

## Research Article

# The Comparison of Lately Proposed Harris Hawks Optimization and Jaya Optimization in Solving Directional Overcurrent Relays Coordination Problem

Jiangtao Yu,<sup>1,2</sup> Chang-Hwan Kim ,<sup>2</sup> and Sang-Bong Rhee <sup>2</sup>

<sup>1</sup>Department of Electronic Information and Electrical Engineering, Anyang Institute of Technology, Anyang 455000, China

<sup>2</sup>Department of Electrical Engineering, Yeungnam University, Kyonsan 38541, Republic of Korea

Correspondence should be addressed to Sang-Bong Rhee; rrsd@yu.ac.kr

Received 7 September 2019; Revised 2 January 2020; Accepted 4 January 2020; Published 22 January 2020

Academic Editor: Mojtaba Ahmadi Khanesar

Copyright © 2020 Jiangtao Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a lately proposed Harris Hawks Optimizer (HHO) is used to solve the directional overcurrent relays (DOCRs) coordination problem. To the best of the authors' knowledge, this is the first time HHO is being used in the DOCRs coordination problem. The main inspiration of HHO is the cooperative behavior and chasing style of Harris' hawks from different directions, based on the dynamic nature of scenarios and escaping patterns of the prey. To test its performances in solving the DOCRs coordination problem, it is adopted in 3-bus, 4-bus, 8-bus, and 9-bus systems, which are formulated by three kinds of optimization models as linear programming (LP), nonlinear programming (NLP), and mixed integer nonlinear programming (MINLP), according to the nature of the design variables. Meanwhile, another lately proposed optimization algorithm named Jaya is also adopted to solve the same problem, and the results are compared with HHO in aspects of objective function value, convergence rate, robustness, and computation efficiency. The comparisons show that the robustness and consistency of HHO is relatively better than Jaya, while Jaya provides faster convergence rate with less CPU time and occasionally more competitive objective function value than HHO.

## 1. Introduction

Relay coordination task is considered of great importance for the operation of power systems. The optimal coordination of relays is supposed to guarantee that faults in the protected zones are cleared firstly by the corresponding primary relays, and if they fail, the corresponding backup relays act after a coordination time interval (CTI). With the development of relays, directional overcurrent 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 decided by two design variables as time dial setting (TDS) and pickup current setting (IP) or plug setting (PS). Optimal coordination between the operating times is able to maintain

the reliability of the overall protection system. Mathematically speaking, this coordination problem is a highly constrained optimization problem, which can be modeled in three ways as follows, according to the nature of the design variables:

- (i) Linear programming (LP)
- (ii) Nonlinear programming (NLP)
- (iii) Mixed integer nonlinear programming (MINLP)

Firstly, when the DOCRs coordination problem is formulated as an LP problem, 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 easily gets stuck in local minima [1].

Secondly, when formulated as an NLP problem, both TDS and IP are considered as design variables and calculated to optimize the relay operating time ( $T_i$ ), where TDS and IP take continuous values. By NLP, the total operational time of the primary relays can be reduced and the coordination can be maintained well. Thirdly, when formulated as an MINLP problem, both 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 problem. Genetic algorithm (GA), Hybrid GA, and Hybrid GA-NLP were used in [2–4]. Modified DE algorithm (MDEA) and opposition-based chaotic DE (OCDE) were used in [5, 6]. Two modified particle swarm optimization (PSO) algorithms were used in [7, 8], where the repair algorithm and a nonrandom technique for initialization were introduced to the standard version, respectively. Teaching learning-based optimization (TLBO) and modified adaptive TLBO (MATLBO) were used in [9, 10]. Firefly algorithm (FA), chaotic firefly algorithm (CFA), and modified swarm firefly algorithm (MSFA) were used in [11–13]. Biogeography-based optimization (BBO) along with a new hybrid BBO with linear programming (BBO-LP) was used in [14]. Furthermore, Jaya algorithm [15], seeker optimization algorithm (SOA), simulated annealing-based symbiotic organism search (SASOS), and improved group search optimization (IGSO) were used in [16–18].

Regardless of the variety of these algorithms, exploration (diversification) and exploitation (intensification) phases are two common phases that should always be considered. In the exploration phase, the optimizer should be able to promote its randomized solutions as many as possible to thoroughly explore the whole space. In the exploitation phase, only the solutions with better fitness values are searched further on its neighborhood to intensify the searching quality. These two phases should be made balanced reasonably; otherwise, the optimizer would be trapped in local optima or suffer from immature convergence drawbacks.

Recently, a number of nature-inspired modern algorithms were proposed to effectively balance the exploration phase and exploitation phase by Mirjalili, such as grey wolf optimizer (GWO) [19], whale optimization algorithm (WOA) [20], ant lion optimizer (ALO) [21], and moth-flame optimization (MFO) [22]. They have achieved good results. Then, in 2019, a new nature-inspired technique named Harris' Hawks Optimizer (HHO) is proposed by Heidari et al. [23], with the same purpose to make fine balance between exploration and exploitation. The main idea of HHO is inspired from the cooperative behaviors of one of the most intelligent birds, Harris's hawks, in hunting escaping preys (rabbits in most cases). Different mathematical models are constructed to mimic different stages of hunts used by Harris's hawks; then, a new stochastic metaheuristic algorithm is proposed and designed based on the constructed models to tackle various optimization problems.

Moreover, another algorithm named Jaya is proposed by Rao in 2016 [24]. The most significant benefit of Jaya is that 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 competitive advantage makes it popularly applied in various real-world problems [25]. The applications including boosted regression trees (BRT) of the data-driven approaches [26], photovoltaic cell and module [27], economic load dispatch problems [28], Li-ion battery model [29], isolated microgrid with electric vehicle battery swapping stations [30], urban traffic light scheduling problem [31], maximum power point tracking (MPPT) problem of PV systems [32], parameter estimation of proton-exchange membrane fuel cells [33], flexible job-shop rescheduling problem (FJRP) [34], discrete optimization of truss structures [35], abrasive waterjet machining process [36], and so on.

In this paper, Jaya algorithm is used to be compared with the HHO algorithm in solving the DOCRs coordination problem, with the purpose of testing the advantages and disadvantages of each other, according to the objective function value, convergence rate, robustness, and computation efficiency. Then, the main contributions of this study can be summarized as follows:

- (i) The DOCRs coordination problem is experimented by 3 kinds of formulations, which are LP, NLP, and MINLP
- (ii) The lately proposed HHO and Jaya are used to solve the DOCRs coordination problem
- (iii) HHO and Jaya are implemented on 3-bus, 4-bus, 8-bus, and 9-bus test systems, and the problems are solved effectively in most cases
- (iv) The result comparisons show that the robustness and consistency of HHO is relatively better than Jaya, while Jaya provides faster convergence rate and occasionally more competitive objective function value than HHO.

Rest of this paper is organized as follows. In Section 2, the formulation of the DOCRs coordination problem is illustrated. Related works on HHO and Jaya are described in Section 3 and Section 4. Experimental results and comparisons are presented in Section 5. Finally, discussion and conclusion are given in Section 6.

## 2. Problem Formulation

*2.1. 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 ( $T_i$ ) of the primary relays in a system, as expressed below:

$$\text{OF} = \sum_{i=1}^N W_i T_i, \quad (1)$$

where  $N$  is the total number of primary relays,  $W_i$  is the weight assigned for relay  $R_i$  which is equal to 1 for all the relays in this study, and  $T_i$  is the operating time of relay  $R_i$  calculated by the following 3 kinds of formulations: LP, NLP, and MINLP.

**2.1.1. LP Formulation.** When the DOCRs coordination problem is formulated as an LP problem, the operating time  $T_i$  is a linear expression related with TDS for a pre-determined value of IP, as shown in equations(2) and (3):

$$T_i = \text{TDS}^i \times \frac{\alpha}{(\text{IF}^i/\text{IP}_{\text{constant}}^i)^\beta - \gamma} + L, \quad (2)$$

$$\text{IP}_{\text{constant}}^i = \text{PS}_{\text{constant}}^i \times \text{CT}^i, \quad (3)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $L$  are constant parameters which, according to the IEC curves, are assumed to be 0.14, 0.02, 1.0, and 0,  $\text{TDS}^i$  is the time dial settings of relay  $R_i$ ,  $\text{IF}^i$  is the fault current, and  $\text{IP}_{\text{constant}}^i$  is the pickup current flowing through relay  $R_i$  for a particular fault located in a particular zone. Since  $\text{PS}_{\text{constant}}^i$  stands for the plug setting which is a known constant and  $\text{CT}^i$  stands for the CT ratio, the pickup current  $\text{IP}_{\text{constant}}^i$  is a known constant by equation (3). Finally, the value of  $T_i$  is linearly related with the value of  $\text{TDS}^i$ ; therefore, the problem is simplified as a linear programming (LP) problem.

**2.1.2. NLP Formulation.** When formulated as an NLP problem, the design variables are TDS and IP, the operating time is calculated by

$$T_i = \text{TDS}^i \times \frac{\alpha}{(\text{IF}^i/\text{IP}_{\text{continuous}}^i)^\beta - \gamma} + L, \quad (4)$$

where the values of  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $L$  are the same as in LP formulation.  $\text{TDS}^i$  and  $\text{IF}^i$  also have the same meaning as in LP formulation. The difference is that  $\text{IP}_{\text{continuous}}^i$  in NLP formulation takes continuous variables, while  $\text{IP}_{\text{constant}}^i$  in LP formulation is a known constant.

**2.1.3. MINLP Formulation.** For MINLP formulation, the design variables are TDS and PS, where PS takes discrete values. Therefore, the operating time is calculated by

$$T_i = \text{TDS}^i \times \frac{\alpha}{[\text{IF}^i/(\text{PS}_{\text{discrete}}^i \times \text{CT}^i)]^\beta - \gamma} + L, \quad (5)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $L$ ,  $\text{TDS}^i$ , and  $\text{IF}^i$  have the same meaning as in LP and NLP formulations. The difference is that the continuous variable of  $\text{IP}_{\text{continuous}}^i$  in equation (4) is replaced by  $\text{PS}_{\text{discrete}}^i \times \text{CT}^i$  in equation (5), where  $\text{PS}_{\text{discrete}}^i$  takes discrete values [37, 38].

## 2.2. Constrained Functions

**2.2.1. Relay Coordination Constraints.** In a power system, when fault happens, it is sensed by primary and backup relays simultaneously. To avoid maloperation, the backup relay should takeover the tripping action, only after the primary relay fails to operate. So, the operating time of the backup relay ( $T_{\text{backup}}^i$ ) is decided by the operating time of the primary relay ( $T_{\text{primary}}^i$ ) 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

$$T_{\text{backup}}^i - T_{\text{primary}}^i \geq \text{CTI}. \quad (6)$$

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

**2.2.2. Relay Characteristic Constraints.** The relay characteristic constraints are the physical and operational bounds of the relay parameters as follows

$$T_i^{\min} \leq T_i \leq T_i^{\max}, \quad (7)$$

$$\text{TDS}_{\min}^i \leq \text{TDS}^i \leq \text{TDS}_{\max}^i, \quad (8)$$

$$\text{IP}_{\min}^i \leq \text{IP}^i \leq \text{IP}_{\max}^i, \quad (9)$$

$$\text{PS}_{\min}^i \leq \text{PS}^i \leq \text{PS}_{\max}^i, \quad (10)$$

where  $T_i^{\min}$  and  $T_i^{\max}$  in equation (7) are the minimum and maximum operating time of relay  $R_i$  for the fault at any point,  $\text{TDS}_{\min}^i$  and  $\text{TDS}_{\max}^i$  in equation (8) are the minimum and maximum values of  $\text{TDS}^i$  of relay  $R_i$ ,  $\text{IP}_{\min}^i$  and  $\text{IP}_{\max}^i$  in equation (9) are the minimum and maximum values of  $\text{IP}^i$  for relay  $R_i$ , and  $\text{PS}_{\min}^i$  and  $\text{PS}_{\max}^i$  in equation (10) are the minimum and maximum values of  $\text{PS}^i$  for relay  $R_i$ .

**2.3. Constraint Handling.** In this study, the 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 [39].

In the DOCRs coordination problem, the coordination constraints and the characteristic constraints are included in the objective function using the penalty method, as shown in equation (11). If any constraint is violated, a value of penalty is added to the objective function. Since the objective function is of minimization type, a large number is taken as the penalty factor:

$$\text{OF} = \sum_{i=1}^N T_i^{\text{primary}} + \sum_{k=1}^M \text{Penalty}(k), \quad (11)$$

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

$$\text{Penalty}(k) = \begin{cases} \xi \times |\text{CTI} - \Delta T_k|, & \text{if } \Delta T_k < \text{CTI}, \\ 0, & \text{if } \Delta T_k \geq \text{CTI}, \end{cases} \quad (12)$$

where  $\Delta T_k = T_k^{\text{backup}} - T_k^{\text{primary}}$  and  $\xi$  is the penalty factor for the penalty method to make the value of the objective function more significant during minimisation.  $\xi$  is usually given a relatively high value, with the aim to achieve zero penalties in optimal solutions [40].

