Chapter 3

Spherical Search and Butterfly Constrained Optimizer for Power Flow of Unbalanced Distribution System

3.1 Introduction

In this chapter, general-purpose PF tools are developed which are handy and robust for all possible applications of power system analysis. Recently, evolutionary algorithms have emerged as robust optimization tools because of their potentials and versatile qualities. These search techniques are adequate to deal with a large collection of problems of various characteristics because they do not constrain the variable types and search-space.

In the traditional Jacobian based methods of PF, the solution proceeds as per the direction gradient (Jacobian or partial derivative). This direction gradient is a kind of static formulation to decide the step direction and step size. In this gradient direction procedure, the step direction and the step size can not be decided as per quality of solution achieved so far. In fact, during the iterations, we do not evaluate any metric related to the quality of solution.

In any optimization based formulation, this is not the case. In general, in an optimization based formulation, step direction and step size are decided based on the quality of the solution. There are two broad classifications of optimization: Conventional Optimization, and Evolutionary (population based) optimization. Conventional optimization proceeds with one solution whereas evolutionaly optimization proceeds, iteration by iteration, with multiple solutions. This population of solution enhances the procedure of evaluation of the quality of solutions and consequently step direction and step size in a better way. This results in a dynamic formulation to decide the step direction and step size. Due to the above fundamental difference in the procedure to evaluate solutions, evolutionary optimization based PF formulation does not succumb to the ill-conditioning of the Jacobian matrix and is able to keep sailing in the search of solution and ultimately provides PF solution.

Traditional Jacobian based correction of solution does not consider the quality of solution in terms of present solution and solution expected in the next iteration. Whereas, conventional optimization based correction of solution considers the quality of solution in terms of present solution and the solution expected in the next iteration considering active and reactive power mismatch as objective function.

Almost all conventional (Jacobian and optimization based) PF algorithms reported in the literature utilize the following structure. The first iteration starts with the initial seed, and mismatches in reactive and active power at all buses are evaluated. Power injection equations or current injection equations at buses with their derivatives are used to calculate a correction for the variables to update the system variables. PF algorithms diverge when the required corrections are determined incorrectly. Thus, the cause of divergence exists in the procedures of an algorithm applied to determine the solutions. To address this issue, two robust algorithms are developed in this chapter. In these algorithms, rather than determining the corrections vectors, a population of solutions is perturbed around the distribution of population on search-space. Only those perturbations are utilized which facilitates the population to converge on the optimum solution. This step-by-step scheme avoids divergence, however, this makes the algorithm slower. Consequently, these proposed algorithms are only valuable when the conventional methods are not able to produce a solution for the PF problem.

The need and rationale of using optimization techniques for solving power flow problem can be appreciated in the following two ways.

- The optimization techniques have advantages of finding all the possible multiple solution accurately required in power system analysis. The optimization algorithm (especially search based) inherently have high chance of convergence due to their formulations and techniques.

- The optimization techniques have also over the times resolved in several theoretical bottlenecks thereby easing out their implementation but also improving performance.

- The above advantages though were obvious at all times, however their practical implementation were subject to contemporary computational advancements. In the present times the computational resources have improved quite significantly (or exponentially increased). This has lead to several successful implementations of computationally demanding algorithms as apparent from literature. Thus it is imperative to revisit and explore the performance of algorithmss vis-a-vis contemporary computational resource scenario.

With advent of new technologies, the modern distribution systems and tools to examine them are updating themselves. Conventional algorithms of power flow problems fail to provide a solution. As an alternative, optimization algorithms are becoming popular in this area.

In this chapter, two different optimization methods are designed to deal with the PF problem of ill-conditional test systems. The first one, referred to as Butterfly Constrained Optimizer (BCO), is a constrained optimization algorithm which employs the multi-order LM based mutation with υ-constrained handling routine to deal with the constraints of PF problems. The other one, named as Spherical Search (SS), is a bound-constrained optimization algorithm which can be utilized to calculate initial seed for conventional PF algorithms.

The major contributions of this work are summarized as follows:

1. It introduces two novel evolutionary-based PF algorithms, SS and BCO, to address the PF problem of ill-conditioned test systems.

2. It includes an evolutionary-based method to determine the initial seed for the conventional PF algorithm whose performance is dependent on the initial seed.

3. It presents a powerful methods to determine the maximum loadability limit for distribution test systems. This can be applied to investigate the voltage stability of

the system buses of distribution test systems.

4. It provides an authentic PF algorithm for the distribution test systems by establishing two optimization algorithms which diminish the active and reactive power discrepancies. The optimization algorithms have been validated on the distribution test systems to establish that these algorithms can reduce the active and reactive power mismatch.

3.2 Power Flow Formulation as a Constrained Optimization Problem

In this section, the PF problem is formulated as Constrained Optimization Problem (COP) based on power injection.

3.2.1 Formulation based on Power Injection.

The PF problem can be represented by the power balance equation at each bus. Reactive and active power are specified at each PQ buses (load buses) and active power is only specified at PV buses (generator buses). These active and reactive powers can also be calculated using bus voltages and Ybus, which are termed as calculated power. The solution to this PF problem is bus voltages where the difference of specified power and calculated power at each bus becomes zero or within the tolerance limit. Consequently, the main objective of PF is to calculate the voltage magnitude and angles of the system buses which reduce the differences between the specified power and the calculated power at each bus of the system. Hence, the PF problem can be treated as a system of non-linear equations.

In polar co-ordinates, the power balance equation at k-th bus can be represented by the following equations.

$$P_k - \sum_{i=1}^{N} |V_k||V_i||Y_{ki}| cos(\delta_k - \delta_i - \theta_{ki}) = 0, \tag{3.1}$$

$$Q_k - \sum_{i=1}^{N} |V_k||V_i||Y_{ki}| sin(\delta_k - \delta_i - \theta_{ki}) = 0, \tag{3.2}$$

where $P_k(= P_{g,k} - P_{l,k})$ and $Q_k(= Q_{g,k} - Q_{l,k})$ are total active and reactive power injected at k-th bus, respectively, $V_k(|V_k|\angle\delta_k)$ represents the bus voltage at k-th bus, and $Y_{ki}(|Y_{ki}|\angle\theta_{ki})$ represents the ki-th element of admittance matrix. Here, $P_{g,k}$ and $Q_{g,k}$ are total generated active and reactive power at k-th bus, respectively, $P_{l,k}$ and $Q_{l,k}$ represent total active and reactive load at k-th bus.

3.2.2 Formulation as a Constrained Optimization Problem

In this section, PF problem of ill-conditioned test system has been formulated as a COP which is proposed to be solved using the evolutionary algorithms.

A PF problem can be formulated as a COP:

$$\text{Minimize } f = \sum_{(i \in S_{pq} \cup S_{pv})} (P_i - P_{i,g})^2 + \sum_{(j \in S_{pq})} (Q_i - Q_{i,l})^2 + \sum_{(k \in S_{pv})} \left(|V_k| - \sqrt{V_{rk}^2 + V_{mk}^2} \right)^2 \tag{3.3}$$

subject to.

$$\begin{gathered} V_{ri} \sum_{j=1}^{N} (V_{rj} G_{ij} - V_{mj} B_{ij}) + V_{mi} \sum_{j=1}^{N} (V_{rj} B_{ij} + V_{mj} G_{ij}) - P_i = 0, \\ V_{mi} \sum_{j=1}^{N} (V_{rj} G_{ij} - V_{mj} B_{ij}) - V_{ri} \sum_{j=1}^{N} (V_{rj} B_{ij} + V_{mj} G_{ij}) - Q_i = 0, \\ Vr_{lb} < V_{rj} < Vr_{ub},\ Vm_{lb} < V_{mj} < Vm_{ub} \\ P_{lb} < P_i < P_{ub},\ Q_{lb} < Q_i < Q_{ub}, \end{gathered} \tag{3.4}$$

where, S_{pq} and S_{pv} are set of PQ and PV buses, respectively. $P_{i,g}$ and $Q_{i,g}$ are generated real and reactive power at i-th bus and $P_{i,l}$ and $Q_{i,l}$ are real and reactive load at i-th bus. G_{ij} and B_{ij} are active and reactive part of ijth element of Ybus matrix and N is the total number of buses in the system. In this problem, the variables are V_{rj}, V_{mj}, P_j, and Q_j where $j = 2, ..., N$. Therefore, total number of variables are $4 \times (N - 1)$.

3.3 Spherical Search

In this section, the mathematical modeling of the SS algorithm is discussed and developed.

SS algorithm is a swarm based meta-heuristics proposed to solve the non-linear bound-constrained global optimization problems. It shows some properties similar to other popular meta-heuristics, particularly PSO and DE. In the SS algorithm, the search space is represented in the form of vector space where the location of each individual in space is a position vector representing a candidate solution to the problem.

In a D-dimension search space, for each individual, $(D-1)$-spherical boundary is prepared towards the target direction in every iteration before generating the trial location of the individual. Here the target direction is the main axis of the spherical boundary and the individual lies on the surface of the spherical boundary. Trial solutions appear on the 1-spherical boundary. Thus, in every iteration, the trial location for each individual is generated on the surface of $(D-1)$-spherical boundary. An objective function value determines the fitness value of a location. On the basis of the fitness value of the trial locations, better locations pass on into the next iteration as individual locations.

In the SS algorithm, solution update procedure and spherical search movement balance the ability of exploration and exploitation. When the $(D-1)$-spherical boundary is small, exploitation of search-space is emphasized in the algorithm. On the other hand, in case of larger $(D-1)$-spherical boundary, exploration of the space gets emphasized. It is evident that when the target location of a respective individual is far-off, the individual has a tendency to explore, as the spherical boundary is large. This is advantageous as in such conditions it is better to explore the larger search space. On the contrary, when the target locations of a respective individual are nearby, the individual has a tendency to exploit as the spherical boundary becomes small. This is advantageous as in such situations it is better to exploit in a small search space.

At the end of every iteration, the location having the best fitness value is saved as the best solution. Stopping criteria is achieved when the number of function evaluations reaches to a specified number or when the value of the best solution reaches near to the predefined solution within a specified tolerance. For the reported experimental work of this chapter, both the stopping criteria have been used.

3.3.1 Initialization of Population

At the k^{th} iteration, population P_x is represented as follows.

$$P_x^{(k)} = [\bar{x_1}^{(k)}, \bar{x_2}^{(k)}, ...\bar{x_N}^{(k)}] \tag{3.5}$$

where

$$\bar{x_i}^{(k)} = [x_{i1}^{(k)}, x_{i2}^{(k)}, ...x_{iD}^{(k)}]^T \tag{3.6}$$

here, x_{ij} is the value of j^{th} element (parameter) of i^{th} solution and D is the total number of elements (parameters). So, $\bar{x_i}$ is actually representing a point on $D-$dimensional search space. Here, x_{ij}^0 is initialized using random uniform distribution between pre-specified lower and upper bounds of j^{th} element as follows:

$$x_{ij}^0 = (x_{hj} - x_{lj}) * rand(0, 1] + x_{lj}, \tag{3.7}$$

where x_{hj} and x_{lj} represent the upper and lower bounds of j^{th} element respectively. Also, $rand(0, 1]$ generates random number from uniform distribution within the limit $(0, 1]$.

3.3.2 Spherical Surface and Trial Solutions

In the case of population-based optimization algorithms, in every iteration, there will be a need to calculate potential new solutions that compete with the old solution to become a part of the population for the next iteration. In this algorithm, the name, trial solution, is used to represent these potential new solutions. In SS algorithm, for each solution, a $(D-1)$-spherical boundary is prepared where the search direction passes through the main axis of boundary i.e. search direction crosses the center of $(D-1)$-spherical boundary.

In SS algorithm, following equation is used to generate a trial solutions for $i-$th solution.

$$\bar{y_i}^{(k)} = \bar{x_i}^{(k)} + c_i^{(k)} P_i^{(k)} \bar{z_i}^{(k)}, \tag{3.8}$$

where, P_i is a projection matrix, which decides the value of $\bar{y_i}^k$ on the $(D-1)$-spherical boundary. For a particular solution, $\bar{x_i^{(k)}}$, different possible values of $P_i^{(k)}$ yield different values of $\bar{y}_i^{(k)}$. Locus of $\bar{y}_i^{(k)}$ gives $(D-1)$ spherical boundary.

To define all the iterative steps of the SS algorithm, the calculation procedure of $\bar{z_i}^{(k)}$, $c_i^{(k)}$ and $P_i^{(k)}$ are discussed in following sections.

Calculation of search direction, $\bar{z_i}^{(k)}$

In optimization algorithms, quality of new solution highly depends upon the balance between exploration and exploitation of search space. Emphasis on exploration of search space increases the diversity of candidate solutions but slows the optimization process resulting in delayed or no convergence. Whereas, emphasis on exploitation may accelerate the optimization process which may lead to premature convergence trapped in local minima.

Search direction, ${\bar{z}_i}^{(k)}$, should be generated in such a way that it guides the $i-$th solution towards the better solutions. Points $\bar{x}_t$, r_1 and r_2 are needed to calculate the search direction as follows.

$$ {\bar{z}_i}^{(k)} = ({\bar{x}_t}^{(k)} + r_1^{(k)} - r_2^{(k)}) - {\bar{x}_i}^{(k)}, \tag{3.9} $$

where, $\bar{x}_t$ is the target point. In equation (3.9), two random solutions r_1 and r_2 are selected from the current set of solutions (population). So, the actual search direction deviates by some angle from target direction.

In this chapter, two methods are introduced to calculate the search direction, namely *towards-rand* and *towards-best*. Method *towards-rand* has a better exploration capability and *towards-best* improves the exploitation capability. So, to provide a good balance between the exploration and exploitation of search space, for the half population of better solution, calculation of search direction can be done by *towards-rand* and for the rest half of the population, *towards-best* is used to calculate the search direction thereby forcing diversity in the set of better solutions and forcing the inferior solutions to strive for improved fitness.

In *towards-rand*, the search direction, ${\bar{z}_i}^{(k)}$, for i^{th} solution at k^{th} iteration is calculated using following equation.

$$ \bar{z}_i^{(k)} = \bar{x}_{p_i}^{(k)} + \bar{x}_{q_i}^{(k)} - \bar{x}_{r_i}^{(k)} - \bar{x}_i^{(k)}, \tag{3.10} $$

where

p_i, q_i, and r_i are randomly selected indices from among 1 to N such that $p_i \neq q_i \neq r_i \neq i$.

In *towards-best*, the search direction, ${\bar{z}_i}^{(k)}$, for i^{th} solution at k^{th} iteration is calculated using following equation.

$$ \bar{z}_i^{(k)} = \bar{x}_{pbest_i}^{(k)} + \bar{x}_{q_i}^{(k)} - \bar{x}_{r_i}^{(k)} - \bar{x}_i^{(k)}, \tag{3.11} $$

where $x_{pbest_i}^{(k)}$ represents the randomly selected individual from among the top p solutions searched so far.

Here, $\bar{x}_{p_i}$ and $\bar{x}_{pbest_i}$ represent the target points in *towards-rand* and *towards-best* respectively. Difference term $(\bar{x}_q - \bar{x}_r)$ is common in both *towards-rand* and *towards-best* which represents $\bar{r}_1 - \bar{r}_2$ (equation 3.9), which is an approximation of distribution of difference of solutions in a population. In calculation of new search direction, difference term $(\bar{x_q} - \bar{x_r})$ makes the population to evolve maintaining the diversity of solution thereby avoiding convergence to local minima.

Projection matrix, P is a symmetrical matrix which is used for linear transformation from search space to itself such that $P^2 = P$ i.e. whenever P transforms a point twice, it provides the same point. Projection matrix, $P = A' diag(\bar{b_i}) A$, has been used in equation (3.8) to linearly transform $c_i \bar{z_i} + \bar{x_i}$ to generate trial solution $\bar{y_i}$ on the circular (1-spherical) boundary. Here, A and $\bar{b_i}$ are the orthogonal matrix and binary vector respectively. The total number of combinations of possible binary vectors is finite but in case of orthogonal matrix, A, the possible combinations are infinite. Therefore, all the possible projections of $c_i \bar{z_i} + \bar{x_i}$ create a $(D-1)$-spherical boundary on search space.

Method to compute elements of P along with c has been illustrated as follows.

