Noname manuscript No.
(will be inserted by the editor)

Solving Security Constrained Optimal Power Flow problems: a hybrid evolutionary approach

Carolina Marcelino · Paulo Almeida · Elizabeth Wanner · Manuel Baumann · Marcel Weil · Leonel Carvalho · Vladimiro Miranda

Received: date / Accepted: date

Abstract A hybrid population-based metaheuristic, Hybrid Canonical Differential Evolutionary Particle Swarm Optimization (hC-DEEPSO), is applied to solve Security Constrained Optimal Power Flow (SCOPF) problems. Despite the inherent difficulties of tackling these real-world problems, they must be solved several times a day taking into account operation and security conditions. A combination of the C-DEEPSO metaheuristic coupled with a multipoint search operator is proposed to better exploit the search space in the vicinity of the best solution found so far by the current population in the first stages of the search process. A simple diversity mechanism is also applied to avoid premature convergence and to escape from local optima. A experimental design is devised to fine-tune the parameters of the proposed algorithm for each instance of the SCOPF problem. The effectiveness of the proposed hC-DEEPSO is tested on the IEEE 57-bus, IEEE 118-bus and IEEE 300-bus standard systems. The numerical results obtained by the proposed hC-DEEPSO are compared with other evolutionary methods reported in this literature to prove the potential and capability of the proposed hC-DEEPSO for solving the SCOPF at acceptable economical and technical levels.

Keywords Evolutionary optimization methods · Optimal Power Flow · hybrid algorithms · hC-DEEPSO algorithm · Fining tunining parameters · Statistical Inference.

Carolina Marcelino, Paulo Almeida
Federal Center of Technological Education of Minas Gerais, Brazil
E-mail: carolimarc@gmail.com, pema@lsi.cefetmg.br

Elizabeth Wanner
Aston University, UK. E-mail: e.wanner@aston.ac.uk

Manuel Baumman, Marcel Weil
Institute for Technology Assessment and Systems, Germany

Leonel Carvalho, Vladimiro Miranda
Institute for Systems and Computer Engineering, Portugal

1 Introduction

The increasing demand for electric power is one of the main challenges being faced by many countries in the World. Among other goals, electric companies aim at supplying power to consumers at a minimal operational cost. In general, this optimization problem known as Optimal Power Flow (OPF) seeks to optimize the overall operation cost of generating and transmitting electric power subject to system constraints and control limits. Many other goals can also be considered in this problem, such as the minimization of modifications in controls, minimization of the active power losses, minimization of power not supplied when load has to be curtailed, or even the minimization of the greenhouse gas emissions, just to name a few [1]. Thus, the software module in Energy Management Systems to solve the OPF problem plays an important role, as the whole operation of the power system must be optimized for each time interval.

OPF can be defined as a mono-objective problem dealing with the aggregation of two or more objective functions, or a multiobjective one trying to optimize each goal separately and simultaneously [2]. Traditionally, the control variables are the active power production and the voltage set point of generating units, the positions of transformer taps and the states of shunt compensator's (reactors and capacitors). The equality constraints stem from the power flow equations and are associated with Kirchhoff's current and voltage laws. Operational limits, such as the active and reactive power boundaries of generators, the bus voltage magnitude and the apparent power flow in the lines are modeled as inequality constraints. Additional security criteria can be included in the OPF problem, such that the operation of the system is kept within pre-defined security margins even when unplanned outages occur [3]. Usually, OPF is considered as an extremely difficult optimization problem to solve due to its highly dimensionality, non-linearity, combinatorial, non-convex and multimodal nature [1] [2] [3].

The OPF problem was modeled and solved for the first time by Carpertier [4] using deterministic techniques. The some common deterministic techniques used to solve this problem are Linear Programming, Quadratic Programming, Gradient Method, Interior Point Methods, to name a few [1] [2] [5]. Some examples of nondeterministic techniques are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Differential Evolution (DE) [6]-[13]. Table 1 describes in a simple manner the main advantages and disadvantages of the most important techniques.

Apart from supplying electric power at minimal operation cost, utilities must comply with high levels of safety and quality. To ensure adequate continuity and security of supply, the operator might want that the inequality constrains are not violated in case of equipment outage (generators, transformers, lines, etc.). When contingencies of equipment are included in the optimization model, the problem is known as Security Constrained Optimal Power Flow (SCOPF) [3]. Several research works handling the SCOPF problem can be found in literature, namely, methodologies that use GA, PSO, ANN, Linear Programming and some hybrid approaches [14]-[20].

Hybrid optimization techniques have been recently proposed to solve this problem. Hybrid methods can be defined as a combination of deterministic and nondeterministic techniques to solve a particular problem. In many cases, hybrid methods have proven to be more efficient and lead to faster convergence to the optimal

Table 1 Summary of solution methods for OPF problems based on [1] [2]

Methods	Main principle	Advantages	Disadvantages
Gradient or Dommel-Tinney	Usage of the gradient of the Lagrangian function to direct the search towards the optimal solution.	Does not require the calculation of second derivatives, making the method simpler.	Slow convergence; Difficulties in handling inequality constraints.
Interior points	Start from a point in the "interior" of the search space and transform the inequality constraints into equality constraints through slack variables in order to build the Lagrangian function by adding all restrictions.	Faster than conventional Linear programming methods; efficient and robust on solving linear and nonlinear problems.	Easily handles inequality constraints; sensitive to initial conditions; sometimes requires linearization.
Linear Programming	Usage of linear or piecewise linear cost functions and DC power flow instead of AC power flow, which provides a linear relationship between injections and line flows.	Convergence is guaranteed; fast convergence; less computational effort; Easily handles inequality constraints.	Inaccuracy due to linearization of cost functions and nonlinear constraints.
Quadratic Programing	Usage of a quadratic objective function; all constraints are linear.	Does not require linearization of cost functions.	Inaccuracy due to linearization of nonlinear constraints.
Newton-Raphson	Application of a second order Taylor series expansion, using the Hessian and gradient matrixes to build an approximation of quadratic function.	Fast convergence when it is close to optimum value; Flexible formulation (Different applications for the problem OPF); efficient and robust.	Computationally heavy; highly sensitive to initial conditions; difficulty of enforcing inequality constraints.
Metaheuristics	Genetic algorithms, Particle Swarm Optimization, Evolutionary Programming, Ant Colony optimization.	No linearization is required; Allows for the use of non-continuous functions, not convex and not differentiable; easy implementation.	The solution obtained is not guaranteed to be optimal and high time to obtain solutions.

solution than methods that optimize solely individual components [2]. Hybrid methods applied to solve the OPF problem are based on the combination of techniques such as GA and PSO, DE as well as PSO, ANN with GA, Interior Point and PSO, Newton Method, PSO and Artificial Bee Colony (ABC), PSO and others [21]-[30].

Hybrid methods are becoming reliable and effective techniques to solve real-world and complex optimization problems with remarkable performance. For instance, a PSO inspired multi-elitist ABC algorithm named PS-MEABC is proposed in [30] for real-parameter optimization. The optimal coordination of directional overcurrent relays in meshed networks using a hybrid PSO-DE approach is detailed in [31]. The economic dispatch problem of generating units is solved in [32] once again using a hybrid technique based on PSO and DE.

This paper proposes a new hybrid approach, the so called Hybrid Canonical Differential Evolutionary Particle Swarm Optimization (hC-DEEPSO), which combines the C-DEEPSO [18] coupled with a multipoint search operator called Spiral Search (SLS) to solve the SCOPF problem. The usage of this search operator aims to better exploit the search space in the vicinity of the best solution found so far by the current population in the first stages of the search process. In addition, a simple diversity mechanism is applied to balance between exploration and exploitation. That mechanism is able to avoid premature convergence and to escape from local optima.

The algorithm is tested on three test systems well-known in literature - IEEE 57 bus, IEEE 118 bus and IEEE 300 bus systems - with two different objective functions: the minimization of active power losses in the transmission network and the minimization of total cost of electricity production for the expected operation scenario. In both scenarios, several unplanned outages of equipment are taken into account, requiring the system operation to be kept within pre-defined margins even in contingency scenarios.

This section has introduced the OPF problem and its SCOPF variation as well as the current state-of-art methods for solving these problems. The remainder of the paper is organized as follows: Section 2 deals with the modeling of the SCOPF problems; Section 3 presents the hybrid algorithm hC-DEEPSO, the problem-specific search operator SLS and the diversity mechanism; Section 4 discusses the experiments and the results obtained; finally, Section 5 concludes the work.

1.1 Problem Formulation

The original OPF problem is a mixed-integer non-linear optimization problem. That formulation does not guarantee that operation of the network remains in a secure state after a sudden equipment outage. Therefore, an improved formulation has been proposed: the SCOPF. This model ensures that the power can be successfully transferred from generators to loads not only under the expected network topology, but also for some unplanned outage caused by sudden loss of sections of the network, such as transmission lines, distribution branches and/or transformers. Therefore, SCOPF problem has a considerable larger number of constraints than OPF problem.

This paper addresses the SCOPF problem considering two different objective functions: the minimization of total operation cost, which is known as Optimal Active Reactive Power Dispatch (OARPD) problem, and the minimization of total active power losses in the transmission network, known as Optimal Reactive Power Dispatch (ORPD) problem. These two objective functions can be expressed as non-linear functions: the OARPD version is a quadratic function of the generators active power output in $\$/h$ while the ORPD version is quadratic function of the bus voltage magnitudes and of the cosine of the difference between bus voltage angles. Note that the active power production of the power plants in the ORPD problem is an input parameter and not a decision variable: the only active power production to be optimized is that corresponding to the generator in the slack bus.