### 3. Harris Hawks Optimization (HHO)

In this section, the exploratory and exploitative phases of Harris Hawks Optimization (HHO) are modeled. HHO is inspired by exploring a prey, surprise pounce, and different attacking strategies of Harris hawks. It is a population-based, gradient-free optimization technique, which can be applied to any optimization problems subject to proper formulations [23].

**3.1. Exploration Phase.** In HHO, Harris' hawks represent the candidate solutions and the intended prey represents the best candidate solution in every iteration. There are two strategies for Harris' hawks to perch and detect the prey: firstly, perch based on the positions of other family members and the prey; secondly, perch on random locations such as random tall trees. The strategies are modeled as follows:

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)|, & q \geq 0.5, \\ (X_{\text{rabbbit}}(t) - X_m(t)) - r_3 (\text{LB} + r_4 (\text{UB} - \text{LB})), & q < 0.5, \end{cases} \quad (13)$$

where  $X(t)$  is the current position of hawks,  $X(t+1)$  is the updated position of hawks,  $X_{\text{rabbbit}}(t)$  is the current position of the prey,  $r_1, r_2, r_3, r_4$ , and  $q$  are random numbers inside  $(0, 1)$ , LB and UB show the upper and lower bounds of variables,  $X_{\text{rand}}(t)$  is a randomly selected hawk from the population, and  $X_m(t)$  is the average position of the hawks which is calculated by the following equation:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t), \quad (14)$$

where  $N$  is the total number of hawks and  $X_i(t)$  is the location of each hawk in iteration  $t$ .

**3.2. Transition from Exploration to Exploitation.** The transition from exploration to exploitation is based on the escaping energy of the prey, which decreases during the escaping behavior. In HHO, the energy of a prey is modeled as

$$E = 2E_0 \left(1 - \frac{t}{T}\right), \quad (15)$$

where  $E$  is the escaping energy of the prey,  $E_0$  is the initial energy randomly changes within  $(-1, 1)$ , and  $T$  is the maximum number of iterations. When  $E_0$  decreases from 0 to  $-1$ , it means that the rabbit is physically flagging; when  $E_0$  increases from 0 to 1, it means that the rabbit is strengthening.

It is obvious that the value of  $E$  is decreasing as  $t$  is increasing. When  $|E| \geq 1$ , the hawks search different regions to explore the rabbit, working as the exploration phase; when  $|E| < 1$ , the algorithm performs the exploitation phase on the neighborhood of the solutions.

**3.3. Exploitation Phase.** The preys are always trying to escape from threatening situations. Based on the escaping behaviors of the prey, the hawks perform four different chasing strategies to catch the prey.

Suppose that  $r$  is the chance of a prey in successfully escaping ( $r < 0.5$ ) or not successfully escaping ( $r \geq 0.5$ ). To catch the prey, the hawks will encircle it softly or hardly, according to the remaining energy of the prey. Here, we use the  $E$  parameter to model this process; when  $|E| \geq 0.5$ , the soft besiege happens; when  $|E| < 0.5$ , the hard besiege occurs.

**3.3.1. Soft Besiege.** When  $r \geq 0.5$  and  $|E| \geq 0.5$ , the prey still has enough energy to escape, so the hawks encircle it softly to make the prey exhausted to perform surprise pounce. It is modeled by the following rules:

$$X(t+1) = \Delta X(t) - E |J X_{\text{rabbbit}}(t) - X(t)|, \quad (16)$$

$$\Delta X(t) = X_{\text{rabbbit}}(t) - X(t), \quad (17)$$

where  $\Delta X(t)$  is the position difference between the prey and the hawks in iteration  $t$ ,  $r_5$  is within  $(0, 1)$ , and  $J = 2(1 - r_5)$  is the random jump strength of the prey during escaping procedure.

**3.3.2. Hard Besiege.** When  $r \geq 0.5$  and  $|E| < 0.5$ , the prey is exhausted with low escaping energy. So, the hawks encircle the prey hardly to perform the surprise pounce. In this situation, the current positions are updated using the following equation:

$$X(t+1) = X_{\text{rabbbit}}(t) - E |\Delta X(t)|. \quad (18)$$

**3.3.3. Soft Besiege with Progressive Rapid Dives.** When  $|E| \geq 0.5$  and  $r < 0.5$ , the prey has enough energy to escape successfully, so a soft besiege is constructed. To perform the soft besiege, we suppose the hawks can evaluate their next movement based on the following equation:

$$Y = X_{\text{rabbbit}}(t) - E |J X_{\text{rabbbit}}(t) - X(t)|. \quad (19)$$

Then, they compare the movement to the previous dive to detect if the previous dive is better. If not, they will dive based on the levy flight (LF) pattern using the following equation:

$$Z = Y + S \times \text{LF}(D), \quad (20)$$

where  $D$  is the problem dimension,  $S$  is a random vector by size  $1 \times D$ , and LF is the levy flight function calculated by using the following equation:

$$\text{LF}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \quad (21)$$

where  $u$  and  $v$  are random values within  $(0, 1)$ ,  $\beta$  equals to 1.5, and  $\sigma$  is calculated by using the following equation:

$$\sigma = \left( \frac{\Gamma(1 + \beta) \times \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \times \beta \times 2^{((\beta-1)/2)}} \right)^{1/\beta}. \quad (22)$$

Finally, the updating strategy of the hawks in this phase is presented as follows:

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)), \\ Z, & \text{if } F(Z) < F(X(t)). \end{cases} \quad (23)$$

It is to be noted that, in all search agents, only the better position  $Y$  or  $Z$  will be selected to the next iteration.

**3.3.4. Hard Besiege with Progressive Rapid Dives.** When  $|E| < 0.5$  and  $r < 0.5$ , the rabbit has not enough energy to escape and a hard besiege is constructed. In the prey side, this step is similar to the soft besiege, but for the hawks, they try to decrease their distances with the escaping prey. So, the following rule is performed with different  $Y$  and  $Z$ :

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)), \\ Z, & \text{if } F(Z) < F(X(t)), \end{cases} \quad (24)$$

where  $Y$  and  $Z$  are obtained using equations (25) and (26):

$$Y = X_{\text{rabbit}}(t) - E|JX_{\text{rabbit}}(t) - X_m(t)|, \quad (25)$$

$$Z = Y + S \times \text{LF}(D), \quad (26)$$

where  $X_m(t)$  is calculated by equation (14).

**3.4. Pseudocode of HHO.** The pseudocode of the HHO algorithm is reported in Algorithm 1.

**3.5. Application of HHO Algorithm.** The main procedures of using HHO to solve the DOCRs coordination problem are illustrated with further details below:

*Step 1.* Set parameters. Three common parameters are initialized; they are number of design variable ( $N_{\text{var}}$ ), population size ( $N_{\text{pop}}$ ), and maximum iteration number ( $\text{Max\_iter}$ ).

*Step 2.* Initialize the location  $X$  of Harris's hawks in the form of

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,N_{\text{var}}} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,N_{\text{var}}} \\ X_{3,1} & X_{3,2} & \cdots & X_{3,N_{\text{var}}} \\ \cdots & \cdots & X_{i,j} & \cdots \\ X_{N_{\text{pop}},1} & X_{N_{\text{pop}},2} & \cdots & X_{N_{\text{pop}},N_{\text{var}}} \end{bmatrix}, \quad (27)$$

where

$$X_{i,j} = X_j^{\min} + (X_j^{\max} - X_j^{\min}) \times \text{rand}(N_{\text{pop}}, N_{\text{var}}), \quad (28)$$

where  $X_{i,j}$  is the  $i^{\text{th}}$  solution of the  $j^{\text{th}}$  variable where  $i \in [1, N_{\text{pop}}]$  and  $j \in [1, N_{\text{var}}]$ .  $X_j^{\min}$  and  $X_j^{\max}$  are the lower and upper limits of the  $j^{\text{th}}$  variable given by relay characteristic constraints of equations (8) and (9) or (10) in Section 2.2.

```

Initialize  $N_{\text{var}}$ ,  $N_{\text{pop}}$ , and  $\text{Max\_iter}$ ;
Generate initial population  $X$ ;
Evaluate the fitness value  $F(X)$ ;
Set  $t = 1$ ;
while  $t < \text{Max\_iter}$  do
  Set  $X_{\text{rabbit}}$  as the prey (best location);
  for each hawk ( $X_i$ ) do
    Update initial energy  $E_0$  and jump strength  $J$ ;
    Update the  $E$  by equation (15);
    if  $|E| \geq 1$  then
      Update the location vector using equation (13);
    end
    if  $|E| < 1$  then
      if  $r \geq 0.5$  and  $|E| \geq 0.5$  then
        Update the location vector using equation (16);
      end
      if  $r \geq 0.5$  and  $|E| < 0.5$  then
        Update the location vector using equation (18);
      end
      if  $r < 0.5$  and  $|E| \geq 0.5$  then
        Update the location vector using equation (19);
      end
      if  $r < 0.5$  and  $|E| < 0.5$  then
        Update the location vector using equation (24);
      end
    end
    if  $F[X_i(t+1)]$  is better than  $F[X_i(t)]$  then
       $X_i(t) = X_i(t+1)$ 
       $F[X_i(t)] = F[X_i(t+1)]$ 
    else
      Keep the old value;
    end
  end
   $t = t + 1$ ;
end

```

ALGORITHM 1: HHO.

*Step 3.* The fitness value of Harris's hawks  $F(X_{i,j})$  is calculated, which is problem dependent. In Section 5, we use equation (2) as the objective function for LP formulation, equation (4) for NLP formulation, and equation (5) for MINLP formulation.

*Step 4.* Check the constraints according to Section 2.2.

*Step 5.* Update the location of the prey  $X_{\text{rabbit}}$  and its energy  $F(X_{\text{rabbit}})$ .

*Step 6.* Calculate the escaping energy  $E$  of the prey by equation (15).

*Step 7.* Update the location  $X_i$  of Harris's hawks according to the value of  $E$ , which is based on the following conditions:

If  $|E| \geq 1$ , then we perform the exploration phase by equation (13);

If  $|E| < 1$ , then we perform the exploitation phase by 4 strategies regarding to the behavior of the rabbit, which are as follows:

If  $r \geq 0.5$  and  $|E| \geq 0.5$ , then we perform soft besiege by equation (16);

If  $r \geq 0.5$  and  $|E| < 0.5$ , then we perform hard besiege by equation (18);

If  $r < 0.5$  and  $|E| \geq 0.5$ , then we perform soft besiege with progressive rapid dives by equation (19);

If  $r < 0.5$  and  $|E| < 0.5$ , then we perform hard besiege with progressive rapid dives by equation (24);

*Step 8.* The updated fitness value  $F[X_i(t+1)]$  is calculated, which is as the same as in Step 3.

*Step 9.* Compare the present fitness value with its former fitness value and update it as follows:

If  $F[X_i(t+1)] < F[X_i(t)]$ ,

then  $F[X_i(t)] = F[X_i(t+1)]$  and  $X_i(t) = X_i(t+1)$ ;

otherwise, keep the old value.

*Step 10.* Check the stopping condition. If Max\_iter is reached, stop the loop and report the best solution; otherwise set  $t = t + 1$  and go to Step 4 to continue the iteration.

## 4. Jaya Algorithm

Jaya algorithm is a lately developed yet powerful heuristic algorithm for solving constrained and unconstrained optimization problems [24]. Compared with most of the other heuristic algorithms requiring for algorithm-specific parameters, Jaya is totally free from algorithm-specific parameters, and only two common parameters named maximum number of iteration (Max\_iter) and population size ( $N_{\text{pop}}$ ) are required, whose values can be initialised easily. In this section, the working principle of Jaya and the application of Jaya in solving DOCRs coordination problem are explained in the following parts.

*4.1. Jaya Algorithm.* Suppose the objective function  $F(X)$  is required to be minimized or maximized. Let the population size be  $N_{\text{pop}}$  where the index  $u \in [1, N_{\text{pop}}]$ , let the total design variable number be  $N_{\text{var}}$  where the index  $v \in [1, N_{\text{var}}]$ , and let the maximum iteration number be Max\_iter where the index  $w \in [1, \text{Max\_iter}]$ . Then, let  $X_{u,v,w}$  be the value of the  $u^{\text{th}}$  candidate population for the  $v^{\text{th}}$  variable during the  $w^{\text{th}}$  iteration, then the new modified value  $X_{u,v,w}^{\text{new}}$  is calculated by

$$X_{u,v,w}^{\text{new}} = X_{u,v,w} + r_1 \times (X_{\text{best},v,w} - |X_{u,v,w}|) - r_2 \times (X_{\text{worst},v,w} - |X_{u,v,w}|), \quad (29)$$

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

It should be explained that, in equation (29), the first term " $X_{u,v,w}$ " represents the original position, which provides the necessary start point for each population (as a moving particle) to roam among the fitness space. The second term " $r_1 \times (X_{\text{best},v,w} - |X_{u,v,w}|)$ " encourages the population to fly toward the spot of the best position found so far. The third term " $-r_2 \times (X_{\text{worst},v,w} - |X_{u,v,w}|)$ "

represents the tendency of the population to run far away from the worst position found so far. Pseudocode of Jaya is shown in Algorithm 2.

