IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 21, 2019, accepted October 13, 2019, date of publication October 16, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947626

Oppositional Jaya Algorithm With Distance-Adaptive Coefficient in Solving Directional Over Current Relays Coordination Problem

JIANGTAO YU[1,2], CHANG-HWAN KIM[2], AND SANG-BONG RHEE[2]
[1]Department of Electronic Information and Electrical Engineering, Anyang Institute of Technology, Anyang 455000, China
[2]Department of Electrical Engineering, Yeungnam University, Kyonsan 38541, South Korea

Corresponding author: Sang-Bong Rhee (rrsd@yu.ac.kr)

The work of J. Yu was supported by the Yeungnam University.

ABSTRACT The model of directional over current relays (DOCRs) coordination is considered as an optimization problem. It is generally formulated as linear programming (LP), non-linear programming (NLP) and mixed integer non-linear programming (MINLP), according to the nature of the design variables. For each kind of formulation, the main goal is to minimize the summation of operating times of primary relays, by setting optimal values for decision variables as time dial setting (TDS) and pickup current setting (IP) or plug setting (PS). In this paper, we proposed an oppositional Jaya (OJaya) algorithm with distance-adaptive coefficient (DAC), to effectively solve the DOCRs coordination problem. Firstly, by oppositional learning (OL), the searching space of Jaya is expanded and the diversity of its population is strengthened; secondly, by DAC, the population's trends of running towards the best position and escaping from the worst position is accelerated. The performance of OJaya is evaluated by 3-bus, 8-bus, 9-bus and 15-bus testing systems, in aspects of convergence rate, objective function value, robustness and computation efficiency. The results indicate the effectiveness and superiority of OJaya in solving DOCRs coordination problems compared with standard Jaya.

INDEX TERMS Jaya, oppositional learning, distance-adaptive coefficient, over current relays coordination.

I. INTRODUCTION

Relays coordination problem is of great importance for the operation of power systems. The aim of relays coordination is to efficiently protect the power systems by quickly isolating the faulted sections to preserve services throughout the remaining sections. Over the last 40 years, great progress has been achieved in the development of relays for the protection of power systems. Directional over current relays (DOCRs) have been applied to the design of economical alternatives for the primary and backup protection of power systems. The operating times of DOCRs are depended on two parameters as time dial setting (TDS) and pickup current setting (IP) or plug setting (PS). Optimal coordination between the DOCRs is able to maintain the reliability of the overall protection system.

The mathematical model of DOCRs coordination problem is generally formulated in three ways. Firstly, as a linear programming (LP) problem. In LP, the value of IP or PS is assumed to be fixed, hence the operating time of each relay (T_i) is calculated as a linear function of TDS. Even though LP is a simple formulation, it requires experts for setting the initial values of IP or PS, and it is easily get stuck in local minima [1]. Secondly, as a non-linear programming (NLP) problem. In NLP, both the TDS and IP are considered as variables and calculated to minimize the relay operating time (T_i), where IP takes continuous values. By NLP, the total operational time of the primary relays can be reduced and the coordination can be maintained well. Thirdly, as a mixed integer non-linear programming (MINLP) problem. In MINLP, both the parameters of TDS and PS are calculated and optimized. The difference between NLP and MINLP is that, the parameter of PS takes discrete values in MINLP, while IP takes continuous values in NLP.

Modern optimization algorithms were used to solve the DOCRs coordination problems. Genetic algorithm (GA), Hybrid GA and Hybrid GA-NLP were used in [2]–[4]. Two modified particle swarm optimization (PSO) algorithms were

The associate editor coordinating the review of this manuscript and approving it for publication was Shaoyong Zheng.

used in [5], [6], where the repair algorithm and non-random technique for initialization were introduced to the standard version. Teaching learning-based optimization (TLBO) and modified adaptive TLBO (MATLBO) were used in [7], [8]. Chaotic firefly algorithm (CFA), modified swarm firefly algorithm (MSFA) and improved firefly algorithm (IFA) were used in [9]–[11]. The new developed whale optimization algorithm (WOA) and hybridized whale optimization algorithm (HWOA) were used in [12], [13]. In [14], two different two-phase solution approaches (IPM-BBM and IPM-IPM) are proposed to solve the coordination problem. Lately, an adaptive coordination scheme of numerical DOCRs is proposed in [15] by utilizing a mathematical programming language (AMPL) based interior point optimization (IPOPT) solver. Furthermore, in [16], optimum settings of DOCRs considering different characteristic curves for AC microgrids is presented. Recently, there are new published articles on DOCRs coordination problems, such as ant lion optimizer (ALO), invasive weed optimization (IWO) and water cycle algorithm (WCA) [17]–[19], all of them achieve good results, but they are faced with disadvantages of adjusting the algorithm-parameters.

Jaya algorithm is a newly developed yet advanced heuristic algorithm proposed by Rao in [20]. It is totally free from algorithm-specific parameters and only two common parameters are required, which are maximum number of iteration (*Max_iter*) and population size (*N_pop*). This significant benefit makes it popularly applied in various real-world optimization problems, such as photovoltaic cell and module [21], economic load dispatch problems [22], Li-ion battery model [23], isolated microgrid with electric vehicle battery swapping stations [24], parameter estimation of proton exchange membrane fuel cells [25] and flexible job-shop rescheduling problem (FJRP) [26].

In this paper, an oppositional Jaya (OJaya) algorithm with distance-adaptive coefficient (DAC) is proposed to solve the optimal coordination problem of DOCRs. Compared with standard Jaya, there are two improvements in OJaya. Firstly, by oppositional learning (OL), the searching space is expanded and the diversity of its population is strengthened. Secondly, with the help of DAC, which is determined by the best position and the worst position in Jaya, the population's trends of running towards the best position and escaping from the worst position is accelerated. The main contributions of this paper can be summarized as follows:

- Jaya algorithm has been used to solve the DOCRs coordination problem;
- OJaya algorithm has been proposed to expand the population diversity and to accelerate the convergence rate of Jaya, without adding any more parameters.
- The performance of OJaya has been assessed by standard test systems of DOCRs with 3-bus, 8-bus, 9-bus and 15-bus;
- The results verified that, with the introduction of OL and DAC, OJaya outperforms Jaya in all testing systems.

Rest of this paper is arranged as follows. In Section 2, the formulation of DOCRs coordination problem is constructed. Related works on Jaya, OJaya and the procedures of solving DOCRs coordination problem are described in Section 3. Experimental results and comparisons are presented in Section 4. Finally, conclusions are given in Section 5.

II. PROBLEM FORMULATION

A. OBJECTIVE FUNCTION

The coordination problem of DOCRs in a ring fed distribution system can be formulated as an optimization problem, where the objective function is the sum of the operating times of the primary relays in a system, as expressed below:

$$OF = \sum_{i=1}^{N} W_i T_i \tag{1}$$

where N is the number of the primary relays, W_i is the weight assigned for relay R_i which is equal to 1 for all the relays, T_i is the operating time of relay R_i calculated by the following formulations:

$$T_i = TDS_i \times \frac{\alpha}{(IF_i/IP_i)^{\beta} - \gamma} + L \tag{2}$$

$$IP_i = PS_i \times CT_i \tag{3}$$

where α, β, γ and L are constant parameters which, according to the IEC curves, are assumed to be 0.14, 0.02, 1.0 and 0. TDS_i is the time dial settings of relay R_i. IF_i is the fault current, IP_i is the pickup current flowing through relay R_i for a particular fault located in a particular zone. PS_i stands for the plug setting, CT_i stands for the CT ratio, so the pickup current IP_i is calculated by Eq.(3).

B. CONSTRAINED FUNCTIONS

1) RELAY COORDINATION CONSTRAINTS

In a power system, when fault happens, it is sensed by primary and backup relays simultaneously. To avoid mal-operation, backup relay should takeover the tripping action, only after primary relay fails to operate. The operating time of backup relay (T^{backup}) is decided by the operating time of primary relay ($T^{primary}$), plus the coordination time interval (CTI). This is necessary for maintaining the selectivity of primary and backup relays. This relay coordination constraint can be stated as:

$$T^{backup} - T^{primary} \geq CTI \tag{4}$$

The value of CTI varies from 0.30s to 0.40s for electromechanical relays while it varies from 0.10s to 0.20s for numerical relays.

2) RELAY CHARACTERISTIC CONSTRAINTS

The relay characteristic constraints are the physical and operational bounds of the relay parameters as follows:

$$T_i^{min} \leq T_i \leq T_i^{max} \tag{5}$$

$$TDS_i^{min} \leq TDS_i \leq TDS_i^{max} \tag{6}$$
$$IP_i^{min} \leq IP_i \leq IP_i^{max} \tag{7}$$
$$PS_i^{min} \leq PS_i \leq PS_i^{max} \tag{8}$$

where T_i^{min} and T_i^{max} in Eq.(5) are the minimum and maximum operating time of relay R_i for the fault at any point; TDS_i^{min} and TDS_i^{max} in Eq.(6) are the minimum and maximum values of TDS_i of relay R_i; IP_i^{min} and IP_i^{max} in Eq.(7) are the minimum and maximum values of IP_i for relay R_i; PS_i^{min} and PS_i^{max} in Eq.(8) are the minimum and maximum values of PS_i for relay R_i.

C. CONSTRAINTS HANDLING

In this paper, penalty method is used to handle the constrained functions. It consists of adding a penalty term to the objective function to penalize the unfeasible solutions that violate the constraints. A comprehensive survey of the most popular penalty functions is given in [27].

In DOCRs coordination problem, the relay coordination constraints and the relay characteristic constraints, are included in the objective function using penalty method, as shown in Eq.(9). If any constraint is violated, a value of penalty is added to the value of objective function. Since the objective function is of minimization type, a large number is taken as the penalty factor.

$$OF = \sum_{i=1}^{N} T_i^{primary} + \sum_{k=1}^{M} Penalty(k) \tag{9}$$

where N is the number of primary relays and M is the number of relay pairs, the penalty term $Penalty(k)$ is given by the following equation:

$$Penalty(k) = \begin{cases} 0, & if\ (T_k^{backup} - T_k^{primary}) \geq CTI \\ \xi|CTI - (T_k^{backup} - T_k^{primary})|, & otherwise \end{cases} \tag{10}$$

where ξ is the penalty factor for penalty method to make the value of the objective function more significant during minimisation. ξ is usually given a relatively high value, with the aim to achieve zero penalties in optimal solutions [28].

III. OJAYA ALGORITHM

A. JAYA ALGORITHM

Jaya algorithm is a newly developed yet powerful heuristic algorithm for solving constrained and unconstrained optimization problems [20]. Compared with most of the other heuristic algorithms that requiring for algorithm-specific parameters, Jaya is totally free from the algorithm-specific parameters, and only two common parameters named maximum number of iteration (Max_iter) and population size (N_pop) are required, whose values can be initialised easily. Pseudo code of Jaya is shown in Algorithm 1. The working principle is explained as follows.