Orthogonal matrix, A

At the start of k^{th} iteration, an orthogonal matrix, A, is generated randomly such that

$$AA' = I. \tag{3.12}$$

Binary diagonal matrix, $diag(\bar{b_i})$

Binary diagonal matrix, $diag(\bar{b_i})$, are calculated randomly in such a way that,

$$0 < \text{rank}(diag(\bar{b_i})) < D \tag{3.13}$$

Step-size control vector, $\bar{c}$

A step-size control vector, $\bar{c}^{(k)} = [c_1^{(k)}, c_2^{(k)}, ...c_N^{(k)}]$, consists of step-size control parameter used for generation of all the possible trial-solutions where $c_i^{(k)}$ represents the step-size control parameter for i^{th} trial-solution at k^{th} iteration.

At the start of k^{th} iteration, the elements of $\bar{c}^{(k)}$ are calculated randomly in range of [0.5 0.7], arrived by experiments.

3.3.3 Selection of New Population for Next Iteration

Greedy selection procedure is applied to select new set of population for next iteration. To update the i^{th} solution of the population, following criteria is applied. If the objective function value of trial solution, $f(\bar{y_i}^{(k)})$, is lower than the objective function value of solution, $f(\bar{x_i}^{(k)})$ then y_i replaces x_i.

Mathematically,

$$\bar{x_i}^{(k+1)} = \begin{cases} \bar{y_i}^{(k)}, & \text{if } f(\bar{y_i}^{(k)}) \leq f(\bar{x_i}^{(k)}) \\ \bar{x_i}^{(k)}, & \text{otherwise} \end{cases} \tag{3.14}$$

3.3.4 Stopping Criteria

Termination of iterations depends upon two criteria: i) the maximum number of function evaluations and ii) convergence of solution i.e solution is not getting updated for specified number of consecutive iterations.

3.3.5 Steps of Spherical Search Algorithm

The main steps of the proposed algorithm are summarized as follows:

- **Step 1:** Initialize the population P.
- **Step 2:** Calculate the objective function of each solution of P.
- **Step 3:** The best solution of population is selected as best solution.
- **Step 4:** Calculate the search direction for each solution of population P.
- **Step 5:** Calculate the orthogonal matrix, A.
- **Step 6:** Calculate parameters: c_i and *rank of projection matrix*.
- **Step 7:** Calculate trial solution for each solution of population P.
- **Step 8:** Update the population using greedy selection operator.
- **Step 9:** If the stopping criterion is satisfied then the algorithm will be stopped, otherwise it will return to **Step 3**.
- **Step 10:** Return the best optimal solutions, after stopping criteria is satisfied.

Table 3.1: Mean and SD of best error value obtained in 51 independent runs by SS, PSO, BB-PSO, CLPSO, APSO, and OLPSO on 30-D CEC2014 problem suite (Mean: Mean of best error, SD: Standard deviation of best error, W: result of Wilcoxon signed rank test).

Nos	SS		PSO			BB-PSO			CLPSO			APSO			OLPSO		
	Mean	SD	Mean	SD	W	Mean	SD	W	Mean	SD	W	Mean	SD	W	Mean	SD	W
f01	**8.75E+03**	**6.70E+03**	5.14E+07	2.24E+07	+	2.40E+06	1.97E+06	+	7.82E+06	2.44E+06	+	2.69E+09	3.28E+08	+	6.12E+06	3.58E+06	+
f02	**0.00E+00**	**0.00E+00**	2.42E+08	4.23E+08	+	1.84E+03	7.68E+03	+	8.93E+01	2.33E+02	+	1.02E+11	2.29E+09	+	1.28E+03	1.48E+03	+
f03	**1.03E-07**	**2.81E-07**	5.97E+04	1.07E+04	+	6.23E+03	1.40E+04	+	2.03E+02	1.85E+02	+	1.19E+06	1.25E+06	+	3.23E+02	5.69E+02	+
f04	**0.00E+00**	**0.00E+00**	1.66E+02	8.29E+01	+	5.99E+01	3.45E+01	+	6.92E+01	2.06E+01	+	2.49E+04	1.54E+03	+	8.64E+01	2.22E+01	+
f05	2.09E+01	5.88E-02	2.08E+01	9.01E+01	=	**2.01E+01**	**5.20E-02**	-	2.04E+01	4.13E-02	-	2.13E+01	5.61E-02	+	2.03E+01	1.28E-01	-
f06	**2.79E-01**	**5.86E-01**	2.58E+01	6.40E+01	+	1.17E+01	2.12E+00	+	1.23E+01	1.47E+00	+	4.80E+01	1.79E+00	+	5.09E+00	1.48E+00	+
f07	**0.00E+00**	**0.00E+00**	9.28E+01	1.23E+01	+	2.73E-02	2.71E-02	+	3.43E-06	4.28E-06	+	1.06E+03	3.85E+01	+	1.02E-13	3.47E-14	=
f08	1.59E+02	1.00E+01	5.09E-02	1.11E+00	-	1.02E-03	8.96E-04	-	**0.00E+00**	**0.00E+00**	-	5.03E+02	3.02E+01	+	**0.00E+00**	**0.00E+00**	-
f09	1.61E+02	1.04E+01	3.18E+01	1.28E+02	-	**3.16E+01**	**8.10E+00**	-	5.22E+01	6.38E+00	-	4.78E+02	6.30E+00	+	4.06E+01	7.02E+00	-
f10	6.32E+03	2.55E+02	7.61E+02	7.61E+02	-	1.59E+00	1.42E+00	-	3.11E+00	1.60E+00	-	9.30E+03	5.68E+02	+	**8.72E-02**	**2.04E-01**	-
f11	6.57E+03	3.04E+02	6.83E+03	1.93E+03	+	**2.11E+03**	**4.38E+02**	-	2.33E+03	3.19E+02	-	9.24E+03	4.86E+02	+	2.28E+03	4.66E+02	-
f12	2.29E+00	3.10E-01	2.86E+00	2.77E+01	+	**1.66E-01**	**5.61E-02**	-	3.94E-01	5.53E-02	-	5.91E+00	1.32E+00	+	2.28E-01	6.38E-02	-
f13	2.49E-01	3.21E-02	6.40E-01	4.44E-01	+	**2.38E-01**	**5.24E-02**	=	3.17E-01	3.75E-02	+	1.03E+01	7.53E-01	+	2.59E-01	3.20E-02	+
f14	2.51E-01	3.53E-02	4.21E-01	1.57E-01	+	5.75E-01	2.78E-01	+	2.51E-01	2.85E-02	=	3.95E+02	2.22E+01	+	**2.41E-01**	**2.66E-02**	=
f15	1.39E+01	7.80E-01	**4.47E+00**	**1.92E+00**	-	7.82E+00	3.03E+00	-	7.49E+00	1.02E+00	-	1.05E+06	0.00E+00	+	6.67E+00	1.62E+00	-
f16	1.19E+01	3.28E-01	1.29E+01	4.82E-01	+	**9.13E+00**	**1.08E+00**	-	1.04E+01	3.72E-01	-	1.42E+01	2.37E-01	+	1.17E+01	5.48E-01	=
f17	**4.80E+02**	**2.34E+02**	3.26E+05	2.57E+05	+	6.44E+05	6.17E+05	+	9.59E+05	4.41E+05	+	2.86E+08	1.28E+08	+	7.98E+05	4.13E+05	+
f18	**7.33E+01**	**2.28E+01**	1.02E+06	2.95E+06	+	6.46E+03	8.74E+03	+	1.01E+02	4.67E+01	+	8.75E+09	3.11E+09	+	3.58E+02	5.12E+02	+
f19	**5.37E+00**	**6.72E-01**	2.82E+02	1.98E+01	+	7.43E+00	1.29E+00	+	7.39E+00	6.27E-01	+	8.45E+02	1.15E+02	+	6.13E+00	8.20E-01	+
f20	**5.77E+01**	**2.14E+01**	1.08E+04	2.18E+03	+	3.02E+03	2.91E+03	+	3.14E+03	1.40E+03	+	1.59E+07	1.37E+07	+	5.58E+03	4.01E+03	+
f21	**4.85E+02**	**1.96E+02**	7.54E+05	4.21E+05	+	2.28E+05	3.29E+05	+	8.46E+04	5.35E+04	+	1.33E+08	7.50E+07	+	1.07E+05	8.33E+04	+
f22	2.29E+02	1.06E+02	8.24E+02	5.24E+02	+	2.43E+02	2.10E+02	+	2.08E+02	7.41E+01	+	1.31E+04	9.38E+03	+	**2.20E+02**	**1.07E+02**	=
f23	3.15E+02	6.61E-03	3.42E+02	6.24E+00	+	3.15E+02	1.87E-12	+	3.15E+02	2.02E-06	+	**2.00E+02**	**0.00E+00**	-	3.15E+02	1.23E-10	+
f24	2.21E+02	7.79E+00	2.05E+02	2.41E-01	-	2.28E+02	4.97E+00	+	2.25E+02	4.53E-01	+	**2.00E+02**	**0.00E+00**	-	2.24E+02	5.47E-01	+
f25	2.03E+02	2.74E-01	2.20E+02	4.51E+00	+	2.06E+02	2.27E+00	+	2.08E+02	1.06E+00	+	**2.00E+02**	**0.00E+00**	-	2.09E+02	1.75E+00	+
f26	1.00E+02	3.24E-02	1.00E+02	2.42E-01	=	1.00E+02	5.49E-02	+	1.00E+02	6.42E-02	+	1.86E+02	2.68E+01	+	1.00E+02	4.44E-02	+
f27	3.34E+02	4.34E+01	2.51E+03	4.82E+02	+	6.53E+02	5.06E+01	+	4.14E+02	4.92E+00	=	**2.00E+02**	**0.00E+00**	-	3.26E+02	3.80E+01	=
f28	8.47E+02	9.35E+01	1.81E+03	4.91E+02	+	9.10E+02	2.59E+01	+	9.08E+02	3.97E+01	+	**2.00E+02**	**0.00E+00**	-	8.73E+02	2.97E+01	+
f29	8.24E+02	6.91E+01	8.77E+07	3.24E+07	+	1.70E+06	3.47E+06	+	9.71E+02	9.14E+01	+	**2.00E+02**	**0.00E+00**	-	1.36E+03	2.82E+02	+
f30	1.74E+03	8.95E+02	4.11E+05	1.87E+04	+	7.94E+03	1.07E+03	+	3.39E+03	7.88E+02	+	**2.00E+02**	**0.00E+00**	-	2.39E+03	5.99E+02	+
+/=/-			23/2/5			21/1/8			20/2/8			23/0/7			18/5/7		

3.3.6 Validation of Spherical Search on Benchmark Problems

In this section, to analyze the performance, SS are benchmarked on 30 real-parameter single objective bound-constrained optimization problems used in a special session of IEEE CEC-2014 [122]. Detailed information and characteristics of these problems are available in [122]. To evaluate the performance of SS on the CEC 2014 problem suite, the results are compared with other state-of-the-art algorithms. State-of-the-art algorithms are divided into four groups:

1. *Group-I:-* Variants of PSO: Basic PSO [123], BB-PSO [124], CLPSO [125], APSO [126], OLPSO [127].

2. *Group-II:-* Variants of DE: CoBiDE [128], FCDE [129], RSDE [130], POBLADE [131], DE-best [132].

3. *Group-III:-* Variants of CMA-ES: Basic CMA-ES [133], I-POP-CMAES [134], LS-CMAES [135], CMSAES [136], (1+1)cholesky-CMAES [137].

4. *Group-IV:-* Recently proposed optimization algorithms: GWO [138], GOA [139], MVO [140], SCA [141], SHO [142], SSA [143], SOA [144], WOA [145].

PSO, DE, and CMA-ES are the popular classical Meta-heuristics. Popular variants of classical algorithms are also taken from the literature to show the effectiveness of the SS.

In this experiment, the population size, N, is set to 80, the dimension of search space for all problems, D, is fixed to 30, and allowed maximum function evaluation, MaxFES, is fixed to 300,000 for 51 independent runs. The parameters of other algorithms are set to their default values as reported in their referred paper.

Tables 3.1-3.9 summarize the mean and standard deviation (SD) of the error values obtained by the algorithms over 51 independent runs for each problem. We also performed the Wilcoxon Signed Ranks Test in this experiment. The statistical results are also summarized in Tables 3.1-3.9, where '+' denotes the performance of the SS is better than other algorithms, '-' denotes the performance of other method is better than the SS, and '=' denotes that there is no significant difference in performance. We also rank all algorithms along with the SS using Friedman ranking test based on the mean of error

Table 3.2: Mean and SD of best error value obtained in 51 independent runs by SS, CoBiDE, FCDE, RSDE, POBL_ADE, and DE_best on 30-D CEC2014 problem suite (Mean: Mean of best error, SD: Standard deviation of best error, W: result of Wilcoxon signed rank test).