*4.2. Application of Jaya Algorithm.* According to the previous work, Jaya algorithm is implemented on the DOCRs coordination problem. As shown in Algorithm 2, it starts by setting values for common parameters. Then, the initial population is created. After this, each population is updated by the Jaya function. Then, we compare the current fitness value with its previous fitness value to keep the better one. At last, if the maximum iteration number is reached, stop the iteration and record the best solution. Otherwise, go to the next iteration. The main procedures of using Jaya to solve the DOCRs coordination problem are illustrated with further details below:

*Step 1.* Set parameters. Three common parameters are initialized; they are number of design variable ( $N_{\text{var}}$ ), population size ( $N_{\text{pop}}$ ), and maximum iteration number (Max\_iter).

*Step 2.* Initialization: initial population  $X$  is generated in the form of

$$X = \begin{bmatrix} X_{1,1,w} & X_{1,2,w} & \cdots & X_{1,N_{\text{var}},w} \\ X_{2,1,w} & X_{2,2,w} & \cdots & X_{2,N_{\text{var}},w} \\ X_{3,1,w} & X_{3,2,w} & \cdots & X_{3,N_{\text{var}},w} \\ \cdots & \cdots & X_{u,v,w} & \cdots \\ X_{N_{\text{pop}},1,w} & X_{N_{\text{pop}},2,w} & \cdots & X_{N_{\text{pop}},N_{\text{var}},w} \end{bmatrix}, \quad (30)$$

where

$$X_{u,v,w} = X_v^{\min} + (X_v^{\max} - X_v^{\min}) \times \text{rand}(N_{\text{pop}}, N_{\text{var}}), \quad (31)$$

where  $X_{u,v,w}$  is the  $u^{\text{th}}$  candidate solution of the  $v^{\text{th}}$  variable and  $w$  is the iteration index number which could be ignored in the initialization step.  $X_v^{\min}$  and  $X_v^{\max}$  are the lower and upper limits of the  $v^{\text{th}}$  variable given by relay characteristic constraints of equations (8) and (9) or (10) in Section 2.2.

*Step 3.* Evaluation: fitness value  $F(X_{u,v,w})$  is calculated by the objective function, which is as the same as Step 3 in Section 3.5.

*Step 4.* Check the constraints according to Section 2.2.

*Step 5.* Identify  $X_{\text{best},v,w}$  and  $X_{\text{worst},v,w}$  according to the best and worst value within  $F(X)$ .

*Step 6.* Update the population. The updated population  $X_{u,v,w}^{\text{new}}$  is calculated by equation (29).

*Step 7.* Evaluation: the updated fitness value  $F(X_{u,v,w}^{\text{new}})$  is calculated as the same as in Step 3.

*Step 8.* Comparison: compare the present fitness value with its former fitness value and keep the better one.

*Step 9.* Check the stopping condition. If Max\_iter is reached, stop the loop and report the best solution;

```

Initialize  $N\_var$ ,  $N\_pop$ , and  $Max\_iter$ ;
Generate initial population  $X$ ;
Evaluate the fitness value  $F(X)$ ;
Set  $w = 1$ ;
while  $w < Max\_iter$  do
  Identify  $X_{best,v,w}$  and  $X_{worst,v,w}$  according to the best and worst value within  $F(X)$ ;
  for  $u = 1 \rightarrow N\_pop$  do
    for  $v = 1 \rightarrow N\_var$  do
      Generate updated population  $X_{u,v,w}^{new}$  by equation (29);
    end
    if  $F(X_{u,v,w}^{new})$  is better than  $F(X_{u,v,w})$  then
       $X_{u,v,w} = X_{u,v,w}^{new}$ 
       $F(X_{u,v,w}) = F(X_{u,v,w}^{new})$ 
    else
      Keep the old value;
    end
  end
   $w = w + 1$ ;
end

```

ALGORITHM 2: Jaya.

otherwise set  $w = w + 1$  and go to Step 4 to continue the iteration.

## 5. Numerical Experiments

To evaluate the effectiveness of HHO and Jaya in solving the DOCRs coordination problem, four test systems such as 3-bus, 4-bus, 8-bus, and 9-bus (one LP case, four NLP cases, and two MINLP cases) have been investigated in this section, where each system has its own different set of design variables. All the cases are developed using MATLAB software (version R2018b) and executed on a computer under windows 7 on Intel(R) Core(TM) i5-6500 CPU 3.20 GHz with 8 GB RAM environment.

In the following tables, “Pop” is denoted as population size, “Iter” is the maximum iteration number, “Time” is the average CPU time spent on one time of independent run, “Std” is the standard deviation of objective function value among 20 times of independent runs, “OF” is the best objective function value, “Feasible” means if this solution satisfies all the constraints, and “✓” means “yes” and “✗” means “no.”

**5.1. Parameter Settings.** In HHO algorithm, there are three control parameters including the population size  $N\_pop$ , the maximum number of iterations  $Max\_iter$ , and the variable number  $N\_var$ . Among the three parameters,  $N\_pop$  and  $Max\_iter$  have direct impacts on the optimal solution and execution time. In this paper, we choose  $N\_pop$  from {5, 20, 30, 50} and we choose  $Max\_iter$  from {20, 50, 200, 1000, 2000}. The choosing principle for these two values is not complex; as for small-scale systems with simple constraints, the values are set relatively smaller, such as Case 1 where  $N\_pop = 5$  and  $Max\_iter = 20$ ; on the contrary, for large-scale systems with complex constraints, they are set relatively higher, such as Case 7 where  $N\_pop = 50$  and

$Max\_iter = 2000$ . However, even though these two parameters are tuned for each case, there are no criteria for a perfect selection of these parameters; the authors choose certain values only for making comprehensive and fair comparison with other conducted studies in the literature.

Jaya algorithm owns the same control parameters:  $N\_pop$ ,  $Max\_iter$ , and  $N\_var$ . To make fair comparison with HHO algorithm in solving the DOCRs coordination problem under the same condition, these parameters are set as the same values as HHO algorithm in each case in the following sections.

**5.2. 3-Bus System.** This 3-bus system consists of 3 buses, 3 generators, 3 lines, and 6 relays, as shown in Figure 1.  $3\phi$  fault at the midpoint of each line is considered. The CT ratio, the listed primary/backup (P/B) relay pairs, and the  $3\phi$  fault current of each line are given in Table 1. The time dial setting (TDS) is considered continuously lying in [0.1, 1.1], the coordination time interval (CTI) is equal to 0.2 s, and the plug setting (PS) lies in [1.5, 5.0]. All the relays have IDMT characteristic. This system is experimented separately by LP, NLP, and MINLP formulations.

**5.2.1. Case 1: 3-Bus System with LP Formulation.** In this case, the DOCRs coordination problem is formulated as an LP problem. The common parameters for HHO and Jaya are both set as follows:  $N\_pop = 5$ ,  $Max\_iter = 20$ , and  $N\_var = 6$ . The IF, PS, and CT are predefined fixed constants. The only variable is TDS, which is a continuous value and needs to be optimized. Data of this case could be found in Table 1. The optimum settings of TDS obtained by Jaya and HHO are given in Table 2. Simultaneously, simplex method [1], LP using matlab [7], PSO [7], and seeker algorithm (SA) [16] have also been presented to compare with the Jaya and HHO.

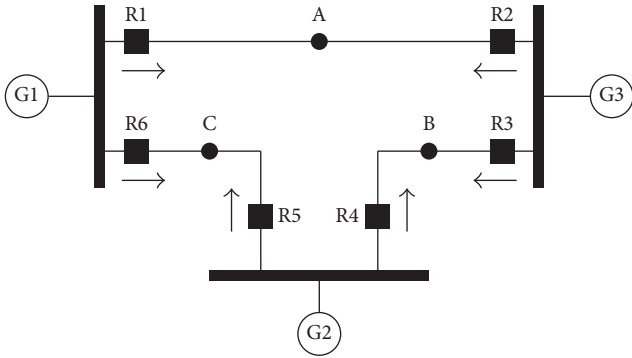


FIGURE 1: IEEE 3-bus DOCRs coordination problem model.

TABLE 1: P/B pairs and related parameters for the 3-bus system [16].

Relay no.	Primary relay		IF(A) (primary)	Backup relay	IF(A) (backup)
	CT	PS			
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

It is obvious from Table 2 that all the compared algorithms give the same objective function value as 1.9258 (s), but Jaya and HHO are able to give more optimized value as 1.7804 (s). It is to be noted that the standard deviation of HHO achieves 0, which means every time of its independent run reaches the global optima. However, the average CPU time by Jaya is 0.0218 (s), which is an extraordinary short time compared with LP, PSO, Seeker, and HHO.

To observe the convergence characteristics of Jaya and HHO more visually, Figure 2 depicts one convergence curve from 20 times of independent runs. We can see that Jaya shows faster convergence rate than HHO at the beginning and reaches its best value within less than 3 iterations. Figure 3 provides the value distributions of OF. We can observe that most of the runs are able to reach optimum results with LP formulation, but there exist some “outliers” with extreme values by Jaya. Those extreme values illustrate that, in this case, Jaya is suffering problems of falling into local optima which is far away from the global optima, but HHO is able to achieve the global optima in all independent runs with Std equals to 0.

Table 3 shows the operating times of primary and backup relays; we can see that the CTI constraints are satisfied in every P/B pair by both Jaya and HHO.

**5.2.2. Case 2: 3-Bus System with NLP Formulation.** In this case, the coordination problem is formulated as an NLP problem. Parameters for HHO and Jaya are both set as follows:  $N_{pop}=20$ ,  $Max\_iter=50$ , and  $N_{var}=12$ . The design variables to be optimized are TDS and IP, and both of them are considered continuous. System data are obtained from Table 1. The optimum settings of TDS and IP are

presented in Table 4. Simultaneously, GSO [18], IGSO [18], and analytic [41] algorithms have been provided to be compared.

It is observed from Table 4 that IGSO achieves the best OF value as 1.2918 (s) and both Jaya and HHO performs worse than IGSO. From Figure 4, we can observe that similar to Case 1, Jaya performs faster convergence rate than HHO. Figure 5 shows the value distribution of OF by 20 running times, from which we can see that the stability of Jaya (with Std equal to 2.2628) is not as good as that of HHO (with Std equal to 0.0995). The authors think that this is the price Jaya has to pay for very fast speed of the convergence rate.

Table 5 illustrates that the CTI constraints are satisfied in all P/B pairs by both Jaya and HHO.

**5.2.3. Case 3: 3-Bus System with MINLP Formulation.** In this case, parameters for HHO and Jaya are set as follows:  $N_{pop}=20$ ,  $Max\_iter=50$ , and  $N_{var}=12$ . The TDS is continuous, but PS is discrete in steps of 0.5 from 1.5 to 5.0, which is different from Case 2. System data are obtained from Table 1. The optimum settings of TDS and PS are showed in Table 6. The standard branch-and-bound (SBB) [16], Seeker [16], BBO [14], and BBO-LP [14] algorithms are provided to be compared.

We can observe from Table 6 that the minimum value of OF is achieved by HHO as 1.4984 (s), followed by Jaya as 1.5477 (s). The average time spent on one time of independent run by Jaya and HHO is 0.0286 (s) and 0.1228 (s), which are super-short times compared with the other algorithms. In Figure 6, Jaya reaches its optima in less than 15 iterations, while HHO needs about 25 iterations to reach its optima, which means Jaya converges faster than HHO. From Figure 7, we can observe intuitively that the OF value varies in a large range by Jaya (with Std equal to 1.7432), but HHO can maintain the OF value very well (with Std equal to 0.1030).

Table 7 illustrates that the CTI constraints are satisfied in all P/B pairs by both Jaya and HHO.

**5.3. 8-Bus System.** This 8-bus system is composed of 8 buses, 2 generators, 2 transformers, 7 lines, and 14 relays, as shown in Figure 8. The near-end  $3\phi$  fault is considered. The minimum and maximum values of TDS are set to be 0.1 and 1.1, while those of PS are set to be 0.5 and 2.5. The CTI is selected as 0.3. This system is experimented by NLP and MINLP formulations.

**5.3.1. Case 4: 8-Bus System with NLP Formulation.** The coordination problem in this case is formulated as an NLP problem. Parameters for HHO and Jaya are set as follows:  $N_{pop}=50$ ,  $Max\_iter=1000$ , and  $N_{var}=28$ . TDS and IP are continuous values. The CT ratio and  $3\phi$  short-circuit current for each P/B pair are given in Table 8. The optimum settings of TDS and IP obtained by Jaya and HHO are displayed in Table 9. The results are compared with EFO [38], MEFO BH [38], EM [38], HS [38], and PSO [38].