Suppose the objective function $OF(X)$ is required to be minimized or maximized. Let the design variable number is N_var where the index $u \in [1, N\_var]$, let the population size is N_pop where the index $v \in [1, N\_pop]$, let the maximum iteration number is Max_iter where the index $w \in [1, Max\_iter]$. Then let $X_{u,v,w}$ be the value of the u^{th} variable for the v^{th} candidate population during the w^{th} iteration, then the new modified value $X_{u,v,w}^{new}$ is calculated by:

$$X_{u,v,w}^{new} = X_{u,v,w} + r_1 \times (X_{u,best,w} - |X_{u,v,w}|) - r_2 \times (X_{u,worst,w} - |X_{u,v,w}|) \tag{11}$$

where $X_{u,v,w}^{new}$ is the updated value of $X_{u,v,w}$. r_1 and r_2 are two uniformly generated random numbers ranged in [0, 1]. $X_{u,best,w}$ is the best population with the best fitness value and $X_{u,worst,w}$ is the worst population with the worst fitness value.

It should be explained that, in Eq.(11), the first term "$X_{u,v,w}$" represents the original position, which provides the necessary start point for each population (each population can be seen as a moving particle) to roam among the fitness space. The second term "$+r_1 \times (X_{u,best,w} - |X_{u,v,w}|)$" encourages the population to fly toward the spot of the best position found so far. The third term " "$-r_2 \times (X_{u,worst,w} - |X_{u,v,w}|)$" represents the tendency of the population to run far away from the worst position found so far.

Algorithm 1 Jaya

```
Initialize N_var, N_pop and Max_iter;
Generate initial population X;
Evaluate the fitness value OF(X);
Set w = 1;
while w < Max_iter do
    Identify X_{u,best,w} and X_{u,worst,w} within current X;
    for v = 1 → N_pop do
        for u = 1 → N_var do
            Generate updated population X^{new}_{u,v,w} by
            Eq.(11);
        end
        Calculate OF(X^{new}_{u,v,w});
        if OF(X^{new}_{u,v,w}) is better than OF(X_{u,v,w}) then
            X_{u,v,w} = X^{new}_{u,v,w}
            OF(X_{u,v,w}) = OF(X^{new}_{u,v,w})
        else
            Keep the old value;
        end
    end
    w = w + 1;
end
```

B. OPPOSITIONAL LEARNING (OL)

Oppositional learning (OL) is usually utilized by population-based algorithm by calculating and evaluating the current population and its opposite population simultaneously, and choose the better one for going to next generation. By OL, the searching space is expanded and the diversity of the population is strengthened. It has successfully obtained better results in biogeography-based optimization (BBO) [29], whale optimization algorithm (VOA) [30] and krill herd algorithm (KH) [31].

Here goes the working principle. Suppose $X = (X_1, X_2, \cdots, X_u)$ and $X_m \in [A_m, B_m]$, where $m = 1, 2, \ldots, u$. Then the oppositional X is represented as $X^o = (X_1^o, X_2^o, \cdots, X_u^o)$, which is calculated by:

$$X_m^o = A_m + B_m - X_m \tag{12}$$

In this paper, Eq.(12) is applied to the current population $\{X\}$ to generate the oppositional population $\{X^o\}$. To illustrate it in details, we suppose the current population $X_{u,v,w} = (X_{1,v,w}, X_{2,v,w}, \cdots, X_{N\_var,v,w})$, the corresponding opposite solution can be defined as $X_{u,v,w}^o = (X_{1,v,w}^o, X_{2,v,w}^o, \cdots, X_{N\_var,v,w}^o)$, which is obtained by the following equation:

$$X_{u,v,w}^o = s \times (A_{u,w} + B_{u,w}) - X_{u,v,w} \tag{13}$$

where s is a random number in $[0, 1]$. $A_{u,w}$ and $B_{u,w}$ are the dynamic bounds of u^{th} variable in the w^{th} iteration for all the population, which can be obtained by the following equations:

$$A_{u,w} = \min(X_{u,v,w}), \quad B_{u,w} = \max(X_{u,v,w}) \tag{14}$$

As we know the searching space is shrinking with iteration, this may cause the population stuck in local minimum. Thus, we will update the dynamic bounds $A_{u,w}$ and $B_{u,w}$ every 50 generations. Even though the dynamic bounds are good at restoring searching experiences, they can make $X_{u,v,w}^o$ jump out of $[X_u^{min}, X_u^{max}]$, where $[X_u^{min}, X_u^{max}]$ are the minimum and maximum limits in constrained functions of the u^{th} relay. If that happens, equation below should be used to reset $X_{u,v,w}^o$:

$$X_{u,v,w}^o = rand(A_{u,w}, B_{u,w}) \tag{15}$$

where $rand(A_{u,w}, B_{u,w})$ is a random number within in $[A_{u,w}, B_{u,w}]$.

In this work, OL is combined with Jaya in two aspects. The first one, when we are generating the initial population, we apply OL simultaneously to get its oppositional population. Then by comparing the current population with its oppositional population, we keep the better one as the initial population. The second one, OL is applied to the current population during the whole iteration process, with the aim of jumping to a new position which may have greater opportunity to get closer to the optimal solution. By comparing its fitness value, the fittest N_pop solutions are saved to the next iteration and the others are removed. Pseudo code of OL learning is shown in Algorithm 2.

Algorithm 2 OL_Learning (X)

```
Calculate the fitness value of current population OF(X) ;
s=rand(0,1) ;
for v = 1 → N_pop do
    for u = 1 → N_var do
        Calculate the opposite population X^o by Eq.(13)
        ;
    end
    Calculate OF(X^o) ;
end
Order{OF(X), OF(X^o)} ;
Select N_pop fittest population from {X, X^o} as new
population X ;
```

C. DISTANCE-ADAPTIVE COEFFICIENT (DAC)

It can be observed from Eq.(11) that, searching process towards better positions by Jaya is mainly guided by two stochastic terms, one is the best position $X_{u,best,w}$ and the other one is the worst position $X_{u,worst,w}$. Therefore, reasonable control of these two terms is of crucial importance in searching for optimum solution efficiently and accurately.

Generally speaking, at the early stage of searching process, the populations are expected to approach the promising regions as fast as possible; at the latter stage, since the populations have converged to the promising regions, fine-tuning should be implemented around the neighborhood to find the global optima. In order to meet this requirements, distance-adaptive coefficient (DAC) (d_w) is introduced. The mathematical representation of d_w is given by:

$$d_w = \begin{cases} \left(\dfrac{OF(X_{u,best,w})}{OF(X_{u,worst,w})}\right)^2, & if\ OF(X_{u,worst,w}) \neq 0 \\ 1, & otherwise \end{cases} \tag{16}$$

where $OF(X_{u,best,w})$ and $OF(X_{u,worst,w})$ are the fitness values of the best solution and worst solution in Eq.(11). Then we introduce Eq.(16) to Eq.(11):

$$\begin{aligned} X_{u,v,w}^{new} = X_{u,v,w} &+ r_1 \times (X_{u,best,w} - |X_{u,v,w}|) \\ &- d_w \times r_2 \times (X_{u,worst,w} - |X_{u,v,w}|) \end{aligned} \tag{17}$$

We can tell that, d_w has self-adaptive feature and its value increases gradually, since the distance between $X_{u,best,w}$ and $X_{u,worst,w}$ is becoming closer as the search process. Therefore, when d_w is small at the early stage, a relatively small term of $X_{u,worst,w}$, compared with $X_{u,best,w}$, will result in significantly accelerated speed in approaching $X_{u,best,w}$. In contrast, when d_w is gradually increasing to 1, it will fairly make the balance between $X_{u,worst,w}$ and $X_{u,best,w}$, so the population would make use of both of the two sides to refine the $X_{u,v,w}$ at the latter stage. In addition, since the value of d_w is calculated adaptively, thus no additional parameter is introduced [32].

D. OJAYA ALGORITHM

According to the previous work, an oppositional Jaya (OJaya) algorithm with distance-adaptive coefficient (DAC) is proposed. Pseudo code of OJaya is shown in Algorithm 3. It starts by setting values for N_var, N_pop and Max_iter. Then the initial population is created by OL_Learning (X) according to Algorithm 2. Then we use DAC to modify the Jaya function. After that, the modified function is applied to update the current population. Then OL_Learning (X) is re-utilised to select the better value. Finally, if Max_iter is reached, stop the iteration and record the best solution. Otherwise, re-calculate d_w and go to the next iteration.

Algorithm 3 OJaya

```
Initialize N_var, N_pop and Max_iter;
Generate initial population X;
X = OL_Learning (X);
Evaluate the fitness value OF(X);
Set w = 1;
while w < Max_iter do
    Identify X_{u,best,w} and X_{u,worst,w} within current X;
    Calculate d_w by Eq.(16);
    for v = 1 → N_pop do
        for u = 1 → N_var do
            Generate updated population X^{new}_{u,v,w} by
            Eq.(17);
        end
        X^{new}_{u,v,w} = OL_Learning (X^{new}_{u,v,w});
        Calculate OF(X^{new}_{u,v,w});
        if OF(X^{new}_{u,v,w}) is better than OF(X_{u,v,w}) then
            X_{u,v,w} = X^{new}_{u,v,w}
            OF(X_{u,v,w}) = OF(X^{new}_{u,v,w})
        else
            Keep the old value;
        end
    end
    w = w + 1;
end
```

The main procedures of using OJaya algorithm to solve the DOCRs coordination problem are illustrated with further details below, and the flowcharts are shown in Fig.1.

1. Set parameters. Common parameters of N_var, N_pop and Max_iter are given.
2. Initialization. Initial population X is generated in the form of:

$$X = \begin{bmatrix} X_{1,1,w} & X_{2,1,w} & \ldots & X_{N\_var,1,w} \\ X_{1,2,w} & X_{2,2,w} & \ldots & X_{N\_var,2,w} \\ \ldots & \ldots & X_{u,v,w} & X_{N\_var,v,w} \\ X_{1,N\_pop,w} & X_{2,N\_pop,w} & \ldots & X_{N\_var,N\_pop,w} \end{bmatrix}$$

where

$$X_{u,v,w} = X_u^{min} + (X_u^{max} - X_u^{min}) \times rand(N\_pop, N\_var)$$

where $X_{u,v,w}$ is the u^{th} variable in the v^{th} candidate solution where $u \in [1, N\_var]$ and $v \in [1, N\_pop]$. w is the iteration index number, which actually can be ignored in the initialization step. X_u^{min} and X_u^{max} are the lower and upper limits of the u^{th} variable given by relay characteristic constraints, as shown in Eq.(6), Eq.(7) or Eq.(8).
3. Apply OL_Learning. The Initial population X is updated according to Algorithm 2.
4. Evaluation. Fitness value $OF(X_{u,v,w})$ is calculated by the objective function given in Eq.(1).
5. Identify $X_{u,best,w}$ and $X_{u,worst,w}$ within X according to the best and worst OF value.

TABLE 1. Primary/Backup relay pairs and related parameters for 3-bus system [33].

Primary Relay			IF(A)	Backup	IF(A)
Relay No	CT	PS	(Primary)	Relay	(Backup)
1	300/5	5.0	1978.90	5	175.00
2	200/5	1.5	1525.70	4	545.00
3	200/5	5.0	1683.90	1	617.22
4	300/5	4.0	1815.40	6	466.17
5	200/5	2.0	1499.66	3	384.00
6	400/5	2.5	1766.30	2	145.34

6. Apply DAC. Calculate d_w by Eq.(16).
7. Update the population. The updated population $X^{new}_{u,v,w}$ is calculated by Eq.(17).
8. Apply OL_Learning. The current population $X^{new}_{u,v,w}$ is updated according to Algorithm 2.
9. Evaluation. The updated fitness value $OF(X^{new}_{u,v,w})$ is calculated by the objective function, which is as the same as in step 4.
10. Comparison. Compare $OF(X^{new}_{u,v,w})$ with $OF(X_{u,v,w})$ and keep the better value.
11. Check the stopping condition. If the Max_iter is reached, stop the loop and report the best solution; otherwise set $w = w + 1$ and go to step 6 to re-calculate d_w and continue the loop.