Both optimization problems can be formulated using three types of variables - the the control or decision variables, the state variables and the parameters - as follows:

- Control: active power and voltage magnitude for the generators; tap status of transformers; switching state of capacitors; power electronic controls and phase position of transformer taps (when available);
- State: voltage magnitude and angle for all buses;
- Parameters: network topology, resistance, reactance, shunt conductance and susceptance of transmission lines and transformers, susceptance of shunt capacitors, physical limits of the control variables, reactive power limits of generators, apparent power limits of transmission lines and transformers, slack bus, and set of N-1 contingency scenarios.

SCOPF problem includes security constraints, namely the N-1 criteria, to uphold the inequality constrains even in case of failure of a single component. The two objective functions, which are represented in Eq. (1) as $\min C$ in $\$/h$ and P_{loss} in MW, are solved for the expected scenario of operation, whereas the constraints are met not only in the case of that scenario but also for all the contingency scenarios. Note that these two problems are solved separately using a mono-objective approach.

$$\min C = \textstyle\sum_{i=1}^{NG} a_i + b_i \times Pg_i + c_i \times Pg_i^2,$$

$$\min P_{loss} = \textstyle\sum_{K=1}^{N_L} G_K[U_i^2 + U_j^2 - 2|U_i||U_j|\cos(\delta_i - \delta_j)],$$

$$\text{s.t. :} \begin{cases} P_i = Pg_i - Pl_i = \\ \sum_{j=1}^{NB} U_i U_j \left[G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)\right], \\ \forall_i \in NB, \forall_s \in NS; \\ Q_i = Qg_i - Ql_i = \\ \sum_{j=1}^{NB} U_i U_j \left[G_{ij}\sin(\delta_i - \delta_j) + B_{ij}\cos(\delta_i - \delta_j)\right], \\ \forall_i \in NB, \forall_s \in NS; \\ \underline{U}_i \leqslant U_i \leqslant \overline{U}_i, \forall_i \in NB, \forall_s \in NS; \\ \left|S_{ij}\right| \leqslant \overline{S_{ij}}, \forall_i \in NC, \forall_s \in NS; \\ \left|S_{ji}\right| \leqslant \overline{S_{ij}}, \forall_i \in NC, \forall_s \in NS; \\ \underline{Pg_i} \leqslant Pg_i \leqslant \overline{P_i}, \forall_i \in NG, \forall_s \in NS; \\ \underline{Qg_i} \leqslant Qg_i \leqslant \overline{Q_i}, \forall_i \in NG, \forall_s \in NS; \\ \underline{t_i} \leqslant t_i \leqslant \overline{t_i}, \forall_i \in NOLTC, t_i \in \forall_s \in NS; \\ 0 \leqslant q_i \leqslant 1, \forall_i \in NSHUNT, q_i \in Z, \forall_s \in NS. \end{cases} \tag{1}$$

in which Pg_i (MW) is the active power generation, Qg_i (MVar) is the reactive power generation, Pl (MW) is the active power load, Ql (MVar) is the reactive power load, U (kV) is the voltage magnitude, δ (degrees) is the voltage angle, S_{ij} (MVA) is the apparent power flow injection at the sending end of the transmission circuit connecting bus i to bus j whereas S_{ij} (MVA) is the apparent power flow injection at the receiving end of the same circuit.

The variable t is the tap setting position of the OLTC (**O**n-**L**oad **T**ap **C**hanger), q is a binary variable that represents the state of the capacitor/reactor banks, a (\$/h), b (\$/MWh) and c (\$/MWh2) are cost coefficients, $Y = G + jB$ is the bus admittance matrix, NG is the number of generators, NB is the number of buses, NC is the number of circuits in the network, $NOLTC$ is the number of $OLTC$ transformers, $NSHUNT$ is the number of capacitor/reactor banks and NS is the number of scenarios that represent the expected operation scenario and contingency states. Note that all constraints must be satisfied for all scenarios.

In this research work, the active power production at PV buses, the voltage magnitude at the REF and PV buses, the OLTC tap setting position and the state of the capacitor banks were modeled as hard constraints, i.e. the limits of these variables were not allowed to be violated. The algorithm proposed in this paper is a metaheuristic devised for continuous search spaces, a simple rounding procedure was used to convert real variables into integer ones.

Soft constraints, i.e. the constraints that were modeled as penalties to two functions presented in Eq. (1), consist of the active power production of the REF bus, the voltage magnitude at PQ buses, the apparent power flow through branches, and the reactive power generation at REF and PV buses. The deviations from the limits were computed in per-unit system. The penalty function adopted f_p is described in Eq. (2) :

$$f_p = [1\ 1\ 1\ 1] \begin{bmatrix} max(0, Pg_{REF} - \overline{Pg_{REF}})^2 + max(0, \underline{Pg_{REF}} - Pg_{REF})^2 \\ \sum_{i=1}^{NB_{PQ}} [max(0, U_i - \overline{U_i}) + max(0, \underline{U_i})]^2 \\ \sum_{i=1}^{NC} [max(0, |S_{ij}| - \overline{S_{ij}}) + max(0, |S_{ji}| - \underline{S_{ij}})]^2 \\ \sum_{i=1}^{NB_{REf+PU}} [max(0, Q_i - \overline{Q_i}) + max(0, \underline{Q_i} - Q_i)]^2 \end{bmatrix} \tag{2}$$

The fitness functions used of the individuals computed by our algorithm depends on the penalty function selected and is defined as $\min C$ and P_{loss} respectively,

$$fit = C + 1 \times 10^7 \times f_p, \tag{3}$$

$$fit = P_{loss} + 1 \times 10^7 \times f_p, \tag{4}$$

in which f_p represents the penalty function.

2 hC-DEEPSO with Spiral Search

Evolutionary algorithms (EAs), popular algorithms in optimization research community, are methods that mimick the processes of Darwinian evolution. In practice, simple EAs are usually outperformed by hybrid EAs when solving complex optimization problems. The concept of hybridization refers to combine EAs with local search algorithms or to combine features of different optimization techniques to produce a more powerful algorithm. In this context, the combination of Differential Evolution (DE) and Particle Swarm Optimization (PSO) represents a promising way to create superior optimzers [33].

While being similar in terms of basic algorithm features and manipulating a population of candidate solutions, PSO and DE differ in terms of population sampling approach. Sampling in PSO is based on individual cognition and social collaboration while sampling in DE relies on differential mutation and crossover [33]. The hybrid algorithm, C-DEEPSO [18], creates a vector of differences inspired by the mutation operator of DE and introduces that vector in the movement equation pf PSO. Moreover, C-DEEPSO addresess the binomial recombination operator presented in the DE.

This metaheuristic constitutes an enhancement over Differential Evolutionary Particle Swarm Optimization (DEEPSO) [17] and can be viewed as an evolutionary algorithm with recombination rules borrowed from PSO or a swarm optimization method with selection and self-adaptiveness properties inherent from DE algorithm. Table 2 describes the main parameters of the C-DEEPSO algorithm.

Table 2 Parameters in C-DEEPSO.

Parameter	Description
t	is the current generation;
X	is the current solution;
X_{gb}	is the best solution found so far;
V	is the velocity of solution;
F	is disturbance rate;
$*$	indicates a parameter is subjected to the mutation process;
st	is a specific strategy by DE algorithm;
C	is a diagonal matrix of random variables sampled at each iteration (with probability P of communication);
w_I, w_A, w_C	are weights on the inertia, memory and communication in C-DEEPSO, respectively.

C-DEEPSO is based on the concept of biological evolution, in which a population of solutions evolves and gradually improves over time according to the adaptation of its individuals to the environment. Selection, recombination and mutation operators are applied to create new solutions. Recombination in C-DEEPSO is governed by the so called *Movement Rule*, which in DEEPSO is given by Eq. (5) and Eq. (6):

$$X_t = X_{t-1} + V_t, \tag{5}$$

$$V_t = w_i^* \times V_{t-1} + w_m^* \times (X_r - X_{t-1}) + w_c^* \times C \times (X_{gb}^* - X_{t-1}) \tag{6}$$

in which X_r is an individual different from X_{t-1} that can be obtained according to one of the following four options [34]:

1. S_g: sampled from all individuals in current generation;
2. P_b: sampled from a Memory B of the best individual found so far;
3. S_g-rnd: sampled as an uniform recombination from the individuals of the current generation, and;
4. P_b-rnd: sampled as an uniform recombination within Memory B.

Typically, the mutation of a generic weight w of an individual follows a simple additive rule as described by (7),

$$w^* = w + \tau \times N(0,1), \tag{7}$$

in which τ is the mutation rate that must be set by the user. $N(0,1)$ is a number sampled from the standard Gaussian Distribution.

Note that the mutated weight must not become negative nor greater than 1. Moreover, not only the weights presented in Eq. (5) are mutated but also X_{gb}. This attracting position is slightly moved in the search space using a Gaussian Distribution to prevent the population from being trapped in a given region of the search space. This is especially evident in those cases when the cooperation term becomes more dominating than the other terms. Mutation of X_{gb}, which is carried out for every particle, is performed according to the following equation:

$$X_{gb}^{*} = X_{gb}[1 + \tau \times N(0,1)]. \tag{8}$$

An analysis of the *Movement Rule* shows that DEEPSO is best described as a metaheuristic stemming from swarm intelligence with selection and self-adaptation abilities rather than from the canonical DE algorithm (see [35]). For the sake of the clarity, the traditional DE mutation operator is shown in Eq. (9):

$$v_{t,i} = x_{t,r1} + F(x_{t,r2} - x_{t,r3}); \quad r_1, r_2, r_3 \in \{1, \ldots, N\}, \tag{9}$$

in which parameters $x_{t,r1}$, $x_{t,r2}$ and $x_{t,r3}$ are different solutions randomly obtained from the population and F is a number that belongs to the interval $[0, 2]$, aiming to control the amplification of differential variation. It is easy to see that three vectors are used in DE mutation process of Eq. 9, while in DEEPSO movement rule, only two vectors are required, represented by X_r and X_{t-1}, which are used in Eq. 6. On the other hand, C-DEEPSO uses the original DE mutation operator, as described by Eq. (9).

