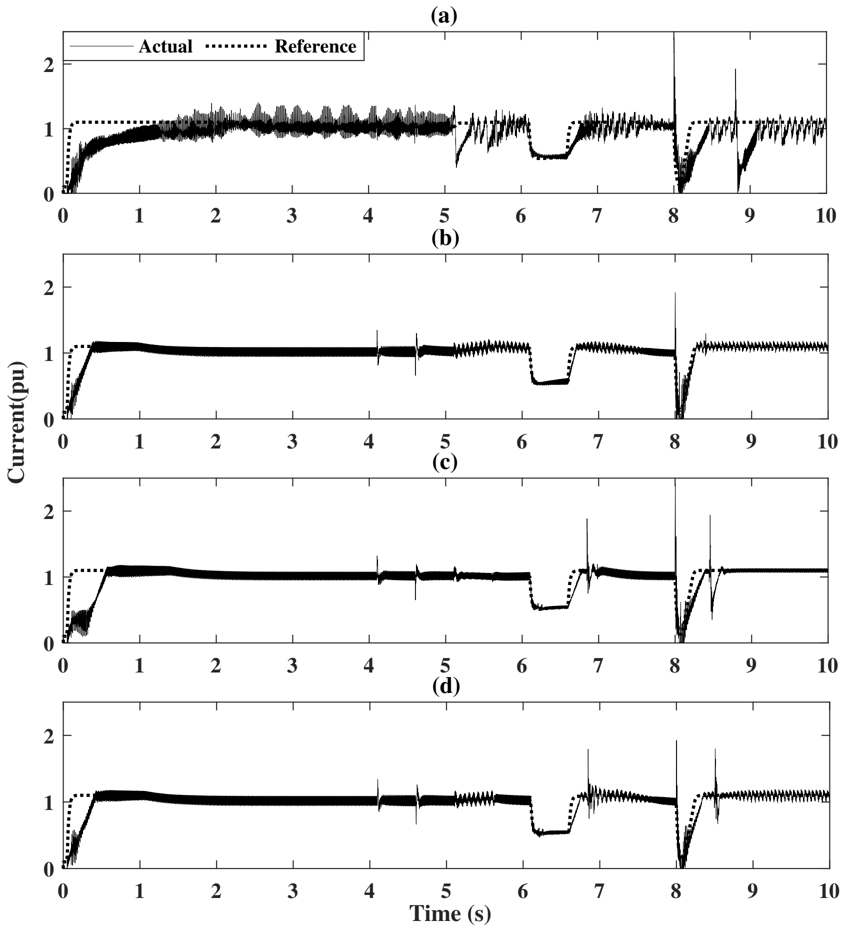

Figure 5.14: Active power output with (a) initial values, (b) sequential GA optimized values, (c) intermediate GA values, (d) final Simplex optimized values.

h to complete the simulation.

5.3.3.4 Optimization Using Hybrid Parallel GA-Simplex Algorithm

In this part, the parallel GA is used in the hybrid algorithm to achieve more computational benefits. The procedure followed here is the same as in sub-section 5.3.3.2 and the only difference is that the parallel GA is used to run the first 2 generations of GA which allows

