Novel Constraint Stratification Ranking Methodology Integrated on Dynamic Balance Imperialist Competition Algorithm for Solving the Multi-Objective Optimal Reactive Power Dispatch Problem

Gang Guo*, Jia Cao, and Shuaiyong Li

Abstract—**Computer technology provides new possibilities for handling the multi-objective optimal reactive power dispatch (MOORPD) problems with high-dimension and non-differentiability. In this paper, a novel constraint stratification ranking methodology integrated on dynamic balance imperialist competition algorithm (ICA) is proposed to deal with MOORPD problem, and this methodology is called NCSR-DBICA. The proposed methodology includes three sub-methods to solve three typical problems encountered when dealing with MOORPD problems with ICA. These typical problems include the original ICA which is easy to fall into local optimum, it is difficult to handle the constraints in MOORPD problem and there is no effective solution set ranking method. Thus, the dynamic balance strategy (DBS) is proposed to improve the searching ability of ICA, the constraint-based country stratification mechanism (CCSM) is proposed to deal with the constraint problem, and a novel ranking method (NRM) is proposed to solve the ranking problem. To verify the effectiveness of the improved method, the NCSR-DBICA, MOICA-FS, NSGA-III, NSGA-II and MOPSO-CD were tested on three test systems. Simulation results show that NCSR-DBICA can find better results, especially in large scale systems. In addition, two indicators: Generational Distance (GD) and Hyper-volume (HV), were selected to evaluate the diversity, stability and convergence of the above algorithms. The evaluation results also verify the superiority of NCSR-DBICA.**

Index Terms—**multi-objective optimal reactive power dispatch problem, imperialist competition algorithm, novel ranking method, constraint-based country stratification mechanism, dynamic balance strategy**

I. Introduction

This work was supported in part by the National Natural Science Foundation Project of China under Grant (61703066), in part by the Natural Science Foundation Project of Chongqing under Grant (cstc2018jcyjAX0536), and in part by the Chongqing Technology Innovation and Application Development Project (cstc2019jscx-fxydX0042, cstc2019jscx-zdztzxX0053).

Gang Guo is with the School of Information Technology, Luoyang Normal University, Luoyang, Henan 471934, China (corresponding author, e-mail: guogangchina @sina.cn).

Jia Cao is with the Key Laboratory of Complex Systems and Bionic Control, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: caojia1995@126.com).

Shuaiyong Li is with the Key Laboratory of Industrial Internet of Things & Networked Control, Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: lishuaiyong@cqupt.edu.cn).

THE optimal reactive power dispatch (ORPD) problem is a sub-problem of the optimal power flow (OPF) research, and it is also a classic optimization problem for power systems [1-5]. This problem is a static nonlinear, non-convex, multi-objective, multi-constrained, multivariable problem which optimizes one or more objective functions under certain equality and inequality constraints. The objective functions that need to be optimized generally include active power losses, voltage deviation, and voltage stability index. And this optimization process can be achieved by adjusting the generator voltage (continuous variable), the transform taps (discrete variable) and the reactive power generations of Volt-Ampere Reactive (VAR) source (discrete variable). According to the number of optimization targets, the ORPD problem can be divided into single-objective ORPD (SOORPD) problem and multi-objective ORPD (MOORPD) problem [6-8]. Different from the single-objective problems (SOPs), the multi-objective problems (MOPs) are solved in order to find a set of Pareto optimal solution sets, and then obtain a best compromise solution (BCS).

Over the last decades, some classic methods such as weight sum method, goal attainment method, and ε-constraint approach have been applied to deal with the MOPs. However, the searching capabilities of these methods are very limited, which require a mass of runs and a large amount of calculation time, and sometimes researchers cannot even find a solution that meets the requirements. In recent years, the emergence of intelligent algorithms provides a new idea for MOPs. The successful application of some algorithms such as non-dominated sorting genetic algorithm II (NSGA-II), reference-point based many-objective non-dominated sorting genetic algorithm (NSGA-III), multi objective particle swarm optimization (MOPSO), multi-objective evolutionary algorithm (MOEA) and other algorithms also shows the effectiveness of intelligent algorithms in dealing with the MOPs [9-13]. In addition, some algorithms also have obvious experimental advantages when dealing with the MOORPD problem. In [14], the author solves the MOORPD problem by using the multi-objective differential evolution (MODE) algorithm. In [15], a two-archive multi-objective grey wolf optimizer is proposed for solving

MOORPD problem. In [16], a two-point estimate method is proposed to model the load uncertainty in MOORPD problem.

ICA is an intelligent algorithm inspired by social competitive behavior [17-19]. A lot of research has shown the superiority of this algorithm in dealing with the optimization problems [20-23]. In [20], an enhanced ICA is proposed to optimum design of skeletal structures. In [21], an efficient hybrid method based on ICA and GA is proposed for solving the optimal sitting and sizing problem of DG and shunt capacitor banks simultaneously. In [23], a multi-objective global best ICA is proposed and successfully applied to solve MOORPD problems. Therefore in this study, ICA is used to deal with the MOORPD problem. Although ICA has obvious advantages in dealing with SOPs, it also has some problems when dealing with the MOORPD problem. For example: ICA is easy to fall into local optimum, the solution may not satisfy the constraint, and there is no effective multi-objective solution set ranking method. For the first problem, a dynamic balance strategy (DBS) is proposed to balance the searching ability of the ICA. For the constraint problem, a new constraint processing method based on Pareto dominant is proposed, instead of using the commonly used penalty coefficient method, which is called constraint-based country stratification mechanism (CCSM). For the sorting problem, a novel ranking method based on country satisfaction and country distance is proposed to solve the ranking problem of the multi-objective solution set.

To verify the effectiveness of the proposed methods, the simulation experiments with different objective function combinations of NCSR-DBICA, MOICA-FS [22], NSGA-III [24, 25], NSGA-II [26] and MOPSO-CD [27] are performed on IEEE 30, 57 and 118 test systems. The results prove that NCSR-DBICA can find better BCS in different cases. In addition, to explore the diversity, convergence and stability of the algorithm, two indicators, GD and HV, are selected to evaluate the performance of the algorithm [28-30].

The sections of this paper are arranged as follows: Section II introduces the mathematical model of the MOORPD problem, Section III describes the original ICA and the proposed NCSR-DBICA to solve the MOORPD problem, Section IV presents the detailed results of the improved algorithm in dealing with the MOORPD problem and the experimental performance of the algorithms. Finally, the conclusions are given in Section V.

II. The MOORPD Problem

In this paper, a MOORPD problem to be solved can be defined as follows:

$$\begin{aligned} &\text{Minimize} \quad F(x,u)=[f_1(x,u), f_2(x,u),\cdots, f_m(x,u)] \\ &\text{subject to} \quad g_j(x,u)=0, \quad j=1,2,...,J \\ &\qquad\qquad h_k(x,u)\le 0, \quad k=1,2,...,K \end{aligned} \tag{1}$$

Where x is the state variables, u is the control variables, $F(x, u)$ is the objective functions, m is the number of objective functions, $g(x, u)$ and $h(x, u)$ are equality and inequality constraints, J and K are the number of equality and inequality constraints. In detail, state variables x include U_L (load bus voltages), Q_G (generator reactive power outputs) and S_L (transmission line loadings). Control variables u include U_G (generator bus voltages), T (transformer taps) and Q_C (reactive power compensation). Hence, x and u can be formed as follows:

$$x^T=[U_{L1},...,U_{LN_{pq}}, Q_{G1},...Q_{GN_{pv}}, S_{L1},...,S_{LN_E}] \tag{2}$$

$$u^T=[U_{G1},...,U_{GN_{pv}}, T_1,...,T_{N_T}, Q_{C1},...,Q_{CN_C}] \tag{3}$$

Where N_{pq} denotes the number of PQ buses, N_{pv} represents the number of PV buses, N_E is the total number of transmission branches, N_C shows the total number of shunt VAR compensators.

A. Objective Functions: F(x, u)

Minimization of active power losses: this objective function is aims to minimize the total active power transmission losses.

$$f_1(x,u)=\min P_{loss}=\sum_{k\in N_E} g_k(U_i^2+U_j^2-2U_iU_j\cos\delta_{ij}) \tag{4}$$

Where f_1 is the first objective function, P_{loss} represents the total active power losses, g_k is the conductance of the kth branch, U_i and U_j are the voltage magnitude of ith node and jth node, δ_{ij} is the voltage angle between node i and node j.