The distinguishing feature of C-DEEPSO is to use instead of the a memory term, as in the classic PSO, a more general term, called *assimilation* term. This term is sensitive to macro-gradients, that lead to improvements in the fitness function. To obtain the assimilation term, C-DEEPSO relies on a collective memory instead of multiple and independent memories corresponding to the search experience of each individual. This memory, called Memory B, contains not only the position of the individual but also its fitness. A new way to generate X_r is proposed to ensure a better assimilation of the search space. This new strategy, named *S_gP_b-rnd*, is a combination of *S_g-rnd* and *P_b-rnd* strategies. In this case, when using *S_gP_b-rnd*, an uniform recombination from the solutions in Memory B and in the current population is used to obtain X_r. The *Movement Rule* for C-DEEPSO is described by Eq. (10) as:

$$V_t = w_I^{*} V_{t-1} + w_A^{*}(X_{st} - X_{t-1}) + w_C^{*} C(X_{gb}^{*} - X_{t-1}). \tag{10}$$

The superscript * in Eq (7) indicates that the corresponding parameter/quantity undergoes evolution under a mutation process. The strategy *st* used in this paper is *current-to-best* [36], which can be expressed by means of Eq. (11),

$$X_{st} = X_r + F(X_{best} - X_r) + F(X_{r1} - X_{r2}). \tag{11}$$

In Eq. (10) and Eq. (11), t denotes the current generation, X the current position or solution. Morever, X_{gb} is the best solution ever found by the population, V is the velocity of the individual, and C represents a $n \times n$ diagonal matrix of Bernoulli random variables that is sampled in every iteration. A communication rate, P, is applied to generate matrix C. The variables w_I, w_A and w_C are weights on the inertia, assimilation and communication terms, respectively. The terms X_{r1}

and X_{r2} are randomly sampled solutions. After the calculation of X_{st}, the corresponding solution is evaluated. If the fitness of X_{st} is better than that of X_r, then X_{st} receives the value of X_r. This operation is similar to the crossover operator in DE algorithm (see [35]). The superscript $*$ indicates that the corresponding parameter/quantity undergoes evolution under a mutation process. Figure 1 shows a planar (2D) representation of the*Movement Rule*, with focus on the different interations between solutions (inertia, cooperation and assimilation). Algorithm 1 describes a pseudo-code for C-DEEPSO.

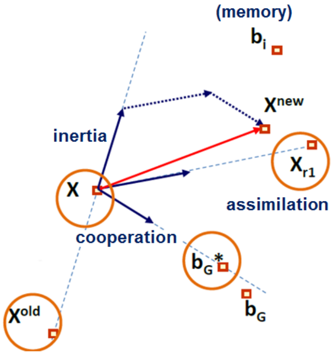

Fig. 1 Illustration of the movement equation for C-DEEPSO, showing similarity with PSO, however not having a memory term as attractor.

```
begin
    Step 1: Set values of control parameters of C-DEEPSO - Population size $NP$,
    Mutation rate $\tau$, Communication rate $P$, Memory rate $MB$, Generations limits for
    Local Search, $N_{gl}$, given a $\gamma$ rate of occurrence and Dimension $(D)$ ;
    Step 2: Set the generation number $t = 0$ and randomly initialize a population of $NP$ ;
    Step 3: Evaluate the current population, $NP$;
    Step 4: Update the global best solution, $X_{bg}$;
    while stopping criterion is not satisfied do
        for each individual i that belongs to population $NP$ do
            Calculate $X_r$ using the strategy, $S_g P_B$-$rnd$;
            Copy the current individual $X_{t-1}$;
            Mutate the strategy parameters $w_I$, $w_A$, $w_C$ using Eq. (7);
            Mutate $X^*_{gb}$ using Eq. (8) ;
            Apply movement rule in current solution using Eq. (10);
            Evaluate the current solution and your copy;
            Update the $X_{best}$ position;
            Select the individual with better fitness to proceed to next population
            $(NP + 1)$. *Using for example Stochastic Tournament;
        end
        Update the best individual $X_{bg}$ and memory $MB$;
        Execute Algorithm 2 until t $=<$ $N_{gl}$;
        $t = t + 1$
    end
end
```

Algorithm 1: *Pseudo-code of hC-DEEPSO.*

2.1 The Multipoint Search Operator and the Diversity Mechanism

Some real-world optimization problems present a high number of variables to optimize and, in this case, the domain search increases exponentially with the dimension size. Algorithms for solving these types of problems need to be carefully designed and more efficient than the ones applied to low dimensional problems. Due to the high dimensionality of SCOPF problems, the evolutionary algorithm will benefit from a inclusion of a multipoint search operator. The proposed search operator is inspired by the Spiral Optimization Algorithm (SOA) [37]. SOA is based on multipoint search for n dimensions applied to the solution of problems that have continuous characteristics, based on an analogy of the spiral shape natural phenomena. The basic idea is to explore the search space using logarithmic spiral shape, which can be observed in nature, such as in shells, some flowers, formation of cyclones and even galaxies [38]. The logarithmic spiral trajectory model, centered at $x^* \in R^n$, starting with an arbitrary point, is given by:

$$x^{(k+1)} = rM(\theta) \cdot x^{(k)} - (rM(\theta) - I_n) \cdot x^*, \tag{12}$$

in which the rotation angle θ belongs to the range $[0,\ 2\pi]$ around the origin in each k. The radius r, $0 < r < 1$ is a convergence rate of the distance between the current point and the origin, for each k. I_n is the n-dimensional identity matrix and $M(\theta)$ is the rotation matrix.

A general rotation in the n-dimensional space can be seen as the rotation of an axis i in direction to an axis j. The plane described by axis i and j is the rotation plane and the general matrix for main rotation, $M_{i,j}^{(n)}(\theta_{i,j})$, is given by,

$$M_{i,j}^{(n)}(\theta_{i,j}) = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & \cos(\theta_{i,j}) & \cdots & -\sin(\theta_{i,j}) & & \\ & & \vdots & \ddots & \vdots & & \\ & & \sin(\theta_{i,j}) & \cdots & \cos(\theta_{i,j}) & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}, \tag{13}$$

in which the blank elements are equal to zero.

It is worthwhile to observe that $M_{i,j}^{(n)}(\theta_{i,j})$ is almost an identity matrix except in the intersection of columns i and j with rows i and j, meaning that only the coordinates i and j will change after a $M_{i,j}^{(n)}(\theta_{i,j})$ rotation. This is consistent with the bi-dimensional and tri-dimensional cases. Since there are $\binom{n}{2}$ main planes in a n-dimensional space, this corresponds to the number of possible main rotations in that space. The rotation matrix in the n-dimensional space is obtained by the composition of all $(n(n-1)/2)$ existing combinations [39]. Therefore, $M^n(\theta)$ can be defined by Eq. (14), in which n is the number of dimensions:

$$M^{(n)}(\theta) = \prod_{i<j} M_{i,j}^{(n)}(\theta_{i,j}). \tag{14}$$

The n-dimensional logarithmic spiral model is given by,

$$x^{(k+1)} = rM^{(n)}(\theta) \cdot x^{(k)} - (rM^{(n)}(\theta) - I_n) \cdot x^*, \tag{15}$$

in which the rotation angle θ belongs to the range $[0,\ 2\pi]$ around the origin in each k. The radius r, $0 < r < 1$ is a convergence rate of the distance between the current point and the origin, for each k. I_n is the n-dimensional identity matrix and the $M(\theta)$ is the rotation matrix. In a 3-dimensional space, Figure 2 shows the trajectory of 50 points starting at $x_0 = (10,\ 10,\ 10)$ and using Eq. (15).

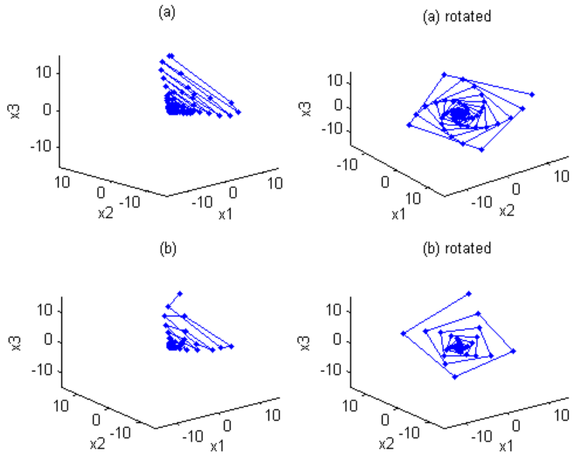

Fig. 2 Examples of logarithmic spiral trajectories using Eq. (12). Different values of convergence rate r produce distinct spiral shape behaviours. (a) trajectory using $r = 0.95$; $\theta = \pi/4$ and (b) trajectory using $r = 0.90$; $\theta = \pi/4$. The right-hand side column shows a rotated version of the corresponding spiral.

It is important to notice that the choice of the number of points used to build the spiral, the radius and angle of the rotation matrix affect the shape of the logarithmic spiral. Four different spirals are shown in Figure 3. Each spiral is constructed using the parameters given in Table 3.