IV. NUMERICAL EXPERIMENTS

To evaluate the effectiveness of Jaya and OJaya in solving DOCRs coordination problem, test systems of 3-bus, 8-bus, 9-bus and 15-bus have been investigated in this section. All the systems are developed using MATLAB software (version R2018b) and executed on a computer under windows 7 on Intel(R) Core(TM) i5-6500 CPU 3.20GHz with 8GB RAM environment.

Moreover, since the proposed OJaya algorithm is the hybridization of Jaya, OL learning and DAC, it is quite necessary to observe the relative effectiveness of each constituent, hence three different algorithms are experimented respectively.

- Jaya: The standard Jaya algorithm.
- DJaya: Jaya with DAC.
- OJaya: Jaya with OL learning and DAC.

A. 3-BUS SYSTEM

This 3-bus system consists of 3 buses, 3 generators, 3 lines and 6 relays, as shown in Fig.2. 3ϕ fault at the midpoint of each line is considered. The CT ratio, the listed primary/backup (P/B) relay pairs and the 3ϕ fault current of each line are given in Table.1. All the relays have IDMT characteristic. This system is experimented by LP, NLP and MINLP formulations to make fair comparison with other conducted studies in the literature.

1) CASE 1: 3-BUS SYSTEM WITH LP FORMULATION

In this case, CTI is 0.2s, IF, PS and CT are fixed constants given in Table.1. The only variable is TDS, which is

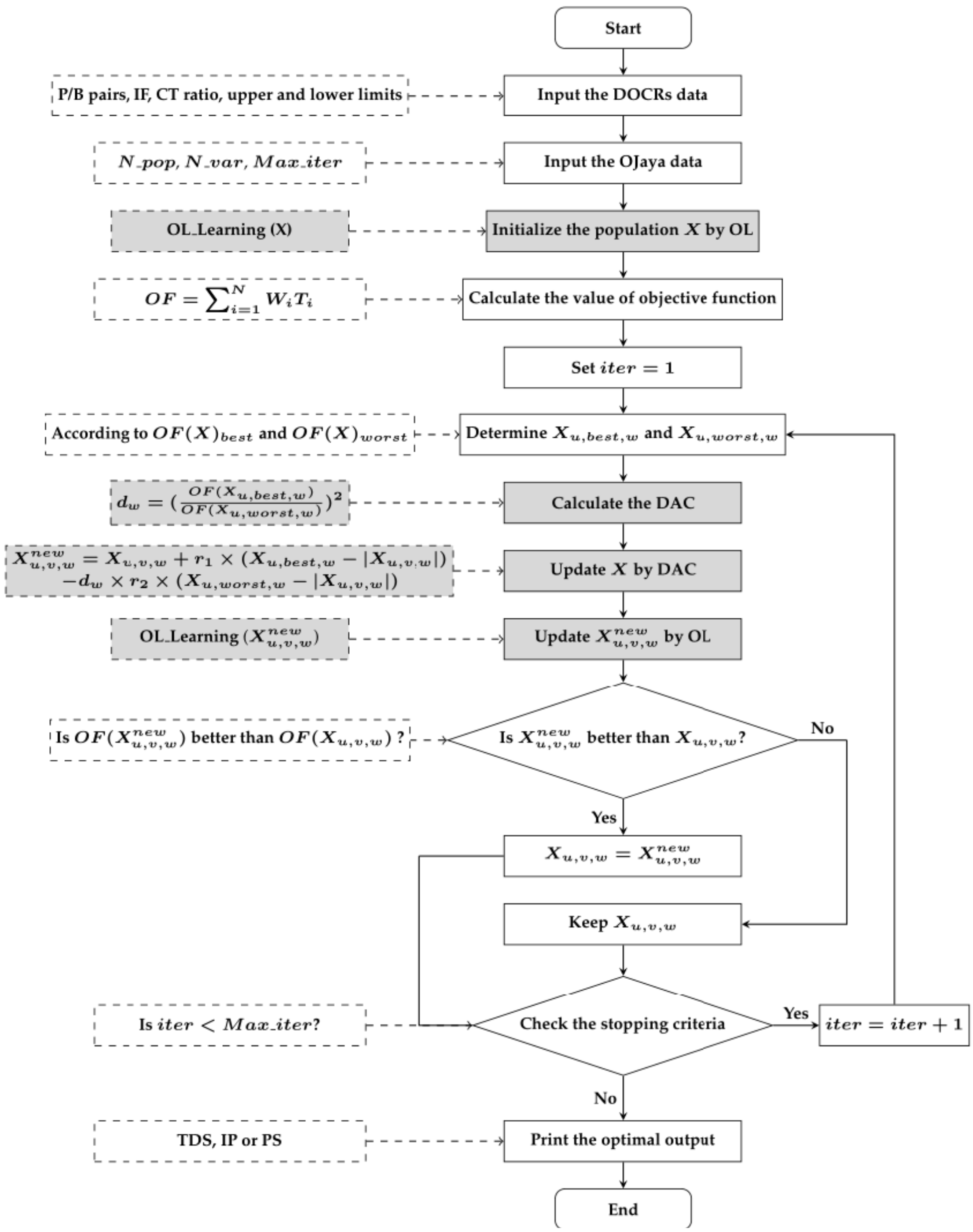

FIGURE 1. Solution process of DOCRs coordination problem by OJaya algorithm.

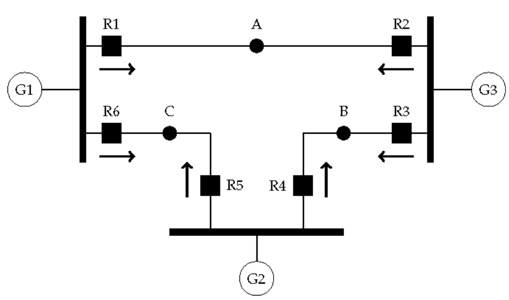

FIGURE 2. IEEE 3-bus DOCRs coordination problem model.

continuous lying in [0.1,1.1]. For Jaya and its variants, the common parameters of variable number (N_var) is 12, population size (N_pop) is 5, maximum iteration number (Max_iter) is 20. The optimum settings of TDS obtained by Jaya, DJaya and OJaya are given in Table.2. Simultaneously, simplex method [1], LP using matlab [5], PSO [5] and seeker algorithm [33] have also been presented to be compared.

Table.2 shows that, all the compared algorithms give the same objective function value as 1.9258(s), but Jaya, DJaya and OJaya are able to give more optimized value as 1.7804(s).

Fig.3 depicts the convergence curves, from which we can observe OJaya shows super fast convergence rate and reaches its best value within 4 iterations. Fig.4 provides the OF values distribution over 20 running times. We can see that, most of the runs are able to reach optimum result in this case. But there exist some "outliers" with extreme values by Jaya

TABLE 2. Time dial setting for 3-bus system by LP formulation.

Relay	Time Dial Setting (TDS)						
	Simplex Method [1]	LP using Matlab [5]	PSO [5]	Seeker [33]	Jaya	DJaya	OJaya
1	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
2	0.1364	0.1364	0.1364	0.1364	0.1000	0.1000	0.1000
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
5	0.1298	0.1298	0.1298	0.1298	0.1000	0.1000	0.1000
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
Pop	-	-	-	50	5	5	5
Iter	-	-	1000	45	20	20	20
Time	-	0.4370	0.5129	6.45	0.0218	0.0114	0.0179
OF	1.9258	1.9258	1.9258	1.9258	1.7804	1.7804	1.7804

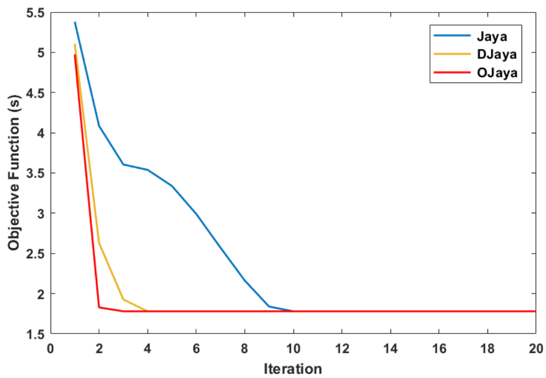

FIGURE 3. Convergence characteristics for 3-bus system by LP formulation.

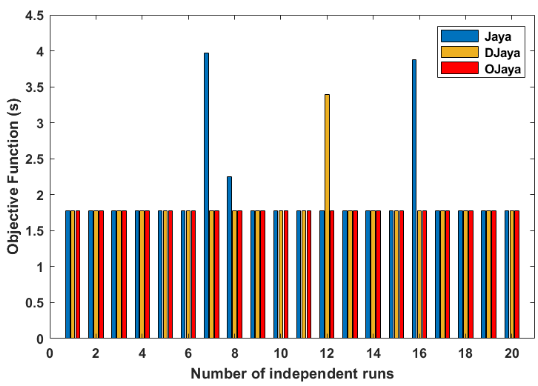

FIGURE 4. Independent runs for 3-bus system by LP formulation.

and DJaya, which illustrates that, different from OJaya, Jaya and DJaya are suffering problems of falling into local optima which is far away from the global optima.

Table.3 shows the value of coordination time interval (CTI), we can see that, the constraints are satisfied in every P/B relay pair.

2) CASE 2: 3-BUS SYSTEM WITH NLP FORMULATION

In this case, the design variables are TDS and IP, which lies in [0.1,1.1] and [1.5,5.0] respectively, and both of them are continuous values. System data is obtained from Table.1. The common parameters of N_var is 12, N_pop is 20, Max_iter is 50. The optimum settings of TDS and IP are presented

TABLE 3. Coordination time interval for 3-bus system by LP formulation.

Primary Relay	Backup Relay	CTI		
		Jaya	DJaya	OJaya
1	5	0.5232	0.5232	0.5232
2	4	0.6371	0.6371	0.6371
3	1	0.6417	0.6417	0.6417
4	6	0.4812	0.4812	0.4812
5	3	0.8342	0.8342	0.8342
6	2	0.4698	0.4698	0.4698

TABLE 4. Time dial setting and pickup current for 3-bus system by NLP formulation.