Minimization of voltage deviation: this objective function aims to enhance the voltage quality of the power system.

$$f_2(x,u)=\min V_d=\sum_{i=1}^{N_{PQ}}\left|U_i-U^{REF}\right| \tag{5}$$

Where f_2 is the second objective function, V_d is the total voltage deviations, U^{REF} is the preferred value of voltage magnitude.

Minimization of voltage stability index: this objective function aims to show the security of the power grid.

$$f_2(x,u)=\min V_d=\sum_{i=1}^{N_{PQ}}\left|U_i-U^{REF}\right| \tag{6}$$

Where $f_3(x, u)$ is the third objective function, L index is the global voltage stability index, L_j is the local voltage stability index and it can be detailed as follows.

$$\begin{cases} L_j=|1-\sum_{i=1}^{N_{PV}} F_{ji}\dfrac{\dot{U}_i}{\dot{U}_j}|, \ j\in N_{PQ} \\ F_{ji}=[-[\mathrm{Y_{LL}}]^{-1}[\mathrm{Y_{LG}}]]_{ji} \end{cases} \tag{7}$$

Where $\dot{U}_i$ and $\dot{U}_j$ are complex voltages of the ith PV bus and the jth PQ bus, Y_{LL} and Y_{LG} are the sub-matrices. Y-bus matrix is the node admittance matrix and it is acquired after separating the PQ buses and PV buses [14].

B. Constraints: g (x, u) and h (x, u)

1) The g (x, u)

In the ORPD problem, the equality constraints $g(x, u)$ include active power and reactive power balance equations, and the violations of $g(x, u)$ are considered by using Newton-Raphson load flow calculation [31, 32].

$$P_{Gi}-P_{Di}-U_i\sum_{j\in N_i}U_j(G_{ij}\cos\delta_{ij}+B_{ij}\sin\delta_{ij})=0,\ i\in N \tag{8}$$

$$Q_{Gi}-Q_{Di}-U_i\sum_{j\in N_i}U_j(G_{ij}\sin\delta_{ij}-B_{ij}\cos\delta_{ij})=0,\ i\in N_{PQ} \tag{9}$$

Where P_{Gi} and P_{Di} are the active power generation and active load demand; N_i is the number of nodes adjacent to node i; G_{ij} and B_{ij} are the real part and imaginary part of the ijth element of Y-bus matrix; N is the number of system buses; Q_{Gi} and Q_{Di} are the reactive power generation and reactive load demand.

2) The h (x, u)

The h (x, u) include state variables and control variables inequality constraints.

State variables inequality constraints: those constraints mainly include U_{Li} (the voltage limits of ith load bus), Q_{Gi} (the reactive power outputs limits of the ith generator bus), and S_{Lij} (the transmission apparent power flow limits of the ijth transmission line). Their lower and upper boundaries can be formulated as below.

$$\begin{cases} U_{Li}^{\min} \le U_{Li} \le U_{Li}^{\max}, & i \in N_{PQ} \\ Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max}, & i \in N_{PV} \\ S_{Lij} \le S_{Lij}^{\max}, & ij \in N_E \end{cases} \tag{10}$$

Control variables inequality constraints: those constraints mainly include U_{Gi} (the voltages limits of the ith generator bus), T_i (the transformer tap settings limits of the ith transformer), and Q_{Ci} (the reactive power compensation capacity limits of the ith capacitor bank). Their corresponding boundaries can be expressed as below.

$$\begin{cases} U_{Gi}^{\min} \le U_{Gi} \le U_{Gi}^{\max}, & i \in N_{PV} \\ T_i^{\min} \le T_i \le T_i^{\max}, & i \in N_T \\ Q_{Ci}^{\min} \le Q_{Ci} \le Q_{Ci}^{\max}, & i \in N_C \end{cases} \tag{11}$$

III. PROPOSED METHODOLOGY

A. Summary of ICA

The ICA, a new intelligent algorithm, was inspired by the imperialistic competition [33]. The initial population of the algorithm is called the country, and the power of the country is determined by the value of the country's objective function. In the minimum value problem, the smaller the value of the objective function, the greater the power of a country. Thus, all countries will be divided into two categories: the colony (the weaker country) and the imperialist country (the stronger country). After the classification, the colonies will be occupied by the imperialist countries through the roulette wheel selection, and the empire will be formed (each empire consists of an imperialist country and several colonies). To improve their power and become an imperialist country, the colonies will learn from their imperialist countries. In addition, the empire will compete with other empire in order to gain more colonial power. In this process, weak empire will be disintegrated by other stronger empire. With the iteration of the algorithm, there is only one strongest empire left, and the position of the imperialist country in the empire is the optimal solution that the algorithm seeks. Thus, the detailed steps of the ICA can be explained as follows:

Step1: Set the primary parameters: N_{pop}, N_{imp}, N_{col}, θ, β, ξ, maximum iteration *Maxit*.

Step2: Create the initial countries. In an N_{var} dimensional problem, an $1\times N_{var}$ array is called country and it can be defined as below.

$$country = [p_1, p_2, p_3, \dots, p_{N_{var}}] \tag{12}$$

Where N_{var} is the number of variables to be considered, p_is is the value of variables that should be optimized. For a country, these variables include politics, culture, economy, religion and other aspects [34]. As shown in the Figure 1.

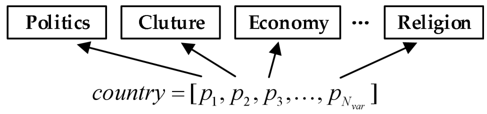


Fig. 1. Socio-political characteristics of a country

Step3: Calculate the cost of countries.

$$\text{cost}_i = f(p_1, p_2, \dots, p_{N_{var}}) \quad i = 1, 2, \dots, N_{pop} \tag{13}$$

In the first step of the algorithm, the total number of countries N_{pop} has been determined. Then, select the strongest initial countries with size of N_{imp} to be the imperialists and the remaining N_{col} of countries form the colonies.

$$N_{col} = N_{pop} - N_{imp} \tag{14}$$

Step4: Form the empires through the roulette wheel selection and calculate the power of each imperialist.

To form empires, all colonies will be divided among the imperialists according to their power. The normalized cost of an imperialist can be determined as below.

$$C_n = \max_i \{c_i\} - c_n \tag{15}$$

Where c_n is the cost of the nth imperialist, C_n is the normalized cost. Therefore, the power of each imperialist can be calculated based on its normalized cost.

$$P_n = |C_n / \sum_{i=1}^{N_{imp}} C_i| \tag{16}$$

Where p_n is the power value of nth imperialist. Then, the initial number of colonies for nth imperialist, NC_n, can be determined.

$$NC_n = round\{p_n N_{col}\} \tag{17}$$

Step5: Start assimilation process. In this movement, all colonies will move to their imperialists to increase their power. Figure 2 shows this process. The colony move distance x is a random variable related to the distance parameter β. The θ is the direction parameter with an angle parameter γ. The x and θ can be defined as blow.

$$x \sim U(0, \beta \times d) \tag{18}$$

$$\theta \sim U(-\gamma, +\gamma) \tag{19}$$

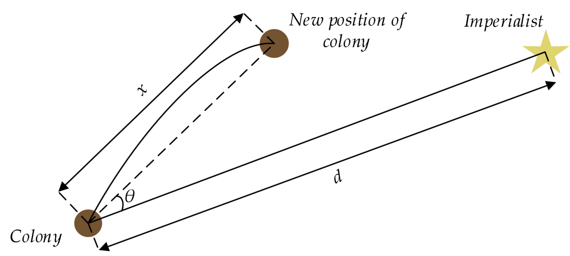


Fig. 2. Assimilation process

Step6: Execute revolution operation. In the history of imperial, the colony can not only change its socio-political characteristics through assimilation, but also strengthen its own power by reforming its own politics, culture, economy and other elements. Revolution, in brief, is that the colonies change their position through reform. This change is similar to the mutation operation in GAs. In ICA, this sudden random change is called the revolution of colony.

Step7: Exchange the position of a colony and an imperialist, if we find any colony has better performance than their relevant imperialist.

Step8: Calculate the total power of an empire. The total power of the empire is mainly related to its imperialist, and the power of the colonies also have a negative effect on the total power. Therefore, the total power of an empire can be calculated by the following formula.

$$\begin{aligned} TC_n &= Cost(imperialist_n) \\ &+ \xi mean\{Cost(colonies\ of\ empire_n)\} \end{aligned} \tag{20}$$

Where ξ is the colonial weight that has value between 0 and 1.

