OPTIMAL POWER FLOW AND NETWORK LOADABILITY USING FEEDBACK-BASED SELF-ADAPTIVE DIFFERENTIAL EVOLUTION AND MULTIOBJECTIVE ALGORITHMS

By

Fares Theyab A Alharbi

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering - Master of Science

2018

ABSTRACT

OPTIMAL POWER FLOW AND NETWORK LOADABILITY USING FEEDBACK-BASED SELF-ADAPTIVE DIFFERENTIAL EVOLUTION AND MULTIOBJECTIVE ALGORITHMS

By

Fares Theyab A Alharbi

In modern electrical grids, planning and operation processes require efficient optimization tools. Optimal placement and sizing of Flexible AC transmission system (FACTS) devices, renewable energy resources, and energy storage units, to name a few, are optimization tasks in the planning process. Minimizing the cost of generated power from committed generators in the operation process is an important part of power system operations. For power system optimization problems, several optimization algorithms have been proposed and used in the past two decades. However, the need for efficient optimization algorithms customized to power system problems still exists. The research reported in this thesis develops novel evolutionary optimization approaches for two applications: optimal power flow (OPF) and optimal placement and sizing of FACTS to enhance electrical network loadability.

For optimal power flow, two new feedback-based self-adaptive differential evolution algorithms are proposed. Prior to applying the proposed methods to the power system test cases, they are tested on standard mathematical benchmark problems. The self-adaptive differential evolution algorithms showed significant improvement in the benchmark problems compared to other algorithms. More importantly, in this work, the feedback-based self-adaptive differential evolution algorithms demonstrated good improvement in results and in convergence rate in several power system test cases.

To enhance the loadability of an electrical network, a new multiobjective-based frame work is proposed for optimal placement and sizing of FACTS devices. The proposed method has been applied to commonly used FACTS devices, thyristor-controlled series controllers (TCSCs), and demonstrated excellent results in the electrical loading margins as well as the investment costs compared to other available methods.

Dedicated To
My parents, Theyab Alharbi and Radhyah Alharbi

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. Joydeep Mitra, for his advice, assistance, and support. His encouragement and valuable advice have motivated me to succeed in my graduate studies and to accomplish this thesis. His great support was not only limited to academics, but also in other aspects of life such as personal and family life. I thank Dr. Mitra for his support and solidarity during the time when a member of my family was experiencing serious health challenges. Without his support, this work could not have been done.

I am thankful to my committee members Dr. Fang Z. Peng and Dr. Bingsen Wang for their time and effort. I would like to thank all professors who taught me and advised me at MSU and before I joined MSU. I also want to thank all the authors I have cited in this thesis; without their inspiring work I could not accomplish this thesis.

Thanks go out to my colleagues in the ERiSe lab, Dr. Mohammed Benidris, Dr. Samer Sulaeman, Dr. Nga Nguyen, Mr. Saleh Almasabi, Ms. Yuting Tian, Mr. Atri Bera, for their help and collaboration.

I express my sincere thanks and gratitude reach to my parents, brothers, sisters, and my wife, for their support, encouragement, and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

Optimization is an essential part of electrical grids planning and operation processes. With the current development towards modern electrical grids, the available optimization methods need to be improved and customized for power system problems. In this thesis, two main optimization problems are considered for improvement namely, optimal power flow (OPF) and optimal placement and sizing of FACTS devices to control power flow and to enhance electrical network loadability.

Optimal power flow is an important optimization problem in power system planning and operation. The full AC model of the OPF is a nonlinear complex problem. The complexity of the OPF problem increases as the size of the electrical grid increases.

In the first part of this thesis, two feedback-based self-adaptive differential evolution algorithms are proposed to solve the OPF. Differential evolution (DE) has been widely used in global optimization due to its effectiveness and simplicity compared to other evolutionary algorithms. However, the performance of evolutionary algorithms is dependent on control parameters of the evolutionary process. Adaptive algorithms have significantly improved the performance of evolutionary algorithms.

Chapter 2 proposes the feedback-based self-adaptive differential evolution algorithms to improve the performance of DE. Two feedback-based self-adapting control parameters algorithms, (FBjDE-I) and (FBjDE-II), are proposed and tested on several mathematical benchmark problems and have been compared to other DE algorithms in the literature. The results show that the proposed feedback-based self-adaptive algorithms have successfully improved the performance the DE algorithm. In Chapter 3, the optimal power optimization problem is presented and the proposed algorithms in Chapter 2, FBjDE-I and FBjDE-II, are introduced to solve the OPF.

In the second part of the thesis, the optimal placement and sizing of FACTS devices is considered to improve electrical network loadability. Enhancing the network loadability is crucial in modern electrical grids because transmission lines are vulnerable to being overloaded beyond their thermal limits in deregulated and highly competitive markets with increasing load demands. Overloading transmission lines will not only increase losses, but also may drive the system into insecure operating conditions. Flexible AC transmission system FACTS devices have been effectively utilized to enhance transmission network loadability.

Chapter 4 presents a new mutliobjective-based framework for optimal placement and sizing of FACTS devices to improve loadaiblity in electrical grids and to relieve congestion. Additionally, Chapter 4 provides a brief summary of FACTS devices, multiobjective optimization problems, and the application of the proposed method using thyristor-controlled series controllers (TCSCs) devices. The thesis is concluded with future work in Chapter 5.

Chapter 2

Feedback Based Self-Adapting Control Parameters for Differential Evolution

2.1 Introduction

Differential evolution algorithms have been under extensive research to improve their reliability, robustness, and performance [1,2]. It has been pointed out that the performance of evolutionary algorithms depend largely on the parameters that are controlling the evolutionary process. Compared to other evolutionarily algorithms, DE has fewer parameters [1]. Yet the performance of the differential evolution is affected by the choice of the control parameters due to the fact that finding suitable parameters set is a problem-dependent task [3].

In DE, the control parameters are the population size NP, crossover rate CR, and the mutation scale factor F. One of the effective methods that has enhanced the performance of DE is controlling or tuning the scale factor F and the crossover rate CR. According to [4], tuning the control parameter takes place before running the evolutionary algorithm, whereas controlling the parameters takes place dynamically during the run of the evolutionary algorithm.

According to [4], the techniques to control the parameter are classified into three types. First, *deterministic parameter control* in which the parameters are updated based on deterministic rules.

Second, *adaptive parameter control* where the parameters are updated based on feedback from the evolutionary process. Third, the *self-adapting Control* in which the control parameters are incorporated with individuals' parameters. The evolution operators are not only applied to the candidate solutions, but are also applied to the control parameters.

In JADE algorithm [5], F and CR are generated from Gaussian (normal) and Cauchy distributions around the mean of the successful parameters from previous generations .The MDEpBX algorithm in [6] proposed that the control parameters should be controlled based on statistical values of the successful population average population. A self-adapting control algorithm for differential evolution by J.Brest et al in [3], known as (jDE), is widely used due to its effectiveness and simplicity compared to other adaptation schemes [7].

In this chapter, two adaptive techniques based on the feedback of the evolutionary process are proposed to improve the performance and robustness of jDE, namely FBjDE-I and FBjDE-II. The proposed feedback-based self-adapting control parameters for DE algorithm ,FBjDE-I and FBjDE-II, will be tested on several benchmark problems and compared to DE and jDE algorithms.

2.2 Differential Evolution (DE)

The general procedure of the classical differential evolution *(DE/rand/1/bin)* as in [1] and [2] can be briefly summarized in the following steps. Step 1, the algorithm starts with a population of random points NP in the search space. Step 2, the objective or fitness function values of all the population are evaluated. Step 3, the mutation operator of the differential evolution is performed to generate a donor vector $V_{i,g+1}$ for each original vector X_i as in (2.1).

$$V_{i,g+1} = X_{r_1,g} + F \times (X_{r_2,g} - X_{r_3,g}) \tag{2.1}$$

where i is the vector index $i = 1...NP$; g is the generation number; r_1, r_2, r_3 are random indices $\in NP$; NP is the population size and $r_1, r_2, r_3 \neq i$. Step 4, the crossover operator in (2.2) builds a trail vector for each $U_{i,g+1}$ individual using the donor vector $V_{i,g+1}$ and the original vector $X_{i,g}$ with a crossover rate CR.

$$u_{ij,g+1} = \begin{cases} v_{ij,g+1}, & \text{If } rand \leq CR \quad \text{or} \quad j = j_r \\ x_{i,j,g}, & \text{Otherwise} \end{cases} \tag{2.2}$$

where $j = 1, ..., D$; and D is the dimension of the decision variables of the objective function. In [1], to ensure that the crossover takes place in at least one dimension of the donor vector $U_{i,g+1}$ an or condition, when the dimension j equals a random dimension index j_r, is added with the crossover rate CR as in (2.2).

Step 5, the selection operator in (2.3) compares the parent $X_{i,g}$ and its offspring $U_{i,g+1}$ and chooses the one that has a better objective function or fitness value to become the parent$X_{i,g+1}$ for the next successive generation. In (2.3) and in most optimization algorithms, the optimization problem is considered to be a minimization problem.

$$X_{i,g+1} = \begin{cases} U_{i,g+1}, & \text{If } f(U_{i,g+1}) < f(X_{i,g}) \\ X_{i,g}, & \text{Otherwise} \end{cases} \tag{2.3}$$

2.3 The self-adapting control parameters in differential evolution (jDE)

Differential evolution (DE) has been widely used in the literature [2]. The DE has three parameters which are the number of population NP, a mutation scale factor F, and a crossover rate CR. The population size NP is usually proportional to the dimension of the problem.

The performance of the differential evolution is highly dependent on the choices of the mutation scale factors F and the crossover rates CR. Empirical or recommended values of the parameters have been used in the literature [8] [9]. In practice, however, the proper choice of parameters is a problem dependent task [3].

In the self-adapting algorithm [3], each candidate solution is encoded with a scale factor $F_{i,g}$

and a crossover rate $CR_{i,g}$. These parameters are updated to possibly better values as follows: the scale factor $F_{i,g}$ is controlled using equation (2.4).

$$F_{i,g+1} = \begin{cases} F_l + rand_1 \times F_u, & \text{If } rand_2 < \tau_1 \\ F_{i,g}, & \text{Otherwise} \end{cases} \tag{2.4}$$

F_l is the lower boundary of the scale factor F. F_u is the upper boundary of the scale factor F. In [3] the values of the lower and upper limits of F are suggested to be $F_l = 0.1$, $F_u = 0.9$. Similarly, the crossover rate associated with each individual is controlled using equation (2.5)

$$CR_{i,g+1} = \begin{cases} rand_1, & \text{If } rand_2 < \tau_2 \\ CR_{i,g}, & \text{Otherwise} \end{cases} \tag{2.5}$$

τ_1 and τ_2 are constants $\in [0,1]$; $rand_1$, $rand_2$ are random numbers, generated uniformly $\in [0,1]$. In [3], τ_1 and τ_2 are recommenced to be $\tau_1 = 0.1$ and $\tau_2 = 0.1$.

2.4 Feedback-based self-adapting control parameters for DE (FBjDE)

jDE adaptive algorithm is a simple yet an efficient adaptive algorithm for F and CR and has shown better results compared to DE and other adaptive algorithms [3, 7]. The adaptation of $F_{i,g+1}$ and $CR_{i,g+1}$ is dependent on the choice of τ_1 and τ_2. These parameters should be defined by the user before running the algorithm. In [3] the authors have used and recommended values of $\tau_1 = 0.1$ and $\tau_2 = 0.1$ which were successful in wide range of single objective problems.

In this work, feedback-based self-adapting control algorithms are proposed to utilize the feedback information from the evolutionary process to improve the performance of DE. The adaptation of the control parameters $F_{i,g+1}$ and $CR_{i,g+1}$ will be based on the feedback from heuristic rules. The proposed adaptive algorithms are explained in the following subsections.