Relay	GSO [34]	IGSO [34]	Analytic [35]	Jaya	DJaya	OJaya
TDS_1	0.100	0.100	0.100	0.100	0.100	0.100
TDS_2	0.100	0.100	0.100	0.100	0.100	0.100
TDS_3	0.100	0.100	0.100	0.1453	0.100	0.100
TDS_4	0.100	0.148	0.100	0.100	0.112	0.100
TDS_5	0.100	0.100	0.100	0.100	0.100	0.100
TDS_6	0.100	0.100	0.100	0.100	0.100	0.100
IP_1	161.6507	117.7491	162.00	90.00	90.00	90.00
IP_2	97.6290	29.995	85.00	119.12	119.12	119.12
IP_3	88.8888	62.9982	115.00	60.00	122.61	124.24
IP_4	133.0546	35.7398	140.00	107.046	90.00	106.75
IP_5	53.7883	36.7397	91.00	74.404	74.40	77.74
IP_6	134.8014	101.5602	140.00	120.00	120.00	120.00
Pop	-	-	-	20	20	20
Iter	-	-	-	50	50	50
Time	-	-	-	0.0225	0.0127	0.0204
OF	1.4807	1.2918	1.5108	1.5019	1.4794	1.4718

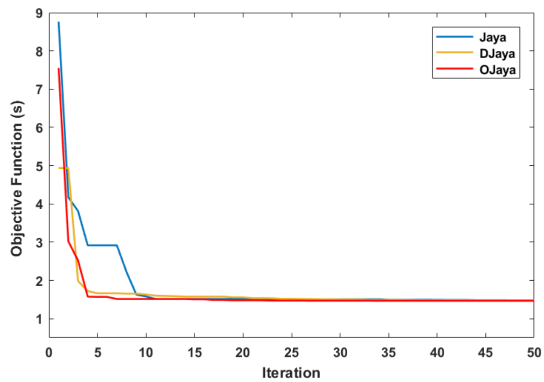

FIGURE 5. Convergence characteristics for 3-bus system by NLP formulation.

in Table.4. Simultaneously, GSO [34], IGSO [34] and Analytic [35] algorithms have been provided to be compared.

From Table.4, we can observe that IGSO [34] achieves the best OF value as 1.2918(s), the proposed OJaya ranks the second place as 1.4718(s). But it needs to mention that, even though IGSO provides better OF value than OJaya, it is not strictly-satisfying all the constraints of CTI, because there are some CTIs a little bit less than 0.2(s), which is underlined in Table.5. However, all the CTI constraints are fully-satisfied by Jaya, DJaya and OJaya.

Fig.5 shows that, both DJaya and OJaya converge faster than Jaya. Fig.6 shows the outlines of OF value by 20 running times. As in Case 1, there are extreme "outliers" by Jaya and DJaya, but none by OJaya. It illustrates the robustness of Jaya and DJaya is not as good as OJaya.

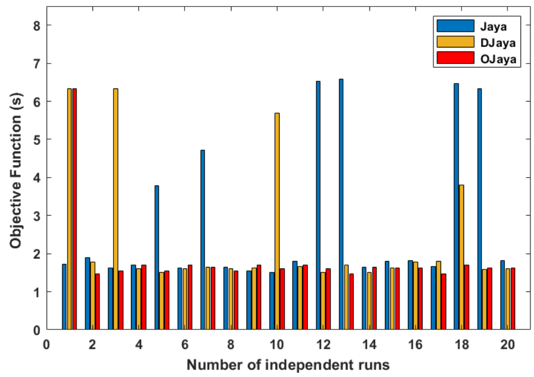

FIGURE 6. Independent runs for 3-bus system by NLP formulation.

TABLE 5. Coordination time interval for 3-bus system by NLP formulation.

Primary Relay	Backup Relay	CTI			
		IGSO [34]	Jaya	DJaya	OJaya
1	5	0.2003	0.2000	0.2000	0.2117
2	4	***0.1987***	0.2012	0.2001	0.2004
3	1	0.2095	0.7507	0.7854	0.7841
4	6	***0.1988***	3.4064	3.3942	3.4066
5	3	***0.1985***	0.2001	0.2000	0.2001
6	2	***0.1985***	0.2000	0.2000	0.2000

TABLE 6. Time dial setting and pickup current for 3-bus system by MINLP formulation.

Relay	SBB [33]	Seeker [33]	BBO [36]	BBO-LP [36]	Jaya	DJaya	OJaya
TDS_1	0.1510	0.1070	0.1043	0.1067	0.100	0.100	0.100
TDS_2	0.1280	0.1080	0.1128	0.1083	0.100	0.100	0.100
TDS_3	0.1300	0.1000	0.1008	0.1000	0.100	0.100	0.100
TDS_4	0.1040	0.1000	0.1080	0.1000	0.1119	0.1119	0.100
TDS_5	0.1060	0.1000	0.1008	0.1000	0.100	0.100	0.100
TDS_6	0.1000	0.1120	0.1030	0.1119	0.100	0.100	0.100
PS_1	1.5	2.5	3.0	2.5	1.5	1.5	1.5
PS_2	1.5	2.0	2.0	2.0	3.0	3.0	3.0
PS_3	2.0	3.0	3.0	3.0	5.0	3.5	3.5
PS_4	2.5	2.5	3.0	2.5	1.5	1.5	2.0
PS_5	2.5	2.5	2.5	2.5	2.0	2.0	2.0
PS_6	2.0	1.5	2.0	1.5	1.5	1.5	1.5
Pop	-	50	50	20	20	20	20
Iter	-	85	1000	20	50	50	50
Time	-	10.45	16.23	2.99	0.0286	0.0275	0.0331
OF	1.727	1.599	1.68375	1.59871	1.5477	1.5006	1.4984

3) CASE3: 3-BUS SYSTEM WITH MINLP FORMULATION

In this case, TDS is continuous in [0.1,1.1], PS is discrete in steps of 0.5 within [1.5,5.0]. System data is obtained from Table.1. Common parameters of N_var is 12, N_pop is 20, Max_iter is 50. Optimum settings of TDS and PS are shown in Table.6. The standard branch-and-bound (SBB) [33], Seeker [33], BBO [36] and BBO-LP [36] algorithms are provided to be compared.

We can observe from Table.6 that, the minimum value of OF is achieved by OJaya as 1.4984(s), followed by DJaya as 1.5006(s). The average time spent by OJaya and DJaya is 0.0331(s) and 0.0275(s), which are super short times compared with Seeker, BBO and BBO-LP.

In Fig.7, DJaya and OJaya show better convergence capability than Jaya, because Jaya needs more times of iteration to reach its optima. In Fig.8, we can observe that, OF value varies in large range by Jaya and Djaya, but it is kept relatively stable by OJaya.

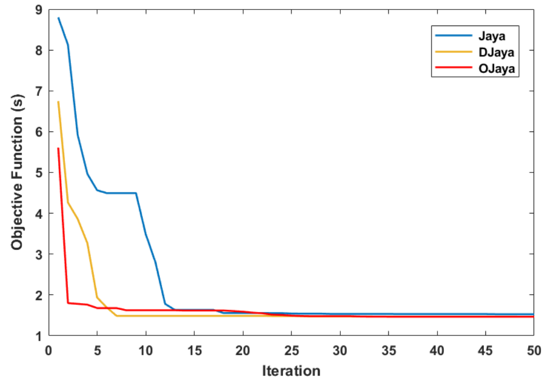

FIGURE 7. Convergence characteristics for 3-bus system by MINLP formulation.

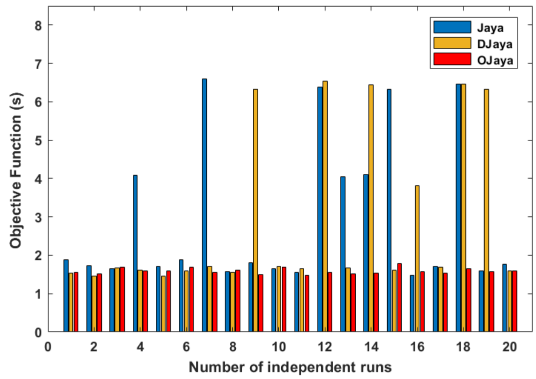

FIGURE 8. Independent runs for 3-bus system by MINLP formulation.

TABLE 7. Coordination time interval for 3-bus system by MINLP formulation.

Primary Relay	Backup Relay	CTI		
		Jaya	DJaya	OJaya
1	5	0.2197	0.2197	0.2197
2	4	0.2001	0.2001	0.2405
3	1	0.7241	0.7712	0.7712
4	6	3.3938	3.3938	3.3960
5	3	0.3823	0.2330	0.2330
6	2	0.2022	0.2022	0.2022

Table.7 illustrates that, the CTI constraints are satisfied in all P/B pairs by Jaya, DJaya and OJaya.

B. CASE 4: 8-BUS SYSTEM

This 8-bus system is considered as MINLP formulation. It is composed of 8 buses, 2 generators, 2 transformers, 7 lines and 14 relays, as shown in Fig.9. The near-end 3ϕ fault is considered. The CT ratio and 3ϕ short circuit current for each P/B pair are given in Table.8. CTI is selected to be 0.3(s).

In this case, the design variables are TDS and PS, where TDS is continuous ranged in [0.1,1.1], PS is discrete from {0.5,0.6,0.8,1.0,1.5,2.0,2.5}. Common parameters of N_var is 28, N_pop is 50, Max_iter is 2000. The optimized TDS

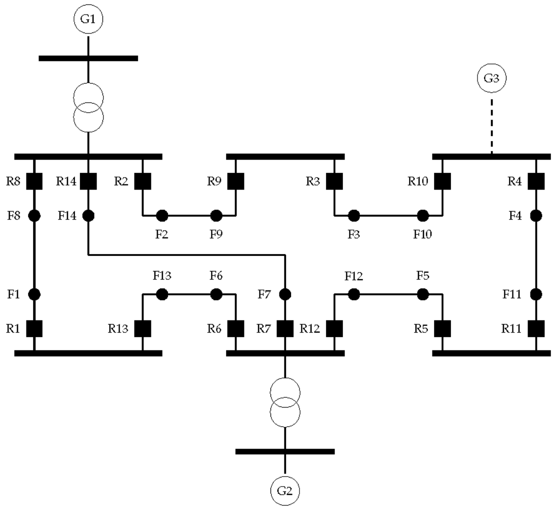

FIGURE 9. IEEE 8-bus DOCRs coordination problem model.

TABLE 8. Primary/Backup relay pairs and related parameters for 8-bus system [37].

Primary Relay		IF(A)	Backup	IF(A)
Relay No	CT	(Primary)	Relay	(Backup)
1	1200/5	3232	6	3232
2	1200/5	5924	1	996
2	1200/5	5924	7	1890
3	800/5	3556	2	3556
4	1200/5	3783	3	2244
5	1200/5	2401	4	2401
6	1200/5	6109	5	1197
6	1200/5	6109	14	1874
7	800/5	5223	5	1197
7	800/5	5223	13	987
8	1200/5	6093	7	1890
8	1200/5	6093	9	1165
9	800/5	2484	10	2484
10	1200/5	3883	11	2344
11	1200/5	3707	12	3707
12	1200/5	5899	13	987
12	1200/5	5899	14	1874
13	1200/5	2991	8	2991
14	800/5	5199	1	996
14	800/5	5199	9	1165

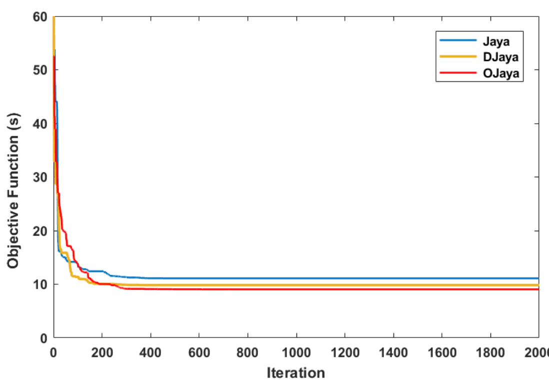

FIGURE 10. Convergence characteristics for 8-bus system by MINLP formulation.

and PS are displayed in Table.9, and the results are compared with Seeker [33], GA [3], GA-LP [3].