From Table 9, it is found that Jaya can converge to its global optimum in this case, but HHO cannot. Different



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

Relay	Time dial setting (TDS)					
	Simplex method [1]	LP [7]	PSO [7]	Seeker [16]	Jaya	HHO
1	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
3	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
5	0.1298	0.1298	0.1298	0.1298	0.1000	0.1000
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
Pop	—	—	—	50	5	5
Iter	—	—	1000	45	20	20
Time	—	0.4370	0.5129	6.45	0.0218	0.1068
Std	—	—	—	—	1.0431	0
OF	1.9258	1.9258	1.9258	1.9258	1.7804	1.7804
Feasible	✓	✓	✓	✓	✓	✓

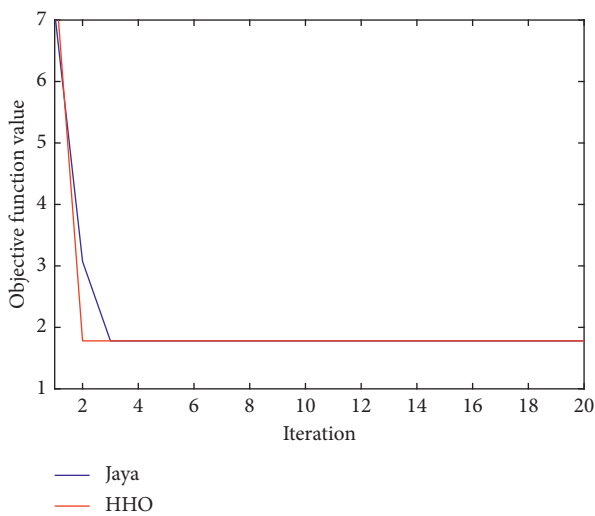


FIGURE 2: Convergence characteristics for the 3-bus system by LP formulation.

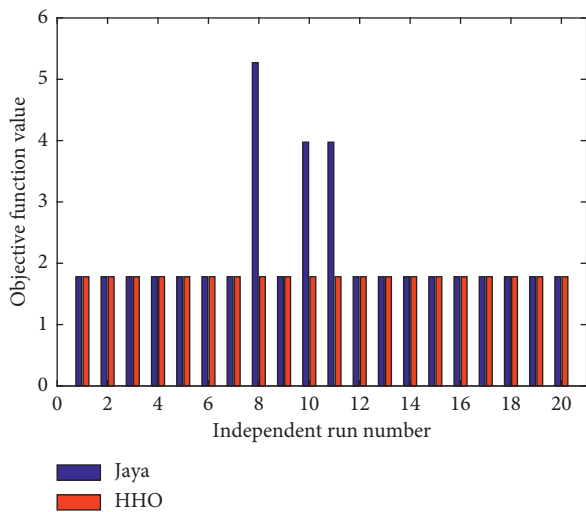


FIGURE 3: Independent runs for the 3-bus system by LP formulation.

TABLE 3: Operating time and CTI for the 3-bus system by LP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	5	0.3641	0.8873	0.5232	0.3641	0.8873	0.5232
2	4	0.2094	0.8465	0.6371	0.2094	0.8465	0.6371
3	1	0.3216	0.9633	0.6417	0.3216	0.9633	0.6417
4	6	0.3390	0.8202	0.4812	0.3390	0.8202	0.4812
5	3	0.2319	1.0661	0.8342	0.2319	1.0661	0.8342
6	2	0.3144	0.7842	0.4698	0.3144	0.7842	0.4698

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

Relay	GSO [18]	IGSO [18]	Analytic [41]	Jaya	HHO
TDS_1	0.100	0.100	0.100	0.100	0.100
TDS_2	0.100	0.100	0.100	0.100	0.100
TDS_3	0.100	0.100	0.100	0.1453	0.100
TDS_4	0.100	0.148	0.100	0.100	0.100
TDS_5	0.100	0.100	0.100	0.100	0.100
TDS_6	0.100	0.100	0.100	0.100	0.100
IP_1	161.6507	117.7491	162.00	90.00	90.00
IP_2	97.6290	29.995	85.00	119.12	119.12
IP_3	88.8888	62.9982	115.00	60.00	126.31
IP_4	133.0546	35.7398	140.00	107.046	165.65
IP_5	53.7883	36.7397	91.00	74.404	74.69
IP_6	134.8014	101.5602	140.00	120.00	120.00
Pop	—	—	—	20	20
Iter	—	—	—	50	50
Time	—	—	—	0.0225	0.1771
Std	—	—	—	2.2628	0.0995
OF	1.4807	1.2918	1.5108	1.5019	1.5157
Feasible	✓	✓	✓	✓	✓

initial values have been tried several times, but it was noticed that most of the results obtained by HHO did not satisfy the constraints or it kept on searching without finding any feasible solutions. The authors think this is because the big number of constraints in the 8-bus system results in big number of decision spaces, which is too hard for HHO to obtain the final solution under the condition of satisfying all

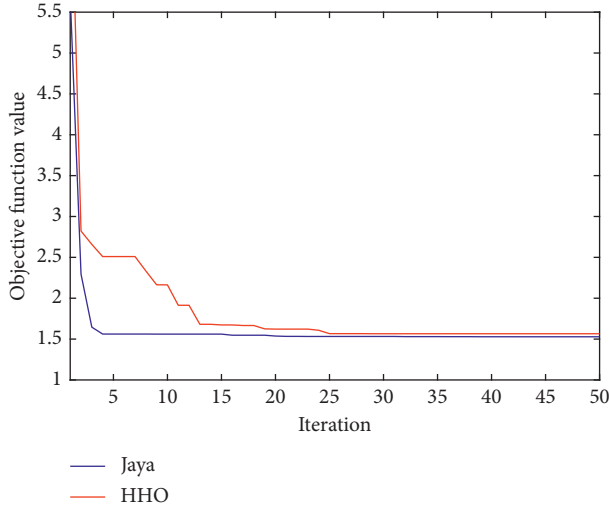


FIGURE 4: Convergence characteristics for the 3-bus system by NLP formulation.

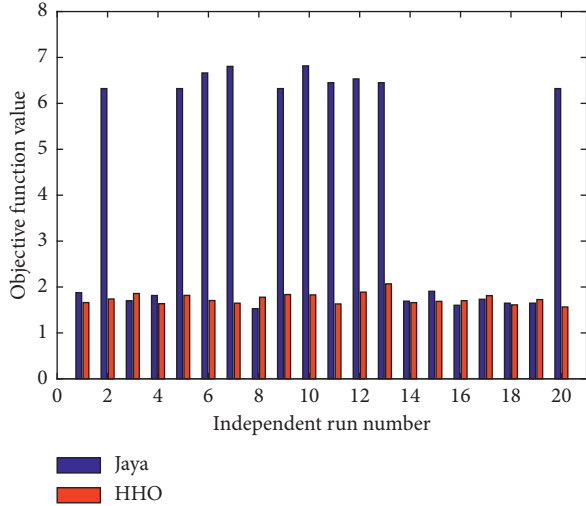


FIGURE 5: Independent runs for the 3-bus system by NLP formulation.

TABLE 5: Operating time and CTI for the 3-bus system by NLP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	5	0.2196	0.4196	0.2000	0.2196	0.4206	0.2010
2	4	0.2676	0.4688	0.2012	0.2676	0.6696	0.4020
3	1	0.2950	1.0457	0.7507	0.2633	1.0457	0.7824
4	6	0.2403	3.6468	3.4064	0.2854	3.6468	3.3613
5	3	0.2261	0.4263	0.2001	0.2264	0.4343	0.2078
6	2	0.2534	0.4534	0.2000	0.2534	0.4534	0.2000

the constraints. Another reason might be that HHO is highly dependent on the initial values; if the initial values are not generated properly (in this paper, all the initial values are generated randomly), it can not converge to the optimal solution or it ends with infeasible solution.

TABLE 6: Time dial setting and plug setting for the 3-bus system by MINLP formulation.

Relay	SBB [16]	Seeker [16]	BBO [14]	BBO-LP [14]	Jaya	HHO
TDS_1	0.1510	0.1070	0.1043	0.1067	0.100	0.100
TDS_2	0.1280	0.1080	0.1128	0.1083	0.100	0.100
TDS_3	0.1300	0.1000	0.1008	0.1000	0.100	0.100
TDS_4	0.1040	0.1000	0.1080	0.1000	0.1119	0.100
TDS_5	0.1060	0.1000	0.1008	0.1000	0.100	0.100
TDS_6	0.1000	0.1120	0.1030	0.1119	0.100	0.100
PS_1	1.5	2.5	3.0	2.5	1.5	1.5
PS_2	1.5	2.0	2.0	2.0	3.0	3.0
PS_3	2.0	3.0	3.0	3.0	5.0	3.5
PS_4	2.5	2.5	3.0	2.5	1.5	2.0
PS_5	2.5	2.5	2.5	2.5	2.0	2.0
PS_6	2.0	1.5	2.0	1.5	1.5	1.5
Pop	—	50	50	20	20	20
Iter	—	85	1000	20	50	50
Time	—	10.45	16.23	2.99	0.0286	0.1228
Std	—	—	—	—	1.7432	0.1030
OF	1.727	1.599	1.68375	1.59871	1.5477	1.4984
Feasible	✓	✓	✓	✓	✓	✓

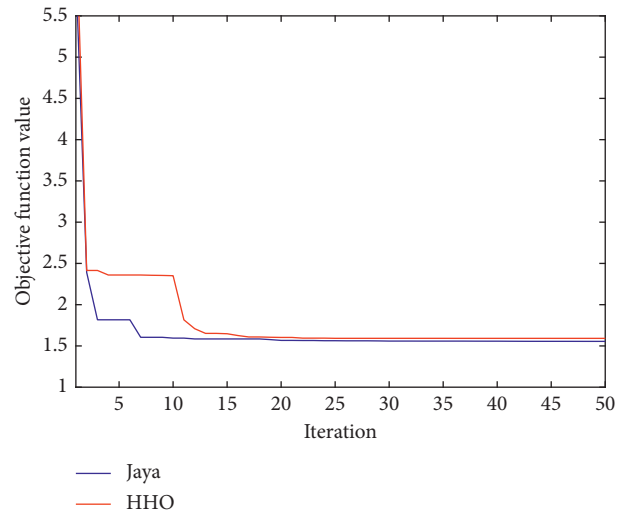


FIGURE 6: Convergence characteristics for the 3-bus system by MINLP formulation.

The convergence curve by Jaya (without HHO because it is unfeasible) is shown in Figure 9. It can be seen that Jaya reaches the optimal setting with no more than 200 iterations. The amplitudes of OF values are shown in Figure 10; it can be seen that the values fluctuate in quite large ranges, which means that the robustness of Jaya still needs to be improved further.

The operating times and CTI are given in Table 10. It is seen that the constraints are all respected by Jaya. But by HHO, there are several CTI values less than 0.3 (s) (underlined by bold format), which violate the constraint of CTI. This illustrates that the solution obtained by Jaya is feasible but that by HHO is unfeasible.

5.3.2. Case 5: 8-Bus System with MINLP Formulation. In this case,  $N_{pop} = 50$ ,  $Max\_iter = 2000$ , and  $N_{var} = 28$ . The

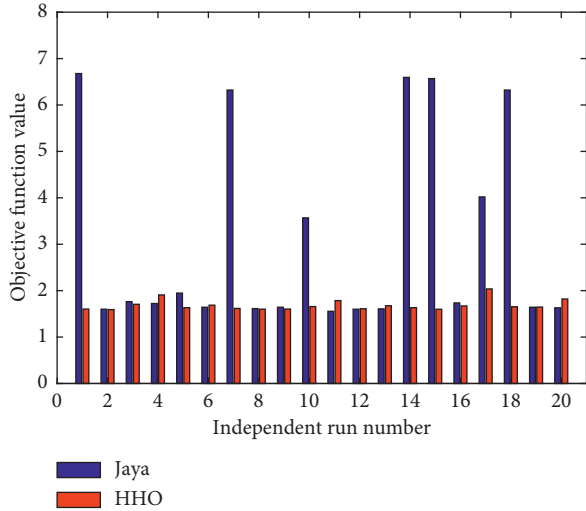


FIGURE 7: Independent runs for the 3-bus system by MINLP formulation.

TABLE 7: Operating time and CTI for the 3-bus system by MINLP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	5	0.2196	0.4393	0.2197	0.2196	0.4393	0.2197
2	4	0.2684	0.4685	0.2001	0.2684	0.5089	0.2405
3	1	0.3216	1.0457	0.7241	0.2745	1.0457	0.7712
4	6	0.2530	3.6468	3.3938	0.2507	3.6468	3.3960
5	3	0.2319	0.6142	0.3823	0.2319	0.4649	0.2330
6	2	0.2534	0.4556	0.2022	0.2534	0.4556	0.2022

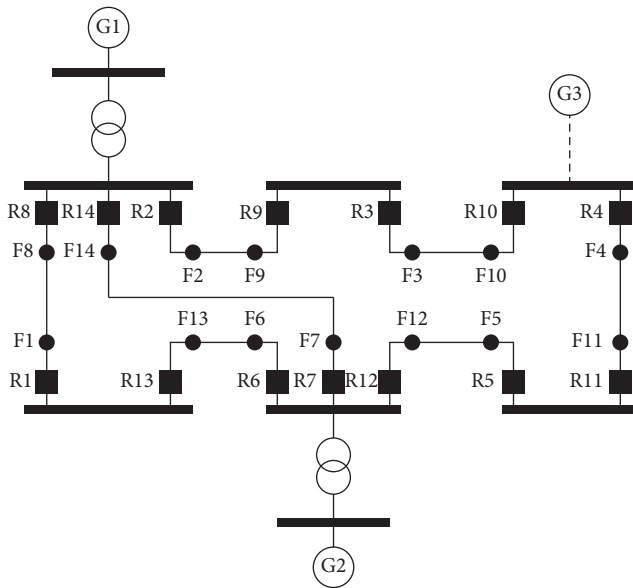


FIGURE 8: IEEE 8-bus DOCRs coordination problem model.

value of TDS is continuous, and PS is discrete from  $\{0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 2.5\}$ . Results obtained by Jaya and HHO are displayed in Table 11, and they are compared with

TABLE 8: P/B pairs and related parameters for the 8-bus system with NLP formulation [42].