Step9: Start imperialistic competition. This process is a competition between the empire and the empire. Depending on the power of the empire, the weakest colony in the weak empire will be encroached by a more powerful empire. Before that, the total power of the empire needs to be measured first. And the power calculation formula can be obtained by the following two formulas.

$$NTC_n = \max_i\{TC_i\} - TC_n \tag{21}$$

$$P_{p_n} = | NTC_n / \sum_{i=1}^{N_{imp}} NTC_i | \tag{22}$$

Where NTC_n is the normalized total cost, TC_n is the total cost of the nth empire, P_{pn} is the total power of the nth empire.

Step10: Eliminate weak empire. The weak empire gradually lost all the colonies to other stronger empires, leaving only one imperialist country in this empire. This imperialist country will become a colony and will be occupied by other empires. Then, the weak empire was eliminated.

Step11: Check the stopping condition. The searching process is terminated when the certain number of iterations are met. Otherwise, return to the **Step5**. The ideal termination criterion is when all the empires have collapsed and only one (the most powerful empires) remains to rule all other countries.

B. Multi-Objective Solution Strategy

The purpose of solving the MOPs is to obtain a group of Pareto optimal solutions and then find one BCS from those solutions [35]. The ICA described above finds the desired results by continuously optimizing the particles, and the Pareto optimization method is the key to guiding ICA to find the Pareto front. After obtaining the optimal Pareto front, the fuzzy mathematics decision method is used to find the BCS. Further description about these two methods is shown below.

1) Pareto optimization method

Different from the SOPs, the solution vector of the multi-objective problem is two-dimensional or higher-dimensional. The core of Pareto optimization method is to effectively judge the quality of the solution of multi-objective problem, and then find the Pareto front. In fact, the Pareto front is a set of solutions that cannot be dominated by other solutions. For any solution u1 and solution u2, u1 dominants u2 if and only if below conditions are satisfied at the same time.

$$\begin{cases} \forall i \in \{1,2,...,m\} : f_i(x,u_1) \le f_i(x,u_2) \\ \exists j \in \{1,2,...,m\} : f_j(x,u_1) < f_j(x,u_2) \end{cases} \tag{23}$$

Finally, a set of Pareto optimal solutions that are non-dominated within the search space can be found. Due to the large number of dominant solutions, in the multi-objective problem, obtain the best one BCS is necessary and is achieved by fuzzy mathematics decision method.

2) Fuzzy mathematics decision method

After obtaining the Pareto front, that is, the Pareto optimal solution sets, the decision maker needs to find the BCS that satisfy the current conditions from these solution sets. In order to obtain the only one BCS, the fuzzy membership of each objective of each solution needs to be calculated first.

$$S_{mi}(x) = \begin{cases} 1 & f_{mi} \le f_{m\min} \\ \dfrac{f_{m\max} - f_{mi}(x)}{f_{m\max} - f_{m\min}} & f_{m\min} < f_{mi} < f_{m\max} \\ 0 & f_{mi} \ge f_{m\max} \end{cases} \tag{24}$$

Where S_{mi} is the satisfaction function of mth objective function of f_m of individual i, f_{mmax} and f_{mmin} represent the maximum and the minimum value of the mth objective function. For each available solution, the normalized value of satisfaction functions is calculated by (25).

$$sf(i) = \frac{\sum_{m=1}^{M} S_{mi}}{\sum_{i=1}^{N_p} \sum_{m=1}^{M} S_{mi}} \tag{25}$$

Where $sf(i)$ is the normalized satisfaction function. The aforementioned dominant solutions with the maximum sf_i can be chosen as the BCS.

C. Multi-Objective Solution Strategy

In this paper, our main purpose is to use ICA to solve the MOORPD problem. The problems we encountered are the improvement of algorithm search ability, the processing of constraints and the processing of multi-objective solutions. To solve the above three problems, three methods are proposed to improve the performance of ICA in dealing with MOORPD problems, namely the dynamic balance strategy (DBS), the constraint-based country stratification mechanism (CCSM) and the novel ranking method based on country satisfaction and country distance (NRM). More details about these improvements are presented as below.

1) DBS: Dynamic balance strategy

In the assimilation process of the original ICA, the development of the colony mainly depends on its relevant imperialist country, but this position update method may lead to the algorithm easily falling into the local optimum. Therefore, in order to improve the global searching ability of the ICA, the strongest imperialist guidance mechanism is introduced based on the original algorithm. This means that the colony is not only influenced by the corresponding imperialist country but also by the most powerful imperialist country in its own development. Thus, the colony's update formula can be improved as below.

$$P_i^{k+1}(Col_{new}) = \omega^k P_i^k(Col_{old}) + \beta^k r_1[P_i^k(Imp_{rel}) - P_i^k(col_{old})] + \alpha^k r_2[P^k(Imp_{strgest} - P_i^k(col_{old}))] \quad (26)$$

Where P_i^k (Col_{new}) and P_i^k (Col_{old}) are the new position and old position of the colony i at iteration k, P_i^k (Imp_{rel}) is the position of relevant imperialist country of colony i at iteration k, P^k $(Imp_{strgest})$ is the position of the strongest imperialist country at iteration k, r_1 and r_2 are random numbers between 0 and 1, ω^k is the inertia weight, β^k and α^k are assimilation coefficients. And, the improved assimilation process can be represented by Figure 3.

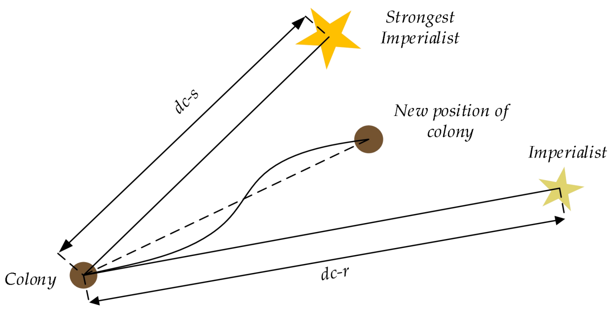


Fig. 3. Improved assimilation process

The successful application of the strongest imperialist guidance mechanism effectively improve the global searching ability of the original algorithm in the early iteration. However, the local searching ability of the algorithm in the middle and late period needs to be strengthened. Therefore, in order to balance the searching ability of different stages of the algorithm, a dynamic balance strategy is proposed. This method can dynamically adjust the assimilation coefficients β^k and α^k according to the number of iterations, and then change the influence of the relevant imperialist country and the strongest imperialist country on the colonies during different periods. And, β^k and α^k are optimized as below.

$$\beta^k = \beta_{\min} + \frac{\beta_{\max} - \beta_{\min}}{k_{\max}} \times k \quad (27)$$

$$\alpha^k = \alpha_{\max} - \frac{\alpha_{\max} - \alpha_{\min}}{k_{\max}} \times k \quad (28)$$

Where β_{max} and β_{min} are the maximum value and minimum value of assimilation coefficient of the relevant imperialist country, k_{max} is the maximum number of iterations, k is the current number of iterations, α_{max} and α_{min} are the maximum value and minimum value of assimilation coefficient of the strongest imperialist country.

2) CCSM: Constraint-based country stratification mechanism

In the SOPs, the quality of the country in ICA can be judged directly by the objective function value. However, the MOORPD problem that needs to be solved is a multi-objective, multi-constrained issue. The traditional methods have failed to meet the requirements of this research. Therefore, based on the Pareto optimization idea, the constraint is regarded as the premise of the country's quality, and the constraint-based country stratification mechanism is proposed to judge the quality of the country.

The essence of the ORPD problem is to obtain a set of control variables u that is, the socio-political position that determines the power of the country's power in ICA. Thus, the constraints of these control variables need to be processed first. For the country i, its position can be adjusted as follows.

$$u_i = \begin{cases} u_{i,\min} & \text{if} \quad u < u_{i,\min} \\ u_i & \text{if} \quad u_{i,\min} \le u \le u_{i,\max} \\ u_{i,\max} & \text{if} \quad u > u_{i,\max} \end{cases} \quad (29)$$

The constraints mentioned above include not only control variable constraints, but also state variable constraints. Here, combined with the Pareto optimization idea and the ICA, a method is proposed to solve the state variable violation constraint in the MOORPD problem, which is called the constraint-based country stratification mechanism.