TABLE 9. Time dial setting and plug setting for 8-bus system by MINLP formulation.

Relay	Seeker [33]	GA [3]	GA-LP [3]	Jaya	DJaya	OJaya
TDS_1	0.113	0.29	0.3043	0.1000	0.1703	0.1000
TDS_2	0.260	0.31	0.2917	0.4409	0.2719	0.3169
TDS_3	0.225	0.26	0.2543	0.4585	0.2971	0.2506
TDS_4	0.160	0.19	0.1851	0.1900	0.1634	0.3572
TDS_5	0.100	0.18	0.1700	0.1030	0.1031	0.1024
TDS_6	0.173	0.26	0.2711	0.3447	0.2664	0.1968
TDS_7	0.243	0.54	0.5316	0.2776	0.3348	0.2998
TDS_8	0.170	0.24	0.2387	0.2638	0.2288	0.3040
TDS_9	0.147	0.17	0.1865	0.2482	0.1790	0.1936
TDS_10	0.176	0.19	0.1895	0.3507	0.2324	0.1952
TDS_11	0.187	0.21	0.2014	0.2665	0.2096	0.4453
TDS_12	0.266	0.30	0.2890	0.3163	0.4386	0.3121
TDS_13	0.114	0.23	0.2207	0.2555	0.2517	0.2529
TDS_14	0.246	0.51	0.5278	0.3205	0.4426	0.2765
PS_1	2.00	1.00	2.00	2.50	1.50	2.50
PS_2	2.50	2.50	1.50	1.00	2.50	2.00
PS_3	2.50	2.50	2.50	0.50	1.50	2.50
PS_4	2.50	2.50	2.50	2.00	2.50	0.50
PS_5	2.50	1.50	1.50	2.50	2.50	2.50
PS_6	2.50	2.50	2.50	0.50	1.50	2.00
PS_7	2.50	0.50	0.50	2.50	1.50	2.00
PS_8	2.50	2.50	2.50	2.00	2.50	1.50
PS_9	2.50	2.00	2.00	1.50	2.50	2.00
PS_10	2.50	2.50	2.50	1.00	2.00	2.50
PS_11	2.50	2.50	2.50	2.00	2.50	0.50
PS_12	2.50	2.50	2.50	2.50	1.00	2.50
PS_13	2.00	1.50	1.50	1.00	1.00	1.00
PS_14	2.50	0.50	0.50	2.00	1.00	2.50
Pop	100	100	100	50	50	50
Iter	-	100000	30	2000	2000	2000
Time	-	36000	300	2.1031	1.5546	2.1021
Std	-	-	-	3.4972	3.1145	1.7749
OF	8.4270	11.001	10.9499	10.2325	9.9661	9.8520
Feasible	Not Fullly	No	No	Fully	Fully	Fully

TABLE 10. Coordination time interval for 8-bus system by MINLP formulation.

Primary	Backup	CTI			
Relay	Relay	Seeker [33]	Jaya	DJaya	OJaya
1	6	0.3002	0.3000	0.3000	0.3000
2	1	***0.2990***	0.4421	0.3472	0.5134
2	7	0.3017	0.3000	0.3001	0.3000
3	2	***0.2995***	0.3000	0.3000	0.3002
4	3	0.3005	0.3000	0.3000	0.3000
5	4	***0.2987***	0.3000	0.3002	0.3000
6	5	0.4967	0.4469	0.3976	0.5028
6	14	0.5881	0.6567	0.5882	0.7061
7	5	0.3614	0.3000	0.3001	0.3000
7	13	0.4538	0.5098	0.4908	0.5036
8	7	0.5769	0.5236	0.4373	0.4296
8	9	0.4507	0.3739	0.4846	0.3041
9	10	0.3017	0.3000	0.3000	0.3003
10	11	0.3001	0.3000	0.3001	0.3000
11	12	***0.2982***	0.3000	0.3001	0.3001
12	13	0.3028	0.3000	0.3000	0.3002
12	14	0.3017	0.3000	0.3000	0.3000
13	8	0.3007	0.3000	0.3001	0.3000
14	1	0.4214	0.5918	0.3002	0.6387
14	9	***0.2980***	0.3000	0.3002	0.3000

Although this case has a small dimension, it is a highly constrained network with limited number of discrete PS values, so it can not get a feasible and optimal solution easily. As shown in Table.9, GA and GA-LP are not capable of achieving feasible solutions, which is also mentioned in [36]. However, Jaya, DJaya and OJaya are able to obtain feasible solutions, and the OF value keeps decreasing from 10.2325 (s) to 9.8520(s). Even though Seeker [33]

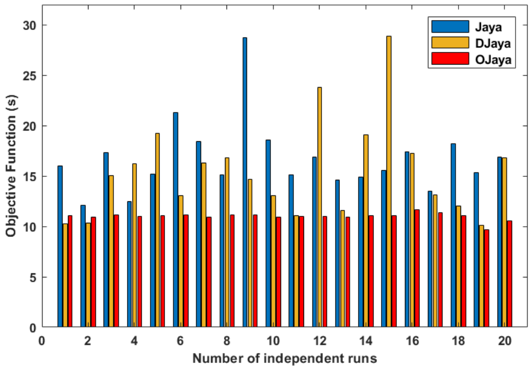

FIGURE 11. Independent runs for 8-bus system by MINLP formulation.

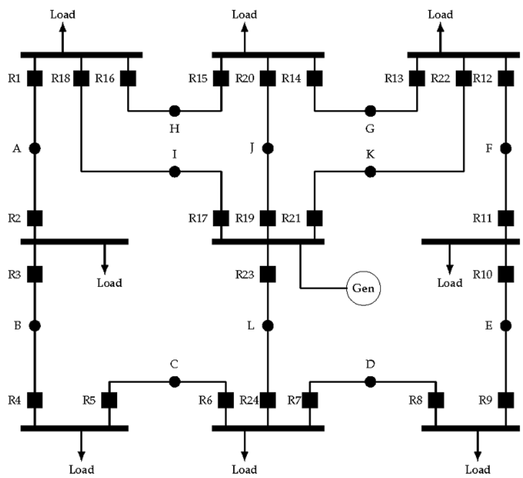

FIGURE 12. IEEE 9-bus DOCRs coordination problem model.

provides the least OF value as 8.4270 (s), it is not strictly-satisfying all the constraints of CTI, because there exist some CTIs a little bit less than 0.3(s), which is underlined in Table.10. On the contrary, all the CTIs are fully-satisfied by OJaya.

The convergence behaviours are represented in Fig.10. We can observe that, all the algorithms converge in similar trends, but OJaya reaches lower OF value than Jaya and DJaya. The amplitudes of OF values are shown in Fig.11, it can be seen that Jaya and DJaya fluctuate in quite large ranges, which means their robustness still need to be improved further. But OJaya is always able to keep the OF value minimum and stable.

C. CASE 5: 9-BUS SYSTEM

In this case, the coordination problem is modeled as NLP problem. It is with one single-end fed and equal impedances for all of the lines, as shown in Fig.12. This system has 3ϕ fault at the midpoint of each line. The P/B pairs, the fault current passing through the relays, the maximum and minimum fault current are given in Table.11. All the DOCRs have same CT ratio of 500:1, the CTI is selected to be 0.2s. It is to be noted that, no backup relay for relays {17, 19, 21, 23}, and

TABLE 11. Related parameters for 9-bus system [38].

Fault point	Primary relay	$I_{L,Max}(A)$	$I_{f,Max}$(A)	$I_{f,Min}$(A)	Backup relay
A	R1	121.74	4863.6	1361.6	R15,R17
	R2	212.74	1634.4	653.6	R4
B	R3	21.74	2811.4	1124.4	R1
	R4	21.74	2610.5	1044.2	R6
C	R5	78.26	1778.0	711.2	R3
	R6	78.26	4378.5	1226.0	R8,R23
D	R7	78.26	4378.5	1226.0	R5,R23
	R8	78.26	1778.0	711.2	R10
E	R9	21.74	2610.5	1044.2	R7
	R10	21.74	2811.4	1124.4	R12
F	R11	121.74	1634.4	653.6	R9
	R12	121.74	2811.4	787.2	R14,R21
G	R13	30.44	3684.5	1031.7	R11,R21
	R14	30.44	4172.5	1168.3	R16,R19
H	R15	30.44	4172.5	1168.3	R13,R19
	R16	30.44	3684.5	1031.7	R2,R17
I	R17	441.3	7611.2	1293.9	-
	R18	441.3	2271.7	1953.7	R2,R15
J	R19	410.87	7435.8	1264.1	-
	R20	410.87	2624.2	2256.8	R13,R16
K	R21	441.3	7611.2	1293.9	-
	R22	441.3	2271.7	1953.7	R11,R14
L	R23	506.52	7914.7	1345.5	-
	R24	506.52	1665.5	1432.3	R5,R8

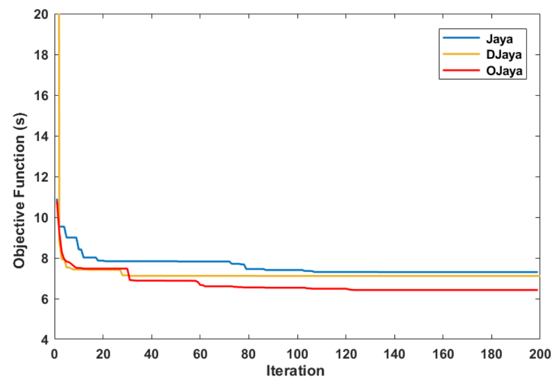

FIGURE 13. Convergence characteristics for 9-bus system by NLP formulation.

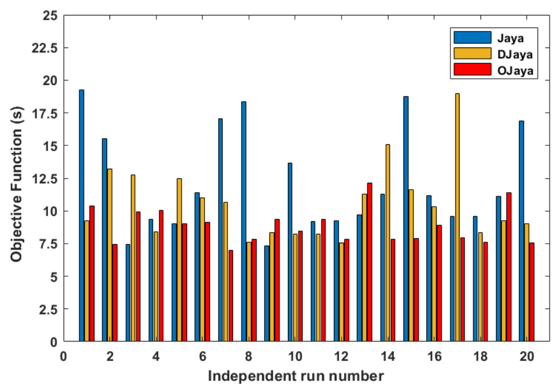

FIGURE 14. Independent runs for 9-bus system by NLP formulation.

the minimum operating time of each relay (T_i^{min}) is taken as 0.2s. For each relay, TDS is continuous ranged in [0.025, 1.2], and the minimum and maximum limits of PS are calculated

TABLE 12. Time dial setting and plug setting for 9-bus system with NLP formulation.