2.4.1 Feedback-based self-adapting control parameters for DE-I (FBjDE-I)

In the FBjDE-I, the adaptation of the control parameters depends on how the current parameters $F_{i,g}$ and $CR_{i,g}$ have contributed to the evolution of an individual X_i. In other words, it gets feedback from the current generation of$F_{i,g}$ and $CR_{i,g}$ and uses the information to decide whether to keep the next successive $F_{i,g+1}$ and $CR_{i,g+1}$ or to change them to possible better values. In FBjDE-I, each individual will be encoded with a mutation scale factor $F_{i,g}$ and a crossover rate $CR_{i,g}$ along with feedback information flag $FB_{i,g}$.

$$FB_{i,g} = \begin{cases} 1, & \text{If } f(U_{i,g+1}) < f(X_{i,g}) \\ 0, & \text{Otherwise} \end{cases} \tag{2.6}$$

$FB_{i,g}$ in equation (2.6) is a feedback information of whether the crossover operation in the current generation g is successful or not for each individual X_i. Based on the feedback information $FB_{i,g}$, the scale factor is controlled as in (2.7).

$$F_{i,g+1} = \begin{cases} F_{i,g}, & \text{If } FB_{i,g} = 1 \\ F_l + rand \times F_u, & \text{Otherwise} \end{cases} \tag{2.7}$$

The crossover rate CR_i is controlled similarly using the previous information of each individual $FB_{i,g}$ as in equation (2.8). To compare FBjDE and jDE, the values of the lower and upper limits of F are as suggested in [3] $F_l = 0.1$ and $F_u = 0.9$.

$$CR_{i,g+1} = \begin{cases} CR_{i,g}, & \text{If } FB_{i,g} = 1 \\ rand, & \text{Otherwise} \end{cases} \tag{2.8}$$

Compared to jDE, the adaptation of the FBjDE-I does not depend on the parameters τ_1 and τ_2. It only depends the performance of the parameters F_i and CR_i in the evolution of each individual. In FBjDE-I, successful parameters will surviver with their corresponding individuals, and they will most likely contribute in creating successful offsprings.

2.4.2 Feedback-based self-adapting control parameters for DE-II (FBjDE-II)

In general the performance of the DE depends highly on the mutation scale factor F compared to the crossover rate CR [5]. In FBjDE-II, more emphasis in controlling the scale factor F is applied, while the crossover rate will be controlled as in FBjDE-I.

In jDE and FBjDE-I, the limits of the scale factor F_l and F_u are fixed in the evolutionary processes, and F_i is uniformly randomly drawn between the upper and the lower bounds of F_i for each individual.

To exploit the information of how an individual has evolved to the current position, the range for generating a scale factor F_i should change according to the performance of the previously generated scale factors. The learning process from the feedback of the scale factor F_i performance can be used as indicator to dynamically control the lower and upper bounds of the scale factor range of each candidate solution in the population.

For instance, if the scale factor has increased and the individual has not improved to a better position, it is likely that the scale factor range needs to be moved to a slightly better region. However, if the current solution is better than its parent and the scale factor that generated the current solution is higher than the scale factor that was used to generate its parent, it is more likely that higher scale factors are more suitable for that individual in this region of the search space.

This information should be utilized to generate potentially better scale factors for the next successive iteration. That could be achieved by dynamically updating the lower bound $F_{l,i}$ and the upper bound $F_{u,i}$ for the scale factor F_i.

In [1], the range of the scale factor is $F_i \in [0, 2]$. In FBjDE-II, however, a more flexible range for the scale factor is suggested $F_i \in [-2, 2]$. Allowing a negative scale factor would make the movement of each individual more flexible in the search space.

The feedback information for the current individual is given in (2.6). Based on that, the range of generating $F_{i,g+1}$ will be updated depending on the improvement that was achieved by preceding values of $F_{i,g}$ and $F_{i,g-1}$ in the previous generations. Controlling the scale factor bounds

Algorithm 1 Controlling the limits of the scale factor range

Evaluate $FB_{i,g}$ using (2.6)
if $FB_{i,g} = 1$ **then**
    if $F_{i,g} \geq F_{i,g-1}$ **then**
        $F_{l,i,g+1} = F_{l,i,g} + \lambda$
        $F_{u,i,g+1} = F_{u,i,g} + \lambda$
    else
        $F_{l,i,g+1} = F_{l,i,g} - \lambda$
        $F_{u,i,g+1} = F_{u,i,g} - \lambda$
    end if
else
    if $F_{i,g} \geq F_{i,g-1}$ **then**
        $F_{l,i,g} = F_{l,i,g} - \lambda$
        $F_{u,i,g} = F_{u,i,g} - \lambda$
    else
        $F_{l,i,g} = F_{l,i,g} + \lambda$
        $F_{u,i,g} = F_{u,i,g} + \lambda$
    end if
end if

$F_{l,i,g+1}$ and $F_{l,i,g+1}$ for the following successive generations is illustrated in the pseudocode presented in Algorithm. 1. $F_{l,i}$ and $F_{u,i}$ will change continuously based on the evolution of the solution X_i.

In Algorithm1, λ defines movement of the scale factor range $[F_{l,i}, F_{u,i}]$. Since the span of the allowed range is 4 units, a reasonable value of λ is in the range $\in [0.01, 0.25]$. As the value of λ gets higher the movement of the range gets faster. A value of $\lambda = 0.1$ is used in this work. The complete algorithm for FBjDE-II is depicted in Algorithm 2. To limit the scale factor $F_i \in [-2, 2]$, $F_{l,i}$ can take values $\in [-1.5, 0.5]$ and $F_{l,i} \in [-0.5, 1.5]$. After controlling $F_{i,l}$ and $F_{i,u}$, $F_{i,g+1}$ is controlled using (2.7). Note that the crossover rate in FBjDE-II is controlled as in FBjDE-I.

2.5 Benchmark problems

In this work the performance of the three adaptive algorithms jDE, FBjDE-I, and FBjDE-II will be tested on well-known numerical benchmark problems [10, 11].

Algorithm 2 Pseudocode of the FBjDE-II Algorithm

```
Initialize: NP Population, Maximum generation g_max, Scale factors F_{i,0}, and Crossover rates CR_{i,0}
while Termination Criteria is not met or g < g_max do
    Build a donor vector V_{i,g} for each individual X_{i,g} using equation (2.1)
    Creat a trial vector U_{i,g} using the parent X_{i,g} and the donor V_{i,g} using equation (2.2)
    Select: Replace the child with the parent if it is better using equation (2.3)
    Get Feedback information for all individuals FB_{i,g} using (2.6)
    Control scale factor F_{i,g+1} as in algorithm 1
    Control crossover rate CR_{i,g+1} as in equation (2.8)
end while
```

1. DeJong's Function: a unimodal function with global minimum $f(X^*) = 0$; at $X^* = (0, ., 0)$

$$f(x) = \sum_{i=1}^{D} x_i^2; \qquad -10 \leq x_i \leq 10; D = 30 \tag{2.9}$$

2. Axis parallel hyper-ellipsoid: a unimodal function with global minimum $f(X^*) = 0$; at $X^* = (0, 0, ., 0)$

$$f(x) = \sum_{i=1}^{D} i.x_i^2; \qquad -10 \leq x_i \leq 10; D = 30 \tag{2.10}$$

3. Rotated hyper-ellipsoid: a unimodal function with global minimum $f(X^*) = 0$; at $X^* = (0, 0, ., 0)$

$$f(x) = \sum_{i=1}^{D} \Big(\sum_{j=1}^{i} x_j\Big)^2; \qquad -10 \leq x_i \leq 10; D = 20 \tag{2.11}$$

4. Rosenborks valley:a multimodel function with a global minimum $f(X^*) = 0$; at $X^* =$

$(1, 1, ., 1)$. The global minimum is in a narrow flat valley.

$$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \\ -2.048 \leq x_i \leq 2.048, D = 30. \tag{2.12}$$

5. Griwank Function: highly multimodal function a with a global minimum $f(X^*) = 0$; at $X^* = (0, 0, ., 0)$.

$$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) \\ -512 \leq x_i \leq 512; D = 30. \tag{2.13}$$

6. Rastrigin Function: highly multimodal function with a global minimum $f(X^*) = 0$; at $X^* = (0, 0, ., 0)$.

$$f(x) = 10D + \sum_{i=1}^{D} (x_i^2 - 10cos(2x_i)), \\ -5.12 \leq x_i \leq 5.12, D = 20 \tag{2.14}$$

7. Ackley Function: highly multimodal function with a global minimum $f(X^*) = 0$; at $X^* = (0, 0, ., 0)$.

$$f(x) = 20 + e + 20e^{-0.2\sqrt{\sum_{i=1}^{D} x_i^2 / D}} - e^{\frac{\sum_{i=1}^{D} cos(2\pi x_i)}{D}}, \\ -30 \leq x_i \leq 30, D = 20 \tag{2.15}$$

8. Schwelfle Function : a highly multimodel function with a global minimum $f(X^*) = 0$; at

$X^* = (420.9687, 420.9687, ..., 420.9687).$

$$f(x) = \sum_{i=1}^{n} (-x_i sin(\sqrt{|x_i|})) + 418.982887;$$
$$-512 \leq x_i \leq 512; D = 20. \tag{2.16}$$

2.6 Simulation

In this work, the feedback-based self-adaptive DE algorithms, FBjDE-I and FBjDE-II, are applied to the previously described benchmark problems in Section 2.5. Both FBjDE-I and FBjDE-II are compared with classical DE and the self-adapting algorithm jDE. The performance of the three adaptive algorithms jDE, FBjDE-I, and FBjDE-II with the differential evolution DE will be compared in controlling the scale factor F and the crossover rate CR based on the number of function evaluations needed to reach a desired value, value to reach (VTR) [12]. For all the benchmark problems that have a global minimum of $f(X^*) = 0$, the VTR is $VTR = 10^{-5}$; for shifted functions, the global minimum is not at zero, the VTR is $VTR = 10^{-2}$.

The convergence behavior of the three adaptive algorithm jDE, FBjDE-I, and FBjDE-II will be compared based on a fixed number of function evaluation (NFE) .In the simulation, the initialization is as follows: $NP = 5 \times D$, D is the dimension of the problem, $F_i = 0.5$, $CR_i = 0.5$. For jDE all the recommended vales will be used: $\tau_1 = \tau_2 = 0.1$, $F_l = 0.1$,$F_u = 0.9$.

To compare the feedback-based self-adaptive algorithms, FBjDE-I, and FBjDE-II, to jDE, they are initialized with the same upper and lower bounds of the scale factor range $F_l = 0.1$,$F_u = 0.9$. The adaptive algorithms are used to control the scale factor F and the crossover rate CR.