Firstly, if any country violates the constraint, calculate the total amount of constraint violation by (30).

$$Tvio(u_i) = \sum_{k=1}^{Ds} \max(h_k(x, u_i), 0) \quad (30)$$

Where $Tvio$ (u_i) is the total amount of constraint violations of country i, D_s is the dimension size of the state variables.

Different from the original Pareto optimization method, the proposed method first needs to judge the country's constraint violations. It means that those countries that do not violate the constraint dominate all other countries that violate the constraint. For any country u_1 and country u_2, their dominant relationship can be expressed as follows:

If $Tvio$ $(u_1) < Tvio$ (u_2), country u_1 dominates country u_2;

If $Tvio$ $(u_1) > Tvio$ (u_2), country u_2 dominates country u_1;

If $Tvio$ $(u_1) = Tvio$ (u_2) and the two following conditions are satisfied at the same time, it can be judged that country u_1 dominates country u_2: Firstly, f_i $(x, u_1) \le f_i$ (x, u_2) for all $i \in \{1, 2, \ldots, M\}$. Secondly, f_j $(x, u_1) < f_j$ (x, u_2) for any $j \in \{1, 2, \ldots, M\}$. Otherwise, u_1 and u_2 belong to the same level.

Using this method, all countries can be divided into n levels. The *Rank* (i) is defined as the level of country i, and those countries are not dominate each other will have the same level. In addition, the smaller the rank value is, the stronger the country is. Countries with higher ranks are more likely to become imperialist countries. Conversely, countries with lower ranks will become colonies. Finally, all countries are divided into n ranks with different power levels according to the constraint-based country stratification mechanism.

3) NRM: Novel ranking method based on country satisfaction and country distance

When dealing with the MOPs, each generation of updated countries will be stored in a repository, and then the repository will sort all countries, leaving the best N_R (the size of repository) solutions for each generation. In the classic NSGA-II algorithm, individuals are sorted by the non-dominated sorting and the crowded distance calculation. In our research, the non-dominated sorting is optimized by the proposed constraint-based country stratification mechanism. For countries at the same level, a novel ranking method based on country satisfaction and distance is proposed.

Country satisfaction sf (i) can be calculated by the aforementioned (25). Similar to the crowded distance calculation method, the country distance can be obtained according to the following equation.

$$Cdis(u_i)=Cdis(u_i)+\frac{f_m(u_{i+1})-f_m(u_{i-1})}{f_{m\max}-f_{m\min}} \quad (31)$$

Where u_i is the position of country i, $Cdis$ (u_i) is the country distance of country i, f_m (u_{i+1}) and f_m (u_{i-1}) represent the objective function values of the corresponding two countries adjacent to country i, $f_{m\ \max}$ and $f_{m\ \min}$ represent the maximum and minimum values of the mth objective function, respectively.

Now, any country i has three attributes: country level *Rank* (i), country satisfaction sf_i and country distance *Cdis* (u_i). Then, make the following judgment:

If (any Country. Rank) ≠1

If *Rank* (i) < *Rank* (j), country i is stronger than country j;

If *Rank* (i) > *Rank* (j), country j is stronger than country i;

If *Rank* (i) = *Rank* (j)

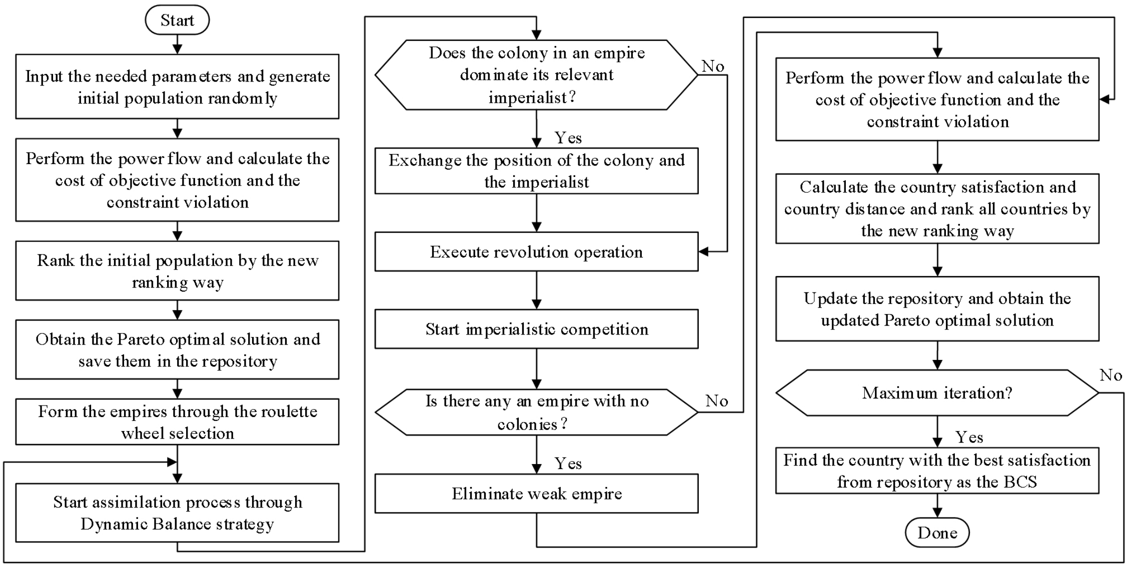


Fig. 4. Flowchart of the proposed methodology for MOORPD problem

If sf (i) > sf (j), country i is stronger than country j;

else country j is stronger than country i;

Else if (all Country. Rank) =1

If *Rank* (i) < *Rank* (j), country i is stronger than country j;

If *Rank* (i) > *Rank* (j), country j is stronger than country i;

If *Rank* (i) = *Rank* (j)

If *Cdis* (i) > *Cdis* (j), country i is stronger than country j;

else country j is stronger than country i.

In this way, country level *Rank* (i) ensures that the country does not violate the constraint, country satisfaction sf (i) can ensure that the country particles move to the BCS at the beginning of the iteration. And the country distance *Cdis* (u_i) ensures that a Pareto front with better distribution can be obtained later in the iteration.

4) The proposed methodology for MOORPD problem

The main purpose of this research is to solve the MOORPD problem using the proposed dynamic balance ICA combined with the novel constraint stratification ranking methodology. In the process of solving this problem, the dynamic balance strategy is mainly to improve the searching ability of the ICA. The constraint-based country stratification mechanism is used for the processing of constraints and the stratification of individuals after each power flow calculation. The new ranking method first solves the judgment of the quality of the multi-objective solution in the algorithm update process, and secondly solves the sorting problem of the particles stored in the repository. Figure 4 shows the flowchart of proposed methodology for MOORPD problem.

In order to evaluate the performance and efficiency of the proposed methodology, it has been applied to solve the MOORPD problem in IEEE30, 57 and 118 bus power systems. In order to substantiate the effectiveness of the proposed methodology, its performance is compared with four multi-objective optimization algorithms and these comparison algorithms are listed as below.

(1) MOPSO-CD [27]: Multi-objective particle swarm optimizers based on the crowding distance calculation method, and in this paper, this algorithm is abbreviated as MOPSO-CD.

(2) NSGA-II [26]: Non-dominated sorting genetic algorithm II.

(3) NSGA-III [24]: Reference-point based non-dominated sorting approach.

(4) MOICA-FS [22]: Multi-objective Imperialist Competition Algorithm based on the fast non-dominated sorting and the Sigma method.