Nos	SS		CoBiDE			FCDE			RSDE			POBL_ADE			DE_best		
	Mean	**SD**	**Mean**	**SD**	**W**	**Mean**	**SD**	**W**	**Mean**	**SD**	**W**	**Mean**	**SD**	**W**	**Mean**	**SD**	**W**
f01	8.75E+03	6.70E+03	**3.24E+00**	**3.24E+00**	-	1.05E+05	9.10E+04	+	1.50E+03	1.70E+03	-	1.60E+04	1.22E+04	+	2.46E+07	9.27E+06	+
f02	**0.00E+00**	**0.00E+00**	1.12E-02	5.54E-03	+	**0.00E+00**	**0.00E+00**	=	**1.19E-09**	**5.99E-09**	=	3.14E+02	7.52E+02	+	**0.00E+00**	**0.00E+00**	=
f03	**1.03E-07**	**2.81E-07**	5.17E-07	4.32E-07	=	1.09E+02	4.04E+02	+	4.74E-02	1.16E-01	+	**6.43E-10**	**4.59E-09**	=	3.32E-05	2.00E-05	-
f04	**0.00E+00**	**0.00E+00**	1.85E+01	6.45E-01	+	2.99E+01	3.05E+01	+	3.05E+00	1.34E+01	+	6.34E+01	2.63E+01	+	7.68E+01	2.42E+01	+
f05	2.09E+01	5.88E-02	2.06E+01	5.12E-02	-	2.09E+01	7.67E-02	=	**2.03E+01**	**9.88E-02**	-	2.06E+01	5.11E-02	-	2.09E+01	4.56E-02	=
f06	2.79E-01	5.86E-01	2.74E+01	1.45E+00	+	2.14E+01	3.56E+00	+	5.16E+00	2.01E+00	+	5.19E+00	1.64E+00	+	1.39E+00	1.31E+00	+
f07	**0.00E+00**	**0.00E+00**	6.74E-07	2.39E-06	+	2.79E-02	2.74E-02	+	8.46E-04	1.59E-03	+	2.37E-02	2.31E-02	+	5.46E-03	6.61E-03	+
f08	1.59E+02	1.00E+01	2.81E+01	3.15E+00	-	9.49E+01	2.23E+01	-	**2.04E+01**	**7.04E+00**	-	5.59E+01	1.10E+01	-	8.90E+01	2.22E+01	-
f09	1.61E+02	1.04E+01	1.42E+02	1.35E+01	-	1.35E+02	3.14E+01	-	**5.80E+01**	**1.65E+01**	-	8.46E+01	9.06E+00	-	1.81E+02	1.12E+01	+
f10	6.32E+03	2.55E+02	2.74E+03	1.84E+01	-	2.27E+03	5.35E+02	-	**3.29E+02**	**2.47E+02**	-	2.17E+03	4.92E+02	-	1.71E+03	1.06E+03	-
f11	6.57E+03	3.04E+02	5.68E+03	2.43E+02	-	3.49E+03	6.80E+02	-	**2.74E+03**	**6.44E+02**	-	3.86E+03	3.52E+02	-	6.41E+03	2.93E+02	=
f12	2.29E+00	3.10E-01	1.05E+00	1.56E-01	-	1.37E+00	5.49E-01	-	**4.44E-01**	**1.66E-01**	-	9.51E-01	1.35E-01	-	2.09E+00	2.42E-01	=
f13	**2.49E-01**	**3.21E-02**	4.56E-01	4.52E-02	+	5.68E-01	1.16E-01	+	3.05E-01	5.50E-02	+	2.86E-01	6.10E-02	+	3.79E-01	4.26E-02	+
f14	2.51E-01	3.53E-02	2.65E-01	2.45E-02	+	4.44E-01	2.27E-01	+	2.36E-01	3.37E-02	=	**2.26E-01**	**4.28E-02**	-	3.90E-01	2.08E-01	+
f15	1.39E+01	7.80E-01	1.35E+01	1.24E+00	=	1.54E+01	6.70E+00	+	**5.92E+00**	**2.59E+00**	-	7.73E+00	1.04E+00	-	1.66E+01	1.26E+00	+
f16	1.19E+01	3.28E-01	1.36E+01	1.85E-01	+	1.22E+01	5.74E-01	+	1.06E+01	7.70E-01	=	**1.04E+01**	**4.58E-01**	=	1.22E+01	2.73E-01	+
f17	**4.80E+02**	**2.34E+02**	1.68E+03	1.45E+02	+	6.54E+03	8.72E+03	+	1.24E+03	3.79E+02	+	1.10E+03	4.14E+02	+	7.64E+05	2.61E+05	+
f18	7.33E+01	2.28E+01	**4.69E+01**	**6.34E+00**	-	1.23E+02	6.70E+01	+	9.54E+01	4.34E+01	+	1.10E+02	3.81E+01	+	1.68E+03	1.30E+03	+
f19	**5.37E+00**	**6.72E-01**	1.25E+01	8.54E-01	+	1.32E+01	1.18E+01	+	5.65E+00	1.46E+00	+	8.88E+00	1.21E+01	+	6.29E+00	1.28E+03	+
f20	5.77E+01	2.14E+01	**2.64E+01**	**3.45E+00**	-	1.33E+02	7.99E+01	+	3.73E+01	2.55E+01	-	3.89E+01	2.21E+01	-	1.14E+02	1.65E+01	+
f21	4.85E+02	1.96E+02	7.16E+02	1.45E+02	+	3.01E+03	3.32E+03	+	4.71E+02	2.34E+02	=	**3.86E+02**	**1.91E+02**	-	3.45E+04	1.49E+04	+
f22	2.29E+02	1.06E+02	2.49E+02	7.98E+01	+	4.57E+02	1.68E+02	+	1.91E+02	1.19E+02	-	2.31E+02	8.16E+01	+	**1.55E+02**	**1.10E+02**	-
f23	3.15E+02	6.61E-03	**3.14E+02**	**2.85E-09**	-	3.15E+02	1.24E-12	+	3.15E+02	1.40E-06	+	3.15E+02	5.74E-14	+	3.15E+02	4.02E-13	+
f24	**2.21E+02**	**7.79E+00**	2.43E+02	1.56E+01	+	2.50E+02	6.86E+00	+	2.24E+02	1.65E+00	+	2.22E+02	7.48E+00	+	2.23E+02	7.13E+00	+
f25	2.03E+02	2.74E-01	**2.00E+02**	**6.58E-04**	-	2.07E+02	4.01E+00	+	2.03E+02	1.17E-01	+	2.04E+02	3.22E+00	+	2.09E+02	1.77E+00	+
f26	**1.00E+02**	**3.24E-02**	1.00E+02	4.56E-02	+	1.01E+02	1.20E-01	+	1.00E+02	4.14E-02	+	1.39E+02	4.91E+01	+	1.00E+02	6.13E-02	+
f27	**3.34E+02**	**4.34E+01**	1.10E+03	6.71E+01	+	6.47E+02	2.50E+02	+	4.69E+02	9.46E+01	+	4.21E+02	4.64E+01	+	3.45E+02	4.44E+01	+
f28	8.47E+02	9.35E+01	**3.72E+02**	**6.37E-01**	-	1.60E+03	5.92E+02	+	9.05E+02	1.21E+02	+	9.16E+02	1.63E+02	+	7.74E+02	1.03E+02	-
f29	8.24E+02	6.91E+01	**2.16E+02**	**9.50E+01**	-	7.55E+05	2.62E+06	+	6.52E+05	2.66E+06	+	3.39E+05	2.41E+06	+	3.57E+03	1.97E+03	+
f30	1.74E+03	8.95E+02	7.36E+02	1.02E+02	-	2.98E+03	1.26E+03	+	1.70E+03	8.67E+02	=	1.29E+03	5.14E+02	-	2.24E+03	6.54E+02	+
+/=/-			14/2/14			23/2/5			15/5/10			19/2/9			21/4/5		

values obtained by the algorithms over 51 independent runs. The statistical results of the Friedman test are reported in Table 3.10.

1. *SS vs Group-I's algorithms:* Table 3.1 summarizes the results obtained by algorithms SS, and Group-I's algorithms: PSO, BB-PSO, CLPSO, APSO, and OLPSO on the CEC-2014 problem suite. It is seen from Table 3.1 that the SS is showing better performance than PSO, BB-PSO, CLPSO, APSO and OLPSO on 23, 21, 20, 23, and 18 problems out of 30, respectively, performance of SS worse than PSO, BB-PSO, CLPSO, APSO and OLPSO on five, eight, eight, seven, and seven problems respectively, and SS is significantly equal to the PSO, BB-PSO, CLPSO, and OLPSO for 2, 1, 2, and 5 problems of CEC-2014 problem suite respectively.

2. *SS vs Group-II's algorithms:* Table 3.2 summarizes the results of the algorithms SS, and Group-II's algorithms: CoBiDE, FCDE, RSDE, POBL_ADE, and DE_best. As shown in Table 3.2, SS have better performance than CoBiDE, FCDE, RSDE, POBL_ADE, and DE_best on 14, 23, 15, 19, and 21 problems out of 30 respectively, performance of SS worse than CoBiDE, FCDE, RSDE, POBL_ADE, and DE_best on 14, five, 10, nine, and five problems respectively, and SS provides performance similar to the CoBiDE, FCDE, RSDE, POBL_ADE, and DE_best for two, two, five, two and seven problems of CEC-2014 problem suite respectively.

3. *SS vs Group-III's algorithms:* Table 3.3 presents the results of the algorithms SS, and Group-III's algorithms: CMA-ES, I-POP-CMAES, LS-CMAES, CMSAES, and (1+1)cholesky-CMAES. when examined the last column of Table 3.3, SS shows better performs then CMA-ES, I-POP-CMAES, LS-CMAES, CMSAES, and (1+1)Cholesky-CMAES on 25, 18, 22, 30, and 23 problems out of 30 respectively, performance of SSO worse than CMA-ES, I-POP-CMAES, LS-CMAES, and (1+1)Cholesky-CMAES on five, nine, eight, and six problems respectively, and SSO is significantly similar to the I-POP-CMAES, and (1+1)Cholesky-CMAES for three, and one problems of CEC-2014 problem suite respectively.

4. *SS vs Group-IV's algorithms:* In Tables 3.4 and 3.9, the outcomes of SS and Group-IV's algorithms are presented. The last rows of Tables 3.4 and 3.9 summarizes the results of WT. It is seen from Tables 3.4 and 3.9 that the performance of SS is

Table 3.3: Mean and SD of best error value obtained in 51 independent runs by SS, CMA-ES, I-POP-CMAES, LS-CMAES, CMSAES, and (1+1)cholesky_CMAES on 30-D CEC2014 problem suite (Mean: Mean of best error, SD: Standard deviation of best error, W: result of Wilcoxon signed rank test).

Nos	SS		CMA-ES			I-POP-CMAES			LS-CMAES			CMSAES			(1+1)cholesky CMAES		
	Mean	SD	Mean	SD	W	Mean	SD	W	Mean	SD	W	Mean	SD	W	Mean	SD	W
f01	8.75E+03	6.70E+03	8.16E+04	2.66E+04	+	**0.00E+00**	**0.00E+00**	-	2.54E+08	1.37E+08	+	7.56E+08	1.40E+08	+	**0.00E+00**	**0.00E+00**	-
f02	**0.00E+00**	**0.00E+00**	4.59E+10	9.72E+09	+	**0.00E+00**	**0.00E+00**	=	6.98E+04	3.25E+04	+	6.85E+10	5.93E+09	+	**0.00E+00**	**0.00E+00**	=
f03	1.03E-07	2.81E-07	2.96E+03	1.29E+03	+	**0.00E+00**	**0.00E+00**	-	2.15E-02	1.90E-02	+	1.26E+05	1.75E+04	+	**0.00E+00**	**0.00E+00**	-
f04	**0.00E+00**	**0.00E+00**	4.71E+03	1.33E+03	+	**0.00E+00**	**0.00E+00**	=	5.50E+00	1.62E+01	+	1.01E+04	1.63E+03	+	1.27E+00	9.10E+00	+
f05	2.09E+01	5.88E-02	**2.00E+01**	**3.70E-04**	-	2.11E+01	7.69E-02	+	2.01E+01	2.18E-01	-	2.10E+01	4.38E-02	+	2.00E+01	5.85E-03	-
f06	**2.79E-01**	**5.86E-01**	6.50E+01	1.57E+00	+	9.48E+00	7.65E+00	+	2.47E+01	3.55E+00	+	3.95E+01	1.04E+00	+	5.02E+01	4.68E+00	+
f07	**0.00E+00**	**0.00E+00**	1.12E+02	1.16E+01	+	**0.00E+00**	**0.00E+00**	=	4.45E-02	1.90E-02	+	6.01E+02	5.27E+01	+	1.11E-02	9.66E-03	+
f08	1.59E+02	1.00E+01	8.98E+02	1.83E+02	+	**1.25E+02**	**1.20E+02**	-	3.41E+02	1.02E+02	+	3.67E+02	1.66E+01	+	4.38E+02	7.76E+01	+
f09	1.61E+02	1.04E+01	9.14E+02	4.24E+00	+	**3.70E+01**	**6.27E+01**	-	3.85E+02	1.84E+02	+	4.31E+02	2.35E+01	+	6.14E+02	1.37E+02	+
f10	6.32E+03	2.55E+02	**1.09E+03**	**5.78E+02**	-	3.47E+03	2.02E+03	-	3.81E+03	5.94E+02	-	7.28E+03	2.80E+02	+	4.96E+03	7.40E+02	-
f11	6.57E+03	3.04E+02	**2.16E+02**	**1.11E+02**	-	2.74E+03	2.92E+03	-	3.38E+03	5.21E+02	-	7.19E+03	2.31E+02	+	5.19E+03	8.58E+02	-
f12	2.29E+00	3.10E-01	**1.20E-01**	**1.16E+00**	-	4.39E+00	1.18E+00	+	2.16E-01	5.88E-02	-	2.29E+00	2.91E-01	+	1.47E+00	7.07E-01	-
f13	**2.49E-01**	**3.21E-02**	1.31E+00	1.70E-01	+	4.67E-01	6.16E-01	+	3.38E-01	4.88E-02	+	6.61E+00	4.79E-01	+	5.86E-01	1.44E-01	+
f14	2.51E-01	3.53E-02	1.57E+01	4.61E+00	+	8.89E-01	2.34E+00	+	**1.79E-01**	**3.12E-02**	-	2.01E+02	2.24E+01	+	4.21E-01	2.12E-01	+
f15	1.39E+01	7.80E-01	1.51E+03	1.48E+00	+	2.95E+05	2.04E+06	+	**8.21E+00**	**1.97E+00**	-	1.30E+06	4.74E+05	+	2.70E+01	1.34E+01	+
f16	1.19E+01	3.28E-01	1.62E+01	6.85E+00	+	**1.09E+01**	**1.99E+00**	-	1.28E+01	4.30E-01	+	1.31E+01	1.79E-01	+	1.43E+01	3.80E-01	+
f17	**4.80E+02**	**2.34E+02**	4.42E+03	9.19E+02	+	5.35E+06	2.59E+07	+	6.98E+04	3.19E+05	+	2.25E+07	8.63E+06	+	1.77E+03	3.91E+02	+
f18	**7.33E+01**	**2.28E+01**	2.50E+03	2.66E+02	+	7.94E+02	3.68E+03	+	4.47E+04	3.15E+05	+	9.95E+08	3.53E+08	+	1.43E+02	4.82E+01	+
f19	**5.37E+00**	**6.72E-01**	2.04E+02	2.19E+01	+	3.68E+01	1.01E+02	+	9.82E+00	1.42E+00	+	2.78E+02	3.94E+01	+	3.42E+01	4.02E+01	+
f20	**5.77E+01**	**2.14E+01**	2.76E+03	2.45E+02	+	1.92E+03	9.92E+03	+	5.40E+02	3.42E+02	+	9.63E+04	3.81E+04	+	3.84E+02	1.24E+02	+
f21	**4.85E+02**	**1.96E+02**	3.53E+03	1.22E+03	+	1.81E+06	1.24E+07	+	4.35E+04	2.65E+05	+	5.48E+06	2.46E+06	+	1.15E+03	3.80E+02	+
f22	2.29E+02	1.06E+02	5.17E+03	1.94E+02	+	**2.19E+02**	**1.00E+02**	-	4.98E+02	1.41E+02	+	1.43E+03	1.85E+02	+	8.63E+02	3.30E+02	+
f23	3.15E+02	6.61E-03	**2.81E+02**	**1.43E+01**	-	3.15E+02	2.30E-13	+	3.14E+02	8.99E+00	-	7.45E+02	7.97E+01	+	3.15E+02	4.09E-08	+
f24	2.21E+02	7.79E+00	2.76E+02	1.22E-01	+	2.28E+02	1.47E+01	+	**2.16E+02**	**2.32E+01**	-	4.06E+02	1.29E+01	+	4.78E+02	3.61E+02	+
f25	**2.03E+02**	**2.74E-01**	2.09E+02	9.14E+00	+	2.05E+02	3.61E+00	+	2.07E+02	9.31E+00	+	2.63E+02	7.69E+00	+	2.26E+02	1.61E+01	+
f26	**1.00E+02**	**3.24E-02**	2.75E+02	4.28E+01	+	1.24E+02	6.59E+01	+	1.00E+02	9.08E-02	+	1.06E+02	6.60E-01	+	1.93E+02	1.73E+02	+
f27	3.34E+02	4.34E+01	4.50E+03	8.76E+02	+	**3.02E+02**	**2.40E+00**	-	4.19E+02	1.40E+02	+	8.82E+02	8.61E+01	+	9.03E+02	4.26E+02	+
f28	**8.47E+02**	**9.35E+01**	5.98E+03	1.36E+02	+	9.72E+02	2.01E+02	+	2.54E+03	9.65E+02	+	3.58E+03	5.23E+02	+	7.88E+03	3.12E+03	+
f29	**8.24E+02**	**6.91E+01**	1.27E+07	6.28E+06	+	2.42E+05	1.71E+06	+	3.59E+03	2.89E+03	+	8.34E+07	3.53E+07	+	4.19E+06	1.17E+07	+
f30	**1.74E+03**	**8.95E+02**	7.18E+05	2.96E+05	+	4.92E+04	2.35E+05	+	7.65E+03	3.29E+03	+	8.96E+05	2.79E+05	+	4.36E+03	1.62E+03	+
+/=/-			25/0/5			18/3/9			22/0/8			30/0/0			23/1/6		

better than GWO, GOA, MVO, SCA, SHO, SSA, SOA, and WOA on 22, 28, 21, 23, 22, 21, 21, and 23 out of 30 problems, respectively. The performance of SS is outperformed by GWO, GOA, MVO, SCA, SHO, SSA, SOA, and WOA on seven, zero, eight, three, six, eight, seven, and seven out of 30 problems , respectively.

In addition, the Friedman Test (FT) is also used to detect the significant differences between SS and the other 23 algorithms on all 30 problems of CEC-2014 problem suite. The detailed results of the FT for all 24 algorithms are shown in Table 3.10. From Table 3.10, it can be found that SS is ranked first by FT among all 24 algorithms. Variants of PSO: BB-PSO, CLPSO, OLPSO, and Variants of DE: CoBiDE, RSDE, POBL-ADE, and DE-best are very competitive with the SS but the performance of SS is slightly better than them. Similarly, variants of CMA-ES: I-POP-CMAES and LS-CMAES are also well performed on the CEC-2014 problem suite, but they could not outperforms the performance of SS. In the case of recently proposed algorithms, MVO, SOA, WOA, SSA, and GWO perform very well, but the performance of SS is significantly better than them. Compared with the rest of other algorithms, SS is significantly outperformed them.

3.3.7 Application of SS Algorithm for Initial Seed for Power Flow

SS Algorithm can be utilized (by minimizing the objective function described in Equation 3.3) to calculate the initial seed for conventional PF algorithm. In ill-conditioned test systems or heavily loaded systems, the steady-state PF solution is far from flat start. Consequently, the conventional algorithms diverges or converges at a very low rate on these problems because of flat start.