Relay no.	Primary relay CT	IF(A) (primary)	Backup relay	IF(A) (backup)
1	1200/5	2666.3	6	2666.3
2	1200/5	5374.8	1	804.7
2	1200/5	5374.8	7	1531.5
3	800/5	3325.6	2	3325.6
4	1200/5	2217.1	3	2217.1
5	1200/5	1334.3	4	1334.3
6	1200/5	4975	5	403.6
6	1200/5	4975	14	1533
7	800/5	4247.6	5	403.6
7	800/5	4247.6	13	805.5
8	1200/5	4973.2	7	1531.5
8	1200/5	4973.2	9	403.2
9	800/5	1420.9	10	1420.9
10	1200/5	2313.5	11	2313.5
11	1200/5	3474.3	12	3474.3
12	1200/5	5377	13	805.5
12	1200/5	5377	14	1533
13	1200/5	2475.7	8	2475.7
14	800/5	4246.4	1	804.7
14	800/5	4246.4	9	403.2

Seeker [16], GA [3], GA-LP [3], BBO [14], and BBO-LP [14]. Here, we should note that system data in this case are different from Case 4, as shown in Table 12, even though they both are 8-bus systems.

Because this case is a highly constrained network with limited number of discrete PS values, it can not get feasible and optimal solutions easily. As shown in Table 11, GA and GA-LP are not capable of achieving feasible solutions, which is also mentioned in [14]. HHO suffers the same problem, but Jaya is able to obtain its optimal result as 10.2325 (s). This illustrates that, in the 8-bus system, no matter PS is continuous or discrete, HHO lacks ability of finding feasible solutions, but Jaya is always able to reach its optima. The reason was explained in Case 4, so it is not repeated here. The convergence behaviour by Jaya is represented in Figure 11, and the distribution of OF value is shown in Figure 12. Table 13 shows the operating time and CTI.

5.4. Case 6: 9-Bus System. This case is modeled as an NLP problem. It is with one single-end fed and equal impedances for all of the lines, as shown in Figure 13. Parameters for

TABLE 9: Time dial setting and pickup current for the 8-bus system by NLP formulation.

Relay	EFO [38]	MEFO [38]	BH [38]	EM [38]	HS [38]	PSO [38]	Jaya	HHO
TDS_1	0.197	0.344	0.299	0.430	0.251	0.077	0.0877	0.1113
TDS_2	0.348	0.263	0.544	0.635	0.412	0.050	0.4291	0.3972
TDS_3	0.285	0.317	0.475	0.622	0.434	0.271	0.3175	0.2821
TDS_4	0.212	0.093	0.295	0.631	0.199	0.110	0.0768	0.0500
TDS_5	0.076	0.080	0.208	0.306	0.156	0.050	0.0500	0.0500
TDS_6	0.329	0.286	0.420	0.608	0.371	0.309	0.2978	0.2801
TDS_7	0.273	0.490	0.448	0.566	0.450	0.320	0.2424	0.2490
TDS_8	0.303	0.456	0.363	0.546	0.425	0.219	0.1624	0.2789
TDS_9	0.098	0.292	0.237	0.348	0.179	0.050	0.2743	0.1025
TDS_10	0.174	0.175	0.264	0.663	0.246	0.050	0.1983	0.1678
TDS_11	0.263	0.319	0.367	0.515	0.348	0.200	0.2309	0.1809
TDS_12	0.343	0.443	0.496	0.723	0.432	0.345	0.3176	0.3419
TDS_13	0.202	0.263	0.278	0.488	0.285	0.152	0.0892	0.1075
TDS_14	0.311	0.548	0.397	0.455	0.428	0.309	0.4865	0.3832
IP_1	237.11	120.00	228.31	163.31	241.57	480.00	479.23	407.14
IP_2	293.32	480.00	240.79	303.26	385.67	120.00	120.00	120.41
IP_3	190.79	110.85	173.52	209.59	146.58	104.36	80.00	80.00
IP_4	120.01	480.00	269.15	149.17	293.48	318.24	480.00	365.11
IP_5	235.93	249.10	192.64	147.02	200.00	297.61	278.62	302.33
IP_6	193.40	414.49	203.05	216.38	228.30	120.00	120.10	179.11
IP_7	283.04	80.00	196.24	209.12	208.13	320.00	320.00	292.08
IP_8	237.31	120.03	288.15	272.14	285.95	480.00	479.33	183.41
IP_9	207.89	80.00	127.96	116.53	177.36	278.30	80.00	204.14
IP_10	225.25	418.41	257.61	146.32	215.11	268.71	336.14	224.18
IP_11	250.91	264.06	249.57	363.46	254.75	120.00	480.00	411.62
IP_12	319.59	248.50	237.60	240.59	404.57	127.16	480.00	254.74
IP_13	233.39	194.45	239.78	141.82	206.76	451.17	480.00	478.32
IP_14	230.26	83.90	216.33	260.14	223.33	154.82	96.64	130.16
Pop	—	—	—	—	—	—	50	50
Iter	—	—	—	—	—	—	1000	1000
Time	—	—	—	—	—	—	1.4741	1.9669
Std	—	—	—	—	—	—	3.0687	3.3709
OF	7.611	6.349	11.401	15.913	11.760	10.421	8.1996	7.2108
Feasible	✓	✓	✓	✓	✓	✓	✓	✗

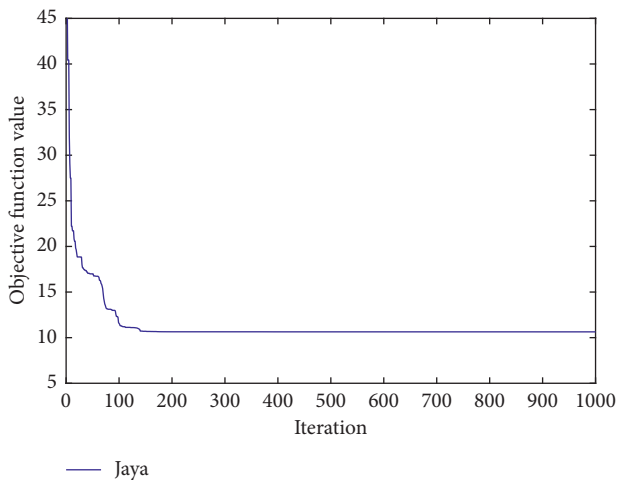


FIGURE 9: Convergence characteristics for the 8-bus system by NLP formulation.

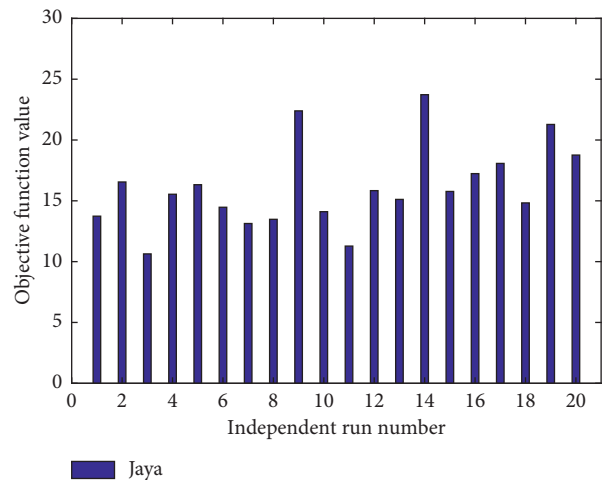


FIGURE 10: Independent runs for the 8-bus system by NLP formulation.

HHO and Jaya are set as follows:  $N_{pop} = 30$ ,  $Max\_iter = 200$ , and  $N_{var} = 48$ . The system has  $3\phi$  fault at the midpoint of each line. The P/B pairs, the fault current, and the maximum and minimum fault current are given in

Table 14. All the DOCRs have the same CT ratio of 500:1; the CTI is selected to be 0.2 s, and the design variables of TMS and PS are considered continuous. There is no backup relay for relays {17, 19, 21, 23}. The minimum operating time

TABLE 10: Operating time and CTI for the 8-bus system by NLP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	6	0.3517	0.6519	0.3001	0.4068	0.7066	<b>0.2998</b>
2	1	0.7604	1.1789	0.4185	0.7045	1.1357	0.4312
2	7	0.7604	1.0669	0.3065	0.7045	1.0346	0.3301
3	2	0.5744	0.8745	0.3001	0.5103	0.8103	0.3001
4	3	0.3458	0.6471	0.3013	0.1906	0.5749	0.3844
5	4	0.2200	0.5202	0.3002	0.2323	0.2666	<b>0.0343</b>
6	5	0.5393	0.9409	0.4017	0.5704	1.2079	0.6375
6	14	0.5393	1.1984	0.6591	0.5704	1.0611	0.4906
7	5	0.6394	0.9409	0.3015	0.6338	1.2079	0.5741
7	13	0.6394	1.2003	0.5609	0.6338	1.4363	0.8025
8	7	0.4746	1.0669	0.5923	0.5723	1.0346	0.4623
8	9	0.4746	1.1679	0.6933	0.5723	1.0470	0.4748
9	10	0.6483	0.9493	0.3010	0.3627	0.6244	<b>0.2617</b>
10	11	0.7059	1.0117	0.3057	0.4916	0.7209	<b>0.2293</b>
11	12	0.8006	1.1011	0.3005	0.5811	0.8922	0.3111
12	13	0.8981	1.2003	0.3022	0.7611	1.4363	0.6752
12	14	0.8981	1.1984	0.3003	0.7611	1.0611	0.3000
13	8	0.3745	0.6810	0.3065	0.4502	0.7308	<b>0.2805</b>
14	1	0.8667	1.1789	0.3122	0.7432	1.1357	0.3926
14	9	0.8667	1.1679	0.3012	0.7432	1.0470	0.3039

TABLE 11: Time dial setting and pickup current for the 8-bus system by MINLP formulation.

Relay	Seeker [16]	GA [3]	GA-LP [3]	BBO [14]	BBO-LP [14]	Jaya	HHO
TDS_1	0.113	0.29	0.3043	0.14239	0.10000	0.1000	0.1000
TDS_2	0.260	0.31	0.2917	0.38159	0.30329	0.4409	0.2484
TDS_3	0.225	0.26	0.2543	0.29326	0.23579	0.4585	0.2691
TDS_4	0.160	0.19	0.1851	0.22081	0.22043	0.1900	0.1000
TDS_5	0.100	0.18	0.1700	0.11834	0.10000	0.1030	0.1000
TDS_6	0.173	0.26	0.2711	0.24271	0.17340	0.3447	0.3484
TDS_7	0.243	0.54	0.5316	0.32704	0.25321	0.2776	0.2901
TDS_8	0.170	0.24	0.2387	0.21560	0.34805	0.2638	0.1616
TDS_9	0.147	0.17	0.1865	0.22933	0.14866	0.2482	0.1000
TDS_10	0.176	0.19	0.1895	0.31500	0.20519	0.3507	0.1050
TDS_11	0.187	0.21	0.2014	0.27815	0.19198	0.2665	0.1636
TDS_12	0.266	0.30	0.2890	0.36940	0.27146	0.3163	0.2402
TDS_13	0.114	0.23	0.2207	0.10363	0.11540	0.2555	0.1000
TDS_14	0.246	0.51	0.5278	0.35736	0.24928	0.3205	0.2043
PS_1	2.00	1.00	2.00	2.50	2.00	2.50	2.50
PS_2	2.50	2.50	1.50	2.00	2.00	1.00	2.50
PS_3	2.50	2.50	2.50	2.00	2.50	0.50	1.50
PS_4	2.50	2.50	2.50	2.00	1.50	2.00	2.50
PS_5	2.50	1.50	1.50	2.50	2.50	2.50	1.00
PS_6	2.50	2.50	2.50	2.00	2.50	0.50	0.50
PS_7	2.50	0.50	0.50	2.00	2.50	2.50	1.50
PS_8	2.50	2.50	2.50	2.00	0.50	2.00	2.50
PS_9	2.50	2.00	2.00	2.00	2.50	1.50	2.50
PS_10	2.50	2.50	2.50	1.50	2.00	1.00	2.50
PS_11	2.50	2.50	2.50	2.00	2.50	2.00	2.50
PS_12	2.50	2.50	2.50	2.00	2.50	2.50	2.50
PS_13	2.00	1.50	1.50	2.50	2.00	1.00	2.50
PS_14	2.50	0.50	0.50	2.00	2.50	2.00	2.50
Pop	100	100	100	50	20	50	50
Iter	—	100000	30	10000	30	2000	2000
Time	—	36000	300	2065.02	6.79	2.1031	14.6635
Std	—	—	—	—	—	3.4972	2.9254
OF	8.4270	11.001	10.9499	10.5495	8.7556	10.2325	7.2849
Feasible	✓	×	×	✓	✓	✓	×

TABLE 12: P/B pairs and related parameters for the 8-bus system with MINLP formulation [3, 16].