IV. Simulation Experiment

TABLE I Objective Function Combinations

Cases	Function Combinations	Trials	Test System
Case I	$f_1 = P_{loss}$ & $f_2 = V_d$	30	System A: IEEE30
Case II	$f_1 = P_{loss}$ & f_3 = L index	30	
Case III	$f_2 = V_d$ & f_3 = L index	30	
Case IV	$f_1 = P_{loss}$ & $f_2 = V_d$	30	System B: IEEE57
Case V	$f_1 = P_{loss}$ & f_3 = L index	30	
Case VI	$f_1 = P_{loss}$ & $f_2 = V_d$	20	System C: IEEE118

TABLE II Algorithm parameter settings of NCSR-DBICA

Pars	Case I	Case II	Case III	Case IV	Case V	Case VI
N_{pop}	100	100	100	100	100	100
N_R	100	100	100	100	100	100
K_{max}	500	500	500	500	500	600

N_{imp}	25	25	25	25	25	25
N_{col}	75	75	75	75	75	75
ξ	0.1	0.1	0.1	0.1	0.1	0.1
ω	1	1	1	1	1	1

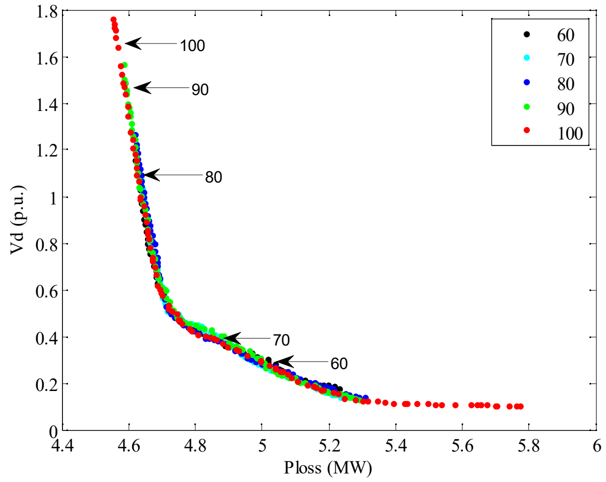


Fig. 5. Pareto front obtained by NCSR-DBICA using different population sizes for Case I

Three optimization objective functions are considered: active power losses (P_{loss}), voltage deviation (V_d) and voltage stability index (L index). Six cases are considered in this paper and the details of these cases are shown in Table I. To find the most suitable population size, we performed simulation experiments on NCSR-DBICA with different population sizes. The Pareto fronts obtained by the NCSR-DBICA with different population sizes and different numbers of imperialist are shown in Figures 5-6. It can be seen that when the number of particles and the number of imperialists are 100 and 25, respectively, we can get a better Pareto front. For different cases, the parameters of the NCSR-DBICA are shown in Table II. And those algorithms have been implemented in MATLAB 2014a and run them on a PC with Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz with 3.41GHz.

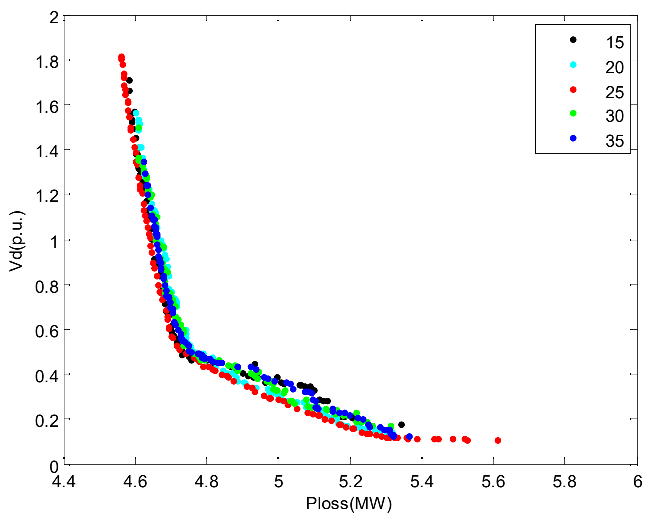


Fig. 6. Pareto front obtained by NCSR-DBICA using different numbers of imperialist for Case I

A. Results for the Test System A

The single line diagram of system A can be seen in Figure 7. The detailed data of the test system A is provided in [36, 37].In this system, the control variable of the ORPD problem is a 19-dimensional vector, which including: 6 generator bus voltages, 4 transformer taps, and 9 reactive power compensation variables. The lower and upper limits for generator bus voltages amplitude are 0.95 p.u. and 1.05 p.u., the limits of transformer taps are set at 0.9-1.1, and the limits of reactive power are set at 0.95 and 1.1 in p.u..

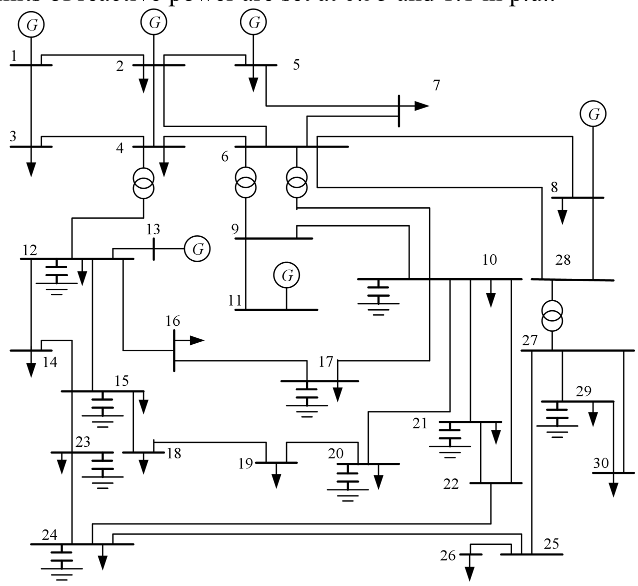


Fig. 7. The single line diagram of the System A

1) Discussion of Case I

In Case I, NCSR-DBICA is utilized to optimizing P_{loss} and V_d simultaneously. The Pareto fronts obtained by NCSR-DBICA, MOICA-FS, NSGA-II, NSGA-III and MOPSO-CD are drawn in Figure 8. The results of BCS achieved by those algorithms are presented in Tables 3. In Table 3, it can be found that the BCS obtained by NCSR-DBICA is better than MOICA-FS, NSGA-III, NSGA-II and MOPSO-CD. Comparing the published MGBICA algorithm, the results of the proposed method dominate the results of the MGBICA algorithm. It can be observed that the P_{loss} and the V_d are reduced by 0.02099 (MW), 0.02543 (p.u.), respectively.

TABLE III Comparison of the BCS for Case I

Variables	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA	MGBICA[33]
VG1(p.u.)	1.0600	1.0563	1.0637	1.0561	1.0632	1.0592
VG2(p.u.)	1.0513	1.0484	1.0542	1.0475	1.0506	1.0493
VG5(p.u.)	1.0285	1.0209	1.0237	1.0257	1.0266	1.0044
VG8(p.u.)	1.0199	1.0212	1.0245	1.0237	1.0219	1.0250
VG11(p.u.)	1.0279	1.0511	1.0897	1.0611	1.0234	0.9969

VG13(p.u.)	1.0510	1.0220	1.0072	1.0254	1.0272	1.0454
T6-9	1.0539	0.9907	1.1000	1.0132	1.0627	1.04
T6-10	0.9000	1.0439	0.9508	0.9977	0.9080	1.0
T4-12	1.0685	0.9995	0.9973	1.0042	1.0229	1.03
T28-27	0.9863	0.9797	0.9801	0.9890	0.9880	0.98
C10(p.u.)	0.0090	0.0413	0.0000	0.0244	0.0287	0.27
C12(p.u.)	0.0293	0.0112	0.0484	0.0108	0.0030	--
C15(p.u.)	0.0429	0.0463	0.0293	0.0198	0.0358	--
C17(p.u.)	0.0021	0.0447	0.0240	0.0136	0.0359	--
C20(p.u.)	0.0277	0.0353	0.0500	0.0465	0.0399	--
C21(p.u.)	0.0500	0.0324	0.0294	0.0274	0.0332	--
C23(p.u.)	0.0272	0.0374	0.0500	0.0324	0.0386	--
C24(p.u.)	0.0254	0.0500	0.0379	0.0477	0.0499	0.13
C29(p.u.)	0.0338	0.0176	0.0072	0.0244	0.0253	--
P_{loss} (MW)	5.11401	5.09168	5.08912	5.08272	**5.07861**	5.0996
V_d (p.u.)	0.25846	0.25559	0.25682	0.25645	**0.23387**	0.2593