In this section, SS algorithm is used to calculate the initial seed for conventional NR algorithm. This approach is named as Spherical Search with Three Phase Current Injection Method (SSTCIM) for further reference. Test systems and algorithms described in Chapter 2 are considered for analyzing the performance of SSTCIM.

Validation of Algorithm

CASE25 is considered to validate the accuracy of the algorithm. The loading condition of this test system is increased to a level (700%) where TCIM diverges.

Table 3.4: Mean and SD of best error value obtained in 51 independent runs by SS, GWO, GOA, MVO, and SCA on 30-D CEC2014 problem suite (Mean: Mean of best error, SD: Standard deviation of best error, W: result of Wilcoxon signed rank test).

Nos	SS		GWO			GOA			MVO			SCA		
	Mean	**STD**	**Mean**	**STD**	**WT**	**Mean**	**STD**	**WT**	**Mean**	**STD**	**WT**	**Mean**	**STD**	**WT**
1	**8.75E+03**	**6.70E+03**	4.22E+07	2.57E+07	+	1.99E+09	1.73E+09	+	1.96E+07	2.54E+07	+	1.58E+08	1.06E+08	+
2	**0.00E+00**	**0.00E+00**	1.06E+09	1.33E+09	+	8.13E+10	6.70E+10	+	4.58E+08	1.17E+09	+	1.01E+10	7.71E+09	+
3	**1.03E-07**	**2.81E-07**	2.52E+04	8.00E+03	+	1.30E+06	2.57E+06	+	9.61E+03	1.27E+04	+	3.16E+04	8.77E+03	+
4	**0.00E+00**	**0.00E+00**	1.85E+02	4.27E+01	+	2.36E+04	2.00E+04	+	1.29E+02	5.98E+01	+	6.69E+02	4.17E+02	+
5	2.09E+01	5.88E-02	2.09E+01	5.92E-02	=	2.12E+01	2.13E-01	+	2.04E+01	4.35E-01	-	2.09E+01	5.67E-02	=
6	**2.79E-01**	**5.86E-01**	1.20E+01	3.08E+00	+	3.37E+01	1.81E+01	+	1.01E+01	2.95E+00	+	2.48E+01	1.09E+01	+
7	**0.00E+00**	**0.00E+00**	7.62E+00	4.23E+00	+	7.56E+02	6.17E+02	+	3.03E+00	4.57E+00	+	8.74E+01	6.84E+01	+
8	1.59E+02	1.00E+01	**6.81E+01**	**1.99E+01**	-	3.21E+02	2.08E+02	+	7.07E+01	2.05E+01	-	1.69E+02	8.60E+01	=
9	1.61E+02	1.04E+01	**8.77E+01**	**2.62E+01**	-	4.11E+02	2.58E+02	+	9.17E+01	2.79E+01	-	2.04E+02	9.14E+01	+
10	6.32E+03	2.55E+02	1.90E+03	5.38E+02	-	6.43E+03	3.73E+03	=	2.33E+03	6.69E+02	-	4.31E+03	2.05E+03	-
11	6.57E+03	3.04E+02	**2.60E+03**	**6.05E+02**	-	6.68E+03	3.37E+03	=	2.79E+03	6.06E+02	-	5.20E+03	2.18E+03	-
12	2.29E+00	3.10E-01	1.70E+00	1.12E+00	-	4.55E+00	2.44E+00	+	**8.37E-01**	**1.06E+00**	-	2.21E+00	8.28E-01	=
13	**2.49E-01**	**3.21E-02**	3.55E-01	8.61E-02	+	6.62E+00	5.12E+00	+	4.03E-01	8.31E-02	+	1.89E+00	1.24E+00	+
14	**2.51E-01**	**3.53E-02**	5.76E-01	5.66E-01	+	2.46E+02	2.05E+02	+	5.08E-01	2.95E-01	+	2.70E+01	2.24E+01	+
15	1.39E+01	7.80E-01	6.22E+01	2.75E+02	+	1.16E+07	1.24E+07	+	**1.09E+01**	**7.81E+00**	-	1.86E+03	2.69E+03	+
16	1.19E+01	3.28E-01	1.07E+01	7.17E-01	-	1.29E+01	1.81E+00	+	1.12E+01	7.05E-01	-	1.19E+01	1.11E+00	=
17	**4.80E+02**	**2.34E+02**	1.06E+06	1.35E+06	+	1.91E+08	1.99E+08	+	4.33E+05	6.98E+05	+	4.22E+06	3.53E+06	+
18	**7.33E+01**	**2.28E+01**	5.69E+06	1.70E+07	+	5.34E+09	4.89E+09	+	1.42E+06	7.13E+06	+	8.57E+07	8.46E+07	+
19	**5.37E+00**	**6.72E-01**	2.59E+01	1.76E+01	+	6.81E+02	6.08E+02	+	1.74E+01	1.47E+01	+	6.19E+01	3.52E+01	+
20	**5.77E+01**	**2.14E+01**	1.28E+04	8.64E+03	+	9.40E+06	1.70E+07	+	5.95E+03	8.75E+03	+	1.34E+04	5.86E+03	+
21	**4.85E+02**	**1.96E+02**	5.42E+05	1.16E+06	+	9.40E+07	9.20E+07	+	2.85E+05	8.76E+05	+	1.14E+06	1.11E+06	+
22	2.29E+02	1.06E+02	3.79E+02	1.87E+02	+	1.34E+04	2.32E+04	+	3.65E+02	1.47E+02	+	6.32E+02	2.64E+02	+
23	3.15E+02	6.61E-03	3.27E+02	4.28E+00	+	1.28E+03	8.31E+02	+	3.20E+02	5.67E+00	+	3.54E+02	2.64E+01	+
24	2.21E+02	7.79E+00	**2.00E+02**	**1.01E-03**	-	4.04E+02	1.67E+02	+	2.15E+02	1.49E+01	=	2.00E+02	2.69E-01	-
25	**2.03E+02**	**2.74E-01**	2.10E+02	3.07E+00	+	3.34E+02	1.07E+02	+	2.07E+02	3.24E+00	+	2.20E+02	1.01E+01	+
26	**1.00E+02**	**3.24E-02**	1.16E+02	3.66E+01	+	2.42E+02	1.35E+02	+	1.16E+02	3.66E+01	+	1.11E+02	2.96E+01	+
27	**3.34E+02**	**4.34E+01**	5.83E+02	1.16E+02	+	1.33E+03	6.36E+02	+	5.41E+02	1.24E+02	+	6.65E+02	2.52E+02	+
28	**8.47E+02**	**9.35E+01**	9.61E+02	1.47E+02	+	5.30E+03	3.58E+03	+	9.97E+02	2.04E+02	+	1.58E+03	6.04E+02	+
29	**8.24E+02**	**6.91E+01**	8.97E+04	1.91E+05	+	4.89E+08	4.37E+08	+	9.42E+05	3.12E+06	+	8.13E+06	8.72E+06	+
30	**1.74E+03**	**8.95E+02**	2.95E+04	2.16E+04	+	6.43E+06	6.54E+06	+	1.54E+04	1.80E+04	+	1.57E+05	1.18E+05	+
+/-/=			22/1/7			28/2/0			21/1/8			23/4/2		

Table 3.5: Initial seed obtained by SS for CASE25 test system

Bus	$\lvert V_a \rvert$	$\angle_a$	$\lvert V_b \rvert$	$\angle_b$	$\lvert V_c \rvert$	$\angle_c$
1	1.0000	0.0000	1.0000	-120.0000	1.0000	120.0000
2	0.9627	-0.2581	0.9634	-120.2113	0.9617	119.7596
3	0.9510	-0.2880	0.9503	-120.2504	0.9503	119.7224
4	0.9454	-0.2769	0.9440	-120.2889	0.9454	119.7125
5	0.9440	-0.2422	0.9421	-120.2575	0.9440	119.7595
6	0.9492	-0.1879	0.9519	-120.1742	0.9465	119.8961
7	0.9382	-0.1416	0.9430	-120.1587	0.9333	119.9966
8	0.9468	-0.1644	0.9495	-120.1403	0.9443	119.9356
9	0.9346	-0.1146	0.9402	-120.1761	0.9285	120.0467
10	0.9328	-0.0914	0.9384	-120.2034	0.9250	120.0902
11	0.9318	-0.0814	0.9375	-120.2194	0.9235	120.1141
12	0.9311	-0.0763	0.9364	-120.2247	0.9225	120.1296
13	0.9318	-0.0741	0.9371	-120.2304	0.9230	120.1223
14	0.9298	-0.1317	0.9356	-120.1310	0.9243	120.0071
15	0.9258	-0.1250	0.9319	-120.1207	0.9207	120.0080
16	0.9374	-0.1362	0.9425	-120.1518	0.9323	120.0185
17	0.9291	-0.1279	0.9354	-120.1407	0.9232	120.0141
18	0.9413	-0.1632	0.9397	-120.0547	0.9404	119.9511
19	0.9326	-0.0640	0.9318	-119.9427	0.9327	120.1612
20	0.9370	-0.1131	0.9355	-119.9851	0.9363	120.0538
21	0.9355	-0.0805	0.9325	-119.9294	0.9336	120.1154
22	0.9322	-0.0403	0.9277	-119.8592	0.9299	120.2074
23	0.9405	-0.1736	0.9386	-120.2086	0.9409	119.8227
24	0.9377	-0.1045	0.9352	-120.1619	0.9380	119.9003
25	0.9340	-0.0334	0.9318	-120.1282	0.9346	119.9908

Table 3.6: Power Flow solution obtained using SSTCIM for CASE25 test system

Bus	$\|V_a\|$	$\angle_a$	$\|V_b\|$	$\angle_b$	$\|V_c\|$	$\angle_c$
1	1.0000	0.0000	1.0000	-120.0000	1.0000	120.0000
2	0.9227	-1.4532	0.9262	-121.0821	0.9378	118.3006
3	0.9051	-1.8118	0.9096	-121.3556	0.9240	117.8898
4	0.8964	-1.9860	0.9017	-121.5027	0.9179	117.7016
5	0.8937	-1.9832	0.8992	-121.4928	0.9156	117.6988
6	0.8827	-1.4361	0.8865	-120.9406	0.9020	118.2451
7	0.8481	-1.4268	0.8523	-120.7969	0.8704	118.1928
8	0.8773	-1.4304	0.8813	-120.9200	0.8972	118.2392
9	0.8321	-1.4260	0.8361	-120.7363	0.8563	118.1686
10	0.8204	-1.4268	0.8234	-120.6928	0.8453	118.1519
11	0.8148	-1.4282	0.8175	-120.6759	0.8403	118.1495
12	0.8121	-1.4243	0.8142	-120.6664	0.8376	118.1635
13	0.8129	-1.4239	0.8150	-120.6730	0.8382	118.1554
14	0.8324	-1.4208	0.8371	-120.7164	0.8553	118.1648
15	0.8266	-1.4144	0.8316	-120.6936	0.8503	118.1588
16	0.8453	-1.4237	0.8496	-120.7859	0.8679	118.1898
17	0.8292	-1.4124	0.8344	-120.7105	0.8518	118.1695
18	0.8902	-1.8089	0.8950	-121.3038	0.9103	117.8768
19	0.8777	-1.7954	0.8844	-121.2805	0.8995	117.9027
20	0.8838	-1.8098	0.8893	-121.2838	0.9045	117.8736
21	0.8812	-1.7924	0.8855	-121.2807	0.9008	117.9060
22	0.8763	-1.7857	0.8794	-121.2644	0.8958	117.9301
23	0.8880	-1.9842	0.8945	-121.4938	0.9115	117.6906
24	0.8829	-1.9836	0.8898	-121.4983	0.9073	117.6852
25	0.8768	-1.9690	0.8852	-121.5176	0.9025	117.6946

The obtained initial seed using SS is reported in Table 3.5. TCIM algorithm uses this initial seed to solve the PF problem of CASE25. Consequently, TCIM converges within 4 iterations. The obtained PF solution is reported in Table 3.6.

From the outcomes, it can be concluded that the proposed approach improves the performance of conventional algorithms. For further analysis, this approach is also analyzed on the test systems with different loading condition with different R/X ratio of lines.

Test systems with high loading conditions

In this section, the stability of the proposed approach is evaluated on various test systems with different loading conditions. The loading level at the buses of the different test systems is gradually increased to their maximum loading limit. Two test systems, CASE37

Table 3.7: Total Number of iterations required for different Power Flow algorithms in heavily loaded ill-conditioned systems.(LF: Loading Factor, NC: Not Converged)

CASE37						
LF(%)	CINR	LMPF	RK4PF	SSTCIM	TCIM	iTCIM
200	2	3	16	3	3	3
600	2	3	17	3	3	3
1000	2	3	18	3	3	3
1400	3	4	18	3	4	4
1800	4	21	59	4	NC	NC
2200	4	24	87	4	NC	NC
2400	NC	26	88	4	NC	NC
2500	NC	37	91	4	NC	NC

CASE84						
LF(%)	CINR	LMPF	RK4PF	SSTCIM	TCIM	iTCIM
100	3	2	26	3	6	4
200	4	4	28	3	13	7
300	6	5	29	3	20	9
400	9	6	29	3	33	11
500	17	9	29	4	69	15
600	NC	40	43	4	NC	NC
700	99	38	63	4	NC	NC
800	31	35	87	4	NC	NC

and CASE84, are considered for this analysis. A total number of iterations required by different algorithms for CASE37 and CASE84 are reported in Table 3.7.

It is observed from this table that the proposed approach improves the performance of TCIM. In CASE37, the performance of SSTCIM is better or at least competitive with the algorithms discussed in chapter 2.

Test systems with high R/X ratio

The sensitivity of the proposed approach is validated for different R/X ratios of the lines of test systems and the performance of the proposed approach is compared with other algorithms. In this study, CASE37 and CASE84 are considered with different R/X ratios.

Number of iterations required to converge by proposed algorithms with other algorithms are reported in Table 3.8. It can be observed that the proposed approach improves the performance of TCIM.

Table 3.8: Total Number of iterations required for different Power Flow algorithms in ill-conditioned systems with high R/X ratios.(NC: Not Converged)

CASE37						
R/X	CINR	LMPF	RK4PF	SSTCIM	TCIM	iTCIM
2	2	3	15	3	3	3
6	2	4	15	3	3	3
10	2	5	15	3	3	3
14	3	6	15	4	4	4
18	3	18	50	4	NC	NC
22	3	29	77	4	NC	NC
24	4	22	78	4	NC	NC
25	8	38	77	4	NC	NC

CASE84						
R/X	CINR	LMPF	RK4PF	SSTCIM	TCIM	iTCIM
1	3	2	26	3	6	4
4	4	4	28	3	14	6
7	5	5	27	4	26	8
10	6	7	27	4	48	10
13	16	16	27	3	NC	20
14	21	43	59	4	NC	NC
15	21	57	87	4	NC	NC
16	21	39	94	4	NC	NC

3.4 Butterfly Constrained Optimizer

BCO is a dual population based method for solving constrained optimization problems. BCO uses υ-Constrained method for handling constraints of the problem. In this section, details of different processes of BCO are described. Afterwards, the complete process of BCO is presented.

3.4.1 Dual-population of BCO

In BCO, there are two populations based on their feasibility. First population, Type-I, contains solutions to minimize the constraint violation and second population, Type-II, contains solution to minimize the constraint violation and optimize the objective function value. Type-I population is defined as a N, D-dimensional vector. If k denotes the iteration, the type-I population, P_1, at k^{th} iteration consist of :

$$P_1^k = [\bar{x}_1^k, \bar{x}_2^k....\bar{x}_N^k] \tag{3.15}$$

$$\bar{x}_i^k = [x_{i1}^k, x_{i2}^k, x_{i3}^k....x_{iD}^k]^T, i = 1, 2, ...N \tag{3.16}$$

The Type-II population is defined as N, D-dimensional vector. If k denotes the iteration, the Type-II population, P_2 at k^{th} iteration consist of :

$$P_2^k = [m\bar{x}_1^k, m\bar{x}_2^k....m\bar{x}_N^k] \tag{3.17}$$

$$m\bar{x}_i^k = [mx_{i1}^k, mx_{i2}^k, mx_{i3}^k....mx_{iD}^k]^T, i = 1, 2, ...N \tag{3.18}$$

where N is the size of population.