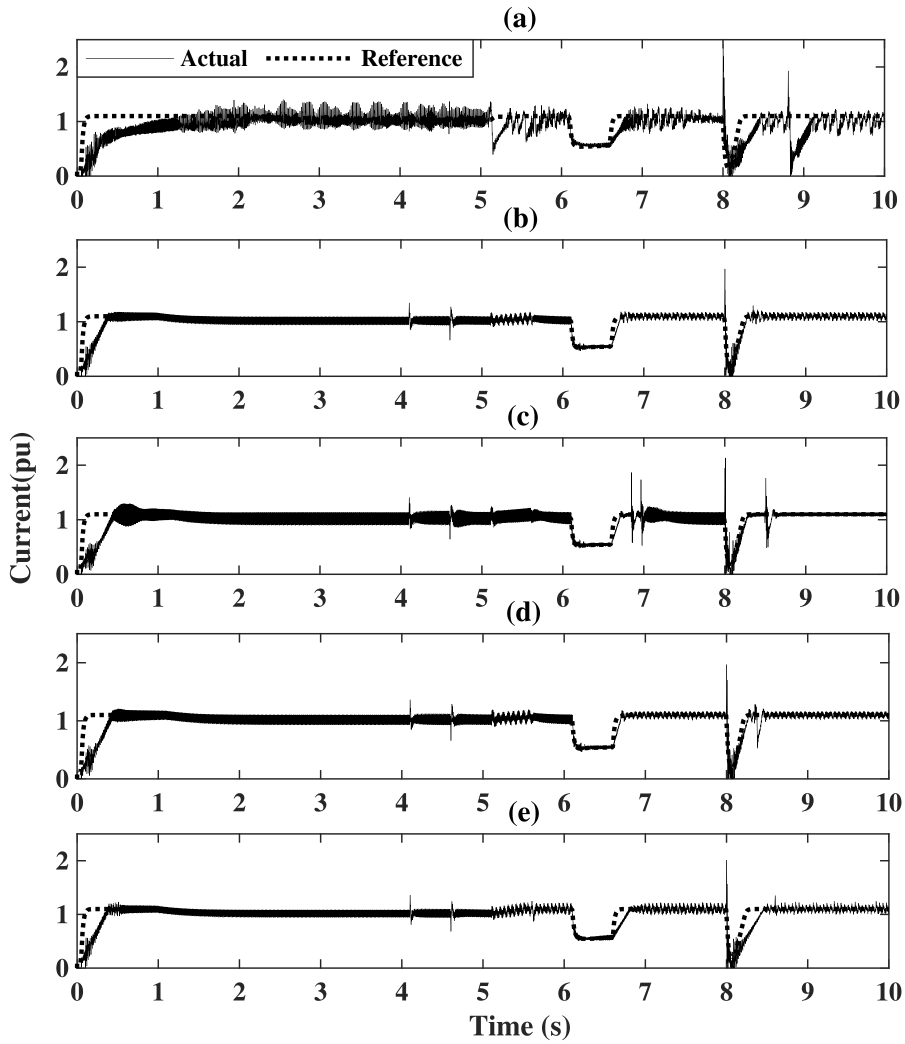

Figure 5.15: Active power output with (a) initial values, (b) parallel GA optimized values, (c) intermediate parallel GA values, (d) final Simplex optimized values, (e) parallel GA optimization of 7 influential parameters.

the designer to achieve better time than in sub-section 5.3.3.2. The optimization results acquired are shown in the 5th and 6th columns of Table 5.10 and the waveforms are in 5.15(c) and 5.15(d). The time taken for optimization is reduced to 2.53 h by this method.

Table 5.11: Time Comparison for Case Study 3

Method	Time (h)
Sequential GA	20.35
Hybrid GA-Simplex	3.13
Parallel GA	13.37
Hybrid Parallel GA-simplex	2.53
Parallel GA + screening	4.98

5.3.3.5 Optimization of Influential Parameters

Using initial screening, it is found that DR_{i}, IR_{r} and DR_{r} can be excluded from the optimization thus the dimension of the design problem is now reduced from 10 to 7. These excluded parameters are assigned their initial values. Then the remaining 7 parameters are optimized by running the parallel GA for 10 generations with 400 and 240 as initial and surviving populations, respectively. The results obtained are shown in the 7^{th} column of Table 5.10 and the waveform is shown in Fig. 5.15(e). Simulations take 4.98 h, which proves that the screening improves the efficiency of the optimization process.

The time taken for the 5 approaches and their population details are compared in Table 5.11 and Table 5.12, respectively. The time taken by the hybrid GA- Simplex algorithm is 16% of the time taken by the sequential GA even though the quality of the output is somewhat lower than the one obtained from the sequential GA. The parallel GA gives the same good results as the sequential GA while consuming only 65% of the time consumed by the sequential GA. Since the HVDC case only takes a few seconds to run in PSCAD/EMTC, significant time reduction cannot be achieved by using the parallel GA as the time taken for communication between the EMT solver and the Python script is considerable when it is

Table 5.12: Comparison of Population Details and Number of Simulation Runs for Case Study 3

	Sequential GA	Hybrid GA-Simplex		Parallel GA	Hybrid parallel GA-Simplex		Parallel GA with screening
		GA	Simplex		Parallel GA	Simplex	
Initial population	504	504	-	504	504	-	400
Surviving population	400	400	-	400	400	-	240
Generations	15	2	-	15	2	-	10
Simulation runs	6105	905	201	6112	912	220	2568

compared with the simulation time of the case. However, the results are improved and show the efficiency of the proposed methods.

This chapter provided example cases that were optimized using the proposed methods. The configurations of the systems in the case studies were described including their control systems. Application of the screening methods, hybrid GA-Simplex algorithm and parallel GA in the optimization process were discussed in detail and finally comparison of the results was presented to demonstrate the success of the novel methods introduced in the thesis.

Chapter 6

Contributions, Conclusions, and Future Directions

6.1 Thesis Contributions and Conclusions

The thesis addressed practical problems that engineers face when using EMT simulators for optimal design of complicated controllers in power systems, e.g., controller tuning in converters in the power systems. These problems stem from the large computational burden of both the EMT and optimization algorithms, and the repetitive nature of the design cycle wherein a large number of simulations need to be conducted. The thesis introduced two screening methods, a hybrid GA-Simplex algorithm, and a parallel GA algorithm to overcome these challenges.

The optimization results of the systems with several converters and multiple control loops revealed that the screening methods were able to correctly identify influential parameters to assist in reducing the dimension of a problem, thereby lowering the burden of the optimization algorithm. Analysis of the simulation time of the three case studies shows that 25% - 60% of the time can be saved (depending on the cases) by using the proposed screening methods.

Comparison of the results revealed that the proposed hybrid GA-Simplex algorithm, which uses a GA to detect the global optimal area and Simplex to conduct local search, performed well in decreasing the simulation time without adversely affecting the quality of the final optimal designs. The results reflect that this approach can save 50% - 85% of the simulation time, which demonstrates the efficacy of the method.