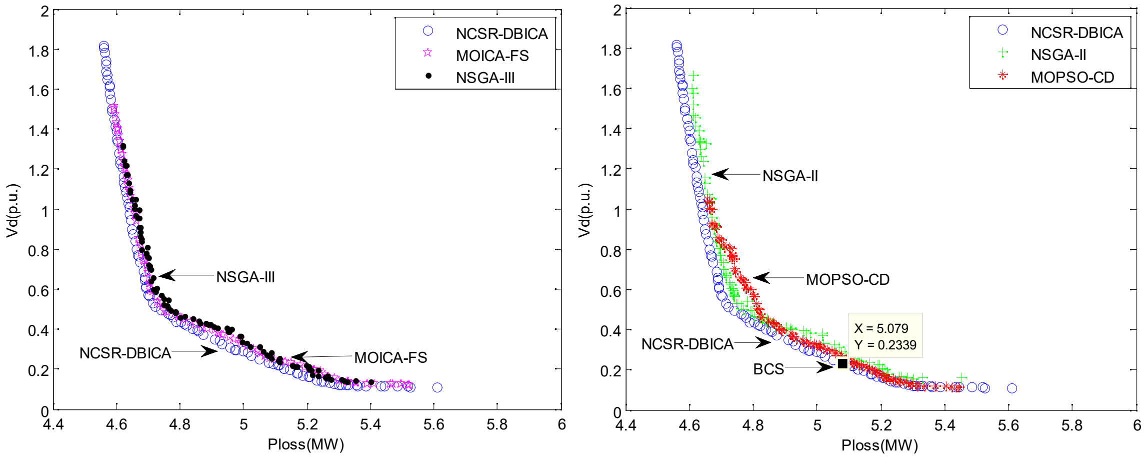


Fig. 8. Pareto fronts obtained by NCSR-DBICA and other four algorithms in Case I

2) Discussion of Case II

In case II, the NCSR-DBICA algorithm is used to optimize the P_{loss} and L index. It can be observed form Figure 9 that the distribution of the Pareto front obtained in case II is not as good as that of case I. which is because the optimization of this objective function combination is more difficult. However, it can be observed that the NCSR-DBICA can still find a better Pareto front than other four algorithms. The detailed data in Table 4 can prove that the BCS obtained by the NCSR-DBICA algorithm is the best. Although the L index obtained by the NCSR-DBICA is a little higher than the MOCIPSO [37], the value of the P_{loss} is much lower than the result of the reported MOCIPSO [37]. This proves that the proposed NCSR-DBICA algorithm has certain optimization capabilities in this case.

3) Discussion of Case III

In case III, NCSR-DBICA is used in order to optimize V_d and L index. Figure 10 presents the results of five algorithms. It can be seen that all the five algorithms can obtain a well distributed Pareto front. Especially the Pareto front of the MOICA-FS and NSGA-III algorithms is very close to NCSR-DBICA. It is proved that the optimization effect of this objective function combination is very obvious. Compared with the NSGA-II and MOPSO-CD algorithms, the Pareto front obtained by the improved algorithm is more effective. This can also be demonstrated by the data in Table 5. According to the results from Table 5, the best BCS was found by the NCSR-DBICA algorithm and it was (0.47465, 0.13277).

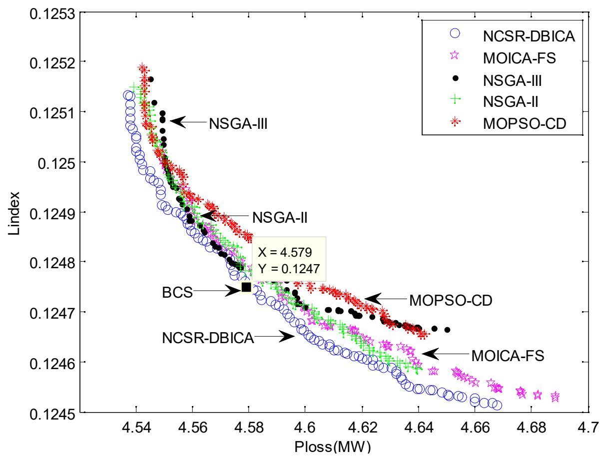


Fig. 9. Pareto fronts obtained by NCSR-DBICA and other four algorithms in Case II.

TABLE IV Comparison of the BCS for Case II

Variables	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA	MOCIPSO[37]
VG1(p.u.)	1.1000	1.1000	1.0992	1.1000	1.1000	1.1000
VG2(p.u.)	1.0970	1.0958	1.0958	1.0956	1.0964	1.1000
VG5(p.u.)	1.0793	1.0813	1.0804	1.0815	1.0781	1.1000
VG8(p.u.)	1.0843	1.0874	1.0848	1.0875	1.0845	1.1000
VG11(p.u.)	1.1000	1.1000	1.0998	1.0998	1.0999	1.1000
VG13(p.u.)	1.1000	1.1000	1.1000	1.0998	1.0996	1.1000
T6-9	1.0252	1.0373	0.9932	1.0379	1.0284	0.9400
T6-10	0.9005	0.9015	0.9329	0.9000	0.9002	1.1000

T4-12	0.9784	0.9822	0.9812	0.9862	0.9769	1.1000
T28-27	0.9596	0.9604	0.9581	0.9617	0.9570	0.9400
C10(p.u.)	0.0489	0.0465	0.0339	0.0395	0.0488	0.2200
C12(p.u.)	0.0094	0.0500	0.0489	0.0479	0.0316	0.3000
C15(p.u.)	0.0493	0.0000	0.0268	0.0356	0.0500	0.1200
C17(p.u.)	0.0455	0.0500	0.0492	0.0486	0.0500	0.0900
C20(p.u.)	0.0138	0.0490	0.0375	0.0357	0.0406	0.0000
C21(p.u.)	0.0461	0.0485	0.0488	0.0468	0.0495	0.1100
C23(p.u.)	0.0483	0.0488	0.0234	0.0483	0.0069	0.0100
C24(p.u.)	0.0484	0.0447	0.0496	0.0490	0.0362	0.0700
C29(p.u.)	0.0022	0.0001	0.0021	0.0013	0.0031	0.3000
P_{loss} (MW)	4.583582	4.589364	4.584608	4.58357	**4.579203**	5.232
L index	0.124814	0.124756	0.124780	0.124768	**0.124749**	0.11821

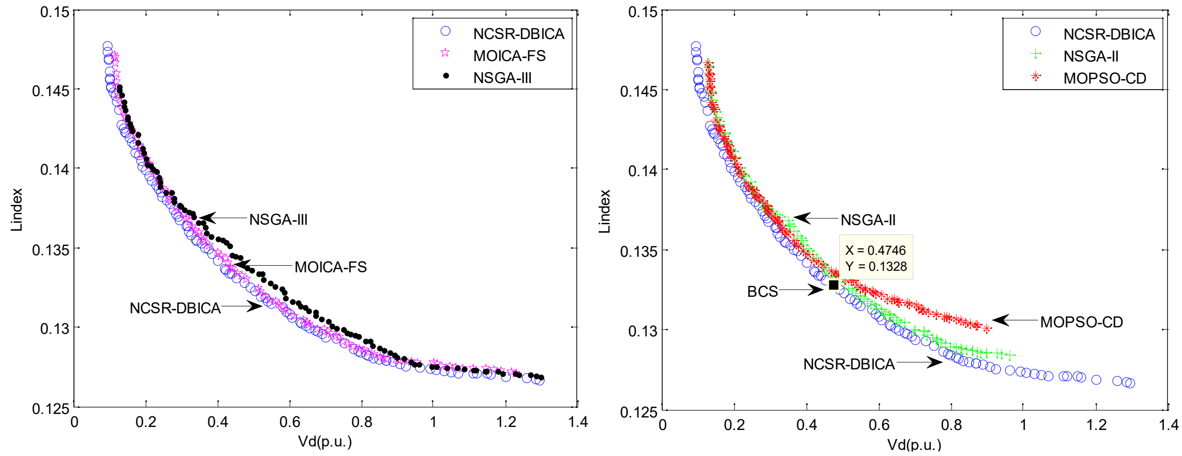


Fig. 10. Pareto fronts obtained by NCSR-DBICA and other four algorithms in Case III

TABLE V Comparison of the BCS for Case III

Variables	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA
VG1(p.u.)	1.0080	1.0390	1.0461	1.0303	1.0386
VG2(p.u.)	0.9987	1.0263	1.0423	1.0230	1.0290
VG5(p.u.)	0.9572	1.0042	0.9966	0.9859	0.9941
VG8(p.u.)	1.0239	1.0283	1.0357	1.0341	1.0299
VG11(p.u.)	1.1000	1.0436	1.0860	1.0955	1.0848
VG13(p.u.)	1.0920	1.0388	1.0239	1.0558	1.0226
T6-9	1.0905	0.9877	1.1000	1.0057	1.0935
T6-10	1.0364	0.996	0.9641	1.0983	0.9110
T4-12	1.0714	1.0792	1.0415	1.0842	1.0250
T28-27	0.9000	0.9001	0.9000	0.9001	0.9010
C10(p.u.)	0.0452	0.0057	0.0235	0.0205	0.0282
C12(p.u.)	0.0264	0.0375	0.0000	0.0021	0.0216
C15(p.u.)	0.0158	0.0406	0.0010	0.0206	0.0446
C17(p.u.)	0.0384	0.0097	0.0500	0.0294	0.0015
C20(p.u.)	0.0298	0.0214	0.0162	0.0499	0.0295
C21(p.u.)	0.0134	0.0079	0.0500	0.0011	0.0033
C23(p.u.)	0.0142	0.0475	0.0486	0.0099	0.0055
C24(p.u.)	0.0500	0.0374	0.0500	0.0483	0.0221
C29(p.u.)	0.0494	0.0484	0.0000	0.0344	0.0500
V_d (p.u.)	0.48950	0.48804	0.49480	0.48107	**0.47465**
L index	0.13339	0.13314	0.13356	0.13298	**0.13277**