The idea behind the BCO is to take advantage of perching and patrolling modes to search new solution for each $\bar{x}_i^k$ and $m\bar{x}_i^k$. The set of solution $m\bar{x}_i^k$ and $\bar{x}_i^k$ is associated with i^{th} individual at k^{th} iteration. Initial dual solutions of each individual are randomly generated with in the search space. At every iteration, before perching or patrolling, a cris-cross neighbor vector $\bar{cc}$ of length N is calculated by reshuffling the integers from 1 to N. The vector $\bar{cc}$ consists of :

$$\bar{cc} = \{cc_1, cc_2, ...cc_i\}, i = 1, 2, N \tag{3.19}$$

Table 3.9: Mean and SD of best error value obtained in 51 independent runs by SS, SHO, SSA, SOA, and WOA on 30-D CEC2014 problem suite (Mean: Mean of best error, SD: Standard deviation of best error, W: result of Wilcoxon signed rank test).

Nos	SS		SHO			SSA			SOA			WOA		
	Mean	STD	Mean	STD	WT	Mean	STD	WT	Mean	STD	WT	Mean	STD	WT
1	**8.75E+03**	**6.70E+03**	1.80E+07	2.65E+07	+	1.94E+07	2.56E+07	+	5.58E+07	1.15E+07	+	1.97E+07	1.81E+07	+
2	**0.00E+00**	**0.00E+00**	4.58E+08	1.17E+09	+	4.58E+08	1.17E+09	+	1.21E+08	1.36E+06	+	1.26E+08	1.20E+07	+
3	**1.03E-07**	**2.81E-07**	9.99E+03	1.24E+04	+	1.15E+04	1.13E+04	+	4.04E+04	1.35E+03	+	1.87E+04	2.03E+04	+
4	**0.00E+00**	**0.00E+00**	1.19E+02	6.55E+01	+	1.34E+02	5.71E+01	+	1.58E+02	4.68E-01	+	1.19E+02	5.68E+01	+
5	2.09E+01	5.88E-02	2.04E+01	4.58E-01	-	**2.04E+01**	**4.28E-01**	-	2.09E+01	3.12E-02	=	2.09E+01	7.09E-02	-
6	**2.79E-01**	**5.86E-01**	2.20E+01	8.92E+00	+	1.63E+01	5.97E+00	+	1.24E+01	2.27E-02	+	1.74E+01	3.35E+00	+
7	**0.00E+00**	**0.00E+00**	3.00E+00	4.59E+00	+	2.99E+00	4.60E+00	+	7.65E+00	1.50E-02	+	1.88E+00	2.68E-02	+
8	1.59E+02	1.00E+01	8.80E+01	2.32E+01	-	9.90E+01	3.48E+01	-	8.16E+01	4.92E-01	-	1.48E+00	2.31E-01	-
9	1.61E+02	1.04E+01	1.25E+02	3.56E+01	-	1.13E+02	3.88E+01	-	8.68E+01	5.12E-01	-	9.63E+01	2.40E+01	-
10	6.32E+03	2.55E+02	1.98E+03	5.42E+02	-	2.99E+03	1.09E+03	-	2.08E+03	2.65E+00	-	**1.34E+03**	**5.27E+02**	-
11	6.57E+03	3.04E+02	3.37E+03	9.00E+02	-	3.34E+03	8.63E+02	-	2.11E+03	5.12E+00	-	2.46E+03	4.95E+02	-
12	2.29E+00	3.10E-01	1.05E+00	9.54E-01	-	1.02E+00	9.93E-01	-	1.54E-01	1.94E-03	-	1.56E-01	4.62E-02	-
13	**2.49E-01**	**3.21E-02**	4.38E-01	1.17E-01	+	4.51E-01	1.23E-01	+	4.44E-01	9.16E-03	+	5.33E-01	1.27E-01	+
14	**2.51E-01**	**3.53E-02**	3.88E-01	2.16E-01	+	4.49E-01	2.52E-01	+	7.34E-01	4.21E-02	+	3.37E-01	1.03E-01	+
15	1.39E+01	7.80E-01	3.47E+01	1.92E+01	+	1.12E+01	7.76E+00	-	1.91E+01	1.02E-01	+	1.75E+01	1.02E+01	+
16	1.19E+01	3.28E-01	1.18E+01	1.02E+00	=	1.14E+01	8.39E-01	-	**1.06E+01**	**1.57E-03**	-	1.12E+01	6.77E-01	-
17	**4.80E+02**	**2.34E+02**	4.17E+05	7.01E+05	+	4.76E+05	6.84E+05	+	6.52E+05	8.23E+04	+	1.34E+06	2.13E+06	+
18	**7.33E+01**	**2.28E+01**	1.41E+06	7.13E+06	+	1.42E+06	7.13E+06	+	1.23E+04	3.84E+01	+	1.99E+03	3.70E+03	+
19	**5.37E+00**	**6.72E-01**	2.18E+01	1.69E+01	+	1.92E+01	1.38E+01	+	1.23E+01	5.87E-03	+	2.67E+01	3.44E+01	+
20	**5.77E+01**	**2.14E+01**	8.22E+03	7.59E+03	+	6.09E+03	8.66E+03	+	8.75E+03	5.84E+01	+	5.84E+03	1.93E+00	+
21	**4.85E+02**	**1.96E+02**	2.81E+05	8.78E+05	+	3.00E+05	8.74E+05	+	6.95E+05	2.15E+04	+	8.60E+05	7.85E+05	+
22	2.29E+02	1.06E+02	5.73E+02	2.48E+02	+	3.91E+02	1.62E+02	+	3.65E+02	2.56E+01	+	5.69E+02	2.06E+02	+
23	3.15E+02	6.61E-03	3.19E+02	5.88E+00	+	3.20E+02	5.67E+00	+	3.54E+02	1.15E-01	+	3.16E+02	1.01E+00	+
24	2.21E+02	7.79E+00	2.17E+02	1.38E+01	=	2.17E+02	1.55E+01	=	**2.00E+02**	**2.34E-04**	-	2.15E+02	5.15E+00	+
25	**2.03E+02**	**2.74E-01**	2.11E+02	3.43E+00	+	2.11E+02	3.70E+00	+	2.09E+02	5.96E-03	+	2.11E+02	4.59E+00	+
26	**1.00E+02**	**3.24E-02**	1.49E+02	4.93E+01	+	1.10E+02	2.99E+01	+	1.00E+02	2.02E-02	=	1.51E+02	5.13E+01	+
27	**3.34E+02**	**4.34E+01**	7.31E+02	3.32E+02	+	6.40E+02	1.53E+02	+	5.62E+02	1.04E+00	+	7.62E+02	1.98E+02	+
28	**8.47E+02**	**9.35E+01**	2.53E+03	1.47E+03	+	9.85E+02	1.41E+02	+	1.72E+03	4.63E-01	+	1.51E+03	4.53E+02	+
29	**8.24E+02**	**6.91E+01**	5.78E+05	2.14E+06	+	1.05E+06	3.49E+06	+	9.68E+03	1.43E+01	+	8.68E+05	2.67E+06	+
30	**1.74E+03**	**8.95E+02**	1.34E+04	1.90E+04	+	1.77E+04	1.73E+04	+	3.16E+04	6.56E+02	+	1.38E+04	1.98E+02	+
+/-/=			22/2/6			21/1/8			21/2/7			23/0/7		

Table 3.10: Ranking of Algorithm according to Friedman ranking based on mean error value. (FR: Friedman Ranking)

S.N.	Algorithm	FR	Rank	S.N.	Algorithm	FR	Rank
1	**SS**	**5.4667**	**1**	13	I-POP-CMAES	11.4000	10
2	PSO	15.5667	19	14	LS-CMAES	10.6500	9
3	BB-PSO	8.4833	6	15	CMSAES	21.4333	23
4	CLPSO	7.4167	4	16	(1+1)cholesky CMAES	14.0167	18
5	APSO	18.2333	21	17	GWO	14.0000	16
6	OLPSO	7.1000	3	18	GOA	22.5500	24
7	CoBiDE	8.7833	7	19	MVO	12.4500	12
8	FCDE	11.6667	11	20	SCA	18.3167	22
9	RSDE	5.6500	2	21	SHO	14.0000	17
10	POBL_ADE	7.4833	5	22	SSA	13.8333	15
11	DE_best	9.7000	8	23	SOA	12.6500	13
12	CMA-ES	15.8333	20	24	WOA	13.3167	14

3.4.2 Perching

Perching can be explained by dividing its functions into five processes: Cris-cross Modification, Exponential Crossover, Solution Repair, Selection-I and Selection-II.

Cris-cross modification

Cris-cross modification process generates the initial form of perching trial positions vector, $\bar{tx_i}^{k+1}$.

$$\bar{tx_i}^{k+1} = m\bar{x}_{cc_i}{}^{k} + F(m\bar{x}_{q_i}{}^{k} - \mathfrak{R}(\bar{x_{r_i}}^{k}, m\bar{x}_{r_i}{}^{k})) \tag{3.20}$$

where F is the scaling factor which controls the amplitude of the search direction, cc_i is the cris-cross neighbour of i^{th} butterfly. Index q_i and r_i are randomly selected neighbour for i^{th} butterfly. Random selection of indices q_i and r_i are done in such a manner that it satisfies equation 3.21

$$i \neq cc_i \neq q_i \neq r_i \tag{3.21}$$

$\mathfrak{R}(a, b)$ is random selection operator, where probability of selection of a and b is equal. $F \in [0, 1]$ is a real constant.

Exponential recombination

Exponential recombination process generates the final form of perching trial position vector, $\bar{tx_i}^{k+1}$. For exponential recombination, the starting recombination point is selected randomly from 1 to D, and l circular consecutive elements are not changed in $\bar{tx_i}$ and other elements are taken from $\bar{mx_i}$ and replaced to $\bar{tx_i}$. The pseudo code to generate l is

$l = 0$

$while((rand(0,1) \leq Cr)\&\&(l \leq D))$

$Do\{l = l + 1\}$

where Cr is recombination probability, D is the problem dimension $rand(0,1)$ is a uniformly distributed random number generator. Final form of $\bar{tx_i}$ is generated as

$$\bar{tx_i}^{k+1} = \begin{cases} \bar{tx}_i^{k+1}, \text{ for } j = r, r+1, r+2.....r+l-1 \\ \bar{mx}_i^k, \text{for other} j \epsilon [1, D] \end{cases} \tag{3.22}$$

Solution Repair

In solution repair, firstly the feasibility of the perching trial solution, $\bar{tx_i}$ is checked. If the solution, $\bar{tx_i}$, is feasible then solution repair is not required and it switches to next process. If solution, $\bar{tx_i}$, is infeasible, then generate a random number in range of $[0,1]$ and compare with p_r. If random number is less than P_r, then this solution is repaired by multi-order Levenberg Marquardt method. Otherwise, it switches to next process without repairing the solution.

Selection-I

Selection-I process generates the Type-I population, P_1, of next iteration. Selection of $\bar{tx_i}^k$ in place of $\bar{x}_i^k$ at k^{th} iteration is decided by the constraint violation. In selection-I, individual having lower constraint violation is selected for the next iteration. Procedure of selection-I is given by equation 3.23

$$\bar{x}_i^{k+1} = \begin{cases} \bar{tx}_i^k, & \text{if } \phi^{(v)}(\bar{tx}_i^k) \leq \phi^{(v)}(\bar{x}_i^k) \\ \bar{x}_i^k, & \text{otherwise.} \end{cases} \tag{3.23}$$

Selection-II

Selection-II is the last process of the perching. It generates the the type-II population, P_2, of next iteration. Selection of $\bar{tx}_i^k$ in place of $\bar{mx}_i^k$ at k^{th} iteration is decided by v-level comparison. It is done by equation 3.24

$$\bar{mx}_i^{k+1} = \begin{cases} \bar{tx}_i^k, & \text{if } (f(\bar{tx}_i^k), \phi^{(v)}(\bar{tx}_i^k)) \leq_v (f(\bar{x}_i^k), \phi^{(v)}(\bar{x}_i^k)) \\ \bar{mx}_i^k, & \text{otherwise.} \end{cases} \tag{3.24}$$

3.4.3 Patrolling

Patrolling can be explained by its function into four processes: Towards-best modification, exponential recombination, Selection-I, Selection-III.
Exponential recombination and selection-I is same as perching. So, Towards-best modification and Selection-III are discussed below.

Towards-best modification

This process generates initial form of patrolling trial position vectors, $\bar{ux}_i^{k+1}$.

$$\bar{ux}_i^{k+1} = \bar{mx}_i^k + F(\bar{mv}_i^k + \bar{mx}_{maxuv_i}^k - \bar{mx}_i^k) \tag{3.25}$$

where F is scaling factor, $maxuv_i$ is the most attractive neighbour of i^{th} butterfly.

Selection-III

Selection-III is same as selection-II, but only difference is selection-III also update the velocity vector, $\bar{mv}_i^k$. Velocity vector updating is done by equation 3.26

$$\bar{mv}_i^{k+1} = \begin{cases} \bar{ux}_i^k - \bar{mx}_i^k, \\ \text{if } (f(\bar{tx}_i^k), \phi^{(v)}(\bar{tx}_i^k)) \leq_v (f(\bar{x}_i^k), \phi^{(v)}(\bar{x}_i^k)) \\ F(\bar{mv}_i^k + \bar{mx}_{cc_i} - \bar{mx}_i^k), \\ \text{otherwise.} \end{cases} \tag{3.26}$$

3.4.4 Selection of Perching and Patrolling Operator

Selection of Perching and Patrolling operator depends upon the probability vector of operator selection, $Prob$.

$$Prob^k = [prob_1^k, prob_2^k.........prob_N^k] \tag{3.27}$$

where $prob_i^k$ is the probability of operator selection of i^{th} individual at k^{th} iteration. it is the ratio of total number of successful Selection-II run, $suc1_i^k$, and sum of total number of successful selection-II run, $suc1_i^k$ and selection-III run, $suc2_i^k$. It is given by:

$$prob_i^k = \frac{suc1_i^k}{suc1_i^k + suc2_i^k} \tag{3.28}$$

For every individual, a uniformly distributed random number of range $[0, 1]$ is generated. If this number is less than $prob_i^k$, then i^{th} individual selects Perching operator. Otherwise, i^th individual selects Patrolling operator.

3.4.5 Selection of Maximum Attractive Butterfly, $maxuv_i$

Selection of max attractive butterfly, $maxuv_i$ is done in patrolling. Three different neighbour is selected for every individual after every 10 patrolling attempts. Their ranking is done by according to objective function values and constraint violation. Best rank of them is selected as the $maxuv_i$ of i^{th} iteration.

3.4.6 Controlling of the υ-level

A simple way of controlling of the ϵ-level is described in [146]. A similar procedure is used in this study to control the υ-level. The initial υ-level is the ratio of constraint violation of top θ^{th} individual and total number of constraint.

$$\upsilon(0) = \frac{\phi(\bar{mx}_\theta)}{n_h + n_g} \tag{3.29}$$

where n_h and n_g are total number of equality and inequality constraint of problem respectively.

The υ-level is decreasing with increase of the iterations and become zero after current iteration T_c. The υ-level updating is done by equation 3.30

$$\upsilon(k) = \begin{cases} \upsilon(0)(1 - \frac{k}{T_c})^{cp}, & \text{if } 0 \leq k \leq T_c \\ 0, & \text{otherwise.} \end{cases} \tag{3.30}$$

where $\theta = 0.2N$, cp is the parameter to control the speed of decreasing of value, $\upsilon(k)$.

3.4.7 Reflecting Back and Cutting-off

If an individual moves outside the search space, there is need to apply operations like reflecting back and cutting off. Reflecting back and cutting off generate new solution inside the search space. In this study both methods are used.

Reflecting back operation

$$x_{ij}^k = \begin{cases} l_j + (l_j - x_{ij}^k) - \lfloor \frac{l_j - x_{ij}^k}{u_j - l_j} \rfloor (u_j - l_j), & \text{if } x_{ij}^k \leq l_j \\ u_j + (x_{ij}^k - u_j) + \lfloor \frac{x_{ij}^k - u_j}{u_j - l_j} \rfloor (u_j - l_j), & \text{if } x_{ij} \geq u_j \\ x_{ij}^k, & \text{otherwise.} \end{cases} \tag{3.31}$$

where $\lfloor z \rfloor$ is a floor function. Reflecting back operations used in Perching.

Cutting off

$$x_{ij}^k = \begin{cases} l_j, & \text{if } x_{ij}^k \leq l_j \\ u_j, & \text{if } x_{ij}^k \geq u_j \\ x_{ij}^k, & \text{otherwise.} \end{cases} \tag{3.32}$$

This operation is applied to Patrolling.