The proposed parallel GA achieved the same quality results as from the sequential GA, consuming considerably lower time than the sequential GA. However, for cases whose simulations (one run) complete in a few seconds, the suggested method does not exhibit significant reduction in time since the time utilized for communication between the Python script and the EMT solver is considerable.

6.2 Recommendations for Future Work

The GA used in this thesis uses a random number generator to produce the initial population. Thus there is a possibility that the initial population consists of similar type of chromosomes, which reduces the diversity of the initial population. Hence, future research can modify the GA to include some filtering methods to eliminate similar chromosomes from the initial population so that it consists of different candidate parameter sets distributed in the search area.

Moreover, future development may focus on developing a single algorithm for hybrid GA-Simplex algorithm to automate the transition from GA to Simplex by processing OF values so that the designer does not have to handle it manually.

References

[1] S. Subedi, M. Rauniyar, S. Ishaq, T. M. Hansen, R. Tonkoski, M. Shirazi, R. Wies, and P. Cicilio, "Review of methods to accelerate electromagnetic transient simulation of power systems," *IEEE Access*, vol. 9, pp. 89 714–89 731, 2021.

[2] K. Kobravi, "Optimization-enabled transient simulation for design of power circuits with multi- modal objective functions," Ph.D. dissertation, University of Manitoba, 2007.

[3] H. Singh, "Development of optimization methods to deal with current challenges in engineering design optimization," *AI Communications*, vol. 29, 11 2014.

[4] I. K. Park, "Real- time application of optimization- enabled electromagnetic transient simulation," Master's thesis, University of Manitoba, 2012.

[5] S. Filizadeh, "Optimization- enabled electromagnetic transient simulation," Ph.D. dissertation, University of Manitoba, 2012.

[6] H. Zhao, "Stability and accuracy of electromagnetic transient simulation algorithms," Ph.D. dissertation, University of Manitoba, 2020.

[7] S. Filizadeh, A. M. Gole, D. A. Woodford, and G. D. Irwin, "An optimization-enabled electromagnetic transient simulation-based methodology for hvdc controller design," *IEEE Transactions on Power Delivery*, vol. 22, no. 4, pp. 2559–2566, 2007.

[8] R. C. Bansal, "Optimization methods for electric power systems: An overview," *International Journal of Emerging Electric Power Systems*, vol. 2, no. 1, 2005.

[9] H. Mejbri, K. Ammous, S. Abid, H. Morel, and A. Ammous, "Bi-objective sizing optimization of power converter using genetic algorithms: Application to photovoltaic systems," *Compel-the International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 33, pp. 398–422, 2013.

[10] A. Ametani, "Electromagnetic transients program: History and future," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 16, 08 2020.

[11] *EMTDC User's Guide*, Manitoba HVDC Reasearch Centre, 2011.

[12] A. GoIe, S. Filizadeh, R. Menzies, and P. Wilson, "Optimization-enabled electromagnetic transient simulation," in *IEEE Power Engineering Society General Meeting, 2004.*, 2004, pp. 1133 Vol.1–.

[13] H. Rasool, A. Rasool, A. Ikram, U. Rasool, M. Jamil, and H. Rasool, "Compatibility of objective functions with simplex algorithm for controller tuning of hvdc system," *Ingeniería e Investigación*, vol. 39, 02 2020.

[14] S. E. H. Randy L. Haupt, *Practical Genetic Algorithms.* John Wiley Sons, Ltd, 2003, ch. 1-3, pp. 1–66.

[15] X.-S. Yang, *Optimization and Metaheuristic Algorithms in Engineering*, 12 2013, pp. 1–23.

[16] T. A. W. Mykel J. Kochenderfer, *Algorithms for Optimization.* The MIT Press Cambridge, Massachusetts London, England, 2019, ch. 1-9, pp. 1–165.

[17] O. Özkaraca, "A review on usage of optimization methods in geothermal power generation," *Mugla Journal of Science and Technology*, pp. 130–136, 06 2018.

[18] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[19] J. Lagarias, J. Reeds, M. Wright, and P. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, pp. 112–147, 12 1998.

[20] R. Chelouah and P. Siarry, "Genetic and nelder–mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions," *European Journal of Operational Research*, vol. 148, no. 2, pp. 335–348, 2003, sport and Computers. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221702004010

[21] E.-G. Talbi and T. Muntean, "Designing embedded parallel systems with parallel genetic algorithms," in *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, 1993, pp. 7/1–7/2.

[22] Z.-x. Wang and G. Ju, "A parallel genetic algorithm in multi-objective optimization," in *2009 Chinese Control and Decision Conference*, 2009, pp. 3497–3501.

[23] N. Adachi and Y. Yoshida, "Accelerating genetic algorithms: protected chromosomes and parallel processing," in *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1995, pp. 76–81.

[24] Z. Wang, X. Yin, Z. Zhang, and J. Yang, “Pseudo-parallel genetic algorithm for reactive power optimization,” in *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491)*, vol. 2, 2003, pp. 903–907 Vol. 2.