Table 3 Illustration parameters

(a)	$r = 0.95$	$\theta = \pi/4$	$popsize = 25$
(b)	$r = 0.95$	$\theta = \pi/4$	$popsize = 50$
(c)	$r = 0.80$	$\theta = \pi/4$	$popsize = 25$
(d)	$r = 0.95$	$\theta = \pi/7$	$popsize = 25$

Such, for coupling the idea inspired on SOA as a local search operator inside C-DEEPSO, some definitions must be given:

- Given an occurrence rate (γ), the new local search operator will be run every generation, until a specified generation N_{gl};
- Radius and angle are randomly chosen and the number of points to be used to construct the spiral is given by the user;
- The number of generated points is directly related to the initialization parameters used in SOA. For convenience, the number of points generated will be the same as the population started in C-DEEPSO;

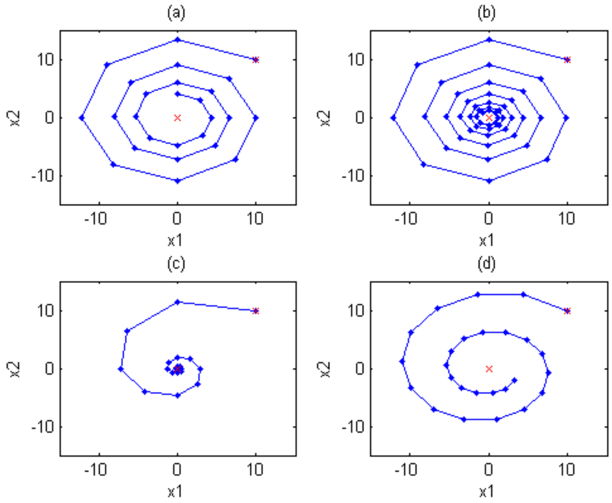

Fig. 3 Examples of logarithmic spiral trajectories using Eq. (12). Different values of convergence rate r produce distinct spiral shape behaviours. (a) trajectory using $r = 0.95$, $\theta = \pi/4$ and $popsize = 25$, (b) $r = 0.95$, $\theta = \pi/4$ and $popsize = 50$, (c) $r = 0.80$, $\theta = \pi/4$, $popsize = 25$ and (d) $r = 0.95$, $\theta = \pi/7$ and $popsize = 25$.

- The center point of the logarithmic spiral is X_{best} since the main idea of the local search operator is to widen the search around a promising space region. Preliminary experiments were conducted using different point as center point of the logarithm spiral and the results showed that using X_{best} as the center point the multipont search is more effective;
- An additive inverse logarithmic spiral is introduced with respect to X_{best}, to intensify the exploration of possible individuals close to X_{best};

In this new approach, a sampled set of individuals, totally randomized of logarithmic spirals is generated, allowing to name this new local search operator as Spiral Search (SLS). For purposes of illustration of a spiral and its additive inverse used in the SLS operator, Figure 4 shows two spirals starting at point at points $x_{0+} = (10, 10, 10)$ and $x_{0-} = (-10, -10, -10)$.

The procedure of SLS operator corresponds to the generation of spiral samples starting at X_{best}. A population sampled by the obtained spiral is evaluated on the original objective function and the individual who has lowest objective function value is named X_{SLS}. The SLS operator is elitist, meaning that the worst individual, X_{worst}, in the current population of C-DEEPSO, is replaced by X_{SLS}.

The proposed search operator helps C-DEEPSO algotithm to find a local optimum in the first generations of the search. Not only the usual selection pressure pushes C-DEEPSO to focus more and more on already discovered better regions, but also does the search operator in-depth. As a result, population diversity declines and premature convergence may occur. Although premature convergence is not caused by the diversity loss, maintaining a certain degree of diversity is widely believed to help to avoid entrapment in non-optimal solutions. Having that in mind, a simple diversity mechanism is proposed. This diversity mechanism uses a

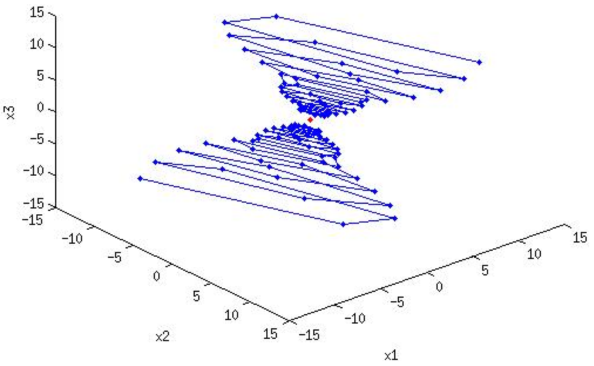

Fig. 4 An example of reverse spiral additive, in a 3-dimensional space, starting at $x_{0+} = (10, 10, 10)$ and $x_{0-} = (-10, -10, -10)$.

distance-to-best measure to alternate between exploration and exploitation. The distance to an average-point, X_d, amongst the population individuals and X_{best}, is obtained:

$$X_d = \frac{1}{NP}\sum_{i=1}^{NP}\sqrt{\sum_{j=1}^{N}(x_{ij} - x_{best})^2}, \tag{16}$$

in which NP is the population size and N is the problem dimension. New individuals $X_{d+} = X + X_d$ and $X_{d-} = X - X_d$ are generated and, then, evaluated on the original objective function. The best NP individuals are then selected for the next generation. Algorithm 2 presents the pseudo-code of the search algorithm, SLS, and the diversity mechanism proposed in this work.

```
if $rand(\ ) > \gamma$ then
    if $(t \leq N_{gl})$ then
        Step 1: Initialize the number of $m$ search points as $m = \#NP$, generate a random value for $\theta$ in the range $[0, 2\pi]$, generate a random value for $r$ in the range $[0, 1]$ to $rM^{(n)}(\theta)$ ;
        Step 2: Generate $m$ individuals (a sample population spiral and your inverse) from $X_{best}$ using Eq. (15);
        Step 3: Evaluate the population $m$;
        Step 4: Replace in $NP$: $X_{SLS}$ by $X_{worst}$;
    else
        for each individual $i$ that belongs to population $NP$ do
            Calculate $X_d$ using Eq. (16) in each dimension;
            Generate $X_{d+} = X_{(i)} + X_d$ and $X_{d-} = X_{(i)} - X_d$;
            Evaluate populations $X_{d+}$ e $X_{d-}$;
            Sort $X_{d+}$, $X_{d-}$ e $NP$;
            Replace new population $\#NP$ by the best fitness individuals from the previous step;
        end
    end
end
```

Algorithm 2: *Pseudo-code of Spiral Local Search (SLS) operator.*

According to [40], it is now well established that pure population-based algorithms are not well suited to refinement of complex combinatorial spaces and that hybridization with other techniques can significantly improve search efficiency. In this way, the proposed algorithm unites different concepts of evolutionary optimization in order to be a viable approach to solve real world problems. The proposed algorithm, the C-DEEPSO coupled with the search operator and diversity mechanism, is called Hybrid Canonical Differential Evolutionary Particle Swarm Optimization (hC-DEESPO).

3 Simulation Results and Discussion

In this section, the performance of the hC-DEEPSO for solving SCOPF problems is evaluated on the IEEE Test Beds. The three IEEE test systems in the ORPD and OARPD are presented and discussed. The experimental setup is divided into the following case studies:

1. to verify the performance of hC-DEEPSO in the preliminary experiment using benchmark functions by literature to compare our algorithm with the canonical PSO and DE methods.
2. the effect of the cannonical DE mutation operator in and the multipoint search operator insertion in the hC-DEEPSO algorithm are analyzed;
3. a parameter fine-tuning for the hC-DEEPSO, regarding the mutation and communication rates, is conducted; and
4. a statistical comparison of the results obtained by hC-DEEPSO, in its best parameters, with three benchmark algorithms available in the literature - DEEPSO, ICDE and MVMO - is performed.

3.1 Preliminar experiment

A preliminary experiment using two benchmark functions of literature - Rastringin and Rosenbrock - was realized. The goal for this experiment was verify the results obtained in hC-DEEPSO in a comparison with the results of canonical PSO and DE algorithms presented in [41] The initialization parameters that hC-DEEPSO used was: Mutation rate 0.5 and Communication rate 0.9. The number of total fitness values was set at 1×10^5. For each dimension (30, 50, 100) the algorithms tested using the population size (30, 50, 100) respectively and was tested 30 times. The Eq. (17) and Eq. (18) shows the benchmark functions:
-Rastringin function - Multimodal - Goal = 0,

$$f(x) = \sum_{i=1}^{D}[x_i^2 - 10cos(2\pi x_i) + 10]. \tag{17}$$

-Rosenbrock function - Unimodal - Goal = 0,

$$f(x) = \sum_{i=1}^{D-1}[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]. \tag{18}$$

Table 4 shows the results of the PSO, DE and hC-DEEPSO algoritmhs. The results showed that hC-DEEPSO obtained a good performance in relationship to canonical DE and PSO algorithms in to solve the Rastringin and Rosenbrock functions in different dimensions. Thus, it is possible show that hC-DEEPSO is able to overcome the results of the base algorithms used for its design in continuous problems.

Table 4 Mean and standard values for PSO, DE and hC-DEEPSO on Rastringin and Rosenbrock functions in 30, 50 and 100 dimensions.