Relay	NLP [4]	GA-NLP [4]	DE [38]	SOA [38]	Jaya	DJaya	OJaya
TDS_1	0.0010	0.0805	0.1241	0.2662	0.0635	0.0642	0.0774
TDS_2	0.0010	0.0266	0.1000	0.2076	0.0521	0.0552	0.0398
TDS_3	0.0362	0.0560	0.1370	0.2928	0.0863	0.0614	0.0434
TDS_4	0.0131	0.0492	0.1089	0.3192	0.0958	0.0710	0.0481
TDS_5	0.0643	0.0472	0.1237	0.2879	0.0559	0.0641	0.0575
TDS_6	0.0203	0.0764	0.1277	0.3677	0.0720	0.0984	0.0739
TDS_7	0.0203	0.0764	0.1277	0.3006	0.0541	0.1026	0.0842
TDS_8	0.0251	0.0472	0.1237	0.2905	0.0800	0.0648	0.0665
TDS_9	0.0131	0.0492	0.1089	0.2476	0.0765	0.0797	0.0451
TDS_10	0.0391	0.0557	0.1370	0.2480	0.0528	0.0748	0.0648
TDS_11	0.0010	0.0305	0.1000	0.2578	0.0809	0.0780	0.0657
TDS_12	0.0010	0.0802	0.1241	0.3665	0.0789	0.0551	0.0556
TDS_13	0.0010	0.0492	0.1000	0.2581	0.1154	0.0672	0.0546
TDS_14	0.0062	0.0637	0.1090	0.3117	0.0662	0.0746	0.0724
TDS_15	0.0062	0.0639	0.1090	0.2921	0.1170	0.0974	0.1152
TDS_16	0.0010	0.0593	0.1000	0.3633	0.1638	0.0876	0.0520
TDS_17	1.2000	0.0974	0.1000	0.2560	0.0749	0.0680	0.0651
TDS_18	0.0016	0.0295	0.1000	0.1038	0.0385	0.0348	0.0452
TDS_19	1.2000	0.0787	0.1000	0.2589	0.0709	0.0743	0.0819
TDS_20	0.0074	0.0964	0.1000	0.1002	0.0515	0.0456	0.0540
TDS_21	0.7669	0.0972	0.1000	0.2758	0.0939	0.0817	0.0872
TDS_22	0.0016	0.0921	0.1000	0.1010	0.0414	0.0353	0.0439
TDS_23	1.2000	0.1011	0.1000	0.1757	0.0922	0.0897	0.0659
TDS_24	0.0108	0.0435	0.1000	0.1014	0.0411	0.0250	0.0405
PS_1	9.0720	1.8150	2.5000	1.2732	1.5008	1.3633	1.2293
PS_2	6.5540	1.2988	2.0899	1.5200	0.7306	0.7448	0.8309
PS_3	1.0687	1.4980	2.5000	1.1975	0.9324	1.1644	1.4368
PS_4	6.9794	1.3920	2.5000	0.6701	0.6729	0.9431	1.3374
PS_5	0.1760	0.9480	2.5000	1.0785	0.7770	0.7575	0.7968
PS_6	8.1739	1.6430	2.5000	0.6311	1.3588	1.0905	1.0480
PS_7	8.1739	1.6430	2.5000	0.9637	1.3636	0.6818	1.4258
PS_8	0.6555	0.9480	2.5000	1.1393	0.7805	0.7586	0.7492
PS_9	6.9794	1.3920	2.5000	1.1994	0.9438	1.0734	1.3399
PS_10	0.7596	1.4980	2.5000	1.7451	1.2611	1.1382	1.3010
PS_11	6.5540	1.1369	2.0899	0.8454	0.6985	0.7297	0.6622
PS_12	9.0940	1.8150	2.5000	0.6461	0.8530	0.8666	1.0194
PS_13	6.8778	1.3740	2.2969	0.9784	0.5695	0.9882	1.3203
PS_14	7.7996	1.5560	2.5000	0.8860	1.3133	1.1905	1.0996
PS_15	7.7838	1.5560	2.5000	0.8993	1.0621	0.9953	0.5193
PS_16	6.8778	0.9639	2.2969	0.5004	0.3827	1.0178	1.3214
PS_17	1.7200	1.7200	2.1606	0.9197	1.4577	1.4878	1.6389
PS_18	1.8715	1.6347	0.5000	0.5003	2.0809	2.1396	1.6927
PS_19	1.6800	1.6800	1.6462	0.7629	1.4125	1.4416	1.6038
PS_20	2.3447	0.2006	0.5000	0.5041	1.9831	2.5285	1.6553
PS_21	1.4735	1.7200	2.1606	0.8902	1.4581	1.4810	1.4358
PS_22	1.8715	0.2000	0.5000	0.5008	1.9307	2.2709	2.0137
PS_23	1.7900	1.7900	1.9435	1.5724	1.5465	1.5835	1.6959
PS_24	0.9989	0.7441	0.5000	0.5017	1.6236	1.6566	1.6917
Pop	-	-	-	-	30	30	30
Iter	-	-	-	-	200	200	200
Time	-	-	7.29	30.20	0.6016	0.4755	0.7930
Std	-	-	0.1233	1.2133	2.8335	2.2792	1.4472
OF	19.4041	6.1786	8.6822	14.2338	7.1378	6.8319	6.3713
Feasible	No	Different	Fully	Fully	Fully	Fully	Fully

by the following equations:

$$PS_{min}^{i} = \frac{I_{n,i} \times OLF}{CTR} \tag{18}$$

$$PS_{max}^{i} = I_{f,i}^{min} \times \frac{2}{3CTR} \tag{19}$$

where $I_{n,i}$ is the nominal current rating of the circuit protected by the relay R_i, OLF is the overload factor equal to 1.25, $I_{f,i}^{min}$ is the minimum fault current detected by R_i.

Common parameters of N_var is 48, N_pop is 30, Max_iter is 200. The optimum settings of TDS and PS are presented in Table.12. It is noticed that, no feasible solution can be found by NLP [4]. The best result is obtained by GA-NLP [4] with values of 6.1786 (s), followed by OJaya, DJaya and Jaya with values as 6.3713(s), 6.8319(s) and 7.1378(s), respectively. But the authors found that, the system data used in [4], is a little different from the

TABLE 13. Coordination time interval for 9-bus system by NLP formulation.

Primary Relay	Backup Relay	CTI		
		Jaya	DJaya	OJaya
1	15	0.7968	0.5683	0.2716
1	17	0.6748	0.6313	0.7367
2	4	0.3455	0.3623	0.5506
3	1	0.4111	0.3757	0.4567
4	6	0.5281	0.5577	0.3595
5	3	0.4270	0.3634	0.4093
6	8	0.6621	0.3935	0.4831
6	23	0.8937	0.8545	0.7560
7	5	0.4436	0.4338	0.3718
7	23	0.9591	0.9040	0.6756
8	10	0.2719	0.4750	0.5300
9	7	0.3343	0.2068	0.8524
10	12	0.6517	0.3200	0.5860
11	9	0.3077	0.4741	0.4243
12	14	0.5119	0.5663	0.4437
12	21	0.8513	0.8161	0.8058
13	11	0.5902	0.7021	0.4537
13	21	0.8311	0.7893	0.8114
14	16	0.4232	0.5991	0.5686
14	19	0.6019	0.6581	1.0085
15	13	0.2305	0.3201	0.5693
15	19	0.4587	0.6070	0.9712
16	2	0.2468	0.3797	0.4041
16	17	0.5319	0.5515	0.7850
17	-	-	-	-
18	2	0.2802	0.3627	0.2952
18	15	0.6873	0.4716	0.2111
19	-	-	-	-
20	13	0.2530	0.2000	0.5278
20	16	0.3025	0.4279	0.4898
21	-	-	-	-
22	11	0.5617	0.5777	0.2973
22	14	0.4632	0.4152	0.2929
23	-	-	-	-
24	5	0.2456	0.4593	0.2747
24	8	0.5296	0.4684	0.3056

TABLE 14. CT ratio for the relays of 15-bus system [33].

Relay No	CT ratio
18-20-21-29	1600/5
2-4-8-11-12-14-15-23	1200/5
1-3-5-10-13-19-36-37-40-42	800/5
6-7-9-16-24-25-26-27-28-31-32-33-35	600/5
17-22-30-34-38-39-41	400/5

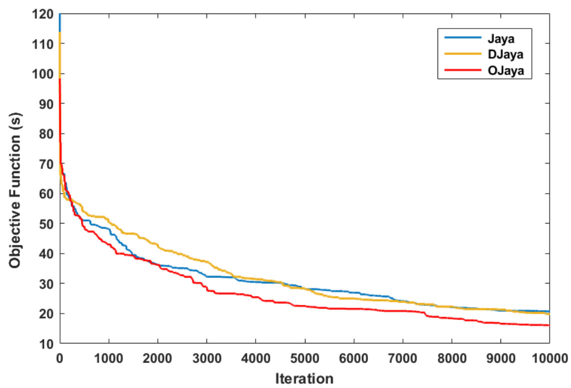

FIGURE 15. Convergence characteristics for 15-bus system with NLP formulation.

TABLE 15. Primary/Backup relay pairs and related parameters for 15-bus system [33].

Primary Relay	IF(A)	Backup Relay	IF(A)	Primary Relay	IF(A)	Backup Relay	IF(A)
R1	3621	R6	1233	R20	7662	R30	681
R2	4597	R4	1477	R21	8384	R17	599
R2	4597	R16	743	R21	8384	R19	1372
R3	3984	R1	853	R21	8384	R30	681
R3	3984	R16	743	R22	1950	R23	979
R4	4382	R7	1111	R22	1950	R34	970
R4	4382	R12	1463	R23	4910	R11	1475
R4	4382	R20	1808	R23	4910	R13	1053
R5	3319	R2	922	R24	2296	R21	175
R6	2647	R8	1548	R24	2296	R34	970
R6	2647	R10	1100	R25	2289	R15	969
R7	2497	R5	1397	R25	2289	R18	1320
R7	2497	R10	1100	R26	2300	R28	1192
R8	4695	R3	1424	R26	2300	R36	1109
R8	4695	R12	1463	R27	2011	R25	903
R8	4695	R20	1808	R27	2011	R36	1109
R9	2943	R5	1397	R28	2525	R29	1828
R9	2943	R8	1548	R28	2525	R32	697
R10	3568	R14	1175	R29	8346	R17	599
R11	4342	R3	1424	R29	8346	R19	1372
R11	4342	R7	1111	R29	8346	R22	642
R11	4342	R20	1808	R30	1736	R27	1039
R12	4195	R13	1503	R30	1736	R32	697
R12	4195	R24	753	R31	2867	R27	1039
R13	3402	R9	1009	R31	2867	R29	1828
R14	4606	R11	1475	R32	2069	R33	1162
R14	4606	R24	753	R32	2069	R42	907
R15	4712	R1	853	R33	2305	R21	1326
R15	4712	R4	1477	R33	2305	R23	979
R16	2225	R18	1320	R34	1715	R31	809
R16	2225	R26	905	R34	1715	R42	907
R17	1875	R15	969	R35	2095	R25	903
R17	1875	R26	905	R35	2095	R28	1192
R18	8426	R19	1372	R36	3283	R38	882
R18	8426	R22	642	R37	3301	R35	910
R18	8426	R30	681	R38	1403	R40	1403
R19	3998	R3	1424	R39	1434	R37	1434
R19	3998	R7	1111	R40	3140	R41	745
R19	3998	R12	1463	R41	1971	R31	809
R20	7662	R17	599	R41	1971	R33	1162
R20	7662	R22	642	R42	3295	R39	896

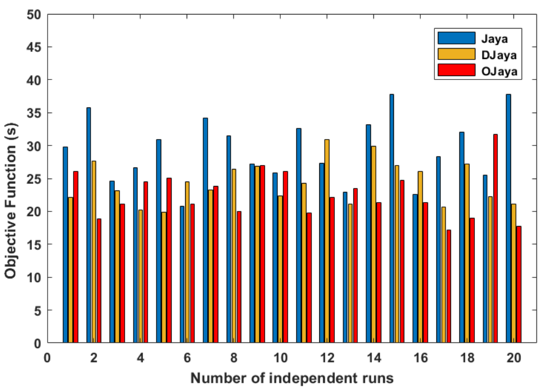

FIGURE 16. Independent runs for 15-bus system with NLP formulation.