Relay no.	Primary relay CT	IF(A) (primary)	Backup relay	IF(A) (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

of each relay ( $T_i^{\min}$ ) is taken as 0.2 s. For each relay, the minimum and maximum values of TMS are 0.025 and 1.2 and the minimum and maximum values of PS are calculated by the following equations:

$$\begin{aligned} PS_{\min}^i &= \frac{I_{n,i} \times \text{OLF}}{\text{CTR}}, \\ PS_{\max}^i &= I_{f,i}^{\min} \times \frac{2}{3\text{CTR}}, \end{aligned} \quad (32)$$

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, and  $I_{f,i}^{\min}$  is the minimum fault current detected by  $R_i$ .

The optimum settings of TDS and PS are presented in Table 15. It is noticed that the best result is obtained by GANLP as 6.1786 (s), followed by HHO and Jaya with values as 7.0297 (s) and 7.1378 (s), respectively. No feasible solutions can be found by NLP.

The convergence characteristics could be seen in Figure 14, from which we can observe that HHO converges a little bit faster than Jaya and obtained slightly lower OF value

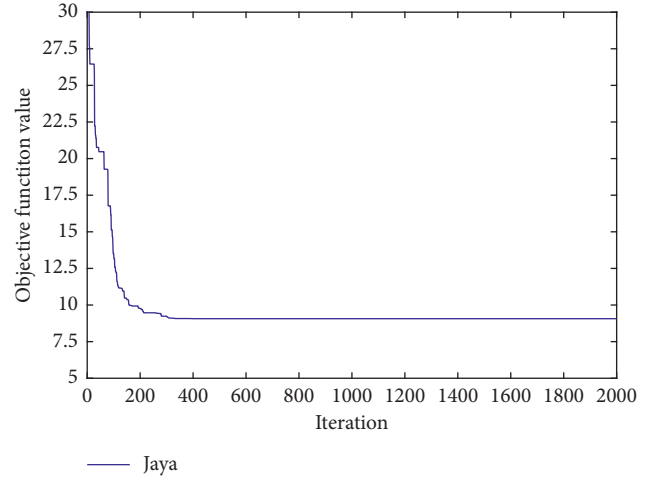


FIGURE 11: Convergence characteristics for the 8-bus system by MINLP formulation.

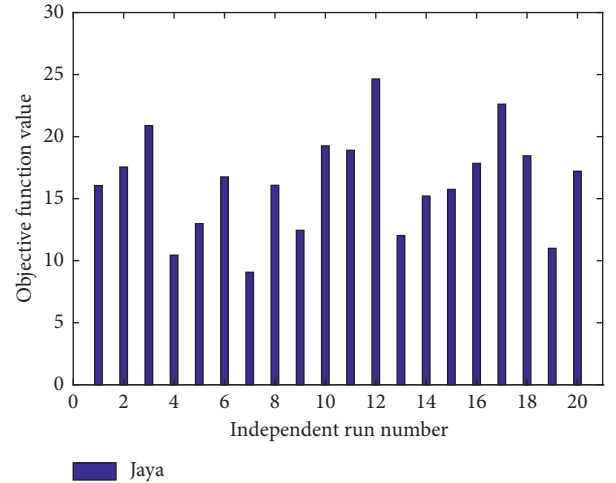


FIGURE 12: Independent runs for the 8-bus system by MINLP formulation.

as well. But generally speaking, there is not much difference between them. However, from Figure 15, we can see the differences become obvious. Among 20 times of independent runs, HHO shows much stronger ability in maintaining the minimum value (with Std equal to 1.9196), while Jaya suffers several times of premature problem (with Std equal to 2.7335).

Table 16 shows the operating time and CTI; we can see that there is no selectivity constraint violated by both Jaya and HHO.

**5.5. Case 7: 4-Bus System.** Different from 3-bus, 8-bus, and 9-bus systems, in the 4-bus system, both near-end and far-end  $3\phi$  fault locations are considered, as shown in Figure 16. The network consists of 4 buses, 4 branches, and 8 DOCRs, and it is formulated as an NLP problem. Parameters for HHO and Jaya are set as follows:  $N_{\text{pop}} = 50$ ,  $\text{Max\_iter} = 2000$ , and  $N_{\text{var}} = 16$ .

TABLE 13: Operating time and CTI for the 8-bus system by MINLP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	6	0.4087	0.7087	0.3000	0.4087	0.7164	0.3077
2	1	0.9321	1.3742	0.4421	0.7421	1.3742	0.6321
2	7	0.9321	1.2321	0.3000	0.7421	0.9638	<b>0.2217</b>
3	2	0.8142	1.1142	0.3000	0.6801	0.9599	<b>0.2798</b>
4	3	0.6310	0.9310	0.3000	0.3732	0.8240	0.4508
5	4	0.5129	0.8129	0.3000	0.2970	0.4978	<b>0.2008</b>
6	5	0.5901	1.0370	0.4469	0.5965	0.4286	<b>-0.1678</b>
6	14	0.5901	1.2468	0.6567	0.5965	0.9118	0.3153
7	5	0.7370	1.0370	0.3000	0.6392	0.4286	<b>-0.2105</b>
7	13	0.7370	1.2468	0.5098	0.6392	1.3994	0.7602
8	7	0.7085	1.2321	0.5236	0.4768	0.9638	0.4871
8	9	0.7085	1.0824	0.3739	0.4768	0.6478	<b>0.1710</b>
9	10	0.7262	1.0262	0.3000	0.3764	0.5100	<b>0.1337</b>
10	11	0.8576	1.1576	0.3000	0.3863	0.8290	0.4427
11	12	0.8939	1.1939	0.3000	0.6175	0.9066	<b>0.2891</b>
12	13	0.9468	1.2468	0.3000	0.7190	1.3994	0.6804
12	14	0.9468	1.2468	0.3000	0.7190	0.9118	<b>0.1928</b>
13	8	0.6911	0.9911	0.3000	0.4288	0.6929	<b>0.2641</b>
14	1	0.7824	1.3742	0.5918	0.5434	1.3742	0.8308
14	9	0.7824	1.0824	0.3000	0.5434	0.6478	<b>0.1044</b>

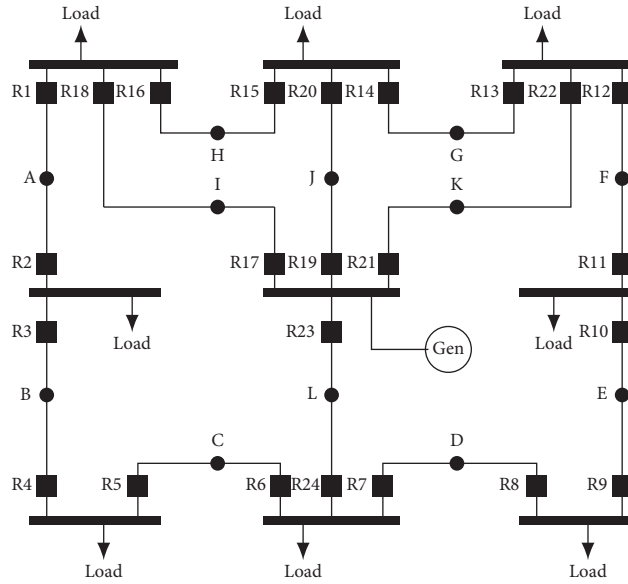


FIGURE 13: IEEE 9-bus DOCRs coordination problem model.

The objective function and the selectivity constraint of the system are mathematically expressed as follows:

$$\text{OF} = \sum_{p=1}^m T_{pr,p}^{\text{near}} + \sum_{q=1}^n T_{pr,q}^{\text{far}}, \quad (33)$$

$$T_{jk} - T_{ik} \geq \text{CTI},$$

where  $T_{pr,p}^{\text{near}}$  and  $T_{pr,q}^{\text{far}}$  are the operating time of the primary relay at the near-end  $3\phi$  fault (at the  $p^{\text{th}}$  location) and the far-end  $3\phi$  fault (at the  $q^{\text{th}}$  location), respectively and  $T_{jk}$  and  $T_{ik}$  are the operating time of the  $j^{\text{th}}$  backup and  $i^{\text{th}}$  primary relay for a  $3\phi$  fault which happens at the  $k^{\text{th}}$  location, respectively. They can be computed by the following equations:

TABLE 14: Related parameters for the 9-bus system [4].

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

TABLE 15: Time dial setting and pickup current for the 9-bus system by NLP formulation.

Relay	GA [4]	NLP [4]	GA-NLP [4]	DE [43]	HS [43]	SOA [43]	Jaya	HHO
TDS_1	0.4766	0.0010	0.0805	0.1241	0.1447	0.2662	0.0635	0.0859
TDS_2	0.0711	0.0010	0.0266	0.1000	0.1000	0.2076	0.0521	0.0682
TDS_3	0.3522	0.0362	0.0560	0.1370	0.1684	0.2928	0.0863	0.0989
TDS_4	0.3245	0.0131	0.0492	0.1089	0.1138	0.3192	0.0958	0.0733
TDS_5	0.2738	0.0643	0.0472	0.1237	0.1309	0.2879	0.0559	0.1064
TDS_6	0.3982	0.0203	0.0764	0.1277	0.1384	0.3677	0.0720	0.1034
TDS_7	0.6148	0.0203	0.0764	0.1277	0.1388	0.3006	0.0541	0.0780
TDS_8	0.6793	0.0251	0.0472	0.1237	0.1300	0.2905	0.0800	0.0868
TDS_9	0.3337	0.0131	0.0492	0.1089	0.1212	0.2476	0.0765	0.0507
TDS_10	0.7991	0.0391	0.0557	0.1370	0.1598	0.2480	0.0528	0.1089
TDS_11	0.2139	0.0010	0.0305	0.1000	0.1000	0.2578	0.0809	0.0948
TDS_12	0.7484	0.0010	0.0802	0.1241	0.1393	0.3665	0.0789	0.1056
TDS_13	0.6977	0.0010	0.0492	0.1000	0.1021	0.2581	0.1154	0.0783
TDS_14	0.7254	0.0062	0.0637	0.1090	0.1141	0.3117	0.0662	0.0735
TDS_15	0.6425	0.0062	0.0639	0.1090	0.1165	0.2921	0.1170	0.1492
TDS_16	0.3107	0.0010	0.0593	0.1000	0.1183	0.3633	0.1638	0.0731
TDS_17	0.7162	1.2000	0.0974	0.1000	0.1000	0.2560	0.0749	0.0840
TDS_18	0.1264	0.0016	0.0295	0.1000	0.1002	0.1038	0.0385	0.0543
TDS_19	0.5226	1.2000	0.0787	0.1000	0.1292	0.2589	0.0709	0.1513
TDS_20	0.1217	0.0074	0.0964	0.1000	0.1001	0.1002	0.0515	0.0517
TDS_21	0.7622	0.7669	0.0972	0.1000	0.1000	0.2758	0.0939	0.0857
TDS_22	0.0803	0.0016	0.0921	0.1000	0.1002	0.1010	0.0414	0.0453
TDS_23	0.8544	1.2000	0.1011	0.1000	0.1334	0.1757	0.0922	0.1022
TDS_24	0.2508	0.0108	0.0435	0.1000	0.1002	0.1014	0.0411	0.0340
PS_1	1.4304	9.0720	1.8150	2.5000	2.1741	1.2732	1.5008	1.1338
PS_2	1.3060	6.5540	1.2988	2.0899	2.2940	1.5200	0.7306	0.5319
PS_3	0.8872	1.0687	1.4980	2.5000	1.8739	1.1975	0.9324	1.0803
PS_4	0.5179	6.9794	1.3920	2.5000	2.4472	0.6701	0.6729	1.0030
PS_5	0.5579	0.1760	0.9480	2.5000	2.4175	1.0785	0.7770	0.5379
PS_6	0.7942	8.1739	1.6430	2.5000	2.2897	0.6311	1.3588	1.1391
PS_7	0.2566	8.1739	1.6430	2.5000	2.3249	0.9637	1.3636	1.1409



TABLE 15: Continued.