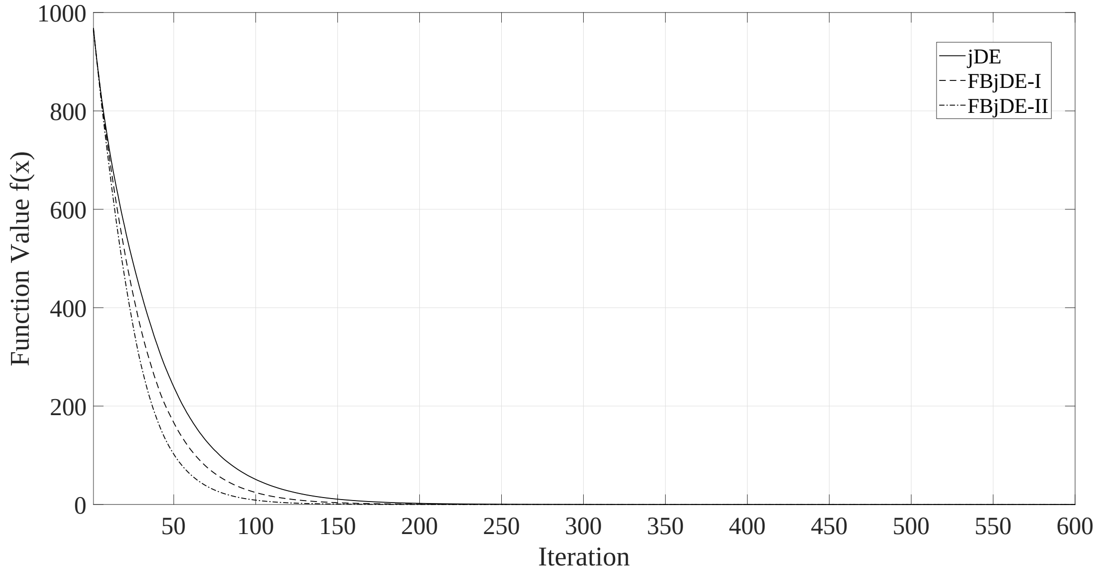

Figure 2.1: De Jongs Function

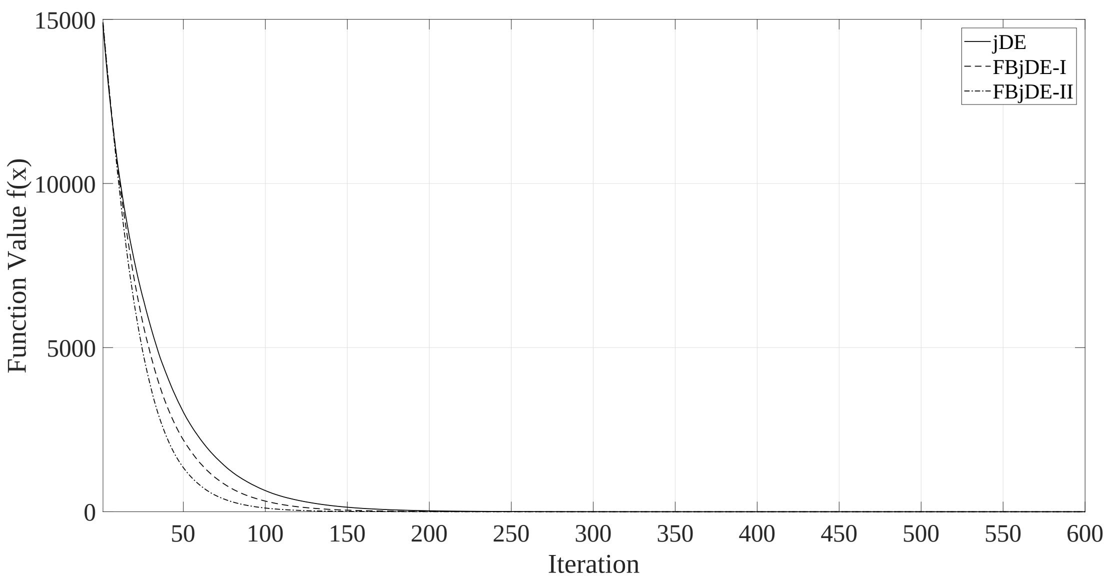

Figure 2.2: Axis parallel hyper-ellipsoid

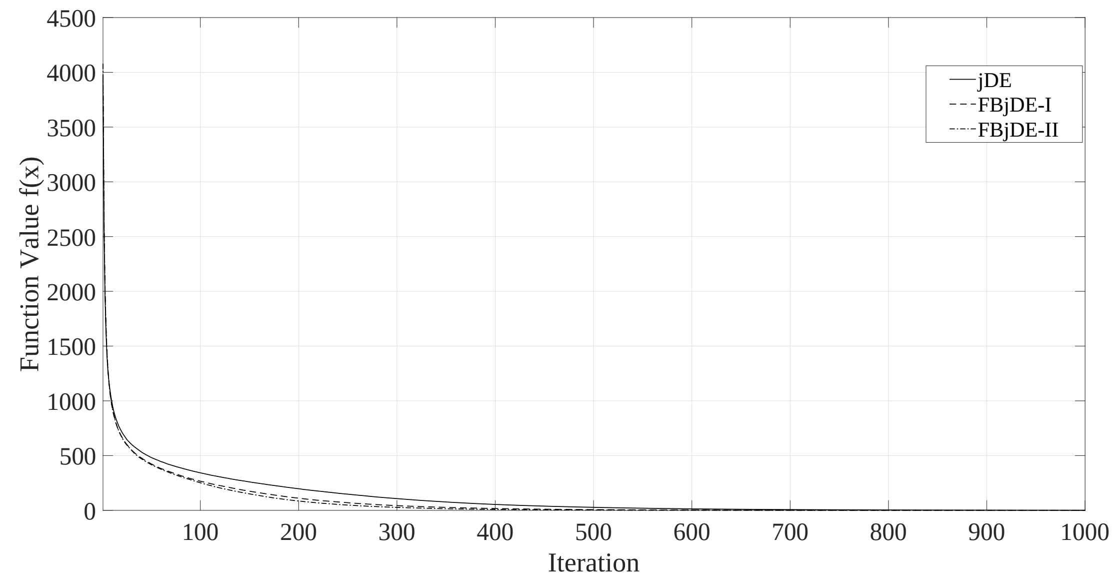

Figure 2.3: Rotated hyper-ellipsoid

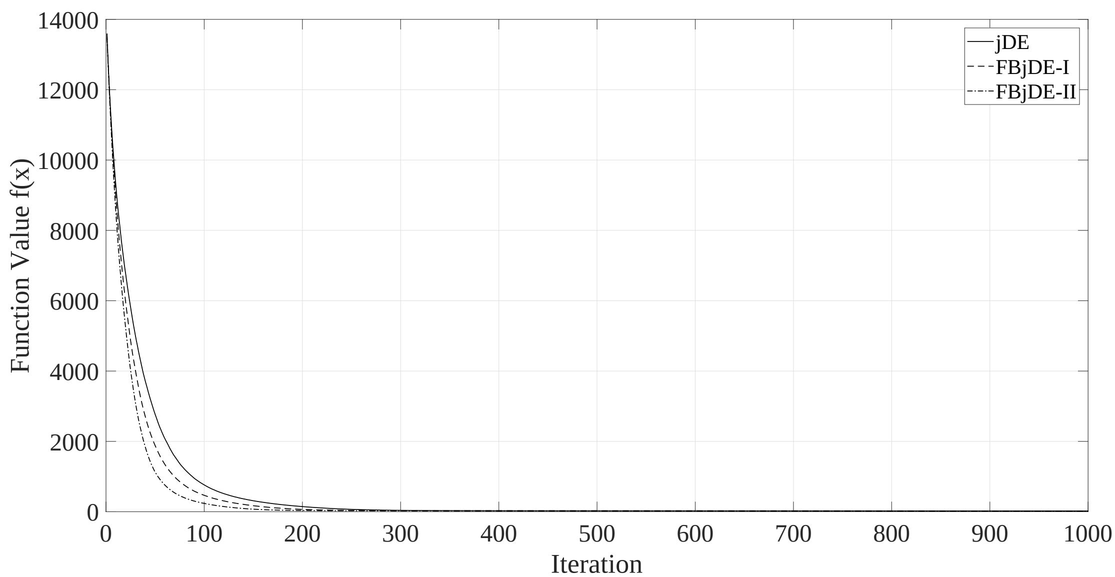

Figure 2.4: Rosenborks valley

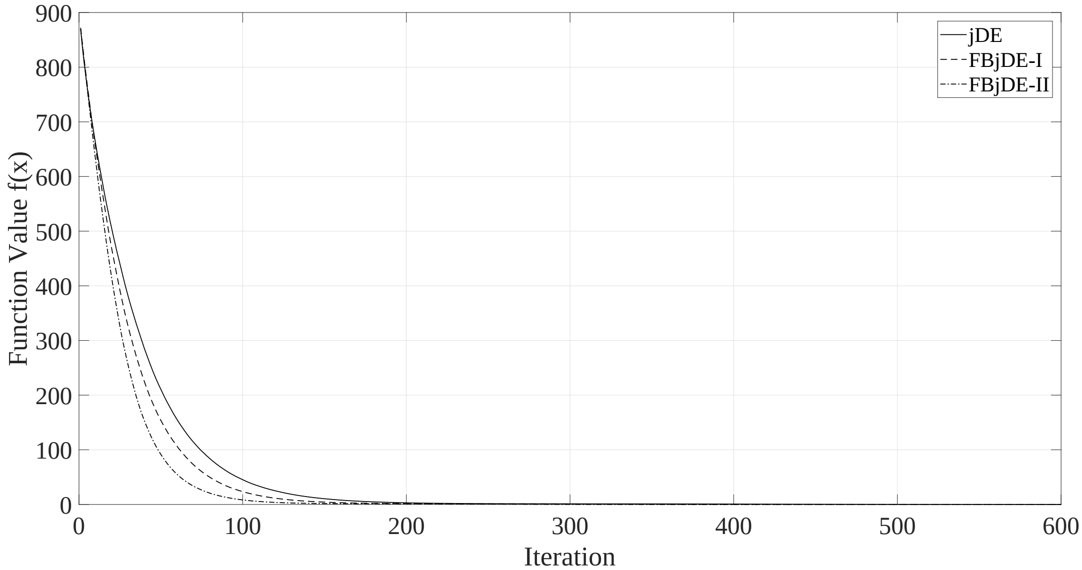

Figure 2.5: Griewank Function

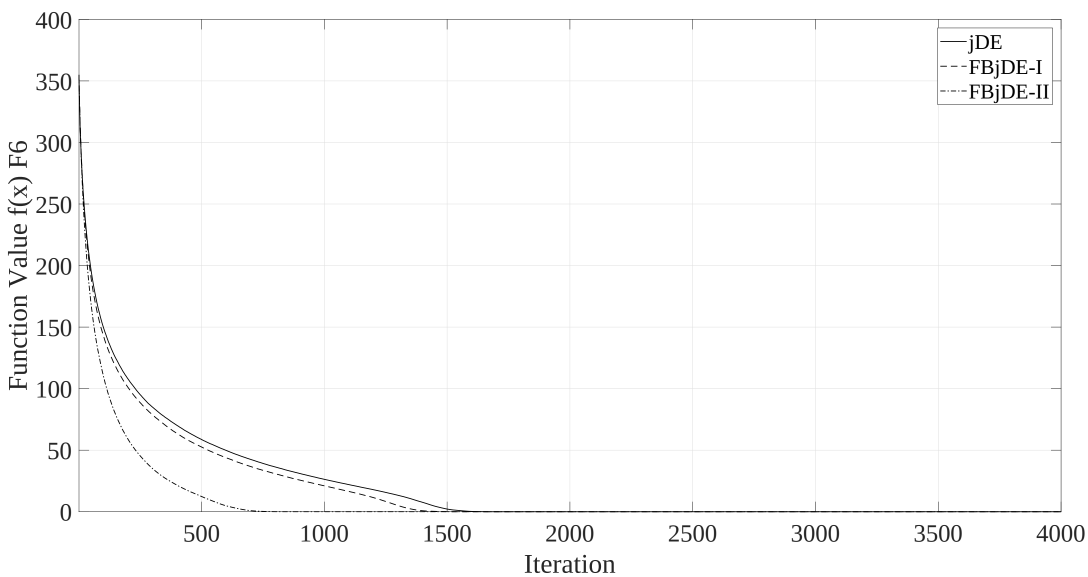

Figure 2.6: Rastrigin Function

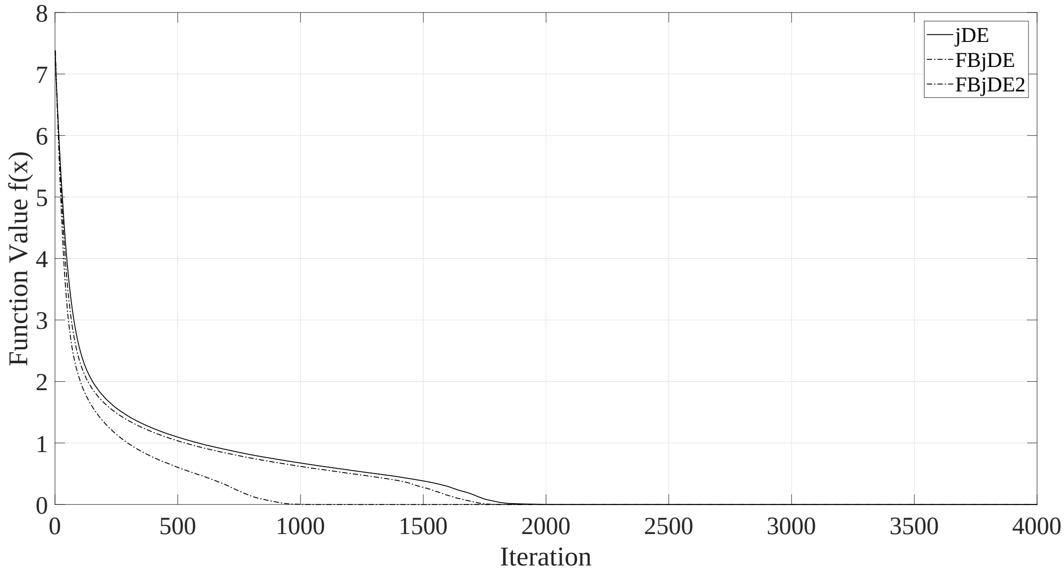

Figure 2.7: Ackley Function

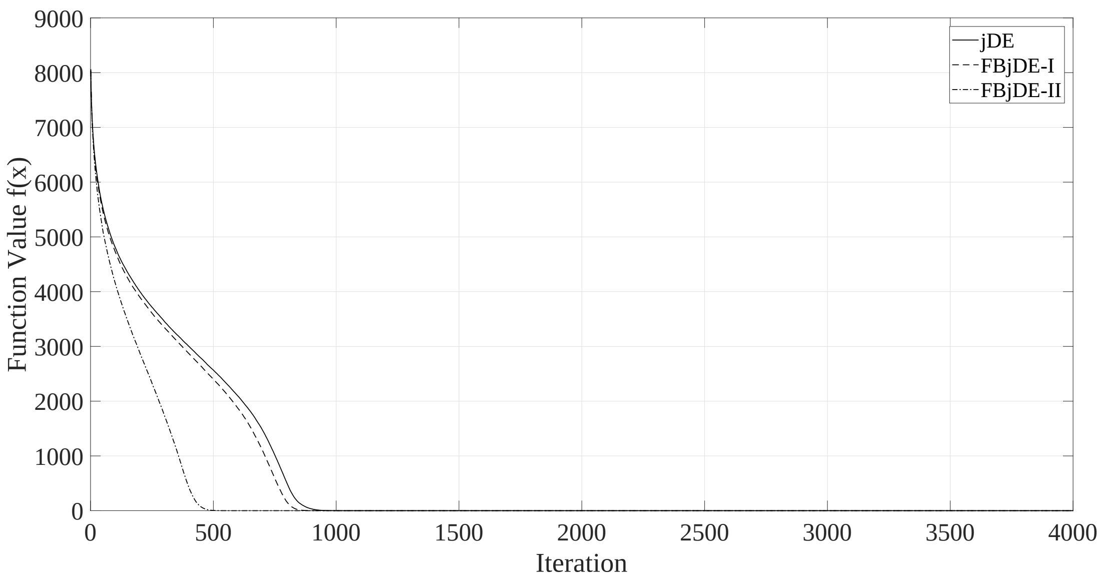

Figure 2.8: Schwefel Function

Table 2.1: Comparison between DE, jDE, FBjDE-I, and FBjDE-II in terms of NFE

BenchMark Problem	D	DE	jDE	FBjDE-I	FBjDE-II
De Jongs Function	30	87618	85245	69873	54921
Axis parallel hyper-ellipsoid	30	99405	97419	79782	63036
Rotated hyper-ellipsoid	20	861374	255646	187768	220314
Rosenborks valley	30	1635663	856554	813555	825255
Griewank Function	30	120111	116004	93132	73938
Rastrigin Function	20	1735586	169606	153186	80096
Ackley Function	20	2077214	200508	184488	103282
Schwefel Function	20	198102	105530	98674	58316

2.7 Discussion

From Table 2.1, it can be clearly concluded that the adaptation algorithms jDE, FBjDE-I and FBjDE-II have outperformed the traditional DE. This shows the importance and the need for controlling the parameters during the evolutionary process.