D = 30	Rastringin		Rosenbrock	
Algorithm	Mean	Std	Mean	Std
PSO	37.81	7.45	81.27	41.21
DE	2.53	5.19	31.13	17.12
hC-DEEPSO	**0.00**	**0.00**	**15.99**	**1.90**
D = 50	Rastringin		Rosenbrock	
PSO	75.30	19.55	174.22	113.63
DE	41.47	8.80	50.33	16.85
hC-DEEPSO	**0.00**	**0.00**	**38.90**	**2.54**
D = 100	Rastringin		Rosenbrock	
PSO	186.04	4.93	250.68	24.64
DE	261.19	1.64	91.23	3.82
hC-DEEPSO	**0.00**	**0.00**	**89.60**	**1.90**

3.2 IEEE Test Systems - study cases

Three well-known IEEE test systems - 57 bus, 118 bus and 300 bus - in the ORPD and OARPD problems are presented. The IEEE 57 bus test and the IEEE 118 bus test represent a portion of the American Electric Power System, in the Midwestern US, as it was in the early 1960's. The IEEE 300 bus test was developed, in 1993, by the IEEE Test System Task force. Figure 5 shows a single-line diagram of each system. IEEE 57 bus system has 25 optimization variables in ORPD, comprising 7 continuous variables associated to generator voltage set-points, 15 discrete variables associated to stepwise adjustable on-load transformer tap positions, and 3 binary variables associated to switchable shunt compensation devices. On the other hand, the number of optimization variables in OARPD is 31, in which the extra 6 continuous variables correspond to the active power production of generators. In both problems, in addition to the expected operation scenario, two contingency scenarios are considered (N-1 criterion) corresponding to the outage of branches 8 and 50. The number of constraints is 178 for non-contingency conditions, and 177 for each N-1 condition.

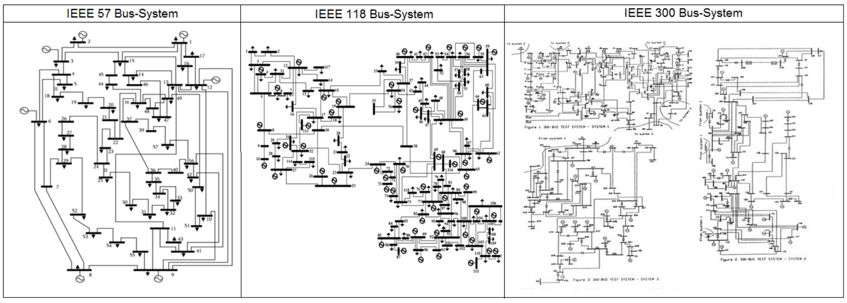

Fig. 5 Diagram systems by IEEE 57, 118 and 300 bus.

IEEE 118 bus system has 77 optimization variables in ORPD, comprising 54 continuous variables associated to generator bus voltage set-points, 9 discrete variables associated to stepwise adjustable on-load transformer tap positions, and 14 binary variables associated to switchable shunt compensation devices. The difference between ORPD and OARPD is the additional number of control variables (total of 130) that represent the active power production of generators. In both problems, considered contingency scenarios result from outages in branches 21, 50, 16, and 48 were considered. The number of constraints is 492 for non-contingency conditions, and 491 for each N-1 condition.

Finally, IEEE 300 bus system has 145 variables in ORPD, comprising 69 continuous variables associated to generator bus voltage set-points, 62 discrete variables associated to stepwise adjustable on-load transformers and 14 binary variables associated to switchable shunt compensation devices. The total number of control variables in the OARPD problem is 213. Table 5 presents a summary of all characteristics for each system. In both problems, three contingency scenarios are considered as a result of outages at branches 187, 176, and 213. The number of constraints is 651 for non-contingency conditions, and 950 for each N-1 condition.

Table 5 Summary of characteristics presented by IEEE 57, 118 and 300 bus systems. The number of additional security constraints is indicated between bracktes (index - 1:ORDP and 2:OARDP).

	57_1	57_2	118_1	118_2	300_1	300_2
Continuous Variables	7	13	54	107	69	137
Discreet Variables	15	15	9	9	62	62
Binary Variables	3	3	14	14	14	14
Dimension	25	31	77	130	145	213
Constraints	178 +(354)	178 +(354)	492 +(1473)	492 +(1473)	651 +(1900)	651 +(1900)

3.3 Study on Efficacy of Hybrid C-DEEPSO

The difference between DEEPSO and C-DEEPSO lies in the usage of the cannonical DE mutation operator inside the Movement Rule. Despite the good performance of DEEPSO when applied to SCOPF problem, the efficacy of hC-DEEPSO needs to be assessed. For that, it is necessary to investigate the benefits of the inclusion of the cannonical DE operator when applied to the SCOPF problems. Moreover, the performance of the Multipoint Search insertion must be analysed. For assessing the performance of DEEPSO, C-DEEPSO and hC-DEEPSO when applied to SCOPF problems, an experiment using the IEEE 57 bus system was conducted. To have a fair comparison, all initialization parameters were the same: NP $= 60$; MB $= 6$; $\gamma = 0.5$; $N_{gl} = 20$; Mutation $= 0.6$; Communication $= 0.2$; Maximum number of function evaluations $= 5 \times 10^3$. Each algorithm was executed 31 times. Figure 6 presents the mean convergence line for DEEPSO, C-DEEPSO and hC-DEEPSO throughout the 5×10^3 function evaluations.

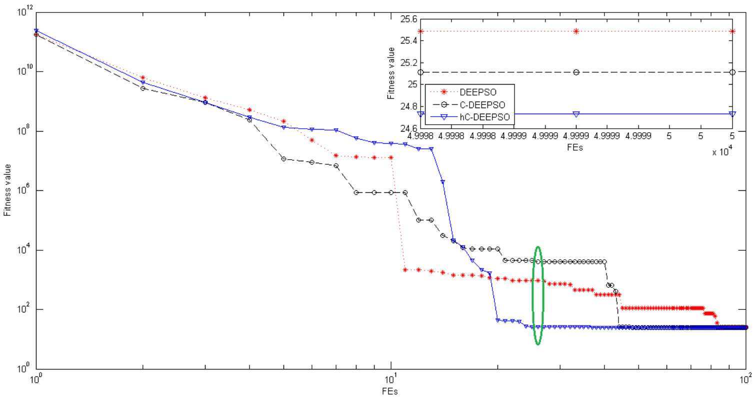

Fig. 6 Mean convergence line for the IEEE 57 bus system/ ORDP. The x-axis represents the function evaluations and the y-axis the mean value of the objective function over the 31 runs.

Analysing Figure 6, it indicates the hC-DEESPO is the first algorithm to converge followed by C-DEEPSO. The magnified area shows the mean value of each algorithm obtained in a specific range of function evaluation counter. The obtained mean value for hC-DEEPSO was 25.91 MW while the obtained mean values of C-DEEPSO and DEEPSO were 4456 MW and 927.6 MW, respectively. The fact indicates the multipoint search operator, SLS operator, plays an important role in the convergence velocity when compared. Figure 7 illustrates the experimental results (corresponding to the total operation cost) of the 31 runs using boxplots for the three algorithms. The Figure 7 shows that hC-DEEPSO performed better as compared to DEEPSO and C-DEEPSO.

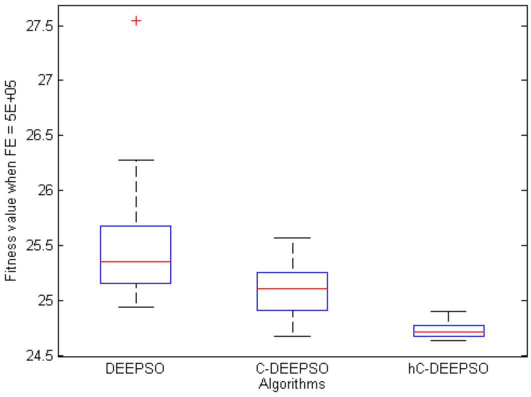

Fig. 7 Comparison of hC-DEEPSO, C-DEEPSO and DEEPSO - IEEE 57 bus system/ ORDP.

This behaviour can be statistically assessed via an ANOVA followed by a Tukey test. The p-value, claculated using a 5% significancy level, was 6.19×10^{-13}. Figure 8 shows the Tuckey test results. The result indicates the hC-DEEPSO presents a lower mean value of the total operation cost when compared to C-DEEPSO and

DEEPSO and C-DEEPSO presents a lower mean value of the total operation cost when compared to DEEPSO. In this way, it is possible to state that the usage of the cannonical DE mutation operator and the multipoint search operator enhances the performance of the algorithm when applied to IEEE 57 bus system. The main idea behind the multipoint search operator is to sample the search space more effectively, exploring some neighbors of the best solution. In large scale optimization problem context, such as SCOPF problems, a more effective search space sampling is a key feature to overcome the dimensionality curse when designing powerful algorithms.

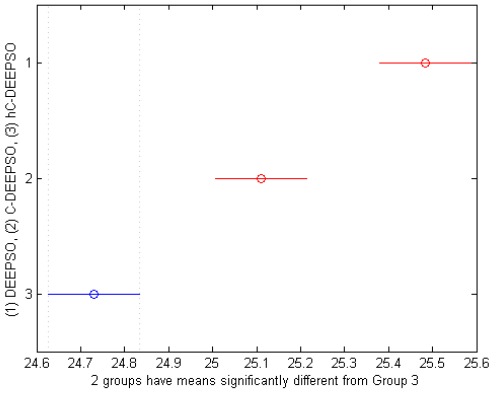

Fig. 8 Tucky Test for each algorithm using IEEE 57 bus system/ ORDP.

3.4 Parameter Fine-Tuning

To determine the optimum value of mutation rate and communication probability, the Response Surface Methodology has been chosen as the experimental setup to perform the fine-tuning of the parameters. This methodology is a collection of mathematical and statistical techniques that are useful for modeling and analysis in applications in which the output variable is influenced by many variables and the goal is to optimize that output variable [42]. Using a specific design of experiments, the goal is to optimize the response (output variable) of the hC-DEEPSO algorithm which, in this work, is influenced by two input variables,the mutation rate and the communication probability.