4) Discussion of BP, BV and BL in System A

When solving multi-objective problems, the simulation results can not only obtain the BCS, but also get the best solution of a single target. Thus, in this subsection, the best values for the five algorithms are listed, and Tables 6-8 present the best active power losses (BP), the best voltage deviation (BV), and the best L index (BL), respectively. In Table 6, from the comparison of BP obtained by NCSR-DBICA with other reported algorithm such as MGBICA [33], IGSA-CSS [38], MOCIPSO [37] and CLPSO [39], it is found that NCSR-DBICA also has better results in BP values. These can effectively prove that the NCSR-DBICA has certain advantages in this case. Furthermore, it can also be seen from Table 7 that the BV of the NCSR-DBICA is much smaller than that of the MGBICA [33]. In addition, Table 8 can also shows the superiority of the NCSR-DBICA in finding the BL.

TABLE VI BP for System A

Algorithms	Ploss(MW)	Vd (p.u.)	L index
MOPSO-CD	4.65732	1.04718	0.13720
NSGA-II	4.60790	1.59881	0.13218
NSGA-III	4.61923	1.31124	0.13455

MOICA-FS	4.59265	1.51342	0.13305
NCSR-DBICA	**4.53109**	2.05353	0.12547
MGBICA[33]	4.937	0.8026	--
IGSA-CSS[38]	4.76601	--	--
MOCIPSO[37]	5.174	0.12664	--
CLPSO[39]	4.5615	--	--

TABLE VII BV for System A

Algorithms	Ploss (MW)	Vd (p.u.)	L index
MOPSO-CD	5.44396	0.11440	0.14892
NSGA-II	5.30310	0.15332	0.14796
NSGA-III	5.32865	0.13747	0.14733
MOICA-FS	5.52422	0.12106	0.14889
NCSR-DBICA	5.61134	**0.10835**	0.14877
MGBICA[33]	5.6379	0.1239	--

TABLE VIII BL for System A

Algorithms	Ploss (MW)	Vd (p.u.)	L index
MOPSO-CD	4.64215	2.10593	0.12466
NSGA-II	4.63979	2.10427	0.12459
NSGA-III	4.65010	2.07901	0.12467
MOICA-FS	4.68857	2.10861	0.12453
NCSR-DBICA	4.66795	2.07789	**0.12451**

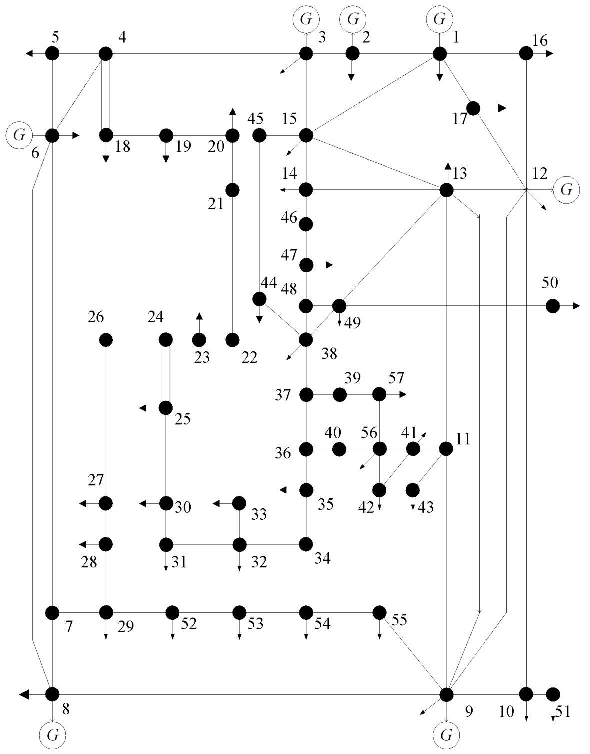


Fig. 11. The single line diagram of the System B

B. *Results for the Test System B*

In this subsection, we will discuss the performance of the proposed NCSR-DBICA algorithm in system B, whose single line diagram can be seen in Figure 11. The detailed data of this system are given in [38, 40]. In system B, the control variable of the ORPD problem is a 27-dimensional vector, it consists of 7 generator bus voltages, 17 transformer taps, and 3 reactive power compensation variables. The minimum and maximum values of all generator buses are set at 0.95 p.u. and 1.05 p.u., the limits of transformer taps are set at 0.9-1.1, and the lower and upper limits of reactive power are set at 0.95 and 1.1 in p.u.

1) *Discussion of Case IV*

In this case, the MOORPD problem is solved through NCSR-DBICA and the other four algorithms with consideration of P_{loss} and V_d. The result of each algorithm in system B is a Pareto figure composed of 100 points. Figure 12 shows the Pareto front of five algorithms mentioned above. The BCS achieved by the NCSR-DBICA, MOICA-FS, NSGA-III, NSGA-II, MOPSO-CD and the reported MGBICA [33] are shown in Table 9. It can be observed that NCSR-DBICA can find a set of BCS, which is better than the results of the MOICA-FS, NSGA-III, NSGA-II, MOPSO-CD. And compared with the MGBICA, the proposed NCSR-DBICA has a larger optimization advantage.

TABLE IX Comparison of the BCS for Case IV

Variables	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA	MGBICA[33]
VG1(p.u.)	1.0836	1.0740	1.0698	1.0761	1.0845	1.0585
VG2(p.u.)	1.0676	1.0590	1.0550	1.0610	1.0688	1.0429
VG3(p.u.)	1.0463	1.0372	1.0364	1.0372	1.0425	1.0231
VG6(p.u.)	1.0390	1.0265	1.0255	1.0224	1.0297	1.0201
VG8(p.u.)	1.0563	1.0425	1.0416	1.0470	1.0459	1.0462
VG9(p.u.)	1.0386	1.0227	1.0238	1.0230	1.0202	1.0115
VG12(p.u.)	1.0295	1.0224	1.0147	1.0103	1.0132	1.0191
T4-18	0.9903	1.0975	1.0814	1.0480	1.0848	0.9300
T4-18	1.0967	0.9991	1.0545	0.9237	0.9164	1.0100
T21-20	1.0329	1.0000	0.9866	1.0083	1.0024	1.0000
T24-25	1.0401	1.0430	1.0999	1.0673	0.9696	--
T24-25	0.9149	0.9266	0.9020	0.9437	1.0108	--
T24-26	1.0113	1.0111	1.0102	1.0141	1.0083	1.0400
T7-29	1.0165	1.0002	1.0053	0.9931	1.0004	0.9800
T34-32	0.9439	0.9398	0.9425	0.9348	0.9402	0.9200
T11-41	0.9099	0.9676	0.9653	0.9000	0.9023	0.9000
T15-45	0.9728	0.9749	0.9639	0.9874	0.9878	0.9500

T14-46	0.9978	0.9714	0.9669	0.972	0.9841	0.9500
T10-51	1.0129	0.9828	0.9797	0.9818	0.9966	0.9800
T13-49	0.9464	0.9379	0.9478	0.9474	0.9139	0.9400
T11-43	0.9855	0.9194	0.9308	0.9721	0.9565	0.9600
T40-56	1.0046	0.9818	1.0732	0.9843	1.0224	1.0400
T39-57	0.9618	0.9787	0.9092	0.9318	0.9430	0.9500
T9-55	1.0235	1.0008	0.9996	0.9883	0.9957	0.9800
C18(p.u.)	0.2102	0.2050	0.2989	0.0068	0.0323	0.0300
C25(p.u.)	0.1266	0.1247	0.1521	0.1554	0.1387	0.0600
C53(p.u.)	0.1718	0.1568	0.1676	0.0890	0.1307	0.0300
Ploss (MW)	24.12873	24.33648	24.65094	24.62639	**24.59119**	25.3664
Vd (p.u.)	0.955922	0.878561	0.830849	0.836184	**0.818256**	0.83711