In comparing FBjDE-I to jDE as in Table2.2, the percentage of improvement in the NFE varies from 5% to 26.5%. The overall improvement of the NFE for the eight benchmark problems is 13.9%. The improvement is measured by of the number of function evaluations needed to reach the desired value with above specified VTR values. The results of comparing the FBjDE-II to jDE are summarized in Table2.3. The overall improvement achieved when using FBjDE-II compared to jDE is 33.8%.

In comparing FBjDE-I to FBjDE-II, we noticed that FBjDE-II outperformed FBjDE-I except in function 3 and function 5. However, for highly multimodel functions, the performance of FBjDE-II is more efficient than FBjDE-I.

Controlling the range of the scale factor has effectively exploited the feedback history of successful scale factors; as a result, more successful scale factor are generated in the evolutionary process. The crossover rate CR_i was controlled in the same manner in both FBjDE-I and FBjDE-II.

Figures 1 to 8 show the comparison of the convergence behavior of the average population averaged over 50 independent runs of the eight benchmark problems. The convergence rate of FBjDE-I and jDE is almost comparable in all the benchmark functions. The convergence rate of FBjDE-II is slightly better than the other two algorithms on functions F_1 to F_5. Figures 6 to 8 show that FBjDE-II has converged faster than jDE and FBjDE-I.

Table 2.2: Comparison between jDE and FBjDE-I in terms of NFE

Function	**D**	**jDE**	**FBjDE-I**	**Improvement**
F1	30	85245	69873	18%
F2	30	97419	79782	18.1%
F3	20	255646	187768	26.5%
F4	30	856554	813555	5%
F5	30	116004	93132	19.7%
F6	20	169606	153186	9.7%
F7	20	200508	184488	8%
F8	20	105530	98674	6.5%

Table 2.3: Comparison between jDE and FBjDE-II in terms of NFE

Function	**D**	**jDE**	**FBjDE-II**	**Improvement**
F1	30	85245	54921	35.6%
F2	30	97419	63036	35.3%
F3	20	255646	220314	13.8%
F4	30	856554	825255	3.6%
F5	30	116004	73938	36.3%
F6	20	169606	80096	52.8%
F7	20	200508	103282	48.5%
F8	20	105530	58316	44.7%

2.8 Conclusion

In this chapter, feedback-based self-adapting control algorithms for differential evolutions are applied to control the differential evolutionary parameters. FBjDE-I and FBjDE-II algorithms have successfully improved the convergence rate and the results of the DE algorithm. The results show that FBjDE-I is comparable to jDE , while FBjDE-II has demonstrated better performance compared to DE, jDE and FBjDE-I.

Chapter 3

Feedback-Based Self-Adaptive Differential Evolution Algorithms for Optimal Power Flow

3.1 Introduction

In electrical power systems, power flow analysis is an essential tool in studying the mechanism of electrical power that is generated, transmitted and consumed in an electrical grid. Power flow is the first step in planning, operation, control, and optimization processes.

Determining the values of the voltages and angles at each bus enables the operators to calculate the real and reactive power needed to be generated from each dispatchable generator. Since the power flow problem is a highly nonlinear problem, nonlinear methods have been used in the power flow problem such as the Gauss-Seidel and Newton-Raphson methods. According to [13], the Newton-Raphson outperformed the Gauss-Seidel in the convergence rate. The aforementioned methods are well explained in [13].

In practice the committed generators need to be optimally dispatached to meet the load demand and to cover the system losses at optimal generation cost. This problem is known as the optimal

Power Flow (OPF). The OPF is a highly nonlinear and constrained optimization problem. In the literature the OPF problem has been addressed using the following methods.

A commonly used method is the linear programming LP [14] [15]. The linear programing could be used when the AC power flow model represented in (3.8) and (3.9) is linearized to a linear model known as DC power flow. The DC model reduces the complexity of the problem and the computation time of the OPF problem at the expense of the accuracy. The linear programming method is also used with a piecewise linearized power flow model which has resulted in better accuracy compared the DC power flow model.

Classical methods, mathematical-oriented methods, have been widely used to solve the AC model of the power flow, such as Newton's method, quadratic programming, and sequential quadratic programming, [16] [17] [18] respectively. In the classical methods used for OPF, the optimization problem is solved using gradient-based search methods starting from an initial guess of the solution.

Even though classical methods produce acceptable results, they have limitations in handling complex features of objective functions and constraints such as non-convexity, discontinuity, and multimodality. Since the OPF is commonly used with other complex optimization problems such as optimal placement and sizing for energy storage, renewable energy resources, and FACTS devices, the use of classical methods is not the suitable choice.

Evolutionary methods such as particle swarm optimization (PSO), genetic algorithms (GA), and differential evolution (DE) provide more flexibility in handling the aforementioned complexities in objective functions and constraints. Moreover, evolutionary algorithms could be expanded for multiobjective optimization problems more efficiently as opposed to classical methods [19]. In the literature the evolutionary algorithms have been used for OPF and have achieved good results as in the flowing references [20] [21] [22]. In evolutionary algorithms, the optimization process starts with multiple points, a population of randomly created solutions, in the search spaces. The population evolve with iterations using evolutionary operators such as crossover, mutation and selection.

Although evolutionary algorithms provide acceptable results, their performance is highly dependent on the parameters of the evolutionary process such as the number of individuals in the population, the crossover rate, and the mutation rate. From practice and based on the no free lunch theorem [23], no specific optimization algorithm is optimal for all optimization problems, it is the case for the control parameters; in other words, within each algorithm, no control parameters set is the best set for all problems. Controlling the parameters of evolutionary algorithms produced better results with faster convergence rates as shown and discussed in Chapter 2.

The feedback-based self-adapting control parameters for differential evolution algorithms, FBjDE-I and FBjDE-II, proposed in Chapter 2 will be used for optimal power flow. The optimal power flow studies are carried over the IEEE 14-bus, IEEE 30-bus, IEEE 57-bus test systems. The results of the FBjDE-I and FBjDE-II are compared with the results of DE and the self-adaptive algorithm jDE. The proposed algorithms resulted in better results and faster convergence speed compared to other algorithms.

3.2 Problem Formulation

The objective of the OPF in an electrical grid is to minimize the cost of the generated power while meeting the operational and security constraints. The cost of the generated electrical power from the committed generators is modeled in (3.1) subject to the generator constrains in (3.2) (3.3), voltage constraints(3.4) (3.5), and meeting load demands and power losses (3.6) (3.7).

$$\text{Minimize} \sum_{i=1}^{G} (a_i P_i^2 + b_i P_i + c_i) \tag{3.1}$$

subject to:

$$P_{gm}^{min} \leq P_{gm} \leq P_{gm}^{max} \tag{3.2}$$

$$Q_{gm}^{min} \leq Qg_m \leq Q_{gm}^{max} \tag{3.3}$$

$$V_m^{min} \leq V_m \leq V_m^{max} \tag{3.4}$$

$$-\pi \leq \delta_m \leq \pi \tag{3.5}$$

$$P_m(\delta, V) + P_{d_m} - P_{g_m} = 0 \tag{3.6}$$

$$Q_m(\delta, V) + P_{d_m} - P_{g_m} = 0 \tag{3.7}$$

The equations of AC power injection at each bus that is used in the solution of the power is given in (3.8) and (3.9). In this work the Newton-Raphson method is used in solving AC power flow model equations.

$$P_m(\delta, V) = |V_m| \sum_{n=1}^{N} |Y_{mn}||V_n| cos(\delta_m - \delta_n - \theta_{mn}) \tag{3.8}$$

$$Q_m(\delta, V) = |V_m| \sum_{n=1}^{N} |Y_{mn}||V_n| cos(\delta_m - \delta_n - \theta_{mn}) \tag{3.9}$$

3.3 Simulation

In this chapter the feedback-based self-adaptive differential evolution algorithms, FBjDE-I and FBjDE-II, explained in 2 along with the self-adaptive differential evolution algorithms jDE and the original DE are used to solve the OPF in three IEEE test cases. To compare the performance of the optimization algorithms, the following procedure is followed.

The optimal power flow using all algorithms is repeated over a number of experiments E_{max}. In each experiment, all algorithms run for a specified number of generations G_{max}. Each algorithm starts with a number of population N_p individuals randomly initialized in the search space. In each experiment, all the optimization algorithms are started from the same initial population in order to reduce the effect of randomness in the evolutionary process. To illustrate, the flow chart depicted in Fig. 3.1 explains the procedure for all algorithms tested in one experiment. In this study $E_{max} = 100$ experiments and $G_{max} = 100$ generations.

For each algorithm, the objective function values of all individuals in N_p at the G_{max} generation over all the E_{max} experiments are sorted to find the minimum (best), median, and maximum (worst) for the both the average population and the best individual in the population. For instance, if the $E_{Max} = 100$, there will be 100 best function values and 100 average population values for each algorithm. From these recorded results we can fairly compare the minimum (best), median, and maximum(worst) values of the algorithms. This procedure is commonly used in comparing optimization algorithms [19]. The average over the total number of experiments is referred to as the mean.

In differential evolution, the control parameters are the number of population NP, the mutation scale factor F, and the crossover rate CR. The number of population could be chosen proportionally to the number of decision variables in the search spaces to achieve good results.

The crossover rate and the mutation scale factor are self-adaptive controlled parameters as explained in Chapter 2. For comparison, all algorithms shown in Fig. 3.1 are initialized with the same number of population, scale factors, and crossover rates. In this case $N_p = 10 \times N$, where N is the number of decision variables; $F = 0.9$ and $CR = 0.1$ for all individuals. The control parameters τ_1 and τ_2 for the jDE algorithm are set as recommended in the original paper $\tau_1 = \tau_2 = 0.1$. The ranges of the scale factor are as follows $f_l = 0.1$ and$f_u = 0.9$.

As explained in Section 2.4, the feed back based self-adaptive differential evolution algorithms, FBjDE-I and FBjDE-II, control the mutation scale factor F_i and the crossover rate CR_i for each individual in the population based on the feed back information about the performance of the previously used the crossover rate for each individual. FBjDE-II takes a further step in controlling the range of the mutation scale factor for each individual as illustrated in the pseudocode presented in Algorithm. 2 in Section 2.4.