Making some changes on mutation rate and communication probability, it is possible to identify the corresponding changes in the output response. In this case, each output response corresponds to each objective function regarding minimization of total system losses and total cost of production. Each objective function f then is a function of mutation (τ) and communication (P) levels. The response expected by the function $f(\tau, P)$ can be represented graphically in the three-dimensional space and is called the response surface. The goal is to find the best parameter values, τ and P, that minimize each objective function.

To construct an approximation model that captures the interaction between those input variables, τ and P, a factorial experiment is necessary [42]. In simple words, in a factorial experiment the input variables are varied at the same time instead of one at a time. The lower and the upper bounds for mutation and communication rates are empirically set to [0.2, 0.9]. Each input variable is defined at only the upper and down level bounds at each discretization of the predefined range. In total, hC-DEEPSO is run 1984 times for each test problem.

Using the input variable values and the corresponding value for each objective function and test scenario in SCOPF, a second-order model for the response surface can be fitted. All executions were performed on a small Cent-OS Cluster composed by 32 Intel Xeon E5-1650 3.5GHz cores and 32GB of RAM memory. The other initialization parameters, empirically chosen, are shown in Table 6. Figure 9 shows the response surfaces, and their corresponding contour plots, obtained using the mean and best objective function values found on ORDP problem - IEEE 57 bus system.

Table 6 Initialization parameters

	IEEE Bus Systems					
Parameters	57_1	57_2	118_1	118_2	300_1	300_2
NP	60	80	80	140	150	200
MB	6	8	8	14	15	20
N_{gl}	10	10	10	10	10	10
γ	0.5	0.5	0.5	0.5	0.5	0.5
MaxFitEval	5×10^3		1×10^5		3×10^5	

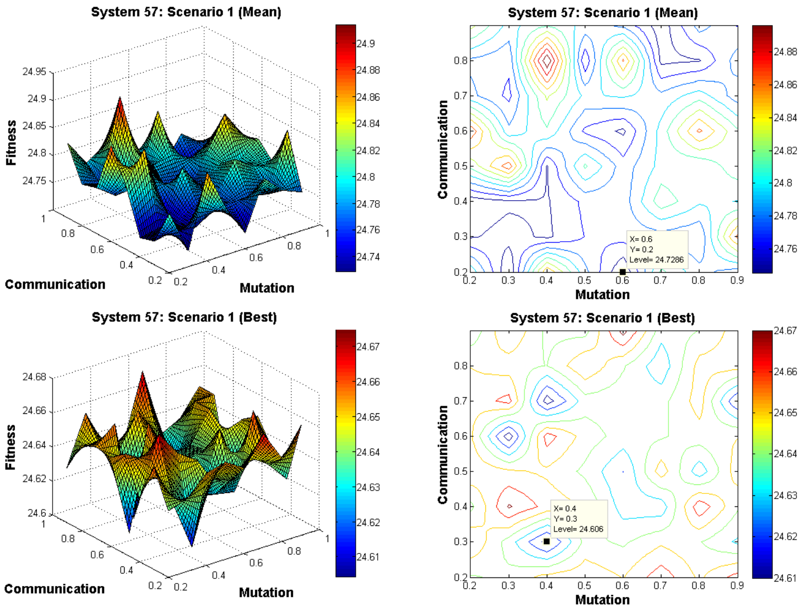

Fig. 9 Response surfaces, and their corresponding contour plots, obtained using the mean and best objective function values of IEEE 57 Bus System in ORPD problem.

It can be observed that the best set of parameters obtained for mutation and communication is [τ, P] =[0.6, 0.2] with a mean value of total active power loss equal to 24.7286 MW. Furthemore, using the response surface generated via the best objective function values, the best set of parameters is [τ, P] = [0.4, 0.3], resulting in a total active power loss of 24.606 MW. Similarly, hC-DEEPSO is used to solve OARDP problems, whose objective is to minimize the total cost of energy production. Figure 10 shows the response surfaces, and their corresponding contour plots, obtained using the mean and best objective function values found on OARDP problem for the IEEE 57 bus system scenario.

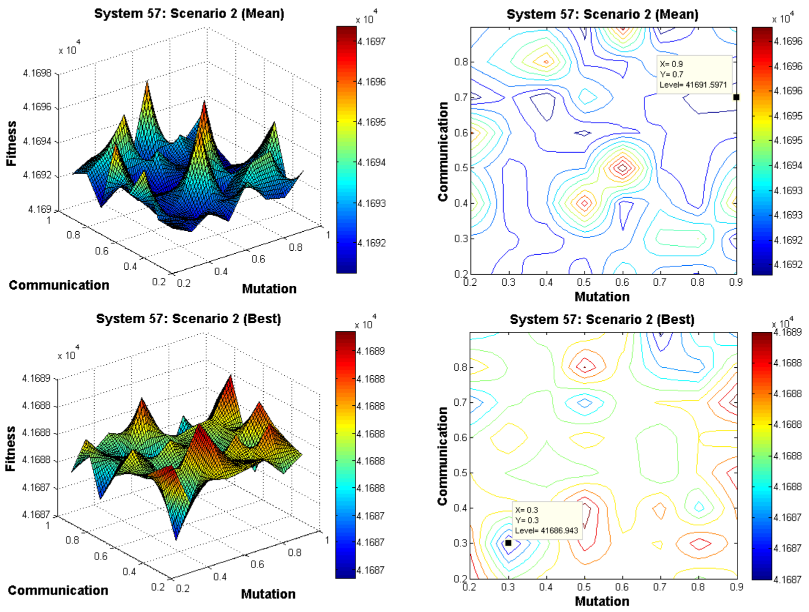

Fig. 10 Response surfaces, and their corresponding contour plots, obtained using the mean and best objective function values of IEEE 57 Bus System in OARPD problem.

In OARPD problem it can be observed that the best set of parameters obtained for mutation and communication is [τ, P] =[0.9, 0.7] with an mean value of total cost of energy production equal to 41691.5 $\$/h$. Using the response surface generated via the best objective function values, the best set of parameters is [τ,P] = [0.3, 0.3], resulting in a total cost of energy production of 41686.9 $\$/h$.

Same methodology was applied to IEEE 118 bus system and IEEE 300 bus system using ORDP and OARDP problems. In all cases, the response surfaces and this corresponding contour plots were generated using the mean and the best objective function values. Similar response surfaces were obtained and the best configuration is [τ, P] = [0.9, 0.4] for a mean result of 117.9 MW regarding the total active power losses and, when using the best objective function values, [τ, P] = [0.6, 0.4] resulting in a total active power losses equal to 117.4 MW.

For OARDP within IEEE 118 bus system scenario and using the mean objective function values, the best parameter configuration is [τ, P] = [0.7, 0.2] resulting in 135560.03 $\$/h$ of a total cost of energy production. Using the best objective function values, the best parameter configuration is [τ, P] = [0.7, 0.2] resulting in 722160.00 $\$/h$ of total cost of energy production. The best set of parameters obtained for mutation and communication is [τ, P] =[0.8, 0.3] with an mean value of total active power losses equal to 394.048 MW. Furthermore, using the response surface generated via the best objective function values, the best set of parameters is [τ, P] = [0.9, 0.6], resulting in a total active power loss of 387.09 MW.

The obtained results in factorial experiment show that the hC-DEEPSO algorithm performs best when the mutation rate used is greater than 0.5 in all cases and that the communication rate was always lower than the communication rate. It can be concluded that the DE mutation operator incorporated the motion equation of hC-DEEPSO produced a beneficial effect to the algorithm. In addition, it is conclusive that the global optimum information at each iteration is not an effective measure since the communication rates were always less than 0.5 (except for IEEE 57 bus system - OARPD).

3.5 Performance Analysis

After having the best parameter configuration, τ and P, for hC-DEEPSO to solve ORDP and OARDP problems in the considered test scenarios, it is necessary to evaluate its performance analysis comparing with some available state-of art algorithms. For this task, three different algorithms, DEEPSO [17], MVMO [19] and ICDE [43] in their best overfit initialization parameters, are chosen. These algorithms were used to solve SCOPF problems and were extracted from *Competition on Application of Modern Heuristic optimization algorithms for solving optimal power flow problems* [44]. The competition represents an initiative for development of power system optimization test beds, providing some comparative analysis, in the field of heuristic optimization. Preliminary tests proved that all test cases are solvable however finding feasible solutions has proven to be a hard-to-solve task.

Although MVMO was not a participant algorithm, it was used to produce reference results for comparisons. It is worthwhile to notice that DEEPSO was the winner of the 2014 Competition. The other algorithms participating on that competition have not been considered in this comparison, because they violated the restrictions of the given problem and given their inferior performance. It is important to highlight that some algorithms in the competition are tailor-made while others utilize specialized mechanism to handle the continuous and discrete variables.

In the experimental setting, the SCOPF problems were treated as black-box task which should be solved for the test cases. Final results for each optimization test bed over 31 independent runs were used in the statistical test to assess the algorithm performance. The only stopping criteria is the completion of the maximum number of function evaluations: in the IEEE 57 bus system, the maximum number of function evaluation is set to 50000, in IEEE 118 bus system it is set to 100000 and, in the IEEE 300 bus system, the maximum number of function evaluations is set to 300000. All the algorithms extracted from the competition have been fine-tuned. So, the best parameters for mutation rate and communication probability obtained in each test case were used in hC-DEEPSO. Table 7 shows the results of the three algorithms used to solve SCOPF problems [44] and results for hC-DEEPSO.

Table 7 Mean and standard values for hC-DEEPSO, DEEPSO, ICDE and MVMO on ORDP and OARDP problems using the three test scenarios.