3.4.8 Validation of Butterfly Constrained Optimizer on Benchmark Problems

In this section, BCO is used to solve the problems presented in CEC 2006 [147] to verify the performance of BCO on different type of constrained problems. Results obtained from experiments are compared with the results of other state of arts constrained optimization techniques for comparative analysis. Constrained variant of other meta-heuristics are grouped based on their underlying techniques: PSO, GA, ES, CMA-ES, and DE etc.

In the experiments, parameter tuning of BCO is done by preliminary parameter analysis and parameter setting of BCO is given in Discussion section.

Experimental Settings

The properties of the benchmark problems given in CEC 2006 are given in Table 3.11. Table 3.11 shows that the benchmark problems containing different types of problems:

$g10, g20, g21, g22, g23$, and $g24$ are linear, $g02, g08, g13, g14, g16, g17$, and $g19$ are non-linear, $g03$, and $g09$ are polynomial, $g05$, and $g06$ are cubic, and $g01, g04, g07, g11, g12, g15$, and $g18$ are quadratic. In the Table 3.11, ρ represent the estimated feasible region in search space and it is calculated by the 1000000 solution samples. Most of the benchmark problems have very low feasible region and also hard to locate the even feasible region. Benchmark problems have different type of constraints which includes linear inequality(LI), non-linear inequality(NI), linear equality(LE), and non-linear equality(NE) constraint. The number of constraint is also very from 1 to 38.

For test function $g20$, an improved best known infeasible solution has been reported in this paper. The best known infeasible solution reported in [147] for test function $g20$ is $x^* = [1.285823e-18, 4.834603e-34, 0, 0, 6.304599e-18, 7.571925e-34, 5.033507e-34, 9.282681e-34, 0, 1.767234e-17, 3.556861e-34, 2.994139e-34, 0.158143, 2.296018e-19, 1.061069e18, 1.319683e-18, 0.530903, 0, 2.891483e-18, 3.348921e-18, 0, 0.311000, 5.412446e-05, 4.849931e-16]$ with $f(x^*) = 0.2049794002$. This solution violates the 11 (10 equality and 1 inequality) constraints with total constraint violation $3.975737e14$. The improved solution found in this paper for test function $g20$ is $x^* = [1.29089e-01, 6.009131e-14, 1.448197e-11, 1.856528e-12, 1.512278e-01, 3.056017e-11, 3.664426e-12, 8.944519e-13, 4.294914e-13, 3.773682e-12, 1.367320e-11, 1.085285e-11, 3.992256e-01, 2.428499e-05, 2.428468e-05, 5.741080e-05, 3.202754e-01, 7.140807e-05, 2.611374e-05, 1.1993356e-10, 1.426029e-11, 3.447475e-05, 1.925004e-05, 2.511113e-05]$ with $f(x^*) = 0.1592683574$. This solution violates only 1 inequality constraint with total constraint violation 0.481498.

General performance of BCO

Twenty five independent runs were performed for each test problems using 5×10^5 FES at most, and the tolerance value δ for the equality constraint was set to 0.0001 as suggested by Liang [147]. The best, median, worst, mean and standard deviation of the error value $(f(x) - f(x^*))$ for the best so far solution after 5×10^3, 5×10^4, and 5×10^5 FEs in each run are recorded in Tables 4.7-3.15.

Tables 4.7-3.15 show that the 3 out of 24 test problems (i.e.$g08$, $g12$, and $g24$) are converged on the feasible optimum solution in every independent run by using 5×10^3 FEs. In 5×10^4 FEs, 12 out of 24 test problems (i.e. $g01$, $g04$, $g06$, $g08$, $g10$, $g11$, $g12$, $g13$, $g15$, $g16$, $g17$, and $g24$) are converged on the feasible optimum solution in every run.

Table 3.11: Properties of benchmark problems given in CEC 2006 [147]

prob	**D**	**Type of Problem**	$\rho(\%)$	**LI**	**NI**	**LE**	**NE**	$\boldsymbol{a}$
$g01$	13	quadratic	0.0111	9	0	0	0	6
$g02$	20	nonlinear	99.9971	0	2	0	0	1
$g03$	10	polynomial	0.0000	0	0	0	1	1
$g04$	5	quadratic	51.1230	0	6	0	0	2
$g05$	4	cubic	0.0000	2	0	0	3	3
$g06$	2	cubic	0.0066	0	2	0	0	2
$g07$	10	quadratic	0.0003	3	5	0	0	6
$g08$	2	nonlinear	0.8560	0	2	0	0	0
$g09$	7	polynomial	0.5121	0	4	0	0	2
$g10$	8	linear	0.0010	3	3	0	0	0
$g11$	2	quadratic	0.0000	0	0	0	1	1
$g12$	3	quadratic	4.7713	0	1	0	0	0
$g13$	5	nonlinear	0.0000	0	0	0	3	3
$g14$	10	nonlinear	0.0000	0	0	3	0	3
$g15$	3	quadratic	0.0000	0	0	1	1	2
$g16$	5	nonlinear	0.0204	4	34	0	0	4
$g17$	6	nonlinear	0.0000	0	0	0	4	4
$g18$	9	quadratic	0.0000	0	13	0	0	0
$g19$	15	nonlinear	33.4761	0	5	0	0	0
$g20$	24	linear	0.0000	0	6	2	12	16
$g21$	7	linear	0.0000	0	1	0	5	6
$g22$	22	linear	0.0000	0	1	8	11	19
$g23$	9	linear	0.0000	0	2	3	1	6
$g24$	2	linear	79.6556	0	2	0	0	2

prob: Benchmark Problem.

D: Number of dimension of problem.

ρ: Feasibility ratio.

LI: Number of linear inequality constraint.

NI: Number of non-linear inequality constraint.

LE: Number of linear equality constraint.

NE: Number of non-linear equality constraint.

a: Number of active inequality constarint.

Rest of the problems are converged on feasible optimum solution with in 5×10^5 FEs for every independent run except $g20$ and $g22$.

To test problem $g20$, the best known solution is slightly infeasible i.e. feasible optimum solution is not available. Proposed method, BCO, is converged on improved solution. For test problem $g22$, 3 runs out of 25 converged on the best known feasible solution by

Table 3.12: Error Values achieved when FEs are 5000, 50000, and 500000 for test function $g01 - g06$

FES		g01	g02	g03	g04	g05	g06
5.0E+03	Min	1.9674E+00	2.0521E-01	4.3547E-01	3.5106E+01	5.1546E+00	**9.0900E-13**
	Median	5.7809E+00	3.1561E-01	7.8524E-01	1.1499E+02	7.6736E+01	**7.2760E-12**
	Worst	8.7350E+00	3.8456E-01	1.0004E+00	2.0169E+02	6.1227E+02	5.7557E+03
	Mean	5.9698E+00	3.0886E-01	7.5429E-01	1.1587E+02	1.3059E+02	1.0047E+03
	Std	1.7661E+00	3.7480E-02	1.7647E-01	4.5865E+01	1.6333E+02	2.1447E+03
5.0E+04	Min	**9.2000E-14**	1.4730E-02	4.5313E-03	**8.4971E-07**	2.1418E-04	**9.0900E-13**
	Median	**9.8400E-13**	3.1546E-02	1.0079E-02	**1.5282E-05**	1.1020E-03	**5.4570E-12**
	Worst	**1.7494E-11**	9.5223E-02	4.0296E-02	2.3949E-04	2.2037E+01	**1.9099E-11**
	Mean	**3.4191E-12**	3.6746E-02	1.2963E-02	**3.6582E-05**	1.3000E+00	**4.6564E-12**
	Std	**4.9143E-12**	2.1116E-02	7.8858E-03	**5.5316E-05**	4.7936E+00	**3.9673E-12**
5.0E+05	Min	**0.0000E+00**	**3.7372E-09**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**9.0900E-13**
	Median	**0.0000E+00**	**1.9797E-08**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**5.4570E-12**
	Worst	**0.0000E+00**	**6.4743E-08**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**1.9099E-11**
	Mean	**0.0000E+00**	**2.6136E-08**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**4.6564E-12**
	Std	**0.0000E+00**	**1.7600E-08**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**3.9673E-12**

Table 3.13: Error Values achieved when FEs are 5000, 50000, and 500000 for test function $g07 - g12$

FES		g07	g08	g09	g10—	g11—	g12—
5.0E+03	Min	3.0814E+01	**3.4510E-12**	1.4601E+01	4.1756E+01	**3.3075E-05**	**1.9717E-08**
	Median	8.5137E+01	**3.9054E-07**	1.2292E+02	2.6334E+03	2.0188E-04	**6.4211E-07**
	Worst	5.6623E+02	7.5612E-04	5.5246E+04	9.8834E+03	1.2911E-01	**4.6084E-06**
	Mean	1.3221E+02	**4.6842E-05**	2.3881E+03	3.2797E+03	8.6365E-03	**9.8570E-07**
	Std	1.2725E+02	1.6050E-04	1.1014E+04	2.9060E+03	2.8965E-02	**1.0977E-06**
5.0E+04	Min	9.1488E-02	**0.0000E+00**	1.1040E-03	**3.5108E-08**	**0.0000E+00**	**0.0000E+00**
	Median	3.4234E-01	**0.0000E+00**	4.3227E-03	**1.5823E-05**	**0.0000E+00**	**0.0000E+00**
	Worst	6.5021E-01	**0.0000E+00**	7.9238E-03	1.2318E-02	**2.0000E-15**	**0.0000E+00**
	Mean	3.6243E-01	**0.0000E+00**	4.1607E-03	6.2102E-04	**1.6000E-16**	**0.0000E+00**
	Std	1.2355E-01	**0.0000E+00**	1.9215E-03	2.4589E-03	**4.7258E-16**	**0.0000E+00**
5.0E+05	Min	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Median	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	Worst	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**2.0000E-15**	**0.0000E+00**
	Mean	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**1.6000E-16**	**0.0000E+00**
	Std	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**4.7258E-16**	**0.0000E+00**

using 5×10^5 FEs.

The result achived by the BCO are very close to or even equal to feasible optimal

Table 3.14: Error Values achieved when FEs are 5000, 50000, and 500000 for test function $g13 - g18$

FES		g13	g14	g15	g16	g17	g18
5.0E+03	Min	1.0850E-02	2.0450E+00	5.9225E-04	8.1472E-02	4.0694E+00	1.1620E-01
	Median	1.2786E-01	5.8555E+00	4.4350E-02	3.0432E-01	5.3215E+01	4.1838E-01
	Worst	9.4729E-01	7.7918E+00	2.2955E+00	7.8726E-01	2.7754E+02	8.5155E-01
	Mean	3.9709E-01	5.6844E+00	3.0686E-01	3.4297E-01	7.4573E+01	4.5509E-01
	Std	3.7967E-01	1.5406E+00	6.3742E-01	2.1835E-01	6.9016E+01	2.3651E-01
5.0E+04	Min	**4.1100E-13**	2.7052E-04	**0.0000E+00**	**3.4311E-06**	**5.2751E-11**	3.2056E-03
	Median	**3.9797E-11**	8.0241E-04	**0.0000E+00**	**1.1654E-05**	**8.5856E-10**	9.3977E-03
	Worst	**2.2570E-08**	2.5207E-03	**9.0900E-13**	**3.7942E-05**	**8.4520E-07**	4.7111E-01
	Mean	**1.3131E-09**	9.1837E-04	**1.4540E-13**	**1.5270E-05**	**5.3349E-08**	2.9085E-02
	Std	**4.6360E-09**	5.8709E-04	**2.2128E-13**	**1.0031E-05**	**1.8432E-07**	9.2273E-02
5.0E+05	Min	**0.0000E+00**	**1.4000E-14**	**0.0000E+00**	**4.0000E-15**	**3.6380E-12**	**0.0000E+00**
	Median	**0.0000E+00**	**1.4000E-14**	**0.0000E+00**	**4.0000E-15**	**3.6380E-12**	**0.0000E+00**
	Worst	**0.0000E+00**	**7.1000E-14**	**9.0900E-13**	**4.0000E-15**	**3.6380E-12**	**0.0000E+00**
	Mean	**0.0000E+00**	**1.8240E-14**	**1.4540E-13**	**4.0000E-15**	**3.6380E-12**	**0.0000E+00**
	Std	**0.0000E+00**	**1.1443E-14**	**2.2128E-13**	**1.6103E-30**	**8.2445E-28**	**0.0000E+00**

Table 3.15: Error Values achieved when FEs are 5000, 50000, and 500000 for test function $g19 - g24$

FES		g19	g20	g21	g22	g23	g24
5.0E+03	Min	4.6250E+01	3.3233E-01	4.5408E+01	2.3643E+02	5.0931E+01	**3.3000E-14**
	Median	7.5536E+01	8.7700E-01	1.1814E+02	1.1161E+04	1.7079E+02	**3.5000E-14**
	Worst	1.4028E+02	2.7133E+00	4.7765E+02	1.9764E+04	3.3621E+02	**6.3000E-14**
	Mean	7.9671E+01	9.5601E-01	1.9748E+02	1.0220E+04	1.8187E+02	**4.0160E-14**
	Std	2.3452E+01	4.6699E-01	1.4443E+02	6.1985E+03	7.0989E+01	**1.0499E-14**
5.0E+04	Min	8.8876E-01	5.2239E-03	1.0848E-02	2.7760E+01	1.1941E+01	**3.3000E-14**
	Median	1.9566E+00	3.0857E-02	2.3503E-02	5.6483E+02	9.3792E+01	**3.3000E-14**
	Worst	3.6198E+00	6.5630E-02	7.5191E-02	4.7803E+03	4.2806E+02	**6.3000E-14**
	Mean	2.0812E+00	2.9412E-02	2.7678E-02	1.0215E+03	1.2093E+02	**3.4200E-14**
	Std	7.4010E-01	1.7922E-02	1.6356E-02	1.2441E+03	9.7208E+01	**6.0000E-15**
5.0E+05	Min	**1.2800E-13**	**1.3662E-10**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**3.3000E-14**
	Median	**4.1254E-11**	**1.4826E-08**	**0.0000E+00**	3.6988E+03	**0.0000E+00**	**3.3000E-14**
	Worst	**4.0478E-08**	**9.7700E-06**	**0.0000E+00**	1.8277E+04	**2.1237E-10**	**6.3000E-14**
	Mean	**4.6628E-09**	**7.7288E-07**	**0.0000E+00**	6.1451E+03	**1.1155E-11**	**3.4200E-14**
	Std	**1.0834E-08**	**2.1255E-06**	**0.0000E+00**	5.6814E+03	**4.3122E-11**	**6.0000E-15**

solution for 23 test problem in all run, except test problem $g22$ (only 3 out of 25 run, result is equal to known feasible optimal solution).

Table 3.16: Best, Median, worst, mean, and standard deviation of NFES to achieve the fixed accuracy level. $((f(x) - f(x^*)) \leq 0.0001)$, feasibility rate, success rate and successful performance over 25 runs of BCO on the CEC 2006 [147]

Prob.	Min	Max	median	Mean	STD	FR	SR	SP
g01	19758	24259	22245	22034	1088	100	100	22034
g02	120145	183451	158284	158293	14441	100	100	158293
g03	65224	89439	81640	80632	5769	100	100	80632
g04	39213	48906	44509	44488	2015	100	100	44488
g05	47230	59218	53852	54437	3287	100	100	54437
g06	258	6166	4050	3784	1425	100	100	3784
g07	179375	206245	197446	196558	7325	100	100	196558
g08	1412	4514	3617	3492	759	100	100	3492
g09	67591	81014	75322	74973	4100	100	100	74973
g10	22443	45100	34998	34192	5103	100	100	34192
g11	468	5902	4083	3705	1381	100	100	3705
g12	120	279	173	198	53	100	100	198
g13	879	18118	14178	13345	4039	100	100	13345
g14	55595	66731	62183	62221	2710	100	100	62221
g15	5415	15677	12365	11761	2911	100	100	11761
g16	34337	41579	38472	38373	2008	100	100	38373
g17	23173	31630	27209	27450	1873	100	100	27450
g18	106449	150370	129525	129224	12341	100	100	129224
g19	212636	265169	245241	241430	14914	100	100	241430
g20	22323	210350	172071	152074	59050	0	100	152074
g21	104454	119499	111879	112507	3858	100	100	112507
g22	250708	297360	284473	277514	92303	32	12	2312617
g23	142823	400000	222973	253267	78025	100	100	253267
g24	132	561	403	360	146	100	100	360

Comparison with other state-of-the-art on CEC 2006 test problems

The NFES required for reaching the optimum value is reported in Table 6.11. Table 6.11 shows the best, worst, median, mean, and standard deviation of NFES to achieve the fixed accuracy level, $(f(\bar{x}) - f(\bar{x}^*)) \leq 0.0001$), Feasible Rate (FR), Success Rate (SR), and Success Performance (SP) over 25 runs of BCO on the cec 2006 [147]. FR denotes the percentage of runs where atleast one feasible solution is found in 5×10^5 by algorithm. SR denotes percentage of run where atleast one solution satisfies success condition. SP denotes the ratio of mean of NFES required to find optimal feasible solution and SR.