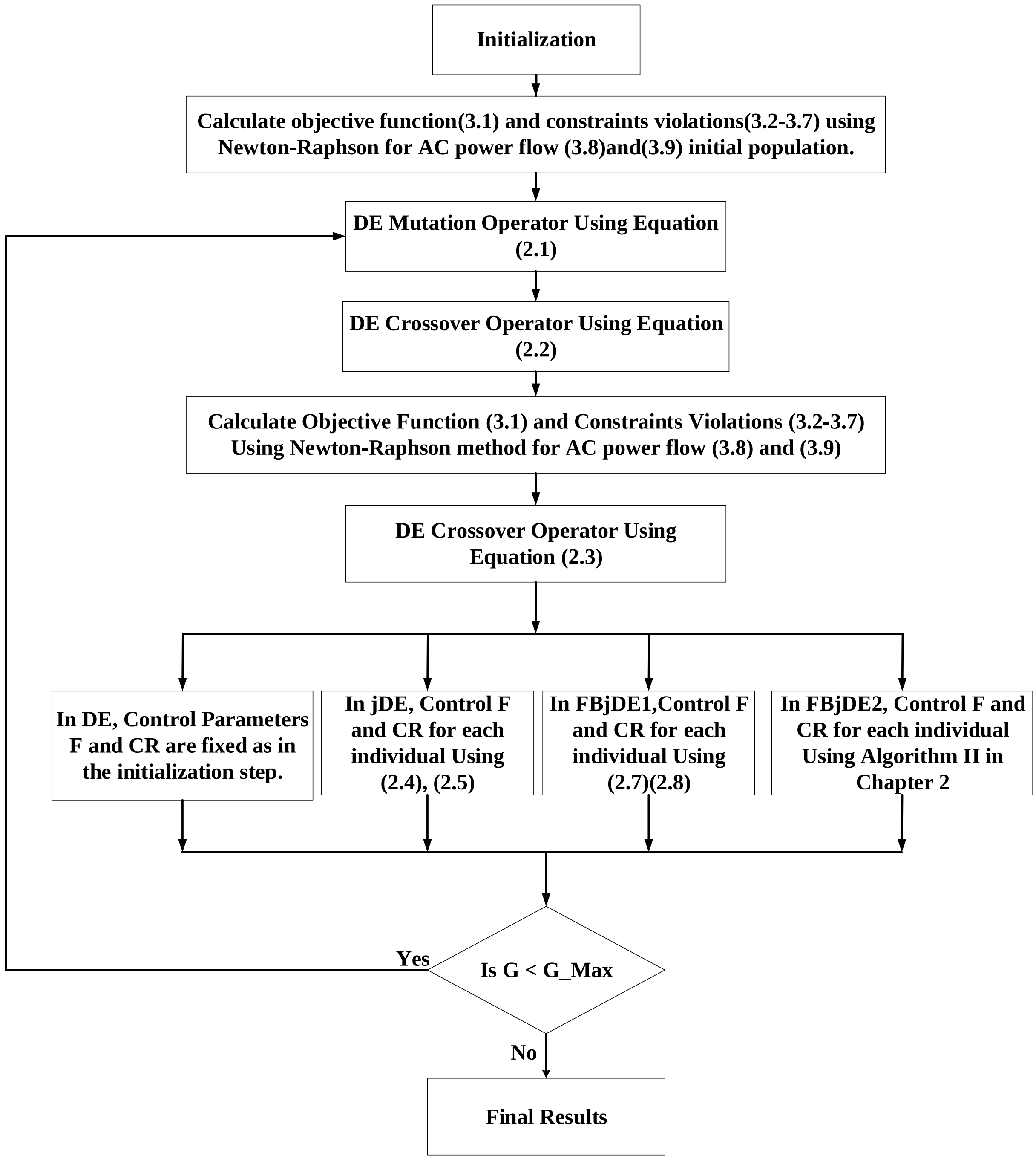

Figure 3.1: Flow chart of DE, jDE, FBjDE-I, FBjDE-II algorithms for OPF

3.4 Results and Discussion

The optimization algorithms, DE, jDE, FBjDE-I, and FBjDE-II, explained in the above section are used to solve the optimal power flow in the following IEEE test case systems: IEEE 14- bus system , IEEE 30-bus system, and IEEE 57-bus system. The cost coefficients a_i, b_i, and c_i of the generated power used in equation (3.1) for all IEEE test systems as well as the generator constraint values in equation (3.2) and (3.3) are taken from the Matpower reference in [24]. The voltage security constraints are within 0.4 of one p.u. for all buses.

For the IEEE 14-bus, the comparison between the convergence rates of the algorithms is depicted in Fig.3.2. The IEEE 14-bus system consists of 14 buses, 20 branches, three transformers, and five generators [25] [24]. The convergence rates shown in Fig.3.2 are averaged over 100 experiments of the best cost of the population. It clearly shows that DE is outperformed by the adaptive algorithms jDE, FBjDE-I, and FBjDE-II. The feedback self-adaptive DE algorithms FBjDE-I, and FBjDE-II show better convergence rates compared to other algorithms.

The mean and the best OPF values of the population over 100 experiments using all algorithms for IEEE 14-bus system are compared in Tables 3.1 and 3.2, respectively. The best, median and worst final results over the 100 experiments of the best cost as well as the average cost are presented in Tables 3.1 and 3.2, respectively.

Table 3.1: Comparison of Mean Cost of the Population over 100 Experiments IEEE 14-bus

Algorithm	Worst of the Mean	Median of the Mean	Best of the Mean
DE	8370.9729	8112.2398	8081.8343
jDE	8091.7723	8081.5715	8081.5262
FBjDE-I	8083.0628	8081.5447	8081.5266
FBjDE-II	**8082.3251**	**8081.5442**	**8081.5261**

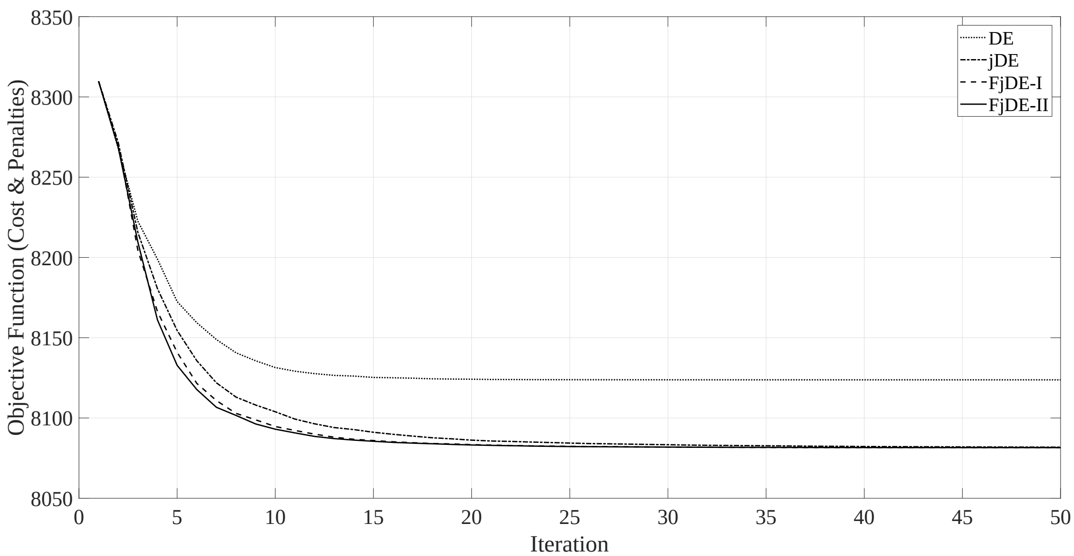

Figure 3.2: Mean of the Best Cost of the Population over 100 Experiments IEEE 14-bus

Table 3.2: Comparison of the Best Cost of the Population over 100 Experiments IEEE 14-bus

Algorithm	Worst of the Best	Median of the Best	Best of the Best
DE	8370.9729	8112.2396	8081.8342
jDE	8089.1285	8081.5357	8081.5251
FBjDE-I	8082.8258	**8081.5288**	8081.5250
FBjDE-II	**8081.9719**	8081.5296	**8081.5251**

The IEEE 30-bus system consist of 30 buses, 41 branches, four transformers, and six generators [25] [24]. Fig.3.3 compares the convergence rate of the algorithms by plotting the mean best cost of the population over 100 experiments of each algorithm for the IEEE 30-bus. The comparison between the final results of the mean and the best OPF values of all algorithms for this test case system are listed in Tables 3.3 and 3.4, respectively.

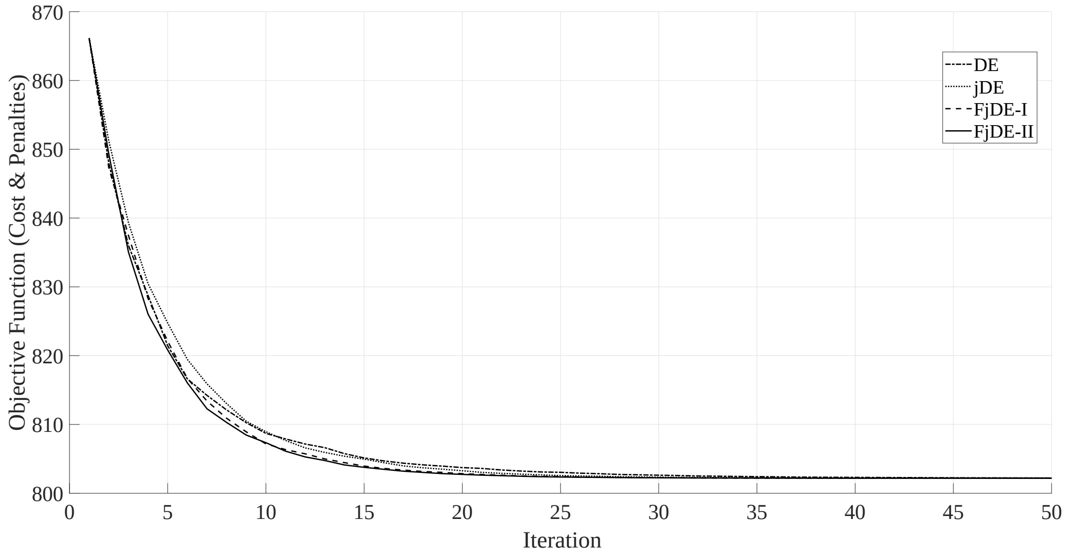

Figure 3.3: Mean of the Best Cost of the Population over 100 Experiments IEEE 30-bus

Table 3.3: Comparison of the Mean Cost of the Population over 100 Experiments IEEE 30-bus

Algorithm	Worst of the Mean	Median of the Mean	Best of the Mean
DE	802.9079	802.4129	802.2641
jDE	802.2823	802.1990	802.1846
FBjDE-I	802.2150	802.1872	802.1843
FBjDE-II	**802.2106**	**802.1871**	**802.1841**

Table 3.4: Comparison of the Best Cost of the Population over 100 Experiments IEEE 30-bus

Algorithm	Worst of the Best	Median of the Best	Best of the Best
DE	802.3179	802.2121	802.1859
jDE	802.2132	802.1850	802.184108
FBjDE-I	**802.1876**	802.1843	802.184109
FBjDE-II	802.1919	**802.1842**	**802.184107**

The last test case is the IEEE 57-bus. It contains 57 buses, 80 branches, 17 transformers, and seven generators [25] [24]. As for the previous test case systems, the comparison is based on the convergence rates as in Fig.3.4 and on the comparison between the final results of the mean and best costs of the population over 100 experiments as Tables 3.5 and 3.6. FBjDE-I and FBjDE-II demonstrated better converges rates as in Fig.3.4 as well as better results as in Tables 3.5and 3.6, respectively.

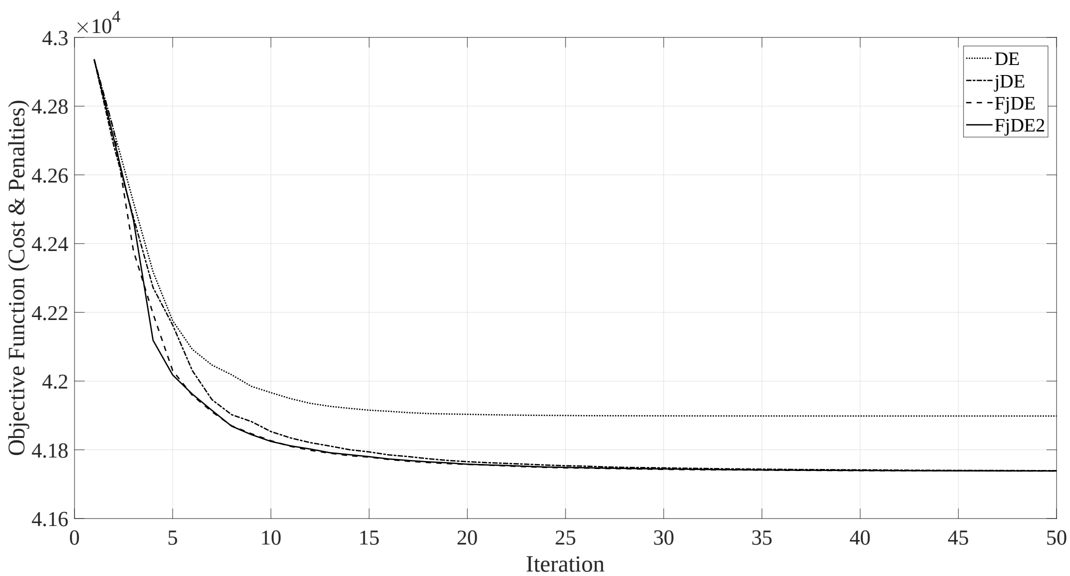

Figure 3.4: Mean of the Best Cost of the Population over 100 Experiments IEEE 57-bus

Table 3.5: Comparison of the Mean Cost of the Population over 100 Experiments IEEE 57-bus

Algorithm	Worst of the Mean	Median of the Mean	Best of the Mean
DE	42891.7011	41820.3454	41745.9749
jDE	41779.3649	41742.2904	41738.0285
FBjDE-I	**41745.2177**	41739.9688	41738.0823
FBjDE-II	41750.1667	**41738.7995**	**41737.8638**

Table 3.6: Comparison of the Best Cost of the Population over 100 Experiments IEEE 57-bus

Algorithm	Worst of the Best	Median of the Best	Best of the Best
DE	42891.6723	41820.3429	41745.9735
jDE	41758.1262	41738.6577	41737.8522
FBjDE-I	**41740.0185**	41738.2268	41737.8287
FBjDE-II	41746.4469	**41738.1402**	**41737.8067**

3.5 Conclusion

By comparing the performance of the adaptive algorithms, jDE, FBjDE-I, and FBjDE-II, to the DE algorithm, it is evident that controlling the parameters of the differential evolution algorithm has resulted in significant improvement not only on the results but also on the convergence rate of the optimal power flow solutions.