IEEE 57 bus	ORDP (MW)		OARDP $(\$/h)$	
Algorithm	Mean	Std	Mean	Std
hC-DEEPSO	**24.7286**	**0.1064**	**41691.4521**	**2.804**
DEEPSO	25.5725	0.4125	41697.58	7.9393
ICDE	25.0414	0.42956	41739.81	27.6884
MVMO	24.8172	0.12504	41707.6893	10.5581
IEEE 118 bus	ORDP (MW)		OARDP $(\$/h)$	
hC-DEEPSO	**117.9541**	**0.3048**	135560.03	320
DEEPSO	119.2799	1.7131	135890.716	221.991
ICDE	128.3549	2.4535	154234.873	12324.2502
MVMO	117.862	0.42379	**135050.21**	**27.5014**
IEE 300 bus	ORDP (MW)		OARDP $(\$/h)$	
hC-DEEPSO	**394.0480**	**3.8037**	**722160.00**	**533.3307**
DEEPSO	414.6239	42.0460	724041.13	2669.9100
ICDE	8669.8730	15255.7100	740013.64	6205.4528
MVMO	394.1294	8.2219	722323.24	869.3588

The mean values and standard deviation, for all the runs, are shown and, since it is a minimzation problem, lower mean values are preferable over the higher ones. The best mean and standard deviation values are marked in boldface. In a very raw analysis, the hC-DEEPSO shows better results than the DEEPSO and ICDE algorithms, when compared in terms of mean values in all the three test scenarios for the ORDP problem. The same can be seen for OARDP problem, except in IEEE 118 bus system scenario. In this case, MVMO has a better mean value when compared to hC-DEEPSO.

For assessing the statistical significance of the results, it is necessary to perform an statistical inference on the difference between each pair of mean values. A T-test [42] can be performed to compare more effectively the obtained results. Such test can be used if one wants to compare results of population samples on a "two by two" basis. The null hypothesis adopted in this work is the equality of means, so the experiment is designed to detect whether this hypothesis is rejected or not using a significance value $\alpha = 5\%$. If the null hypothesis is reject, a *post hoc* comparison can be done to determine which one is the best algorithm. Tables 8 and 9 show the results of the T-test, using the P-Value calculated, and the results after the *post hoc* comparison for ORDP and OARDP problems, respectively. Blank values mean that the *post hoc* comparison is not able to detect any statistical difference between the algorithms.

Analyzing ORDP problem results, T-test indicated that, with 95% of confidence, hC-DEEPSO shows better results compared to DEEPSO, ICDE and MVMO in IEEE 57 bus system. For IEEE 118 bus system and IEEE 300 bus system scenarios, hC-DEEPSO is better than DEEPSO and ICDE. However, the statistical test is not able to detect any difference between hC-DEEPSO and MVMO. Examining OARDP problem results, the T-test indicated that, with 95% of confidence, hC-DEEPSO shows better results compared to DEEPSO, ICDE and MVMO in IEEE 57 bus system. For IEEE 118 bus system and IEEE 300 bus system scenarios, hC-DEEPSO is better than DEEPSO and ICDE. However, MVMO is better than hC-DEEPSO in IEEE 118 bus system and the statistical test is not

Table 8 Comparing results using T-test and a *post hoc* comparison in ORDP problems

ORDP			
IEEE 57 bus	**P-Value**	**Action**	**Winner**
hC-DEEPSO x DEEPSO	4.44E-16	reject	hC-DEEPSO
hC-DEEPSO x ICDE	2.18E-04	reject	hC-DEEPSO
hC-DEEPSO x MVMO	3.87E-04	reject	hC-DEEPSO
IEEE 118 bus	**P-Value**	**Action**	**Winner**
hC-DEEPSO x DEEPSO	2.77E-05	reject	hC-DEEPSO
hC-DEEPSO x ICDE	00E+00	reject	hC-DEEPSO
hC-DEEPSO x MVMO	0.329	not reject	-
IEEE 300 bus	**P-Value**	**Action**	**Winner**
hC-DEEPSO x DEEPSO	8.67E-03	reject	hC-DEEPSO
hC-DEEPSO x ICDE	3.70E-03	reject	hC-DEEPSO
hC-DEEPSO x MVMO	0.960	not reject	-

Table 9 Comparing results using T-test and a *post hoc* comparison in OARDP problems

OARDP			
IEEE 57 bus	**P-Value**	**Action**	**Winner**
hC-DEEPSO x DEEPSO	1.23E-03	reject	hC-DEEPSO
hC-DEEPSO x ICDE	1.58E-13	reject	hC-DEEPSO
hC-DEEPSO x MVMO	1.22E-10	reject	hC-DEEPSO
IEEE 118 bus	**P-Value**	**Action**	**Winner**
hC-DEEPSO x DEEPSO	1.55E-11	reject	hC-DEEPSO
hC-DEEPSO x ICDE	8.94E-12	reject	hC-DEEPSO
hC-DEEPSO x MVMO	1.85E-12	reject	MVMO
IEEE 300 bus	**P-Value**	**Action**	**Winner**
hC-DEEPSO x DEEPSO	2.92E-04	reject	hC-DEEPSO
hC-DEEPSO x ICDE	00E+00	reject	hC-DEEPSO
hC-DEEPSO x MVMO	0.3764	not reject	-

able to detect any difference between hC-DEEPSO and MVMO in IEEE 300 bus system.

The experimental results showed indicates that the proposed hC-DEEPSO was able of operating and controlling the operation of large power systems successfully, in an economical and technical solution for optimizing the fuel generation cost and power losses and voltage profile. In particular, if we make a future projection of the economy for the dispatch in the IEEE 300 bus system, it is possible to observe an economy of approximately \$1.4 million using hc-DEEPSO instead of MVMO for the period of one year.The proposed hC-DEEPSO demonstrated the effectiveness comparable or better than those presented by algorithms in the literature. The hC-DEEPSO has complied with all restrictions imposed by the problems if it shows a safe approach to the operation.

4 Conclusion

The competitive structure of modern energy available resources. Following this, new techniques are being developed for optimization of the operation in large scale networks. In this paper, a hybridization of DE and PSO coupled with a specialized search operator was presented for solving the non-linear, highly constrained, mixed-integer, multimodal Optimal Power Flow problems. To ensure adequate continuity and security of the energy supply, some contingencies of equipment are included in the model in order to preserve the inequality constraints in case of any equipment outage. In this case, the problem is known as Security Constrained Optimal Power Flow problem. The proposed hC-DEEPSO, a hybrid metaheuristic that incorporates distinct features of DE and PSO, is applied to solve SCOPF problems, ORPD and OARPD. Due to a large number of constraints and contingencies in SCOPF problems, the mathematical solution to these problems can become hard to find and finding a feasible solution is a very hard task. To overcome the inherent difficulties of these problems, a multipoint search operator is proposed aiming to find a local optimum in the beginning of the search and, to maintain a certain degree of diversity, a simple diversity mechanism is also applied. The proposed multipoint search operator, in order to carry out an in-depth investigation from the best individual in the population, guarantees hC-DEEPSO to achieve a better performance than its original version.

The effectiveness of the proposed algorithm was investigated through a comparative study using the algorithm version without the canonical DE mutation operator and the multipoint search operator. Using a test case, the proposed hC-DEEPSO has been proven to be more efficient when compared with the previous version. A parameter fine-tunning for hC-DEEPSO was also carried out using the Response Surface procedure and, using the optimal parameters, hC-DEEPSO was tested against three different algorithms, DEEPSO, ICDE nd MVMO, to solve SCOPF problems using three test scenarios. Experimental results showed that hC-DEEPSO obtained competitive results with those present in the literature. According to statistical comparison, the results for IEEE 57 bus system showed that, with 95% significance level, hC-DEEPSO proven to be better than ICDE, DEEPSO and MVMO algorithms in all test cases. In ORDP problem, for IEEE 118 bus system and IEEE 300 bus system scenarios, hC-DEEPSO was better than DEEPSO and ICDE. However, the statistical test was not able to detect any difference between hC-DEEPSO and MVMO in both scenarios. In OARDP problem, for IEEE 118 bus system and IEEE 300 bus system scenarios, hC-DEEPSO was better than DEEPSO and ICDE. However, MVMO was better than hC-DEEPSO in IEEE 118 bus system and the statistical test was not able to detect any difference between hC-DEEPSO and MVMO in IEEE 300 bus system. Therefore, hC-DEEPSO is a relevant hypothesis tested in this research as a method of good performance and suitable to be used to solve problems of this nature.

Acknowledgements

The authors would like to thank CEFET-MG, ITAS and INESC TEC for the infrastructure used by this project and also CAPES, CNPq and FAPEMIG for the financial support. This work is financed by the BE MUNDUS Project, the Helmholtz-Project Energy System 2050.

References

1. S. Frank, I. Steponavice, and S. Rebennack. Optimal Power Flow: A Bibliographic Survey (I) - Formulations and deterministic methods. *Energy Systems. Springer. vol .3, pp.:221-*

258, 2012.