As shown in table 6.11, FR of BCO for every test cases is 100% except problems $g20$, and $g22$. In case of problem $g20$, the optimal solution is slightly infeasible. Problem $g22$ is hard to solve, so in case of BCO only 32% FR is recorded. Regarding SR of BCO,

Table 3.17: Mean NFES to achieve the accuracy level $(f(x) - f(x*)) \leq 0.0001$ and SR over 25 runs on the CEC 2006 [147]

Proposed Work			DE															
Prob.	BCO		EPS-DE [146]		MPDE [132]		GDE [148]		MDE [149]		jDE-2 [150]		DPDE [151]		ICDE [152]		(mu+lm)CDE [153]	
	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR
g01	**22034**	100	59308	100	43430	100	40519	100	75373	100	50386	100	42180	100	105776	100	89000	100
g02	158293	100	149825	100	280573	92	107684	72	96222	16	123490	92	**95102**	94	283528	100	277379	96
g03	80632	100	89407	100	209298	84	143086	4	**44988**	100	–	0	88260	100	212657	100	111025	100
g04	44488	100	26216	100	20883	100	**15281**	100	41562	100	40728	100	25160	100	36770	100	30620	100
g05	54437	100	97431	100	216469	100	178023	92	**21306**	100	206620	68	100506	100	27933	100	165079	100
g06	**3784**	100	7381	100	10574	100	6503	100	5202	100	29488	100	13400	100	13040	100	11032	100
g07	196558	100	74303	100	**57400**	100	123996	100	194202	100	127740	100	99060	100	134789	100	141038	100
g08	3492	100	1139	100	1515	100	1469	100	**918**	100	3236	100	1960	100	1943	100	2010	100
g09	74973	100	23121	100	21044	100	30230	100	**16152**	100	54919	100	31820	100	37929	100	39953	100
g10	**34192**	100	105234	100	48628	100	82604	100	164160	100	146150	100	143300	100	325007	100	188725	100
g11	3705	100	16420	100	22422	96	8460	100	**3000**	100	49700	96	90310	100	4404	100	79475	100
g12	**198**	100	4124	100	4238	100	3149	100	1308	100	6356	100	5626	100	6488	100	4908	100
g13	**13345**	100	34738	100	356433	48	336306	40	21732	100		0	81980	100	34325	100	148237	100
g14	62221	100	113439	100	**42715**	100	220921	96	291642	100	97845	100	107480	100	85758	100	176671	100
g15	11761	100	84216	100	200174	100	71889	96	10458	100	222460	96	94600	100	**10074**	100	130622	100
g16	38373	100	12986	100	13063	100	13224	100	**8730**	100	31695	100	18650	100	25001	100	19154	100
g17	27450	100	98861	100	204791	28	343740	16	26364	100	**17971**	4	128690	100	103230	100	183962	100
g18	129224	100	59153	100	**44045**	100	364861	76	103482	100	104460	100	80280	100	138998	100	215068	100
g19	241430	100	**35635**	100	118274	100	202648	88	–	0	199850	100	163080	100	296145	100	268374	100
g20	152074	100	–	0		0	–	0	–	0	–	0	–	0	–	0	**148506**	100
g21	112507	100	135143	100	142159	68	347653	60	112566	100	**107080**	92	164068	92	317447	100	209896	92
g22	**277514**	12	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0
g23	253267	100	**200765**	100	210661	100	425342	40	360420	100	302550	92	204450	94	364806	100	263695	100
g24	**360**	100	2952	100	4342	100	3059	100	1794	100	10196	100	5860	100	574	100	5059	100

Proposed Work			PSO						Others							
Prob.	BCO		PSO		COPSO [154]		PESO [155]		AP-CMA-ES [156]		ASR-ES [157]		PCX [158]		Shade [159]	
	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR	NFES	SR
g01	**22034**	100	73314	100	95397	100	101532	100	181110	52	35406	100	55204	100	50386	100
g02	158293	100	–	0	179395	73	231193	56	–	0	–	0	**87900**	64	123490	92
g03	80632	100	–	0	315123	100	450644	100	**18302**	100	–	0	34937	100	–	0
g04	44488	100	37802	100	65087	100	79876	100	**4992**	100	15104	100	30989	100	40728	100
g05	54437	100	366824	100	315257	100	452256	100	82365	100	**19281**	100	94765	100	206620	68
g06	3784	100	37802	100	53410	100	56508	100	**3269**	100	9603	100	33821	100	29488	100
g07	196558	100	405156	100	233400	100	352592	96	**14445**	100	76782	8	117121	100	127740	100
g08	3492	100	3656	100	6470	100	6124	100	1661	100	**1027**	100	2826	100	3236	100
g09	74973	100	103677	100	79570	100	97544	100	**5882**	100	30618	100	46527	100	54919	100
g10	34192	100	487525	100	224740	100	452575	16	**24891**	100	–	0	89028	100	146150	100
g11	3705	100	**33073**	100	315000	100	450100	100	25803	100	**2792**	100	38688	100	49700	96
g12	**198**	100	6906	100	6447	100	8088	100	31247	100	2996	100	8960	100	6356	100
g13	13345	100	–	0	315547	100	450420	100	–	0	**11292**	84	53735	100	–	0
g14	62221	100	–	0	326900	3	–	0	**14477**	100	92820	8	59237	100	97845	100
g15	11761	100	267821	100	315100	100	450100	100	131822	100	**8519**	100	46936	100	222460	96
g16	38373	100	56612	100	40960	100	49040	100	**6106**	100	16179	100	30395	100	31695	100
g17	27450	100	–	0	**316609**	76	–	0	–	0	**21491**	76	136110	100	17971	4
g18	129224	100	238706	100	167089	90	214322	92	**68741**	100	40840	92	70027	100	104460	100
g19	241430	100	426101	100	264414	47	–	0	**75669**	100	–	0	129676	100	199850	100
g20	**152074**	100	–	0	–	0	–	0	–	0	–	0	–	0	–	0
g21	112507	100	–	0	–	0	–	0	184302	56	–	0	**38217**	100	107080	92
g22	277514	12	–	0	–	0	–	0	–	0	–	0	–	0	–	0
g23	253267	100	–	0	–	0	–	0	200158	84	–	0	**167119**	100	302550	92
g24	360	100	**1986**	100	19157	100	19980	100	**1663**	100	3638	100	11646	100	10196	100

All the results are taken from their corresponding papers.

−: NFES is not available.

Table 3.18: Ranking based on the SP of all algorithm over 25 runs on the CEC 2006 [147]

Prob.	BCO	EPS-DE	MPDE	GDE	MDE	jDE-2	DPDE	ICDE	(mu+lm)CDE	PSO	COPSO	PESO	AP-CMA-ES	ASR-ES	PCX	Shade
g01	1	9	5	3	11	6	4	15	12	10	13	14	16	2	8	6
g02	7	6	11	5	13	2	1	9	10	14	8	12	14	14	4	3
g03	4	6	9	12	3	13	5	8	7	13	10	11	1	13	2	13
g04	6	14	4	3	13	11	5	9	7	10	15	16	1	2	8	11
g05	4	7	11	10	2	12	8	3	9	15	14	16	5	1	6	13
g06	2	5	7	4	3	11	10	9	8	14	15	16	1	6	13	11
g07	12	3	2	6	11	7	4	9	10	15	13	14	1	16	5	7
g08	13	3	5	4	1	11	8	7	9	14	16	15	6	2	10	11
g09	13	4	3	5	2	11	7	8	9	16	14	15	1	6	10	11
g10	2	6	3	4	10	8	7	13	11	14	12	15	1	16	5	8
g11	3	6	7	5	2	12	14	4	13	9	15	16	8	1	10	11
g12	1	5	6	4	2	9	8	12	7	13	11	14	16	3	15	9
g13	1	5	11	12	3	13	7	4	8	13	9	10	13	2	6	13
g14	4	9	2	11	12	6	8	5	10	15	14	15	1	13	3	6
g15	4	7	11	6	3	12	8	2	9	14	15	16	10	1	5	13
g16	13	3	4	5	2	11	7	9	8	16	14	15	1	6	10	11
g17	2	4	12	13	1	10	6	5	8	14	9	14	14	3	7	10
g18	10	3	1	16	7	8	6	11	13	15	12	14	4	2	5	8
g19	9	1	3	8	14	6	5	11	10	12	13	14	2	14	4	6
g20	2	3	3	3	3	3	3	3	1	3	3	3	3	3	3	3
g21	2	6	8	12	3	5	7	10	9	13	13	13	11	13	1	4
g22	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
g23	6	2	3	12	10	8	4	11	7	13	13	13	5	13	1	9
g24	1	6	9	7	4	12	11	2	10	5	15	16	3	8	14	12
sum	123	125	142	172	137	209	155	181	207	292	288	319	140	162	157	211
rank	1	2	5	9	3	12	6	10	11	15	14	16	4	8	7	13

BCO is capable to provide 100% SR in every test cases except problem $g22$. In problem $g22$, BCO only provide 12% SR. In Table 6.11, SP indicates that the BCO requires less than 4×10^3 FEs for 5 problems, 1×10^5 FES for 16 problems, 2.6×10^5 for 23 problems, to achieve the success condition of optimum.

SR and Mean FEs of BCO required to solve CEC 2006 problems against state-of-the-art that do not use the traditional techniques to solve constrained problems are compared. For convenience, we grouped all the state-of-the-art in three groups: DE-based, PSO-based, and other metaheuristic which is based on CMA-ES and hybrid variant of DE. Most of the state-of-art methods, has taken for comparison, are DE and PSO based because perching and patrolling follows the similar procedure to DE and PSO respectively. Besides metaheuristics, these state-of-the- arts also use different type of constraint-handling techniques to solve COP. Most of the state-of-arts use ϵ-constraint handling, three feasibility rules, and penalty function to handle the constraint. So, effectiveness of proposed constraint handling techniques is also analysed as compared to other constraint handling technique.

Table-3.17 shows the comparison of performance of BCO with other state-of-arts in

Table 3.19: Average ranking of the Friedman test

Algorithm	Average Ranking
BCO	5.4375
EPS-DE	5.8333
MPDE	6.4375
GDE	7.6875
MDE	6.3125
jDE-2	9.6875
DPDE	6.9792
ICDE	8.0625
$(\mu+\lambda)$CDE	8.875
PSO	13.0833
COPSO	12.6875
PESO	14.0833
AP-CMA-ES	6.5
ASR-ES	7.5833
PCX	7.0625
Shade	9.6875

χ^2- distribution : 114.64614.

P-value : 0.

terms of NFES and SR. In table-3.17, the bold face in NFES shows the best performance in terms of NFES. Table-3.18 shows the ranking basad on the SP of all algorithms. In Table-3.19, result of Friedman ranking test is reported. Average ranking of the Friedman test is done based on the SP of all the algorithms on every problem of CEC 2006 [147].

Comparison with DE-based state-of-arts

To compare the performance of BCO with respect to the popular DE-based state-of-arts, eight different algorithms EPS-DE, MPDE, GDE, MDE, jDE-2, DPDE, ICDE, and (mu+lm)-CDE are chosen. The results of these algorithms are used in comparison is same as reported in their original paper.

A closer examination of Table-3.17 at the first sub-table, reveals that the performance of BCO in terms of NFES and SR is better than the DE-based state-of-art design. Performance of BCO as compared to other DE-based state-of-arts are summarized below:

- *Comparison based on NFES:* In case of seven problems $g01$, $g06$, $g10$, $g12$, $g13$, $g22$, and $g24$, BCO are converged fastly on feasible optimum value i.e. takes less number of NFES to achieve the accuracy level $(f(x) - f(x^*)) \leq 0.0001$, as compared to others algorithms. Other algorithms like EPS-DE, MPDE, GDE, MDE, jDE-2, DPDE, ICDE and (mu+lm)-CDE take less number of NFES in case of 2 $(g19, g23)$, 3 $(g07, g14, g18)$, 1 $(g04)$, 6 $(g03, g05, g08, g09, g11, g16)$, 2 $(g17, g21)$, 1 $(g02)$, 1 $(g15)$ and 1 $(g20)$ problems respectively.

- *Comparison based on SR:* In case of all test problems except $g22$ (12%), BCO achieve 100% SR. It can seen from Table- 3.17, BCO outperforms all the DE-based state-of-arts in terms of SR. It is interesting to note that only BCO can found feasible optimal solution for test problem $g22$. Owing to its special characteristic, test problem $g22$ is very hard to solve for different DE-based state-of-arts.

Through this comprehensive comparison with chosen DE-based state-of-art designs, BCO can be considered very competitive with respect to the chosen DE-based state-of-art in case of COPs.

Comparison with PSO-based state-of-art designs

Three PSO-based state-of-the-art design PSO, COPSO, and PESO, are chosen to compare the performance of BCO. The results of these algorithm is directly taken from their original paper. All the chosen PSO-based algorithms use the three feasibility rule as a constraint handling technique.

Second sub-table of Table-3.17, shows that the BCO performs better than chosen PSO-based state-of-art design in terms of NFES and SR. Performance of BCO as compared to chosen PSO-based state-of-the-arts are summarized below:

- *Comparison based on NFES:* In case of test problem except $g04$, BCO are converged fastly i.e. takes less number of NFES to find feasible optimal value. In case of test problem $g04$, PSO converges fastly than BCO.

- *Comparison based on SR:* As shown in second sub-table of Table-3.17, for test problems $g20$, $g21$, $g22$, and $g23$, PSO-based algorithms did not converge on feasible optimum value for any run out of 25 independent run i.e. SR is zero. The performance of BCO is effectively much better than PSO-based algorithms in terms of SR is case of test problems $g02$, $g03$, $g13$, $g14$, $g17$, $g19$, $g20$, $g21$, $g22$, and $g23$.

Based on the above comparison result, we conclude that the performance of BCO is far better than the performance of PSO-based state-of-art designs in case of COPs.

Table 3.20: Multiple solution obtained using BCO for ill-conditioned CASE13

solution-1								solution-2						
Bus	V_a	$\angle_a$	V_b	$\angle_b$	V_c	$\angle_c$		Bus	V_a	$\angle_a$	V_b	$\angle_b$	V_c	$\angle_c$
1	1.00	0.00	1.00	-120.00	1.00	120.00		1	1.00	0.00	1.00	-120.00	1.00	120.00
2	0.99	-0.25	0.99	-120.18	0.99	119.98		2	0.99	-0.25	0.99	-120.18	0.99	119.98
3					0.62	102.02		3					0.92	122.58
4			0.94	-120.73	1.01	119.59		4			0.94	-120.73	1.01	119.59
5			0.92	-121.05	1.02	119.46		5			0.92	-121.05	1.02	119.46
6	0.93	-13.74						6	0.27	-33.10				
7	0.96	-13.96	1.07	-118.37	0.64	104.44		7	0.39	-34.92	1.21	-131.26	0.93	123.30
8	0.93	-15.18	1.08	-118.95	0.63	104.21		8	0.31	-41.56	1.22	-132.73	0.94	123.20
9	0.96	-13.96	1.07	-118.37	0.64	104.44		9	0.39	-34.92	1.21	-131.26	0.93	123.30
10	0.95	-14.05			0.63	103.39		10	0.35	-35.99			0.93	123.21
solution-3								solution-4						
Bus	V_a	$\angle_a$	V_b	$\angle_b$	V_c	$\angle_c$		Bus	V_a	$\angle_a$	V_b	$\angle_b$	V_c	$\angle_c$
1	1.00	0.00	1.00	-120.00	1.00	120.00		1	1.00	0.00	1.00	-120.00	1.00	120.00
2	0.99	-0.25	0.99	-120.18	0.99	119.98		2	0.99	-0.25	0.99	-120.18	0.99	119.98
3					0.97	96.51		3					0.26	90.58
4			0.94	-120.73	1.01	119.59		4			0.94	-120.73	1.01	119.59
5			0.92	-121.05	1.02	119.46		5			0.92	-121.05	1.02	119.46
6	1.14	9.45						6	0.09	-39.62				
7	1.16	9.39	0.15	-164.30	0.98	97.52		7	0.42	-44.17	1.28	-124.24	0.32	97.82
8	1.15	9.41	0.15	-183.44	0.99	96.91		8	0.37	-50.61	1.29	-125.05	0.29	95.65
9	1.16	9.39	0.15	-164.30	0.98	97.52		9	0.42	-44.17	1.28	-124.24	0.32	97.82
10	1.16	9.23			0.97	97.08		10	0.32	-48.98			0.29	97.47

Comparison with Other State-of-Art Designs

To broaden our comparative analysis, we considered some other state-of-art designs AP-CMA-ES, ASR-ES, and PCX and SHADE that combine a population-based algorithm (DE, PSO, and GA) with other programming techniques for example SQP. The performance of BCO as compared to other state-of-art designs are summarized below.