Feedback-based self-adapting differential evolution algorithms FBjDE-I ,and FBjDE-II for optimal power flow have demonstrated better performance compared to jDE in terms of convergence rate and results especially in large systems. The performance of FBjDE-I ,and FBjDE-II are very comparable in terms of the convergence rate.

In the largest test case used in this study, IEEE 57-bus, the results of FBjDE-II are slightly better than the results of the FBjDE-I. From Chapter 2, it was noted that the performance FBjDE-I and FBjDE-II are comparable in solving benchmark problems. It was also observed that the results of FBjDE-II are better than the results of FBjDE-I in complex multimodel problems. These observations suggest that FBjDE-II should be used in solving complex and large scale problems.

Chapter 4

Multiobjective Optimal TCSC Placement and Sizing for Enhancing Network Loadability

4.1 Introduction

This chapter presentes a mutiobjective-based approach to optimally size and allocate FACTS devices to enhance electrical network loadability. The proposed method has been applied to commonly used FACTs devices, hyristor-controlled series capacitor (TCSC).

TCSC devices have been utilized for enhancing transmission network loadability. The technical benefit of the TCSC devices comes at the expense of their high investment cost. In this paper, a new approach is proposed to provide a profound insight into the compromises between technical and economical aspects of installing TCSC devices in a transmission network.

The proposed approach is a multiobjective optimization based algorithm used to maximize the loadability of the network and to minimize the investment cost of installing TCSCs under secured

The content of this chapter has been reproduced with permission from *Fares T. Alharbi, Saleh Almasabi, and Joydeep Mitra, Enhancing Network Loadability Using Optimal TCSC Placement and Sizing, 2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), Denver, CO,2018.*

operation conditions. The results provide an efficient set of nondominated solutions to maximize the loadability at minimal cost for the decision-making process.

The IEEE 118-bus system is used as a test case to validate the effectiveness of the proposed method. Furthermore, the results are analyzed and compared with available results in the literature.

The rest of the chapter is organized as follows: the multiplicative algorithms are briefly summarized in Section. 4.2. Section 4.3 introduces FACTS devices, describes their effect on the system, and illustrates the modeling of the TCSCs. The objective functions and constraints are presented in Section 4.5. The simulation of the proposed approach is shown in Section 4.6. Results and discussion are in Section 4.7, followed by conclusions in Section 4.8.

4.2 Multi-Objective Optimization Algorithms

Evolutionary algorithms (EA), such as genetic algorithm, particle swarm, and differential evolution have been commonly used in power system computation [1, 26, 27]. These algorithms were originally used for optimizing single objective problems. Single objective algorithms were commonly used to solve multiobjective problems by assigning weights to the objective functions. Depending on the weight of each objective a potential optimal solution could be found.

There are, however, certain difficulties with the wighted sum approach. First, the weights used to scale the contradicting objective functions need to be properly studied and chosen beforehand. Choosing multiple weights requires more computations for each set of weights. Second, wighted sum-based algorithms are not able to discover the non-convex parts of the Pareto-optimal solutions [19].

For multiobjective problems, evolutionary algorithms are a more suitable in generating well compromised efficient solutions (a set of non-dominated solutions in the objective spaces); their corresponding values in the decision space are called Pareto optimal solutions [19].

A nondominated sorting genetic algorithm (NSGA-II) has demonstrated an efficiency in solving multiobjective functions, i.e. two objective functions. Thus, in this study, the NSGA-II [28] is used to solve the optimal placement and sizing of TCSCs to enhance the loadability of the system at minimum investment cost.

In multiobjective algorithms, the dominance principle is used as a selection operator to favor solutions that can have better compromise between the objective functions, which are called non dominated solutions. According to [19], a solution x dominates a solution y, if the solution x is not worse than y in all objectives, and it has a better value than y in at least one objective.

The NSGA-II emphasizes the non-dominated solutions, and uses a diversity preservation mechanism to explore the entire search space. It also has an elitism preservation algorithm [28]. The proposed procedure using NSGA-II is illustrated in the flow chart in Fig. 4.3. In NSGA-II, after creating a uniform random population N in the decision space and evaluating the objective functions, the population are sorted according to the level of domination in objective space. That would classify the population as a number of sets called fronts.

To maintain the diversity, solutions in each front are ranked according to their crowding distance, which gives higher rank to solutions that are apart from each other in the objective space. Once the population is sorted in non dominated fronts and ranked according to the crowding distance within each front, a tournament selection operator is used to choose parents for the mating pool.

The NSGA-II algorithm handles constraints violation in the the non-dominated sorting process; it favors feasible over all infeasible solutions. In the infeasible regions it gives higher rank to solutions that have smaller constraints violations. This will force the infeasible solutions to converge to a feasible area or to the constraints boundary, where most likely the actual Pareto-optimal solutions are.

Following this, the selected parents undergo genetic crossover and mutation operators to produce the offspring population. The resulting population of the current parents and their offspring are sorted again based on dominance and ranked according to crowding distance again.

The first nondominated sets are then selected to pass to the next generation. When the size of the first nondominated fronts is larger than the population size N, only solutions that have better, higher, crowding metrics are selected to pass to the next generation [28].

4.3 Flexible AC Transmission System devices (FACTS)

FACTS devices are commonly used to enhance the capabilities of the transmission network. Besides enhancing the system loadability, FACTS devices are utilized to improve both transient and steady state stability, to dampen power oscillation, and to limit short circuit currents [29]. The use of FACTS devices can allow controlling the voltage and angle differences between two connected buses as well as regulating active and reactive power flow.

Different FACTS devices are used to control different parameters [29]. The thyristor-controlled series capacitor (TCSC) controls the effective reactance of the transmission line; the thyristor-controlled phase shifting transformer (TCPST) adjusts the angle difference between two connected buses; the thyristor-controlled voltage regulator (TCVR) is used to control the voltage difference. The static var compensator (SVC) and the static synchronous compensator (STATCOM) control the reactive power and regulate the voltage magnitudes at the terminals. The unified power flow controller (UPFC) is used to control the active and reactive power flow and to regulate the voltage magnitudes at the same level.

FACTS devices could be classified, according to their type of compensation, as shunt controlled, series controlled, and combined (shunt-series controlled) [29]. For instance, SVC and STATCOM are shunt controlled, whereas TCSC, TCVR, and TCPST are series controlled FACTS devices. The unified power flow controller (UPFC) is a combined controlled FACTS device. Series controlled FACTS devices are the most commonly used type [30]. In particular, the TCSC is widely used among them due to its performance and relatively low cost [30]. Series controlled FACTS are utilized to control the effective transmission line impedance X_l, which has a direct effect on the real and reactive power transmitted from a sending bus to a receiving bus in the transmission network [29]. Compared to the SVC, which is shunt controlled, TCSC has demonstrated more effectiveness and efficiency in maximizing the system loadability [31]. The TCSC was the second best for enhancing the loadability next to the UPFC, but has a better investment cost than the UPFC [32].

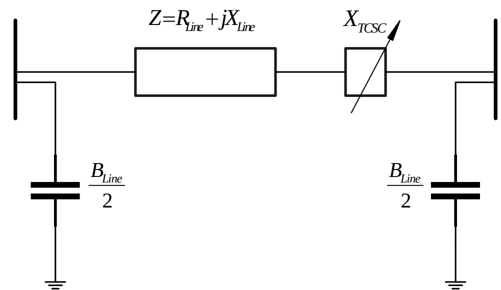

Figure 4.1: TCSC Model

When a TCSC device is installed on a transmission line, it could be considered as a capacitor or an inductor series compensation depending on its impedance jX_{TCSC}. In networks where the transmission lines are represented as a π model, installing the TCSC device in the line is modeled as variable reactance as depicted in Fig.4.1. This model affects the corresponding elements in the admittance matrix of the system.

4.4 Related Work

In a deregulated and highly competitive market with increasing load demands, transmission lines are vulnerable to being overloaded beyond their thermal limits. Overloading transmission lines will not only increase losses, but also may drive the system into insecure operating conditions. Flexible AC transmission system (FACTS) devices have been effectively utilized to enhance transmission networks loadability. The benefits of the FACTS devices are realized if they are optimally installed in the network with proper sizes [29].

The problem of optimal location and sizing of FACTS devices is a highly constrained nonlinear problem due to the nonlinearity of the AC power flow, the model of FACTS devices, and the cost function model of the FACTS devices. Linear and nonlinear programming [33] as well as heuristic methods [34] have been used to overcome the complexity of the problem. Heuristic methods are the most popular procedures used in solving the optimal placement and sizing of FACTS devices due to their effectiveness in handling mixed variables, discontinuities, and nonlinear objectives and constraints [19].

The best locations and respective sizes of multi-type FACTS devices were determined using a genetic algorithm (GA) to maximize the system loadability [35] and to enhance system security [36]. In [32], the authors proposed a multiobjective weighted sum approach based on particle swarm optimization (PSO) to maximize the loadability of the system and to minimize the cost of installing multi-type FACTS devices. Optimal placement of multi-type FACTS devices with a graphic user interface was proposed in [34] for a selected number of FACTS devices, by the user, to maximize the power system loadability.

A self-adaptive firefly algorithm for multi-type FACTS placement was presented in [37] to minimize real power loss and to enhance voltage stability. The study in [30] proposed a multiobjective adaptive differential evolution (ADE) algorithm based on a weighted sum approach to minimize the following: the real and reactive power losses over the transmission lines, voltage deviation on the buses, the installation cost of the TCSCs and the number of TCSCs. In [31], the effect of optimizing the locations of the TCSCs and SVCs was studied to maximize the loadability in both normal and contingency operations. A real genetic algorithm was used to optimize the location and the settings of the TCSC and the SVC. According to [31], TCSCs are more effective in improving the loadability under both normal and contingency conditions.

In the aforementioned studies, the number of FACTS devices is predetermined as in [31, 32, 34–36], and the optimization is focused on the locations and the parameter settings of the FACTS devices. The multiobjective nature of the problem was addressed using a weighted sum approach, a compromised technique, for this multiobjective problem in [30, 32, 37].

The allocation of FACTS devices was addressed as a multiobjective algorithm in [38] to optimally allocate multi-type FACTS to enhance the system security and minimize the investment cost of FACTS devices. However, the optimization was done for a predetermined number of FACTS devices and applied to a small test system. An epsilon-constrained method based on mathematical programming is introduced in [39] to handle the multiobjective nature of the optimal allocation problem with a fuzzy logic procedure for the decision-making process. In all noted references, the loadability was tested on discrete values of loading margins.

Considering related work in the available literature, a profound insight is needed into the trade-off between maximizing the system loading margin and the investment cost of the FACTS devices using intelligent computation methods. To this end, this paper proposes a new multiobjective approach to maximize the power system loadability using TCSC devices and to minimize the investment cost of the TCSC units under secured operating conditions. The nondominated sorting genetic algorithm II, NSGA-II [28], is used to generate the nondominated set of solutions in this work. The number, locations, and sizing of TCSC units, as well as the loading margin of the system, are optimized simultaneously in order to determine the efficient set of nondominated solutions. The presented method provides a flexible and efficient procedure for handling the constrained mixed integer decision variables. In this work, the loading margin is treated as both an objective and a decision variable to explore the search space more effectively.