TABLE 16. Time dial setting and plug setting for 15-bus system by NLP formulation.

Relay No	Jaya TDS	Jaya PS	DJaya TDS	DJaya PS	OJaya TDS	OJaya PS
1	0.1360	1.6418	0.1534	1.0202	0.1000	2.5000
2	0.1281	1.4431	0.1145	1.4195	0.2024	0.5000
3	0.2509	1.3918	0.1982	1.1228	0.1898	0.9608
4	0.1991	0.5000	0.1000	1.7873	0.1000	1.9786
5	0.2471	1.4670	0.1767	2.5000	0.1766	1.0276
6	0.1343	2.0841	0.1517	1.4721	0.1317	1.7451
7	0.4165	0.5000	0.4189	0.5000	0.1118	2.5000
8	0.1889	2.0939	0.1000	2.5000	0.1139	1.5890
9	0.3224	1.2985	0.2299	0.5000	0.2016	0.5027
10	0.2666	1.0800	0.1811	1.9185	0.2212	0.5000
11	0.1570	1.2341	0.1190	1.6042	0.1828	0.5000
12	0.1693	1.7420	0.1306	1.5998	0.1367	1.2163
13	0.2847	0.5000	0.1336	1.4281	0.1036	1.8025
14	0.1353	1.7261	0.1815	0.8812	0.1000	1.7457
15	0.1003	1.8131	0.1000	2.5000	0.1000	2.5000
16	0.2701	1.3631	0.2335	0.5000	0.1873	0.8423
17	0.2080	1.2502	0.2886	0.5000	0.1955	0.5000
18	0.2506	0.8661	0.1160	1.3582	0.1000	1.5670
19	0.4109	0.5000	0.2649	0.5000	0.2054	0.5000
20	0.1000	2.5000	0.1910	0.5000	0.1000	1.6987
21	0.2555	0.5776	0.1000	2.4099	0.1000	1.6129
22	0.1994	1.3496	0.4440	0.5000	0.1000	2.5000
23	0.1649	1.2801	0.1000	2.5000	0.1011	1.9898
24	0.2973	0.5000	0.1745	0.9559	0.1335	1.4626
25	0.2740	1.1780	0.2467	0.7803	0.1536	1.8839
26	0.4096	0.5000	0.1426	1.9197	0.1629	1.2080
27	0.2046	1.7175	0.1837	1.0760	0.2263	0.5000
28	0.1952	2.4996	0.1644	1.9984	0.1972	1.7005
29	0.2630	1.0095	0.1000	2.5000	0.1000	2.5000
30	0.2499	0.7583	0.1533	1.1962	0.1000	2.5000
31	0.3096	1.0174	0.1316	2.5000	0.1000	2.5000
32	0.2342	0.8759	0.1332	1.5971	0.1000	2.5000
33	0.3666	0.5083	0.4259	0.5000	0.2815	0.5000
34	0.4147	0.5000	0.2258	2.5000	0.1852	1.3265
35	0.2593	1.5769	0.1654	1.7953	0.2887	0.5000
36	0.1417	2.5000	0.1470	1.5630	0.1024	2.4614
37	0.3029	1.3072	0.2094	1.5171	0.2685	0.5000
38	0.2022	1.4777	0.1703	1.5302	0.2507	0.5000
39	0.2676	1.5963	0.2409	1.2030	0.1620	1.3475
40	0.2475	1.3711	0.1536	2.0988	0.1983	1.1987
41	0.2486	1.2939	0.2449	1.8541	0.1644	1.7779
42	0.3215	1.0521	0.2061	1.4120	0.1000	2.5000
Pop	50		50		50	
Iter	10000		10000		10000	
Time	30.42		24.39		22.70	
Std	4.332		2.579		1.996	
OF	23.5579		18.8404		15.5233	
Feasible	Fully		Fully		Fully	

TABLE 17. Comparison of the results for 15-bus system with NLP formulation.

Method	OF (s)
MINLP(SBB) [33]	15.335
Seeker [33]	12.227
GA [38]	18.9033
PSO [38]	26.8093
DE [38]	11.7591
HS [38]	12.6225
SOA [38]	20.4068
Jaya	23.5579
DJaya	18.8404
OJaya	15.5233

commonly-used system data given in Table.11. Because the primary relay 13 and 14 in [4] has only one backup relay; actually, the primary relay 13 and 14 has two backup relays, as showed in Table.11. This difference may lead to the OF value of GA-NLP is less than OJaya. In fact, [38] uses the same system data as Table.11, which shows that, OJaya is not only better than the algorithms of DE and SOA, but also better than GA (14.5426), PSO (13.9472) and HS (9.2339) [38]. Because of limited space of the table, we did not show all the algorithms from [38], but the comparison illustrates that, OJaya is still the best performer in this case.

The convergence characteristics could be seen in Fig.13, from which we can observe that, both OJaya and DJaya converge faster than Jaya, and obtained lower OF values as well, while OJaya obviously achieves the lowest OF value.

TABLE 18. Coordination time interval for 15-bus system by NLP formulation.

P	B	CTI			P	B	CTI		
		Jaya	DJaya	OJaya			Jaya	DJaya	OJaya
1	6	0.2265	0.2001	0.2000	20	30	0.4032	0.2038	0.3067
2	4	0.2035	0.2591	0.2353	21	17	0.3478	0.4400	0.2479
2	16	0.8926	0.3333	0.2696	21	19	0.5324	0.3477	0.2475
3	1	0.2073	0.2047	0.5223	21	30	0.2548	0.2498	0.3202
3	16	0.6389	0.1994	0.2492	22	23	0.5160	0.6538	0.6783
4	7	0.5964	0.6814	0.2819	22	34	0.4132	0.2163	0.2726
4	12	0.5607	0.3800	0.2761	23	11	0.2686	0.2858	0.2007
4	20	0.4779	0.2439	0.2674	23	13	0.3484	0.2767	0.2561
5	2	0.2711	0.2248	0.2816	24	21	0.3394	0.8889	0.3807
6	8	0.7730	0.3498	0.2069	24	34	0.3313	0.5898	0.2188
6	10	0.6006	0.5988	0.2212	25	15	0.2002	0.9302	0.9996
7	5	0.1994	0.2192	0.2039	25	18	0.4366	0.2000	0.2625
7	10	0.2365	0.2231	0.2139	26	28	0.2187	0.2823	0.3672
8	3	0.3505	0.3231	0.2740	26	36	0.2048	0.2567	0.2840
8	12	0.3557	0.3410	0.2745	27	25	0.4007	0.2892	0.3303
8	20	0.2729	0.2048	0.2658	27	36	0.3483	0.2250	0.2498
9	5	0.2069	0.5792	0.2164	28	29	0.4161	0.3627	0.3051
9	8	0.4165	0.3341	0.2121	28	32	0.2225	0.2355	0.2883
10	14	0.3011	0.2240	0.2794	29	17	0.2510	0.4349	0.2005
11	3	0.5307	0.3208	0.2397	29	19	0.4356	0.3425	0.2001
11	7	0.5716	0.6401	0.2460	29	22	0.2211	0.7973	0.3017
11	20	0.4531	0.2026	0.2316	30	27	0.3669	0.2442	0.2228
12	13	0.2513	0.2295	0.2032	30	32	0.3460	0.3532	0.5064
12	24	0.2999	0.2638	0.2828	31	27	0.2059	0.2049	0.2366
13	9	0.6740	0.2172	0.2003	31	29	0.3787	0.4411	0.5369
14	11	0.2898	0.2121	0.2123	32	33	0.3114	0.5938	0.2899
14	24	0.4182	0.2372	0.3473	32	42	0.7799	0.6409	0.4925
15	1	0.5110	0.3060	0.5847	33	21	0.2087	0.4968	0.2143
15	4	0.2536	0.2265	0.2772	33	23	0.3033	0.6351	0.4582
16	18	0.4009	0.2866	0.3051	34	31	0.3808	0.1998	0.2456
16	26	0.3226	0.2834	0.2002	34	42	0.5701	0.3037	0.3950
17	15	0.3876	0.9482	1.1112	35	25	0.2783	0.2473	0.2169
17	26	0.5459	0.2145	0.2691	35	28	0.2398	0.2087	0.2196
18	19	0.4872	0.3682	0.2505	36	38	0.2289	0.2024	0.2191
18	22	0.2727	0.8231	0.3521	37	35	0.3902	0.2450	0.2365
18	30	0.2096	0.2703	0.3232	38	40	0.3583	0.2643	0.2078
19	3	0.2222	0.2006	0.2303	39	37	0.3243	0.2031	0.2056
19	7	0.2631	0.5199	0.2366	40	41	0.2306	0.5748	0.2007
19	12	0.2274	0.2185	0.2308	41	31	0.5511	0.2737	0.2723
20	17	0.4961	0.3941	0.2345	41	33	0.2720	0.3308	0.2190
20	22	0.4662	0.7565	0.3356	42	39	0.2085	0.2151	0.1996

Fig.14 shows 20 times of independent runs, we can observe that OJaya shows the strongest ability in maintaining the minimum value of OF (with *Std* equals to 1.4472), while DJaya suffers several times of premature problem (with *Std* equals to 2.2792), and Jaya has the worst robustness (with *Std* equals to 2.8335).

Table.13 shows the operating time and CTI, we can see that there is no selectivity constraint is violated.

D. CASE 6: 15-BUS SYSTEM

This 15-bus system is experimented as NLP formulation, which consists of 15 buses, 21 branches, 42 DOCRs and 82 P/B relay pairs. 3ϕ close-in fault is considered in all the lines. This case is a highly distributed generation (DG) penetrated distribution networks, where CTI is 0.2 (s), TDS is from 0.1 to 1.1, PS is from 0.5 (A) to 2.5 (A). The CT ratios, P/B relay pairs and currents for 3ϕ faults are available in Table.14 and Table.15.

Common parameters of N_var is 84, N_pop is 50, Max_iter is 10000. The optimum settings of TDS and PS are given in Table.16. We can observe that, OJaya is the best performer among Jaya, DJaya and OJaya in terms of OF value (15.5233). However, when we compare OJaya with other published algorithms for this case, OJaya is not the best one, as shown in Table.17. It means that, OJaya still has space for improvements.

The convergence characteristics are given in Fig.15, we can observe that, all the algorithms converge in a similar trend, but OJaya achieves much lower OF value than Jaya and DJaya. The distribution of OF value by 20 times runs is given in Fig.16, the comparison confirms that, OJaya maintains the best robustness with Std equals to 1.996.

Table.18 shows the operating time and CTI, we can see that there is no selectivity constraint is violated.