Relay	GA [4]	NLP [4]	GA-NLP [4]	DE [43]	HS [43]	SOA [43]	Jaya	HHO
PS_8	0.2792	0.6555	0.9480	2.5000	2.4176	1.1393	0.7805	0.5380
PS_9	0.7516	6.9794	1.3920	2.5000	2.2509	1.1994	0.9438	1.0029
PS_10	0.3578	0.7596	1.4980	2.5000	2.0335	1.7451	1.2611	0.5525
PS_11	0.7855	6.5540	1.1369	2.0899	2.3288	0.8454	0.6985	0.3232
PS_12	1.3179	9.0940	1.8150	2.5000	2.2590	0.6461	0.8530	0.5352
PS_13	0.4762	6.8778	1.3740	2.2969	2.3465	0.9784	0.5695	0.9908
PS_14	0.7903	7.7996	1.5560	2.5000	2.4932	0.8860	1.3133	1.1221
PS_15	0.3914	7.7838	1.5560	2.5000	2.4666	0.8993	1.0621	1.1214
PS_16	0.6788	6.8778	0.9639	2.2969	1.9360	0.5004	0.3827	0.9907
PS_17	1.1122	1.7200	1.7200	2.1606	2.3568	0.9197	1.4577	1.1033
PS_18	0.4918	1.8715	1.6347	0.5000	0.6198	0.5003	2.0809	1.1033
PS_19	1.4124	1.6800	1.6800	1.6462	1.2409	0.7629	1.4125	1.0272
PS_20	1.9569	2.3447	0.2006	0.5000	0.7395	0.5041	1.9831	1.1532
PS_21	1.3553	1.4735	1.7200	2.1606	2.4717	0.8902	1.4581	1.1033
PS_22	0.5859	1.8715	0.2000	0.5000	0.7203	0.5008	1.9307	1.1033
PS_23	1.3365	1.7900	1.7900	1.9435	1.4234	1.5724	1.5465	1.2663
PS_24	0.2067	0.9989	0.7441	0.5000	0.5572	0.5017	1.6236	1.2663
Pop	—	—	—	—	—	—	30	30
Iter	100	—	—	—	—	—	200	200
Time	—	—	—	7.29	122.15	30.20	0.6016	0.8849
Std	—	—	—	0.1233	0.2152	1.2133	2.7335	1.9196
OF	32.6058	19.4041	6.1786	8.6822	9.2339	14.2338	7.1378	7.0297
Feasible	✓	✗	✓	✓	✓	✓	✓	✓

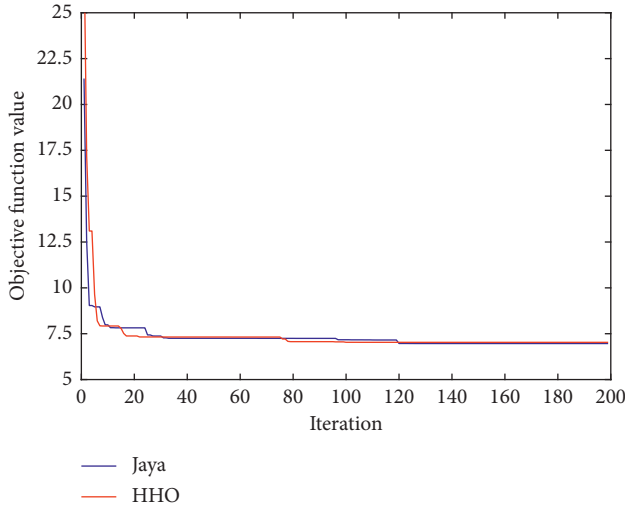


FIGURE 14: Convergence characteristics for the 9-bus system by NLP formulation.

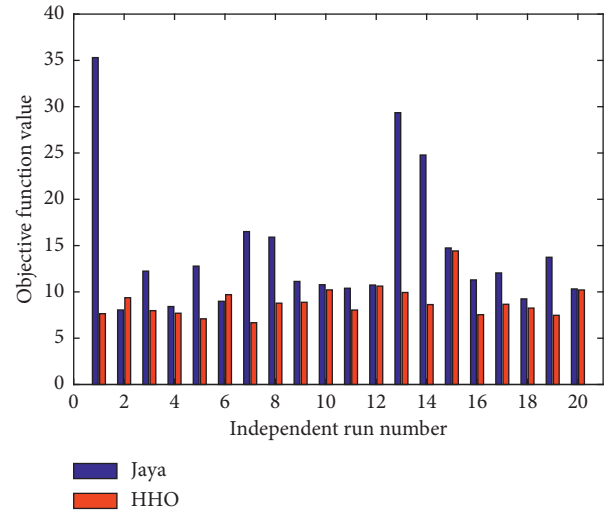


FIGURE 15: Independent runs for the 9-bus system by NLP formulation.

$$\begin{aligned}
T_{pr,p}^{\text{near}} &= \text{TDS}_p \times \frac{0.14}{\left[ a_p / (\text{PS}_p \times b_p) \right]^{0.02} - 1}, \\
T_{pr,q}^{\text{far}} &= \text{TDS}_q \times \frac{0.14}{\left[ c_p / (\text{PS}_q \times d_p) \right]^{0.02} - 1}, \\
T_{jk} &= \text{TDS}_j \times \frac{0.14}{\left[ e_p / (\text{PS}_j \times f_p) \right]^{0.02} - 1}, \\
T_{ik} &= \text{TDS}_i \times \frac{0.14}{\left[ g_p / (\text{PS}_i \times h_p) \right]^{0.02} - 1}.
\end{aligned} \tag{34}$$

In this case, TDS and PS are considered continuous, while TDS lies in  $[0.05, 1.1]$  and PS lies in  $[1.25, 1.5]$ . The CTI is selected to be 0.3 (s). Related constants are given in Tables 17 and 18. The primary operating time of each relay ( $T_i$ ) should be bounded between in  $[0.05, 1.0]$ .

Table 19 shows the optimized results by Jaya and HHO, as well as TLBO [9], TLBO(MOF) [9], DE [5], MDE [5], BBO [14], and BBO-LP [14]. Although MDE ranks the first place as 3.6694 (s), it failed to achieve a feasible solution because there are some constraints unsatisfied, as mentioned in [9]. However, Jaya and HHO are able to achieve feasible solutions with no violations in every independent

TABLE 16: Operating time and CTI for the 9-bus system by NLP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	15	0.2334	1.0302	0.7968	0.2739	1.4118	1.1379
1	17	0.2334	0.9082	0.6748	0.2739	0.6835	0.4096
2	4	0.2397	0.5852	0.3455	0.2584	0.6974	0.4364
3	1	0.3304	0.7415	0.4111	0.4128	0.6805	0.2677
4	6	0.3205	0.8487	0.5281	0.3060	0.9369	0.6308
5	3	0.2535	0.6806	0.4270	0.3869	0.9373	0.5504
6	8	0.2655	0.9275	0.6621	0.3477	0.6189	0.2712
6	23	0.2655	1.1591	0.8937	0.3477	0.9415	0.5938
7	5	0.2000	0.6436	0.4436	0.2625	0.7584	0.4958
7	23	0.2000	1.1591	0.9591	0.2625	0.9415	0.6790
8	10	0.3637	0.6357	0.2719	0.3157	0.5357	0.2200
9	7	0.3077	0.6421	0.3343	0.2115	0.7083	0.4968
10	12	0.2437	0.8954	0.6517	0.3211	0.6778	0.3566
11	9	0.3612	0.6689	0.3077	0.2801	0.4801	0.2000
12	14	0.2873	0.7992	0.5119	0.3070	0.6963	0.3893
12	21	0.2873	1.1386	0.8513	0.3070	0.6975	0.3905
13	11	0.3076	0.8978	0.5902	0.2677	0.4681	0.2004
13	21	0.3076	1.1386	0.8311	0.2677	0.6975	0.4298
14	16	0.2459	0.6690	0.4232	0.2513	0.6921	0.4408
14	19	0.2459	0.8477	0.6019	0.2513	1.1654	0.9141
15	13	0.3890	0.6196	0.2305	0.5098	0.7416	0.2318
15	19	0.3890	0.8477	0.4587	0.5098	1.1654	0.6556
16	2	0.3762	0.6230	0.2468	0.2498	0.5265	0.2766
16	17	0.3762	0.9082	0.5319	0.2498	0.6835	0.4337
17	—	—	—	—	—	—	—
18	2	0.3429	0.6230	0.2802	0.2650	0.5265	0.2615
18	15	0.3429	1.0302	0.6873	0.2650	1.4118	1.1468
19	—	—	—	—	—	—	—
20	13	0.3665	0.6196	0.2530	0.2350	0.7416	0.5066
20	16	0.3665	0.6690	0.3025	0.2350	0.6921	0.4571
21	—	—	—	—	—	—	—
22	11	0.3360	0.8978	0.5617	0.2209	0.4681	0.2471
22	14	0.3360	0.7992	0.4632	0.2209	0.6963	0.4754
23	—	—	—	—	—	—	—
24	5	0.3979	0.6436	0.2456	0.2441	0.7584	0.5143
24	8	0.3979	0.9275	0.5296	0.2441	0.6189	0.3748

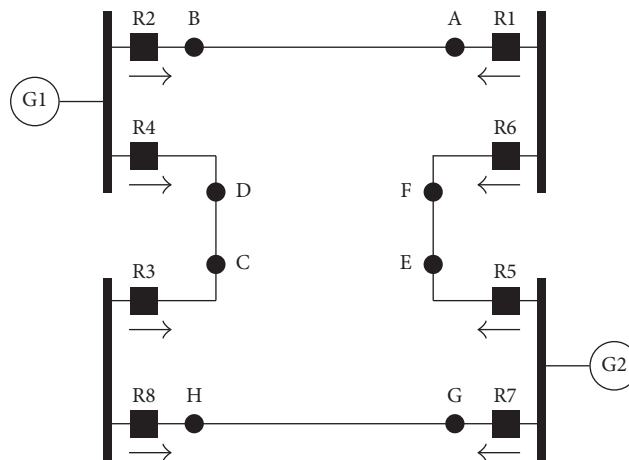


FIGURE 16: IEEE 4-bus DOCRs coordination problem model.

TABLE 17: Constant values of  $a_i$ ,  $b_i$ ,  $c_i$ , and  $d_i$  for the 4-bus system [5, 9].

TDS <sub><i>i</i></sub>	$T_{\text{pri\_cl\_in}}^i$	$b_i$	TDS <sub><i>i</i></sub>	$T_{\text{pri\_far\_bus}}^i$	$d_i$
	$a_i$			$c_i$	
TDS_1	20.32	0.48	TDS_2	23.75	0.48
TDS_2	88.85	0.48	TDS_1	12.48	0.48
TDS_3	13.61	1.1789	TDS_4	31.92	1.1789
TDS_4	116.81	1.1789	TDS_3	10.38	1.1789
TDS_5	116.70	1.5259	TDS_6	12.07	1.5259
TDS_6	16.67	1.5259	TDS_5	31.92	1.5259
TDS_7	71.70	1.2018	TDS_8	11.00	1.2018
TDS_8	19.27	1.2018	TDS_7	18.91	1.2018

TABLE 18: Constant values of  $e_i$ ,  $f_i$ ,  $g_i$ , and  $h_i$  for the 4-bus system [5, 9].

$P$	$T_{\text{backup}}^i$	$f_i$	$q$	$T_{\text{primary}}^i$	$h_i$
	$e_i$			$g_i$	
5	20.32	1.5259	1	20.32	0.48
5	12.48	1.5259	1	12.48	0.48
7	13.61	1.2018	3	13.61	1.1789
7	10.38	1.2018	3	10.38	1.1789
1	1.16	0.48	4	1.16	1.1789
2	12.07	0.48	6	12.07	1.1789
2	16.67	0.48	6	16.67	1.5259
4	11.00	1.1789	8	11.00	1.2018
4	19.27	1.1789	8	19.27	1.2018

TABLE 19: Time dial setting and pickup current for the 4-bus system by NLP formulation.

Relay	TLBO [9]	TLBO (MOF) [9]	DE [5]	MDE [5]	BBO [14]	BBO-LP [14]	Jaya	HHO
TDS_1	0.2348	0.1870	0.0500	0.0500	0.0537	0.0500	0.0500	0.0500
TDS_2	0.5023	0.2294	0.2248	0.2121	0.2257	0.2123	0.2122	0.2297
TDS_3	0.1897	0.3394	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500
TDS_4	0.2560	0.4769	0.1515	0.1515	0.1698	0.1516	0.1539	0.1634
TDS_5	0.3572	0.2661	0.1264	0.1264	0.1439	0.1264	0.1267	0.1420
TDS_6	0.0431	0.0359	0.0500	0.0500	0.0501	0.0500	0.0500	0.0500
TDS_7	0.3996	0.4977	0.1337	0.1337	0.1494	0.1339	0.1350	0.1467
TDS_8	0.0128	0.3043	0.0500	0.0500	0.0540	0.0500	0.0500	0.0500
PS_1	0.3560	0.6398	1.2734	1.2734	1.2631	1.2736	1.3207	1.2969
PS_2	0.2586	0.6440	1.2500	1.4999	1.2501	1.4969	1.5000	1.2500
PS_3	0.2963	0.6479	1.2500	1.2500	1.2549	1.2503	1.3238	1.2500
PS_4	0.3652	0.6168	1.4997	1.4999	1.2503	1.4998	1.4760	1.2500
PS_5	0.2884	0.6606	1.4997	1.5000	1.2522	1.4989	1.5000	1.2500
PS_6	0.3144	0.6563	1.2500	1.2500	1.2529	1.2501	1.2500	1.2500
PS_7	0.2488	0.6237	1.5000	1.5000	1.2581	1.4981	1.5000	1.2500
PS_8	0.2739	0.6530	1.2500	1.2500	1.2599	1.2501	1.3142	1.2500
Pop	---	—	—	—	50	10	50	50
Iter	—	—	95400	35330	1000	1000	2000	2000
Time	—	—	—	—	—	—	2.9355	7.2515
Std	—	—	—	—	—	—	1.5749	0.1701
OF	5.5890	8.7088	3.6774	3.6694	3.8282	3.6698	3.7020	3.7539
Feasible	✓	✓	✓	✗	✓	✓	✓	✓

run time, even though they obtained higher OF values compared to MDE.

Convergence characteristic is represented in Figure 17. Jaya reaches its optima within 200 iterations, while HHO needs about 800 iterations. This proves the convergence rate

of Jaya is much faster than HHO. However, the robustness and consistency of Jaya is not as good as that of HHO, as shown in Figure 18.