4.5 Problem Formulation

The mathematical formulation of the objective functions are as follows. First, enhancing the loadability (λ) of the transmission network is tested by uniformly increasing the active and reactive power demands on all buses as given in (4.1) and (4.2).

$$P_{d,i} = \lambda \times P_{d,i,0} \tag{4.1}$$

$$Q_{d,i} = \lambda \times Q_{d,i,0} \tag{4.2}$$

In the above equations, $P_{d,i,0}$ and $Q_{d,i,0}$ are the real and reactive power demands at bus i under normal conditions, respectively.

$$P_{g,i} = \lambda \times P_{g,i,0} \tag{4.3}$$

The power produced by each generator is also scaled by λ to distribute the extra load demands and the losses on the generators in proportion to their sizes (4.3); where $P_{g,i}$ is the power generated at the PV bus i. Even though for small λ load demands could be supplied by the generator at the slack bus, the objective is to inject power from all generators to allow testing all transmission network branches.

$$\max \lambda \tag{4.4}$$

In equation (4.4), λ is the system loadability factor with a base case of $\lambda = 1$.

Second, minimizing the total investment cost C_T of N_{TCSC} TCSC units is minimized as given in equation (4.5) [32, 33].

$$\min C_T = \sum_{j=1}^{N_{TCSC}} S_{TCSC,j} \times C_{TCSC,j} \times 1000 \qquad \$ \tag{4.5}$$

$$C_{TCSC,j} = 0.0015 S_{TCSC}^2 - 0.713 S_{TCSC} + 153.75 \tag{4.6}$$

The investment cost of installing N_{TCSC} units, given by (4.5), depends on the reactive power capacity of each TCSC unit (4.7) and on their total number.

$$S_{TCSC,j} = \mathrm{Im}(I_j^2 \times X_{TCSC,j}) \tag{4.7}$$

In equation (4.7), $S_{TCSC,j}$ is the capacity of the TCSC unit in MVAR on transmission line j. The compensation level of the TCSC device is constrained by the reactance of the transmission line as in (4.8) [32, 35].

$$-0.8 \times X_{line} \leq X_{TCSC} \leq 0.2 \times X_{line} \tag{4.8}$$

To maintain secure operation while maximizing the system loading margin, the voltage magnitudes are limited within 5% of one per unit of the nominal values (4.9). The power flow over a transmission line is restricted to its rated capacity as in (4.10). The rest of the power flow constraints are presented in (4.11-4.14); V_i and δ_i are the voltage magnitude and angle, respectively, at bus i. P_i and Q_i are active and reactive power injections at bus i, which are calculated using (4.15) and (4.16), respectively.

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{4.9}$$

$$S_l \leq S_l^{max} \tag{4.10}$$

$$-\pi \leq \delta_i \leq \pi \tag{4.11}$$

$$Q_{i,g}^{min} \leq Q_{i,g} \leq Q_{i,g}^{max} \tag{4.12}$$

$$P_i(\delta, V) + P_{i,d} - P_{i,g} = 0 \tag{4.13}$$

$$Q_i(\delta, V) + Q_{i,d} - Q_{i,g} = 0 \tag{4.14}$$

$$P_i(\delta, V) = |V_i| \sum_{j=1}^{N} |Y_{ij}||V_j| \cos(\delta_i - \delta_j - \theta_{ij}) \tag{4.15}$$

$$Q_i(\delta, V) = |V_i| \sum_{j=1}^{N} |Y_{ij}||V_j| \sin(\delta_i - \delta_j - \theta_{ij}) \tag{4.16}$$

4.6 Simulation

In this work the simulation is applied on the IEEE 118-bus test system, the data can be found in [40, 41]. The IEEE 118-bus test system has a total number of 596 constraints in the optimization problem. Some constraints must be handled in the power flow algorithm to make sure that the power flow converges for the suggested solutions; other constraints are handled using the optimization algorithm. The total number of equality and inequality constraints of the IEEE 118-bus system are shown in Table. 4.1.

Table 4.1: IEEE 118-bus Constraints

Constraint Type	Optimization	Power Flow
Inequality constraints	304	54
Equality constraints	0	238
Total number of constraints	304	292

In the optimization problem, the decision variables are the loading factor λ, the number of TCSC units N_{TCSC}, the optimal location $\in [1, N_L]$, and respective size X_{TCSC} of each TCSC unit. The locations and the number of the TCSCs units are considered as integer numbers, while the loading margin and the size for each device is represented in real continuous variables.

For the continuous variables, λ and X_{TCSC}, the simulated binary crossover operator (SBX) [42] is used to perform the crossover; the crossover probability is $P_c = 0.9$ and the distribution index of the (SBX) operator is $\eta_c = 20$. The polynomial mutation operator is used to perform the mutation over the continuous variables [43] with a mutation probability of $P_m = 0.3$ and a polynomial order of $\eta_m = 20$.

The integer variables (i.e the number of TCSC units and their locations) are decoded in the binary space to permit the use of genetic operators, such as crossover and mutation. In the binary decoded space of integer variables, a double site crossover technique is applied with the probability of $P_c = 0.9$, and a bit wise mutation of a probability of $P_m = 0.3$.

Since the genetic operators have some mutation probabilities, unrealistic offspring solutions could be generated. For instance there can be a candidate location Loc_i for the TCSC with zero size $X_{TCSC,i} = 0$, or a size for the a TCSC unit $X_{TCSC,i}$ where there is no location ($Loc_i = 0$). To avoid such cases, the recombination of population is modified as follows:

1) The number of TCSC variables N_{TCSC} controls the total number of TCSC candidate locations.

2) There can not be a candidate location Loc_i, where the size of the TCSC parameter $X_{TCSC,i}$ is zero, and vice versa.

3) Only one TCSC device can be installed at a line; no repeated locations in the same individual are allowed.

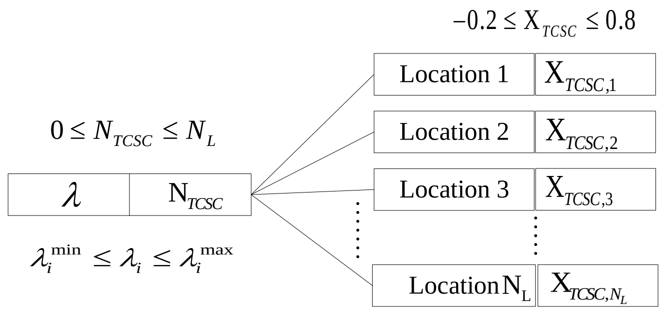

Figure 4.2: Constructing an Individual in the Population

The optimization starts with a population of candidate solutions, individuals. Each individual has a loading factor λ, number of TCSC units N_{TCSC}, locations, and respective sizes for all TCSC units in the individual as shown in Fig. 4.2.

A full AC power flow using NewtonRaphson algorithm is used to evaluate the power flow for each candidate solution; then the loadability of the system and the investment cost of the TCSCs are evaluated as in (4.4) and (4.6).

$$\lambda_i^{\min} = \begin{cases} \lambda_{i-1}^{\min} + \tau, i > 1 \\ 1, i = 1 \end{cases} \tag{4.17}$$

$$\lambda_i^{\max} = \begin{cases} \lambda_{i-1}^{\max} + \tau, i > 1 \\ \lambda_{i=1}^{\min} + \tau, i = 1 \end{cases} \tag{4.18}$$

Optimizing over a continuous range of λ has resulted in less diverse solutions due to the nature of the mixture and the large diversity of the decision variables. To overcome this problem and better explore the search space, the optimization is controlled over the loadability domain λ . A control variable $\tau \in [0, 1]$ is introduced to restrict the optimization over the loadability domain. The optimization process is divided into N_i segments, where i is the segment's index. More computations are needed for the smaller control variable τ. In this work the control variable is

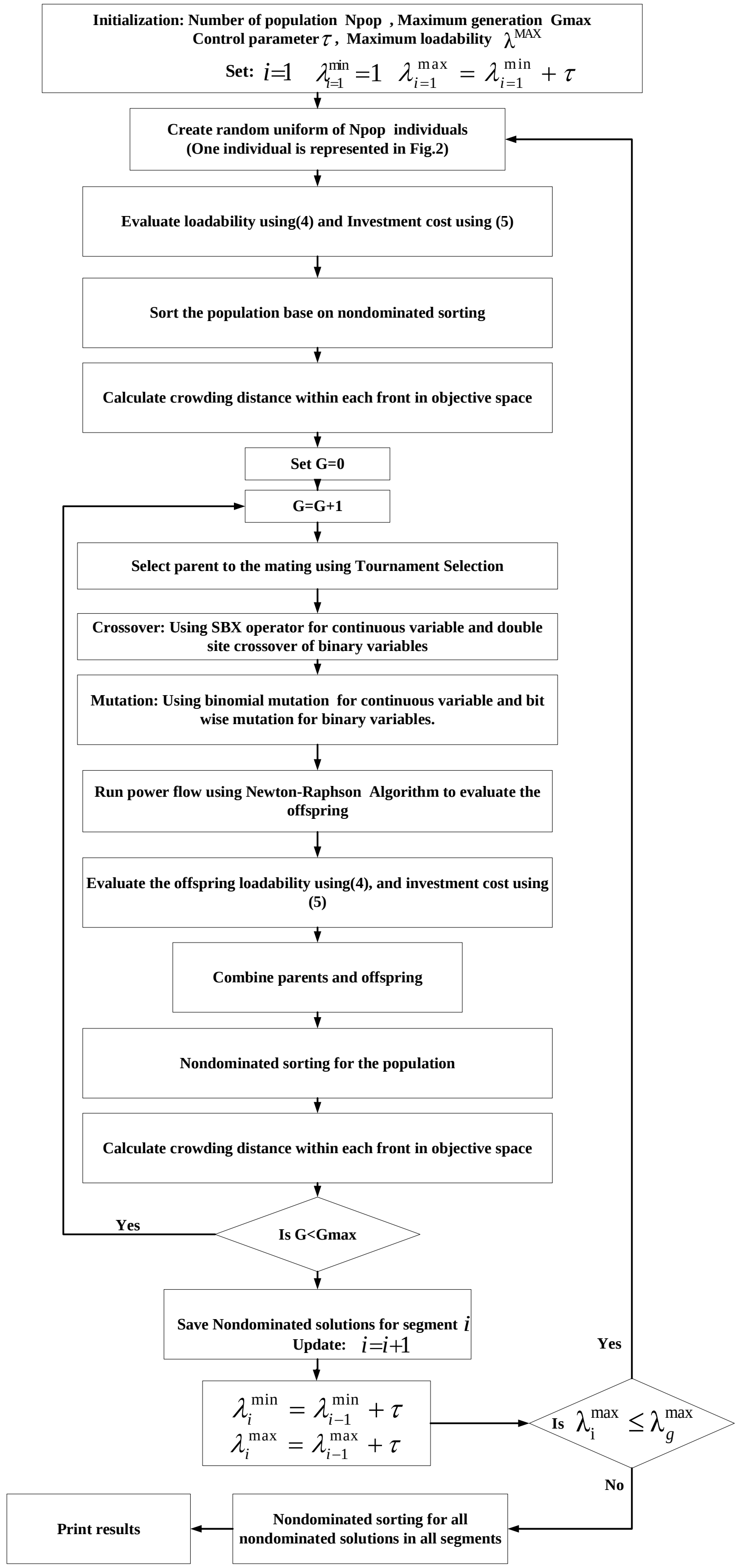

Figure 4.3: Flow chart of the proposed approach

chosen as $\tau = 0.2$, and $\lambda_g^{max} = 2$.

The minimum and maximum loadability in each segment are determined depending on τ as in (4.17) and (4.18). The flow chart of the proposed method is depicted in Fig. 4.3.

4.7 Results and Discussion

The proposed procedure optimizes the loading factor, the number, locations and sizes of TCSC units for maximum loadability (4.4) at minimal cost (4.5). The maximum loadability that could be achieved without needing to install TCSC units is 1.089; where the total load is increased by 8.9% without a congestion. The obtained results are compared with the available results in the literature as in Table 4.2. References [32, 35] have reported the results of the IEEE 118-bus system with operating conditions similar to the test system used in this paper. In [32, 35] and other reported work, the increment of the loadability is discrete, at 0.1 increments. Because the loadability was not tested over over $\lambda \in [1, 1.1)$, others assumed FACTS devices are needed at $\lambda = 1.1$; however, considering the loadability as a continuous variable showed that FACTS devices are needed at $\lambda = 1.089$. The maximum loadability with TCSCs has increased to 1.73 of normal loading capacity as compared to 1.35 in [32, 35]. Moreover, the presented results have effectively utilized fewer TCSCs to achieve a better loading margin at a lower investment cost as shown in Table 4.2.