V. CONCLUSION

This paper proposed an oppositional Jaya (OJaya) algorithm with distance-adaptive coefficient (DAC). With the help of oppositional learning (OL) and DAC, the searching space of standard Jaya is expanded, the diversity of its population is strengthened, the convergence speed in approaching promising regions is accelerated as well. To compare the performances of Jaya and OJaya in solving real-world optimization problems, they are applied to the DOCRs coordination problem including 3-bus, 8-bus, 9-bus and 15-bus. Then we get conclusion that, OJaya has improved Jaya's performance in aspects of convergence rate, objective function value, robustness and computation efficiency in all the testing cases.

It worth mentioning that, there are three attractive properties of OJaya. The first one is, even though the concepts of OL and DAC are introduced, no more parameter is added throughout the whole implementation. The second one is, the working principle of OJaya is easy to understand. Thirdly, the overall frame of OJaya can be easily transported to other population-based evolutionary algorithms (EAs), such as PSO, teaching-learning-based optimization (TLBO), cuckoo search (CS) and artificial bee colony algorithm (ABC), which is one of the authors' interests in research in the future.

In the future study, the authors will mainly focus on two aspects. Firstly, how to improve the performances of OJaya in larger test systems as 30-bus or 42-bus in DOCRs coordination problem. Secondly, how to expand OJaya's applications in power system, such as apply OJaya to the overcurrent protection of AC microgrids by DOCRs.

REFERENCES

[1] A. J. Urdaneta, R. Nadira, and L. G. P. Jimenez, "Optimal coordination of directional overcurrent relays in interconnected power systems," *IEEE Trans. Power Del.*, vol. 3, no. 3, pp. 903–911, Jul. 1988.

[2] F. Razavi, H. A. Abyaneh, M. Al-Dabbagh, R. Mohammadi, and H. Torkaman, "A new comprehensive genetic algorithm method for optimal overcurrent relays coordination," *Electr. Power Syst. Res.*, vol. 78, pp. 713–720, Apr. 2008.

[3] A. S. Noghabi, J. Sadeh, and H. R. Mashhadi, "Considering different network topologies in optimal overcurrent relay coordination using a hybrid GA," *IEEE Trans. Power Del.*, vol. 24, no. 4, pp. 1857–1863, Oct. 2009.

[4] P. P. Bedekar and S. R. Bhide, "Optimum coordination of directional overcurrent relays using the hybrid GA-NLP approach," *IEEE Trans. Power Del.*, vol. 26, no. 1, pp. 109–119, Jan. 2011.

[5] M. M. Mansour, S. F. Mekhamer, and N. E.-S. El-Kharbawe, "A modified particle swarm optimizer for the coordination of directional overcurrent relays," *IEEE Trans. Power Del.*, vol. 22, no. 3, pp. 1400–1410, Jul. 2007.

[6] H. H. Zeineldin, E. F. El-Saadany, and M. M. A. Salama, "Optimal coordination of overcurrent relays using amodified particle swarm optimization," *Electr. Power Syst. Res.*, vol. 76, no. 11, pp. 988–995, Jul. 2006.

[7] M. Singh, B. K. Panigrahi, and A. R. Abhyankar, "Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 50, pp. 33–41, Sep. 2013.

[8] A. A. Kalage and N. D. Ghawghawe, "Optimum coordination of directional overcurrent relays using modified adaptive teaching learning based optimization algorithm," *Intell. Ind. Syst.*, vol. 2, no. 1, pp. 55–71, 2016.

[9] S. S. Gokhale and V. S. Kale, "An application of a tent map initiated chaotic firefly algorithm for optimal overcurrent relay coordination," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 336–342, Jun. 2016.

[10] M. H. Hussain, I. Musirin, A. F. Abidin, and S. R. A. Rahim, "Modified swarm firefly algorithm method for directional overcurrent relay coordination problem," *J. Theor. Appl. Inf. Technol.*, vol. 66, no. 3, pp. 741–755, Aug. 2014.

[11] T. Khurshaid, A. Wadood, S. G. Farkoush, C.-H. Kim, J. Yu, and S. B. Rhee, "Improved firefly algorithm for the optimal coordination of directional overcurrent relays," *IEEE Access*, vol. 7, pp. 78503–78514, Jun. 2019.

[12] A. Wadood, T. Khurshaid, S. G. Farkoush, J. Yu, C. H. Kim, and S.-B. Rhee, "Nature-inspired whale optimization algorithm for optimal coordination of directional overcurrent relays in power systems," *Energies*, vol. 12, no. 12, p. 2297, Jun. 2019.

[13] T. Khurshaid, A. Wadood, S. G. Farkoush, J. Yu, C. H. Kim, and S.-B. Rhee, "An improved optimal solution for the directional overcurrent relays coordination using hybridized whale optimization algorithm in complex power systems," *IEEE Access*, vol. 7, pp. 90418–90435, Jul. 2019.

[14] M. N. Alam, B. Das, and V. Pant, "An interior point method based protection coordination scheme for directional overcurrent relays in meshed networks," *Int. J. Elect. Power Energy Syst.*, vol. 81, pp. 153–164, Oct. 2016.

[15] M. N. Alam, "Adaptive protection coordination scheme using numerical directional overcurrent relays," *IEEE Trans. Ind. Inf.*, vol. 15, no. 1, pp. 64–73, May 2018.

[16] M. N. Alam, "Overcurrent protection of AC microgrids using mixed characteristic curves of relays," *Comput. Electr. Eng.*, vol. 74, pp. 74–88, Mar. 2019.

[17] A. Hatata and A. Lafi, "Ant lion optimizer for optimal coordination of DOC relays in distribution systems containing DGs," *IEEE Access*, vol. 6, pp. 72241–72252, 2018.

[18] S. T. P. Srinivas and S. K. Swarup, "Application of improved invasive weed optimization technique for optimally setting directional overcurrent relays in power systems," *Appl. Soft Comput.*, vol. 79, pp. 1–13, Jun. 2019.

[19] A. Korashy, S. Kamel, A.-R. Youssef, and F. Jurado, "Modified water cycle algorithm for optimal direction overcurrent relays coordination," *Appl. Soft Comput.*, vol. 74, pp. 10–25, Jan. 2019.

[20] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *Int. J. Ind. Eng. Comput.*, vol. 7, no. 1, pp. 19–34, 2016.

[21] K. J. Yu, B. Y. Qu, C. T. Yue, S. L. Ge, X. Chen, and J. Liang, "A performance-guided JAYA algorithm for parameters identification of photovoltaic cell and module," *Appl. Energy*, vol. 237, no. 1, pp. 241–257, Mar. 2019.

[22] J. T. Yu, C. H. Kim, A. Wadood, T. Khurshiad, and S. B. Rhee, "Jaya algorithm with self-adaptive multi-population and Levy flights for solving economic load dispatch problems," *IEEE Access*, vol. 7, pp. 21372–21384, Feb. 2019.

[23] L. Wang, Z. Zhang, C. Huang, and K. L. Tsui, "A GPU-accelerated parallel Jaya algorithm for efficiently estimating Li-ion battery model parameters," *Appl. Soft Comput.*, vol. 65, pp. 12–20, Apr. 2018.

[24] Y. Li, Z. Yang, G. Li, Y. Mu, D. Zhao, D. Zhao, C. Chen, and B. Shen, "Optimal scheduling of isolated microgrid with an electric vehicle battery swapping station in multi-stakeholder scenarios: A bi-level programming approach via real-time pricing," *Appl. Energy*, vol. 232, pp. 54–68, Dec. 2018.

[25] S. Xu, Y. Wang, and Z. Wang, "Parameter estimation of proton exchange membrane fuel cells using eagle strategy based on JAYA algorithm and Nelder-Mead simplex method," *Energy*, vol. 173, pp. 457–467, Apr. 2019.

[26] K. Gao, F. Yang, M. Zhou, Q. Pan, and P. Suganthan, "Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1944–1955, May 2018.

[27] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, pp. 1245–1287, Jan. 2002.

[28] J. Moirangthem, K. R. Krishnanand, S. S. Dash, and R. Ramaswami, "Adaptive differential evolution algorithm for solving non-linear coordination problem of directional overcurrent relays," *IET Gener. Transm. Distrib.*, vol. 7, no. 4, pp. 329–336, Apr. 2013.

[29] A. Bhattacharya and P. K. Chattopadhyay, "Biogeography-based optimization for different economic load dispatch problems," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 1064–1077, May 2010.

[30] W. L. Wang, W. K. Li, Z. Wang, and L. Li, "Opposition-based multiobjective whale optimization algorithm with global grid ranking," *Neurocomputing*, vol. 341, no. 14, pp. 41–59, May 2019.

[31] G.-G. Wang, S. Deb, A. H. Gandomi, and A. H. Alavi, "Opposition-based krill herd algorithm with Cauchy mutation and position clamping," *Neurocomputing*, vol. 177, pp. 147–157, Feb. 2016.

[32] K. Yu, J. J. Liang, B. Y. Qu, X. Chen, and H. Wang, "Parameters identification of photovoltaic models using an improved JAYA optimization algorithm," *Energy Convers. Manage.*, vol. 150, pp. 742–753, Oct. 2017.

[33] T. Amraee, "Coordination of directional overcurrent relays using seeker algorithm," *IEEE Trans. Power Del.*, vol. 27, no. 3, pp. 1415–1422, Jul. 2012.

[34] M. Alipour, S. Teimourzadeh, and H. Seyedi, "Improved group search optimization algorithm for coordination of directional overcurrent relays," *Swarm Evol. Comput.*, vol. 23, pp. 40–49, Aug. 2015.

[35] A. Mahari and H. Seyedi, "An analytic approach for optimal coordination of overcurrent relays," *IET Gener., Transmiss. Distrib.*, vol. 7, no. 7, pp. 674–680, Jul. 2013.

[36] F. A. Albasri, A. R. Alroomi, and J. H. Talaq, "Optimal coordination of directional overcurrent relays using biogeography-based optimization algorithms," *IEEE Trans. Power Del.*, vol. 30, no. 4, pp. 1810–1820, Aug. 2015.

[37] M. Ezzeddine, R. Kaczmarek, and M. U. Iftikhar, "Coordination of directional overcurrent relays using a novel method to select their settings," *IET Gener. Transm. Distrib.*, vol. 5, no. 7, pp. 743–750, Jul. 2011.

[38] M. N. Alam, B. Das, and V. Pant, "A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination," *Electr. Power Syst. Res.*, vol. 128, pp. 39–52, Nov. 2015.

JIANGTAO YU received the B.S. and M.S. degrees from Northeast Forestry University, Harbin, China, in 2009 and 2012, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Yeungnam University, South Korea. She was with the Department of Electronic Information and Electrical Engineering, Anyang Institute of Technology, China. Her current research interests include power system optimization and evolutionary algorithms.

CHANG-HWAN KIM received the B.S. and M.S. degrees from Yeungnam University, Gyeongbuk, South Korea, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. His research interests include power system control and operation, optimal power flow and evolutionary computation, and EMTP and ATP application.

SANG-BONG RHEE received the B.S., M.S., and Ph.D. degrees from Hanyang University, South Korea, in 1994, 1999, and 2004, respectively. He was a Research Professor with the School of Electrical and Computer Engineering, Sung-Kyunkwan University, South Korea. He is currently an Assistant Professor with the Department of Electrical Engineering, Yeungnam University, Gyeongbuk, South Korea. His research interests include distribution system control and operation, and artificial intelligence applications to power system protection.