Table 20 shows the operating time and CTI; we can see that CTI constraints are satisfied in all P/B pairs.

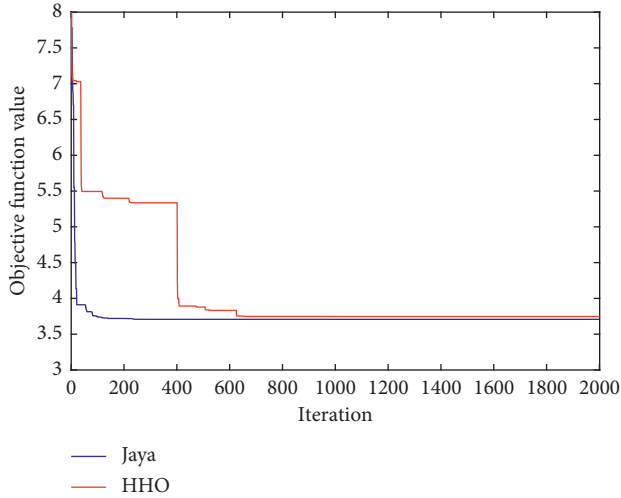


FIGURE 17: Convergence characteristics for the 4-bus system by NLP formulation.

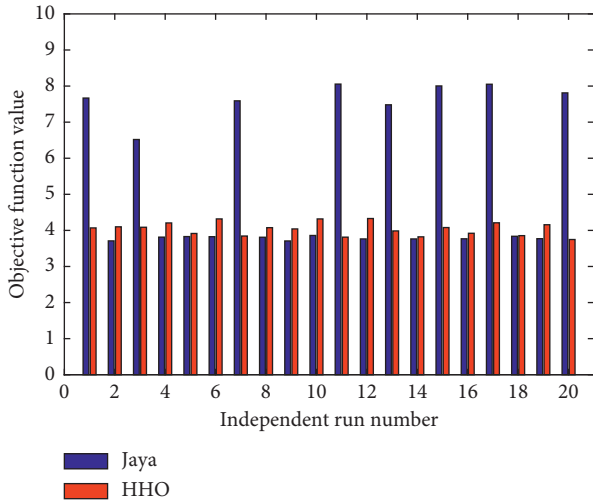


FIGURE 18: Independent runs for the 4-bus system by NLP formulation.

TABLE 20: Operating time and CTI for the 4-bus system by NLP formulation.

P/R	B/R	Jaya			HHO		
		Time	Time	CTI	Time	Time	CTI
1	5	0.0975	0.3975	0.3000	0.0970	0.4103	0.3134
1	5	0.1140	0.5141	0.4001	0.1133	0.5193	0.4061
3	7	0.1581	0.4581	0.3000	0.1540	0.4558	0.3018
3	7	0.1812	0.5304	0.3492	0.1758	0.5211	0.3453
4	1	0.2455	0.5758	0.3303	0.2504	0.5588	0.3085
6	2	0.1630	0.5122	0.3492	0.1630	0.5197	0.3567
6	2	0.1580	0.4580	0.3000	0.1580	0.4677	0.3097
8	4	0.1769	0.5735	0.3966	0.1723	0.5578	0.3855
8	4	0.1364	0.4373	0.3009	0.1337	0.4337	0.3000

## 6. Conclusion

This paper compares the performances of the lately proposed Harris Hawks Optimization (HHO) and Jaya optimization in solving the directional overcurrent relays (DOCRs) coordination problem. Especially HHO, which to the best of the authors' knowledge, is being used for the first time in the DOCRs coordination problem. Four test systems including 3-bus, 4-bus, 8-bus, and 9-bus (one LP case, four NLP cases, and two MINLP cases) are experimented. The conclusion is that the robustness and consistency of HHO is relatively better than Jaya, while Jaya provides faster convergence rate and occasionally more competitive objective function value than HHO.

It should be addressed that, in both Jaya and HHO, different population size ( $N_{pop}$ ) results in different solution quality. If  $N_{pop}$  is too small, the population diversity will be limited and will cause premature problem. On the other hand, if  $N_{pop}$  is too big, there will be unnecessary calculations which reduce the computational efficiency. In fact, when solving real-world problems, which we cannot know the specific scale for  $N_{pop}$ , there is no way to decide the most perfect value for  $N_{pop}$ . In this paper, we set the value of  $N_{pop}$  as 5, 20, 30, or 50, just for the convenience of comparison. However, the authors are thinking about a new way to determine  $N_{pop}$ , which may adopt a self-adaptive mechanism determined by the change strength of the population, without setting the  $N_{pop}$  value in advance.

It also worth mentioning that according to the No-Free-Lunch theorem, there is no perfect algorithms for solving all the optimization problems. It means that an algorithm may perform better than another algorithm in a set of problems, but may fail to perform better in another different set of problems. As in this paper, the authors compared the performances of HHO and Jaya in different test cases, and we can not theoretically conclude that HHO is superior to Jaya or inferior to Jaya because each of them has advantages and disadvantages and they can never be the universally-best optimizer.

In future works, two research directions will be studied. Firstly, what can be done to improve the robustness and consistency of Jaya without being trapped into local optima; secondly, how to accelerate the convergence rate of HHO as well as to achieve better objective function value. Moreover, larger test systems of the DOCRs coordination problem such as 15-bus, 30-bus, and 42-bus are supposed be studied in the following research, to deeper expand HHO and Jaya's applications in this field.

## Data Availability

All data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank Yeungnam University for all the supports in terms of fellowship to Jiangtao Yu.

## References

- [1] A. J. Urdeneta, R. Nadira, and L. G. Perez Jimenez, "Optimal coordination of directional overcurrent relays in interconnected power systems," *IEEE Transactions on Power Delivery*, vol. 3, no. 3, pp. 903–911, 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," *Electric Power Systems Research*, vol. 78, no. 4, pp. 713–720, 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 Transactions on Power Delivery*, vol. 24, no. 4, pp. 1857–1863, 2009.
- [4] P. P. Bedekar and S. R. Bhide, "Optimum coordination of directional overcurrent relays using the hybrid GA-NLP approach," *IEEE Transactions on Power Delivery*, vol. 26, no. 1, pp. 109–119, 2011.
- [5] R. Thangaraj, M. Pant, and K. Deep, "Optimal coordination of over-current relays using modified differential evolution algorithms," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 820–829, 2010.
- [6] T. R. Chelliah, R. Thangaraj, S. Allamsetty, and M. Pant, "Coordination of directional overcurrent relays using opposition based chaotic differential evolution algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 341–350, 2014.
- [7] 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 Transactions on Power Delivery*, vol. 22, no. 3, pp. 1400–1410, 2007.
- [8] H. H. Zeineldin, E. F. El-Saadany, and M. M. A. Salama, "Optimal coordination of overcurrent relays using a modified particle swarm optimization," *Electric Power Systems Research*, vol. 76, no. 11, pp. 988–995, 2006.
- [9] M. Singh, B. K. Panigrahi, and A. R. Abhyankar, "Optimal coordination of directional over-current relays using Teaching Learning-Based Optimization (TLBO) algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 50, pp. 33–41, 2013.
- [10] A. A. Kalage and N. D. Ghawghawe, "Optimum coordination of directional overcurrent relays using modified adaptive teaching learning based optimization algorithm," *Intelligent Industrial Systems*, vol. 2, no. 1, pp. 55–71, 2016.
- [11] M. Sulaiman, S. Muhammad, and A. Khan, "Improved solutions for the optimal coordination of docrs using firefly algorithm," *Complexity*, vol. 2018, Article ID 7039790, 15 pages, 2018.
- [12] S. S. Gokhale and V. S. Kale, "An application of a tent map initiated chaotic firefly algorithm for optimal overcurrent relay coordination," *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 336–342, 2016.
- [13] M. H. Hussain, I. Musirin, A. F. Abidin, and S. R. A. Rahim, "Modified swarm firefly algorithm method for directional overcurrent relay coordination problem," *Journal of Theoretical and Applied Information Technology*, vol. 66, no. 3, pp. 741–755, 2014.
- [14] F. A. Albasri, A. R. Alroomi, and J. H. Talaq, "Optimal coordination of directional overcurrent relays using biogeography-based optimization algorithms," *IEEE Transactions on Power Delivery*, vol. 30, no. 4, pp. 1810–1820, 2015.
- [15] J. Yu, C.-H. Kim, and S.-B. Rhee, "Oppositional Jaya algorithm with distance-adaptive coefficient in solving directional over current relays coordination problem," *IEEE Access*, vol. 7, pp. 150729–150742, 2019.
- [16] T. Amraee, "Coordination of directional overcurrent relays using seeker algorithm," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1415–1422, 2012.
- [17] M. Sulaiman, A. Ahmad, A. Khan, and S. Muhammad, "Hybridized symbiotic organism search algorithm for the optimal operation of directional overcurrent relays," *Complexity*, vol. 2018, Article ID 4605769, 11 pages, 2018.
- [18] M. Alipour, S. Teimourzadeh, and H. Seyedi, "Improved group search optimization algorithm for coordination of directional overcurrent relays," *Swarm and Evolutionary Computation*, vol. 23, pp. 40–49, 2015.
- [19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [20] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [21] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [22] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [23] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [24] R. V. Rao, "Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.
- [25] R. V. Rao, *Jaya: An Advanced Optimization Algorithm and its Engineering Applications*, Springer, Cham, Switzerland, 2019.
- [26] C. Huang, L. Wang, and L. L. Lai, "Data-driven short-term solar irradiance forecasting based on information of neighboring sites," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9918–9927, 2018.
- [27] K. Yu, B. Qu, C. Yue, S. Ge, X. Chen, and J. Liang, "A performance-guided Jaya algorithm for parameters identification of photovoltaic cell and module," *Applied Energy*, vol. 237, no. 1, pp. 241–257, 2019.
- [28] J.-t. Yu, C.-H. Kim, A. Wadood, T. Khurshaid, and S.-B. Rhee, "Jaya algorithm with self-adaptive multi-population and lévy flights for solving economic load dispatch problems," *IEEE Access*, vol. 7, pp. 21372–21384, 2019.
- [29] L. Wang, Z. Zhang, C. Huang, and K. L. Tsui, "A GPU-accelerated parallel Jaya algorithm for efficiently estimating Li-ion battery model parameters," *Applied Soft Computing*, vol. 65, pp. 12–20, 2018.
- [30] Y. Li, Z. Yang, G. Li et al., "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," *Applied Energy*, vol. 232, no. 15, pp. 54–68, 2018.
- [31] K. Gao, Y. Zhang, A. Sadollah, A. Lentzakis, and R. Su, "Jaya, harmony search and water cycle algorithms for solving

- large-scale real-life urban traffic light scheduling problem,” *Swarm and Evolutionary Computation*, vol. 37, pp. 58–72, 2017.
- [32] C. Huang, L. Wang, R. S. C. Yeung, Z. Zhang, H. S. H. Chung, and A. Bensoussan, “A prediction model-guided Jaya algorithm for the PV system maximum power point tracking,” *IEEE Transactions on Sustainable Energy*, vol. 9, no. 1, pp. 45–55, 2017.
- [33] 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, no. 15, pp. 457–467, 2019.
- [34] K. Gao, F. Yang, M. Zhou, Q. Pan, and P. N. Suganthan, “Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm,” *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1944–1955, 2018.
- [35] S. O. Degertekin, L. Lamberti, and I. B. Ugur, “Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm,” *Applied Soft Computing*, vol. 79, pp. 363–390, 2019.
- [36] R. V. Rao, D. P. Rai, and J. Balic, “Multi-objective optimization of abrasive waterjet machining process using Jaya algorithm and PROMETHEE Method,” *Journal of Intelligent Manufacturing*, vol. 30, no. 5, pp. 2101–2127, 2019.
- [37] A. Korashy, S. Kamel, A.-R. Youssef, and F. Jurado, “Modified water cycle algorithm for optimal direction overcurrent relays coordination,” *Applied Soft Computing*, vol. 74, pp. 10–25, 2019.
- [38] H. R. E. H. Bouchekara, M. Zellagui, and M. A. Abido, “Optimal coordination of directional overcurrent relays using a modified electromagnetic field optimization algorithm,” *Applied Soft Computing*, vol. 54, pp. 267–283, 2017.
- [39] C. A. C. Coello, “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11–12, pp. 1245–1287, 2002.
- [40] J. Moirangthem, S. S. Dash, R. Ramaswami, and K. R. Krishnanand, “Adaptive differential evolution algorithm for solving non-linear coordination problem of directional overcurrent relays,” *IET Generation, Transmission & Distribution*, vol. 7, no. 4, pp. 329–336, 2013.
- [41] A. Mahari and H. Seyedi, “An analytic approach for optimal coordination of overcurrent relays,” *IET Generation, Transmission & Distribution*, vol. 7, no. 7, pp. 674–680, 2013.
- [42] M. Ezzeddine, R. Kaczmarek, and M. U. Iftikhar, “Coordination of directional overcurrent relays using a novel method to select their settings,” *IET Generation, Transmission & Distribution*, vol. 5, no. 7, pp. 743–750, 2011.
- [43] M. N. Alam, B. Das, and V. Pant, “A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination,” *Electric Power Systems Research*, vol. 128, pp. 39–52, 2015.