Table 4.2: IEEE 118-bus Maximum Loadability

Criteria	Ref. [35]	Ref. [32]	Proposed Approach
Max λ Without TCSCs	1	1	1.089
Max λ With TCSCs	1.35	1.35	1.737
Number of TCSCs for Max λ	30	32	10
Cost of TCSCs for Max λ	–	$\$15.1 \times 10^6$	$\$3.765 \times 10^6$

The efficient set of nondominated solutions in the objective space is shown in Fig. 4.4. For some loadability conditions there are no efficient solutions in the Pareto efficient set, Pareto front.

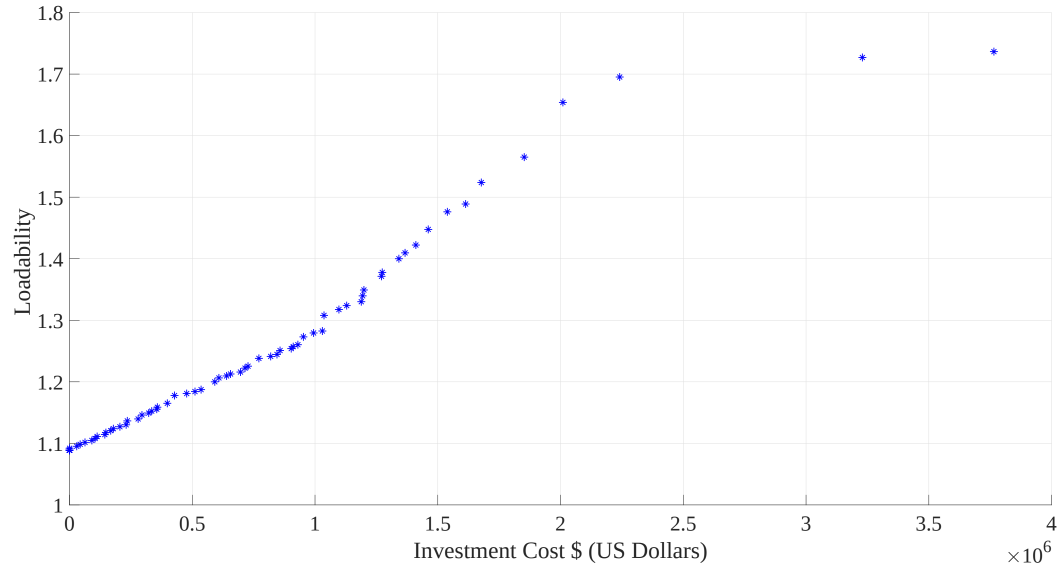

Figure 4.4: Pareto Efficient Set

This observation implies that solutions for these specific loading margins are either unfeasible or are dominated by other feasible solutions, which provide higher loadability with better investment cost.

In considering the trade-off between the efficient solutions in Fig. 4.4, increasing the loadability beyond $\lambda = 1.7$ results in low gain in terms of loadability and a huge sacrifice in the investment cost. Therefore, increasing loadability beyond 1.7 using TCSCs is not recommended.

The relationship between the number of TCSC units and the achieved loadability in the efficient set of solutions in the Pareto front are presented in Fig. 4.5. Different loadability conditions could be achieved with the same number of TCSC units. However, they are nondominated solutions in terms of loadability and the investment cost. As a result, a specific loadability margin could be achieved using a different number of TCSC units at different locations with different settings. This observation could be significant in the decision-making and planning process since some locations may have infrastructural or environmental constraints.

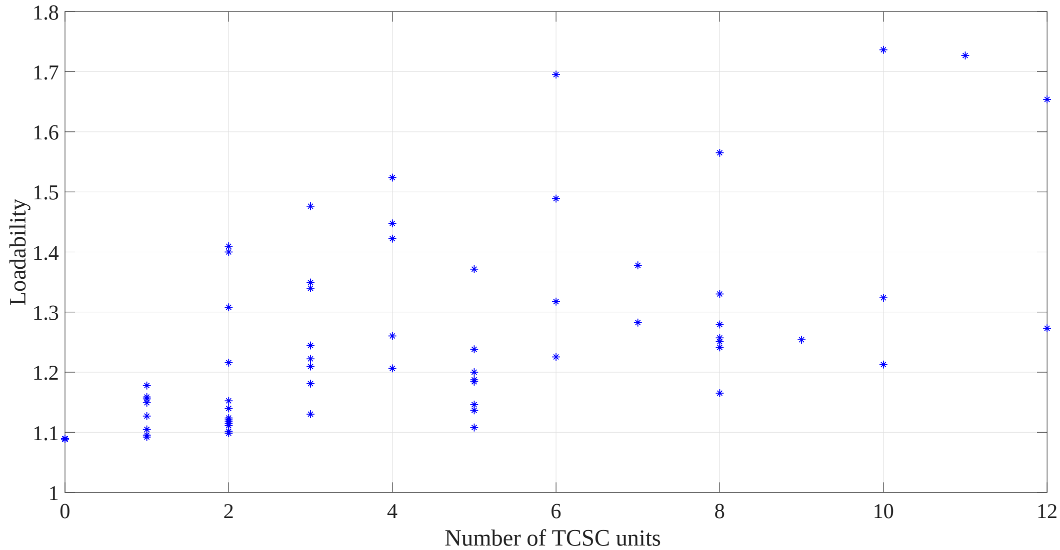

Figure 4.5: Number of TCSC Vs. Loadability

4.8 Conclusion

A multiobjective optimization based approach for enhancing network loadability using TCSCs was proposed in this chapter, that optimizes the loading margin, the number, locations, and respective sizes of the TCSC units simultaneously to maximize the loadability and to minimize the investment cost under secured operating conditions. The proposed method has solved the nonlinear mixed integer complex problem of the TCSC allocation in a flexible and efficient procedure. The results provide competitive options, over the whole domain of the loading margin, for decision-making and the planning process. The applied method shows that a desired loading factor could be reached using a different optimal number of TCSC units at different optimal locations and sizes. It was observed that increasing the loadability beyond a certain point is not beneficial because the investment cost sacrifice is high and does not justify the gain in loadability.

Chapter 5

Future Work

In the first part of this thesis, feedback-based self-adapting control parameters algorithms are proposed to improve the performance of DE in Chapter 2. The proposed algorithms FBjDE-I and FBjDE-II have shown considerable improvements in the results and on the convergence speed. The feedback information was used to control the scale factor and the the crossover rate in both FBjDE-I and FBjDE-II. In FBjDE-II, a further step is taken to control the range of the scale factor. As an extension to this work, the range of the crossover rate can be similarly controlled using the same procedure by which the mutation scale factor was controlled.

The feedback-based self adaptive algorithms were proposed for the differential evolution. These algorithms should be applied to other evolutionary algorithms for potential improvement in the performance of the results and the convergence rates.

The feedback-based self-adaptive differential evolution algorithms, FBjDE-I and FBjDE-II, were introduced to solve the optimal power flow problem in several test systems. The proposed methods have shown better results and faster convergence rates. For future work, FBjDE-I and FBjDE-II are to be used for optimal power flow in the presence of renewable energy resources and energy storage units.

In the second part of the thesis, a multiobjective based algorithm presented in Chapter 4 was applied to the optimal placement and sizing of TCSCs units. The proposed method provided outstanding results for decision making in planning process. The benefits of installing other FACTS devices, such as STATCOM and UPFC, should be utilized using the proposed method. In addition, the proposed method should be extended to include the optimal placement and sizing of renewable energy resources and energy storage units.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Storn and K. Price, "Differential evolutiona simple and efficient adaptive scheme for global optimization over continuous spaces, berkeley," *CA: International Computer Science Institute*, 1995.

[2] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2011.

[3] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 646–657, Dec 2006.

[4] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 124–141, Jul 1999.

[5] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.

[6] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, pp. 482–500, April 2012.

[7] X. Qiu, W. Xu, J.-X. Xu, and K. C. Tan, "A new framework for self-adapting control parameters in multi-objective optimization," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 743–750, ACM, 2015.

[8] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[9] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 485–492, ACM, 2006.

[10] G. C. Onwubolu and B. Babu, *New optimization techniques in engineering*, vol. 141. Springer, 2013.

[11] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2, pp. 1980–1987, IEEE, 2004.

[12] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 64–79, Feb 2008.

[13] J. Mitra, *Understanding Power Flow*. Michigan State University, 2017.

[14] B. . Stott and J. L. Marinho, "Linear programming for power-system network security applications," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, pp. 837–848, May 1979.

[15] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of ac power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.

[16] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by newton approach," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, pp. 2864–2880, Oct 1984.

[17] I. M. Nejdawi, K. A. Clements, and P. W. Davis, "An efficient interior point method for sequential quadratic programming based optimal power flow," *IEEE Transactions on Power Systems*, vol. 15, pp. 1179–1183, Nov 2000.

[18] J. Nanda, D. P. Kothari, and S. C. Srivastava, "New optimal power-dispatch algorithm using fletcher's quadratic programming method," *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 136, pp. 153–161, May 1989.

[19] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Wiley, 2005.

[20] L. Lai, J. Ma, R. Yokoyama, and M. Zhao, "Improved genetic algorithms for optimal power flow under both normal and contingent operation states," *International Journal of Electrical Power and Energy Systems*, vol. 19, pp. 287–292, 6 1997.

[21] S. Paranjothi and K. Anburaja, "Optimal power flow using refined genetic algorithm," *Electric Power Components and Systems*, vol. 30, no. 10, pp. 1055–1063, 2002.

[22] S. Kumar and D. Chaturvedi, "Optimal power flow solution using fuzzy evolutionary and swarm optimization," *International Journal of Electrical Power & Energy Systems*, vol. 47, pp. 416–423, 2013.

[23] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[24] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, Feb 2011.

[25] U. W. D. of Electrical Engineering at the University of Washington, "Power Systems Test Case Archive." `https://www2.ee.washington.edu/research/pstca/`.

[26] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[27] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, pp. 760–766, Springer, 2011.

[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-ii," *IEEE Trans. on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[29] N. G. Hingorani and L. Gyugyi, *Understanding FACTS*. IEEE press, 2000.

[30] W. S. Sakr, R. A. El-Sehiemy, and A. M. Azmy, "Optimal allocation of TCSCs by adaptive DE algorithm," *IET Generation, Transmission & Distribution*, vol. 10, no. 15, pp. 3844–3854, 2016.

[31] T. Nireekshana, G. K. Rao, and S. S. N. Raju, "Enhancement of ATC with FACTS devices using real-code genetic algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 1276–1284, 2012.

[32] M. Saravanan, S. M. R. Slochanal, P. Venkatesh, and J. P. S. Abraham, "Application of particle swarm optimization technique for optimal location of FACTS devices considering cost of installation and system loadability," *Electric Power Systems Research*, vol. 77, no. 3, pp. 276–283, 2007.

[33] O. Ziaee and F. F. Choobineh, "Optimal location-allocation of TCSC devices on a transmission network," *IEEE Trans. Power Syst*, vol. 32, no. 1, pp. 94–102, 2017.

[34] E. Ghahremani and I. Kamwa, "Optimal placement of multiple-type FACTS devices to maximize power system loadability using a generic graphical user interface," *IEEE Trans. Power Syst*, vol. 28, no. 2, pp. 764–778, 2013.

[35] S. Gerbex, R. Cherkaoui, and A. J. Germond, "Optimal location of multi-type FACTS devices in a power system by means of genetic algorithms," *IEEE Trans. Power Syst*, vol. 16, no. 3, pp. 537–544, 2001.

[36] S. Gerbex, R. Cherkaoui, and A. Germond, "Optimal location of FACTS devices to enhance power system security," in *Power Tech Conference Proceedings, 2003 IEEE Bologna*, vol. 3, pp. 7–pp, IEEE, 2003.

[37] S. Ranganathan, M. S. Kalavathi, *et al.*, "Self-adaptive firefly algorithm based multi-objectives for multi-type FACTS placement," *IET Generation, Transmission & Distribution*, vol. 10, no. 11, pp. 2576–2584, 2016.

[38] D. Radu and Y. Besanger, "A multi-objective genetic algorithm approach to optimal allocation of multi-type facts devices for power systems security," in *Power Engineering Society General Meeting, 2006. IEEE*, pp. 8–pp, IEEE.

[39] A. L. Ara, A. Kazemi, and S. N. Niaki, "Multiobjective optimal location of facts shunt-series controllers for power system operation planning," *IEEE Transactions on Power Delivery*, vol. 27, no. 2, pp. 481–490, 2012.

[40] R. Christie, "Power systems test case archive," *Electrical Engineering dept., University of Washington*, 2000.

[41] S. Blumsack, *Network topologies and transmission investment under electric-industry restructuring*. ProQuest, 2006.

[42] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[43] K. Deb and D. Deb, “Analysing mutation schemes for real-parameter genetic algorithms,” *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 1, pp. 1–28, 2014.