- *Comparison based on NFES:* In the case of five problems $g01$, $g12$, $g20$, $g22$, and $g24$, BCO has provided optimum feasible solutions with minimum Mean NFES as compared to other algorithms. The performance of AP-CMA-ES and ASR-ES on the basis of Mean NFES is better than the performance of BCO. Mean NFES of AP-CMA-ES and ASR-ES is better than BCO for 9 ($g03$, $g04$, $g06$, $g07$, $g09$, $g10$,

$g14$, $g16$, and $g19$) and 6 ($g05$, $g08$, $g11$, $g13$, $g15$, and $g18$) problems respectively. BCO outperforms the PCX and Shade on the basis of Mean NFES.

- *Comparison based on SR:* It can seen from Table-3.17, BCO outperform all the state-of-art designs on the basis of SR. AP-CMA-ES and ASR-ES, PCX, and Shade provides 100% SR on 16, 11, 21, and 13 problems respectively.

Based on the above discussion, we can conclude that the ASR-ES and AP-CMA-ES are having better convergence speed as compared to BCO, but the problem-solving capability of these algorithms are poorer than the BCO. PCX algorithm has the better problem-solving capability with good convergence speed, but BCO outperforms this algorithm on the basis of both property. The BCO also outperforms the performance of Shade.

Ranking of all the selected state-of-art algorithms along with BCO on the basis of Success Performance

Rank based on SP of all competitive algorithms are reported in Table 3.18. It is clear from the Table 3.18, the overall normalized rank of BCO is 1, and the performance of the BCO is better than all other competitive algorithms. In the table 3.18, the number in boldface represents the normalized rank of all ranks. From Table 3.18, Performance of BCO on test problems $g01$, $g12$, $g13$, $g22$, and $g24$ based on SP, is better than all another state of art designs and takes the top position in the ranking table 3.18 of these test problems.

ϵ-DE is obvious at the second rank in Ranking Table 3.18, because it is the winner of CEC-2006. In support of Ranking, non-parameter Friedman test is reported in Table 3.18. In Table 3.18, BCO is globally at the first rank with the minimum value. P-value and χ^2 distribution of Friedman test are also reported at the bottom of Table 3.19. It is clear from these values; there is a significant difference in the performance of all competitive algorithm and Ranking is valid.

3.4.9 BCO based Power Flow for Loadability Evaluation

BCO can be used to obtain multiple solution of PF problems. Multiple PF solutions can be useful in voltage stability analysis of power system. In this section, validation of performance of BCO is done on CASE13 test system.

Four different PF solutions of CASE13 are obtained using BCO and are reported in Table 3.20. It can be seen from Table 3.20 that the solutions are low voltage solutions of PF problem which can be used in voltage stability analysis.

For further analysis of performance of BCO, voltage profile of less stable buses of the system at different loading condition is obtained. In this analysis, CASE11 and CASE25 are considered as test systems. In CASE13, bus 7c, 8c, 9c, and 10c are the least stable buses in which voltage level reduces rapidly with increase of load. Similarly, bus 7a, 9a, 10a, 11a, 12a, 13a, 14a, 15a, 16a, and 17a are least stable buses in CASE25. Voltage profile of these buses with different loading condition are depicted in Figure 3.1, 3.2 and 3.3.

It can be seen from Figure 3.1 that after the loading factor of 10, bus voltage start decreasing due to insufficient power generation. After loading factor of 11.291, the voltage collapses because this point is critical point of CASE13 under defined operating conditions.

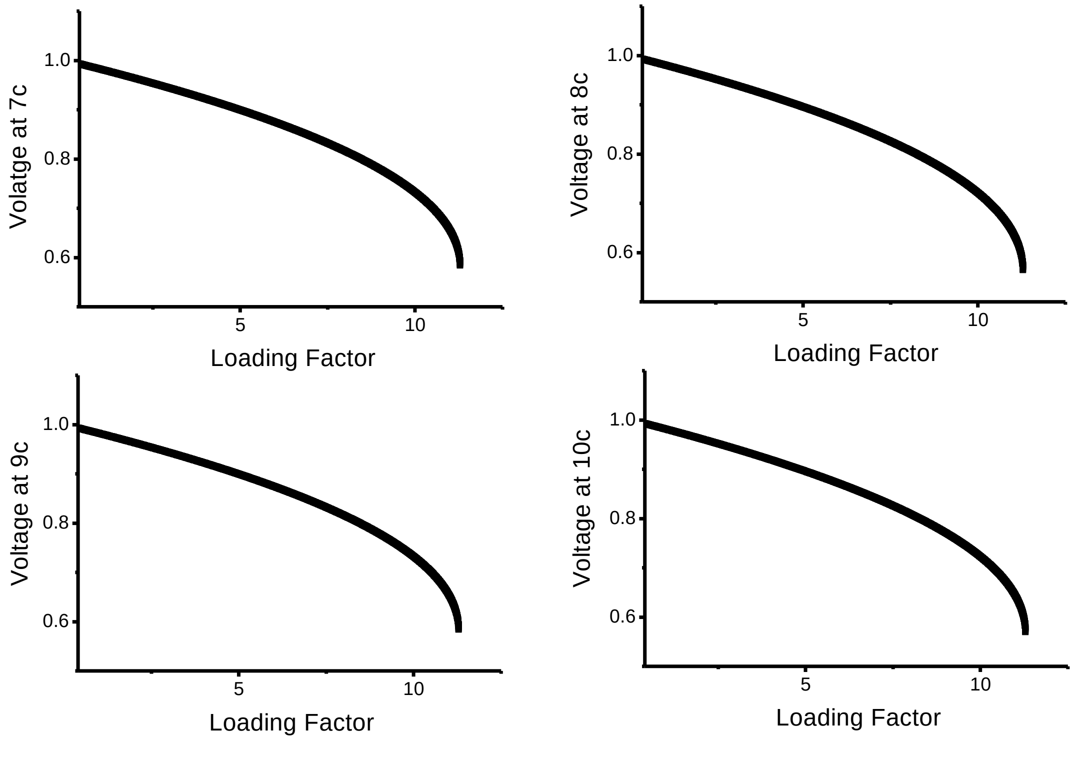

Figure 3.1: Voltage profile of Bus 7c, 8c, 9c, and 10c of CASE13 for different loading condition.

From Figures 3.2 and 3.3, it can be seen that the voltages start reducing rapidly after loading factor of 11 and voltage collapses at loading factor of 12.612. Therefore, loading factor of 12.612 is the critical point of CASE25 under given operating condition.

3.5 Summary

The SS algorithm has been proposed in this chapter. The SS algorithm is an unconstrained optimization algorithm that can be used to calculate initial seed for the conventional PF algorithms. SS based initial seed can not be the competitor to the flat start used in conventional algorithms but it becomes necessary to use this in the situations where flat start does not converge. From the extensive analysis of the proposed approach, it can be concluded that this approach improves the performance of the conventional algorithm on the heavily loaded and ill-conditioned test systems.

In this chapter, a PF problem is also formulated as a COP to compute the PF solutions of the system at critical points (notch points). To solve this non-convex and highly non-linear COP, the BCO algorithm is proposed for PF. BCO is able to provide multiple solutions specially low voltage solutions for the PF problems. From the extensive study of the performance of BCO as a PF tool, it can be concluded that the BCO can provide continuation PF solutions for the distribution networks.

In the case of the islanded operation of microgrids, the system frequency and power generations at a distribution generation are not pre-defined before PF analysis. Therefore, the proposed algorithms of this chapter cannot be applicable to the PF problems of islanded microgrids. In the next chapter, novel algorithms are proposed to solve the PF problems of islanded microgrids.

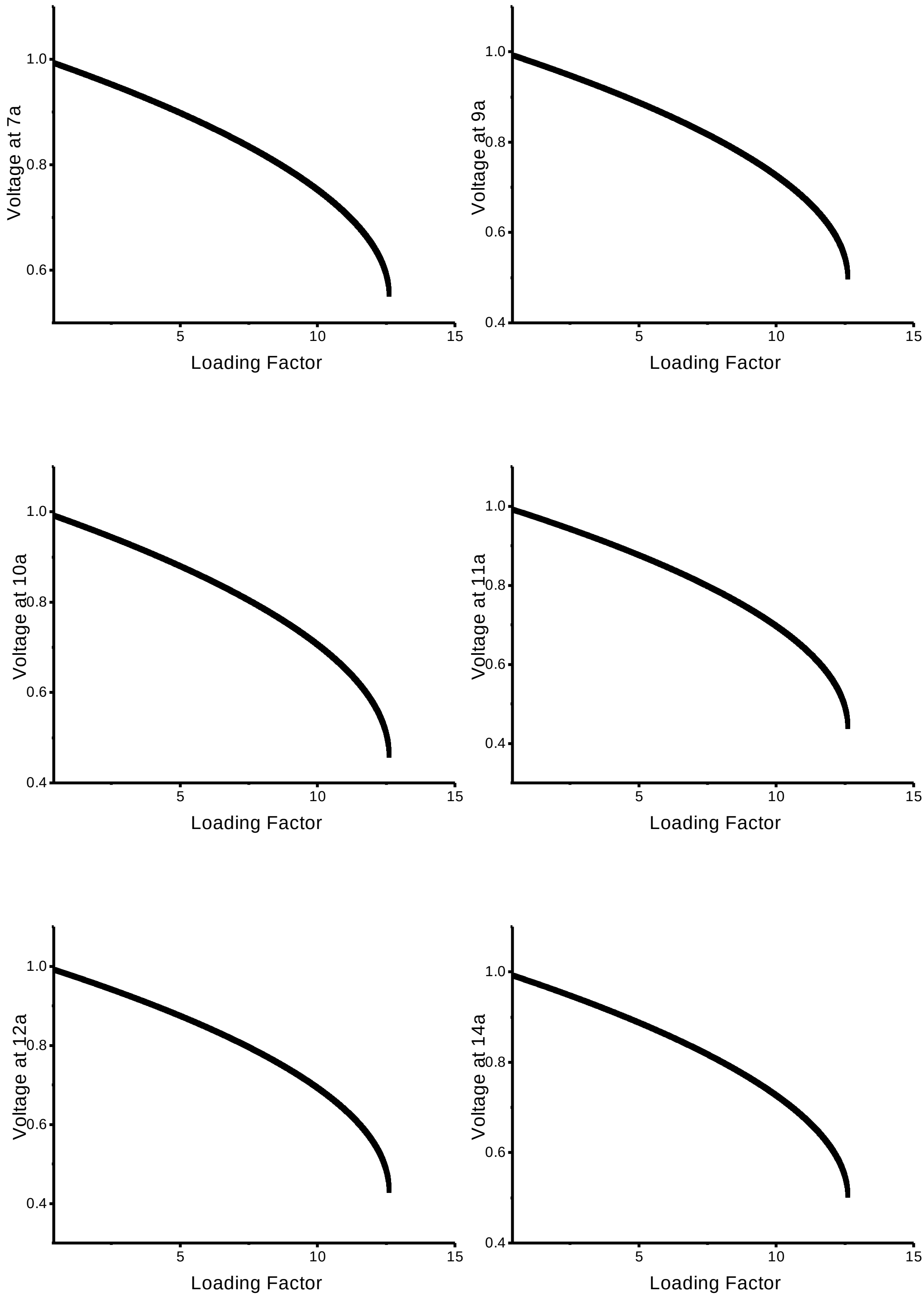

Figure 3.2: Voltage profile of Bus 7a, 9a, 10a, 11c, 12a and 14c of CASE25 for different loading condition.

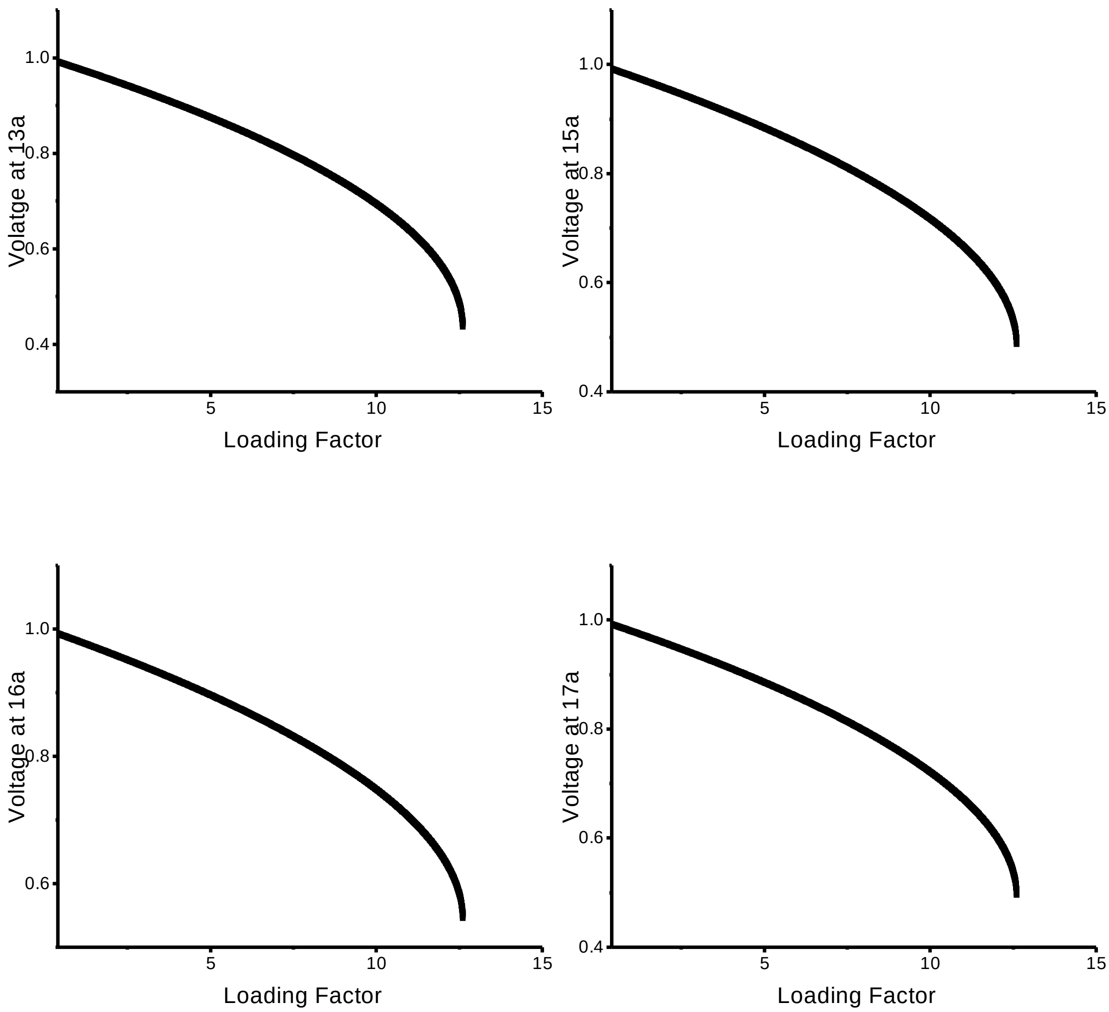

Figure 3.3: Voltage profile of Bus 13a, 15a, 16a, and 17a of CASE25 for different loading condition.