2. S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: a bibliographic survey II: Non-deterministic an Hybrid methods. *Energy Systems, Springer, vol 3, pp.:259-289*, 2012.
3. M. Bhaskar, S. Muthyala, and S. Maheswarapu. Security Constraint Optimal Power Flow (SCOPF) - a compreensive Suvery. *International Journal of Computer Applications, vol. 2, n.6, pp.:42-52*, 2010.
4. J. Carpertier. Contribution to the economic dispatch problem. *Bull. Soc. Fr. Electri. vol. 8, no. 3, pp.: 431-447*, 1962.
5. D. Phan and J. Kalagnanam. Some Efficient Optimization Methods for Solving the Security-Constrained Optimal Power Flow Problem. *IEEE Trans. Power Syst., vol. 29, no. 3, pp. 863-872,*, 2014.
6. A. Ela, M. Abido, and A. Spea. Optimal power flow using differential evolution algorithm. *Electric Power System Research, vol 80, pp.: 878-885*, 2010.
7. M. N. Suharto, M. Y. Hassan, M. S. Majid, M. P. Abdullah, and F. Hussin. Optimal Power Flow Solution Using Evolutionary Computation Techniques. *Proc. in: IEEE Region 10 Conference TENCON, v.1. pp.: 1-8*, 2011.
8. M. Abido and N. Ali. Multi-objective Optimal Power Flow Using Differential Evolution. *Arabian Journal for Science and Engineering, vol. 37, no. 4, pp.: 991-1005*, 2012.
9. J.J.Liang, X.B. Mao, B.Y. Qu, B. Niu3, and T.J. Chen. Elite Multi-Group Differential Evolution. *WCCI 2012 IEEE World Congress on Computational Intelligence*, 2012.
10. Q. Kang, M. Zhou, J. An, and Q. Wu. Swarm Intelligence Approaches to Optimal Power Flow Problem With Distributed Generator Failures in Power Networks. *IEEE Transactions on Automation Science and Engineering, vol. 10, pp.: 343-353*, 2013.
11. S. Dixit, L Srivastava, and G. Agnihotri. Minimization of Power Loss and Voltage Deviation by SVC Placement Using GA. *International Journal of Control and Automation, Vol.7, No.6, pp.95-108*, 2014.
12. J. Radosavljevic, D. Klimenta, M. Jevtic, and N. Arsic. Optimal Power Flow Using a Hybrid Optimization Algorithm of Particle Swarm Optimization and Gravitational Search Algorithm. *Electric Power Components and Systems, 00(00):1-13,*, 2015.
13. A. Shaheen, R. El-Sehiemy, and S. Farrag. Solving multi-objective optimal power flow problem via forced initialised differential evolution algorithm. *IET Generation, Transmission & Distribution, vol.10, Iss..7, pp: 1634-1647*, 2016.
14. W. Nakawiro and I. Erlich. A Combined GA-ANN Strategy for Solving Optimal Power FLow with Voltage Security Constraint. *Proc: in Asia-Pacific Power and Energy Engineering Conference*, 2009.
15. D. Phan and J. Kalagmanam. Some Efficient Optimization Methods for Solving the Security-Contrained Optimal Power Flow Problem. *IEEE Transaction on Power Systems, Vol 29, no. 2, 863-872*, 2014.
16. R. Zhang, Z. Y. Dong, Y. Xu, K. P. Wong, and M. Lai. Hybrid computation of corrective security-constrained optimal power flow problems. *IET Gener. Transm. Distrib., Vol. 8, Iss. 6, pp. 995-1006*, 2014.
17. L. Carvalho, F. Loureiro, J. Sumali, H. Keko, V. Miranda, C. Marcelino, and E. Wanner. Statistical Tuning of DEEPSO Soft Constraints in the Security Constrained Optimal Power Flow Problem. *Proc: in 18th International Conference on Intelligent System Application to Power Systems, vol.1. pp: 1-7*, 2015.
18. C. Marcelino, P. Almeida, E. Wanner, L. Carvalho, and V. Miranda. Fundamentals of the C-DEEPSO Algorithm and its Aplication to the Reactive Power Optimization of Wind Farms. *Proc: in IEEE Congress on Evolutionary Computation, vol. 1, pp.:*, 2016.
19. H. Pham, J. Rueda, and I. Erlich. Online Optimal control or Reactive sources in Wind Power Plants. *IEEE Transactions on Sustainble Energy, vol. 5, no.2, pp.:608-616*, 2014.
20. K. Teeparthi and D. M. Kumar. Multi-objective hybrid PSO-APO algorithm based security constrained optimal power flow with wind and thermal generators. *Engineering Science and Technology, an International Journal, acepted March, pp.: 2-16*, 2017.
21. C. P. Raju, K. Vaisakh, and S. S. Raju. An IPM-EPSO Based Hybrid Method for Security Enhancement Unsing SSSC. *International Journal of Recent Trends in Engineering, vol. 2, no. 5, pp.: 208-2012*, 2009.
22. M. Kumari and S. Maheswarapu. Enhaced genetic algotithm based computation technique for multi-objective optimal power flow solution. *International Journal of Electrical Power & Energy Systems, vol. 32, no. 6, pp.: 736-742*, 2010.

23. C. Y. Chung, C. H. Liang, K. P. Wong, and X. Z. Duan. Hybrid algorithmm of Differentialial evolution and evolutionary programming for optimal reactive power flow. *IET Generation, Transmission & Distribution, vol.4, no.1, pp: 84-93*, 2010.
24. J. RadosavljeviÃ© and N. ArsiÃ©. Optimal Power Flow Using Hybrid PSOGSA Algorithm. *Proc: in 55th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, 2014.
25. Y. Zeng and Y. Sun. Comparasion of multiobjecive particle swarm optimization and evolutionary algorithms for optimal reactive power dispatch problem. *Proc: in IEEE Congress on Evolutionary Computation, vol.1, pp.: 258-265*, 2012.
26. L. Shi, C. Wnag, L. Yao, N. Yixin, and M. Bazargan. Optimal power flow solution incorporating wind power. *IEEE Systems Journal, vol.6, no. 2, pp: 233-241*, 2012.
27. L Srivastava and H. Singh. Hybrid multi-swarm particle swarm opitimisation based multiobjective reactive power dispatch. *IET Generation, Transmission & Distribution, vol.9, no.8, pp: 727-739*, 2015.
28. W. Bai, I. Eke, and K. Lee. Heuristic optimization for wind energy integrated optimal power flow. *Proc: in IEEE Power & Energy Society General Meeting, vol.1, pp: 1-5*, 2015.
29. L. Carvalho, F. Loureiro, J. Sumali, H. Keko, C. Marcelino, and E. F. Wanner. Statistical Tuning of DEEPSO Soft Constraints in the Security Constrained Optimal Power Flow Problem. *Proc in: 18h Conference on Intelligent Systems Apllications to Power Systems*, 2015.
30. Y. Xiang, Y. Peng, Y. Zhong, Z. Chen, X. Lu, and X. Zhong. A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization. *Comput Optim Appl 57:493-516*, 2014.
31. Zellagui. M and Abdelaziz A. Optimal Coordination of Directional Overcurrent Relays using Hybrid PSO-DE Algorithm. *International Electrical Engineering Journal (IEEJ) Vol. 6 No. 4, pp. 1841-1849*, 2015.
32. K. Vaisakh, P. Praveena, R. Rao, and K. Meah. Solving dynamic economic dispatch problem with security constraints using bacterial foraging PSO-DE algorithm. *Electrical Power and Energy Systems 39, 56-67*, 2012.
33. B. Xin, J. Chen, J. Zhang, H. Fang, and Z-H. Peng. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy. *IEEE Transactions on Systems Man and Cybernetics: applications and reviews, vol. 42, n.5 , September*, 2012.
34. V. Miranda and R. Alves. Differential evolutionary particle swarm optimization (DEEPSO): a successful hybrid. *In Proc: 11th Brazilian Congress on Computational Intelligence (BRICS-CCI). pp 368-374*, 2013.
35. K. Price, R. Storn, and J. Lampinen. Differential Evolution- A Pratical Aprooach to Global Optimization. *Berlin, Germany: Springer*, 2005.
36. J. Zhang and A. Sanderson. JADE: Adaptative Differential Evolution with optional external archive. *IEEE Transactions on Evolutionary Computation, vol. 13, no.5, pp. 945-958*, 2009.
37. K. Tamura and K. Yasuda. Primary study of spiral ddynamic inspired optimization. *IEEJ Transactions on Electrical end Eletronic Engineering, vol.6, no. S1, pp.: S98-S100*, 2011.
38. L. Benasla, A. Belmadani, and M. Rahli. Spiral Optimization Algorihm for solving Combined Economic and Emission Dispatch. *Electrical Power and Energy Systems, vol. 62, pp.: 163-174*, 2014.
39. K. Tamura and K. Yasuda. Spiral dynamics inspired optimmization. *Journal Advanced Computational Intteligence and Intelligence Informatics, vol. 15, no 8, pp.: 1116-1122*, 2011.
40. N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: Model, taxonomy and design issues. *IEEE Transactions on Evolutionary Computation, V.9, N.5, pp: 474-488*, 2005.
41. M. Pant, R. Thangaraj, C. Grosan, and A. Abraham. Hybrid Differential Evolution - Particle Swarm Optimization algorithm for Solving Global Optimization Problems. *Proc: In 3Th International Conference on Digital Information Management. pp:18-24*, 2011.
42. D. Montgomery. *Design and analysis of Experiments.* 8th edition, 2012.
43. M. Niu, Y. Jia, Z. Xu, and Wong. K. P. Differential evolution algorithm with a modified archiving-based adaptive tradeoff model for optimal power flow. Technical report, Techreport, disponível em: https://www.uni-due.de/imperia/md/content/ieee-wgmho/icde.pdf, https://www.uni-due.de/imperia/md/content/ieee-wgmho/panel.pdf, 2014.

44. I. Erlich, K. Lee, J. Rueda, and S. Wildenhues. Competition on application of modern heuristic optimization algorithms for solving optimal power flow problems. In *Technical report, Working Group on Modern Heuristic Optimization, Intelligent Systems Subcommittee Power System Analysis, Computing, and Economic Committee, (February)*, 2014.