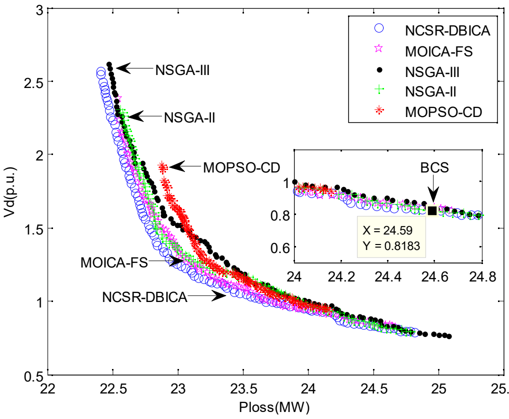


Fig. 12. Pareto fronts obtained by NCSR-DBICA and other four algorithms in Case IV

2) *Discussion of Case V*

As shown in Figure 13, the distribution of the Pareto front obtained by the five algorithms is obviously not as good as the distribution of the Pareto fronts obtained in the previous cases. It can be clearly observed that the MOPSO-CD falls into local optimum, which is also the direct cause of the algorithm cannot find better BCS. Although the NSGA-II, NSGA-III and MOICA-FS have found the Pareto front, it is obvious that their distribution and approximation are inferior to the NCSR-DBICA methodology.

The detailed data in Table 10 can further demonstrate the superiority of the NCSR-DBICA methodology. It can be seen that the BCS obtained by the NCSR-DBICA methodology are better than the results of other four algorithms. Compared with the BCS of the published MOCIPSO [37], although the L index of NCSR-DBICA methodology is higher than MOCIPSO by 0.01728, the P_{loss} of NCSR-DBICA methodology is reduced by about 4.788(MW). The simulation results of this case can effectively demonstrate the superiority of NCSR-DBICA methodology in dealing with complex objective function combinations and large-scale ORPD systems.

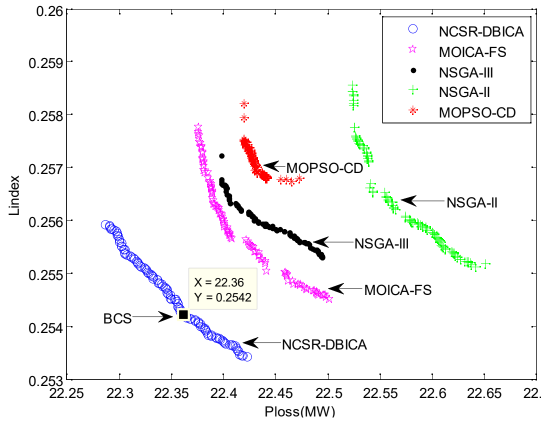


Fig. 13. Pareto fronts obtained by NCSR-DBICA and other four algorithms in Case V

3) *Discussion of BP, BV and BL in System B*

Compared to system A, the optimization effect of the algorithm will be more obvious in system B. It can be observed from Tables 11-13 that the proposed algorithm shows a greater advantage in finding BP, BV and BL. It can be found that BP, BV and BL are 22.40864 (MW), 0.785003 (p.u.) and 0.253418 for NCSR-DBICA. Compared with the reported MGBICA [32], MCS-DE [40] and MOCIPSO [36] in Table 11, it point out that the BP obtained by the NCSR-DBICA is reduced by 2.47766 (MW), 0.86036 (MW) and 4.66636 (MW), respectively. The performance of NCSR-DBICA in finding the optimal solution can also be confirmed by the BV in Table 12. In addition, the BL values in Table 13 also verify the optimization ability of the NCSR-DBICA.

TABLE X Comparison of the BCS for Case V

Variables	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA	MOCIPSO[33]
VG1(p.u.)	1.0999	1.0999	1.1000	1.1000	1.1000	1.1000
VG2(p.u.)	1.0892	1.0878	1.0911	1.0893	1.0925	0.9352
VG3(p.u.)	1.0772	1.0723	1.0849	1.0775	1.0901	0.9000
VG6(p.u.)	1.0710	1.0658	1.0807	1.0731	1.0833	0.9000
VG8(p.u.)	1.0904	1.0823	1.0998	1.0891	1.1000	1.0700
VG9(p.u.)	1.0703	1.0528	1.0820	1.0607	1.0869	0.9000
VG12(p.u.)	1.0670	1.0595	1.0761	1.0685	1.0819	0.9624
T4-18	1.0498	1.0704	1.0902	0.9663	1.0276	0.9600
T4-18	1.0364	0.9054	0.9050	1.0204	0.9462	0.9000
T21-20	1.0917	1.0166	1.0029	1.0241	1.0271	1.0100
T24-25	1.0643	1.0170	1.0809	0.9693	1.0366	1.1000
T24-25	1.0694	1.0384	1.0698	1.0943	1.0684	1.1000
T24-26	1.0203	0.9867	1.0125	1.0059	1.0089	1.0100
T7-29	0.9801	0.9728	0.9858	0.9762	0.9874	0.9300
T34-32	0.9157	0.9286	0.9202	0.9248	0.9429	0.9000
T11-41	0.9720	0.9000	1.0829	0.9341	0.9378	0.9700
T15-45	0.9708	0.9634	0.9779	0.9688	0.9814	0.9500

T14-46	0.9504	0.9458	0.9583	0.9491	0.9629	0.9000
T10-51	0.9599	0.9540	0.9676	0.9561	0.9740	0.9700
T13-49	0.9233	0.9187	0.9287	0.9256	0.9330	0.9000
T11-43	0.9499	0.9432	0.9458	0.9470	0.9683	0.9000
T40-56	1.0020	0.9942	1.0812	1.0069	1.0174	1.0800
T39-57	0.9692	0.9832	1.0467	0.9842	0.9904	1.0200
T9-55	0.9822	0.9545	0.9826	0.9670	0.9883	0.9200
C18(p.u.)	0.0713	0.0715	0.0612	0.1296	0.0185	0.0000
C25(p.u.)	0.1785	0.1542	0.1887	0.1413	0.1716	0.1800
C53(p.u.)	0.1242	0.1154	0.1124	0.1277	0.1215	0.0480
Ploss (MW)	22.42967	22.54289	22.40651	22.40778	**22.3598**	27.122
L index	0.256937	0.256536	0.256322	0.255641	**0.254233**	0.23695

TABLE XI BP for System B

Algorithms	Ploss (MW)	Vd (p.u.)	L index
MOPSO-CD	22.87811	1.928749	0.271031
NSGA-II	22.56162	2.306274	0.270848
NSGA-III	22.47279	2.615120	0.267883
MOICA-FS	22.54859	2.384117	0.272176
NCSR-DBICA	**22.40864**	2.566672	0.268255
MGBICA[33]	24.8863	1.0283	--
MCS-DE[41]	23.269	--	--
MOCIPSO[37]	27.075	0.24274	--

TABLE XII BV for System B

Algorithms	Ploss (MW)	Vd (p.u.)	L index
MOPSO-CD	24.16134	0.938457	0.290774
NSGA-II	0.247881	0.795240	0.293412
NSGA-III	25.07677	0.764097	0.293808
MOICA-FS	24.67251	0.822601	0.293023
NCSR-DBICA	24.81992	**0.785003**	0.293090

TABLE XIII BL for System B

Algorithms	Ploss (MW)	Vd (p.u.)	L index
MOPSO-CD	22.46472	3.332892	0.256734
NSGA-II	22.64132	3.512528	0.255118
NSGA-III	22.49395	3.653143	0.255305
MOICA-FS	22.50166	3.543053	0.254511
NCSR-DBICA	22.42215	3.707225	**0.253418**

C. Results for the Test System C

To demonstrate the performance of the NCSR-DBICA methodology in dealing with the ORPD problem in large system, the aforementioned five algorithms are applied to system C to solve the ORPD problem. The single line diagram of system C can be seen in Figure 14. The detailed data of this system is provided in [40].The improved reactive power generation limits and the transmission apparent power flow limits can be found in [42]. In this system, the control variable of the ORPD problem is a 75-dimensional vector, including: 54 generator bus voltages, 9 transformer taps, and 12 reactive power compensation variables. The lower and upper limits for generator bus voltages amplitude are 0.95 p.u. and 1.05 p.u., the limits of transformer taps are set at 0.9-1.1, and the limits of reactive power are set at 0.95-1.1 in p.u.

1) Discussion of Case VI

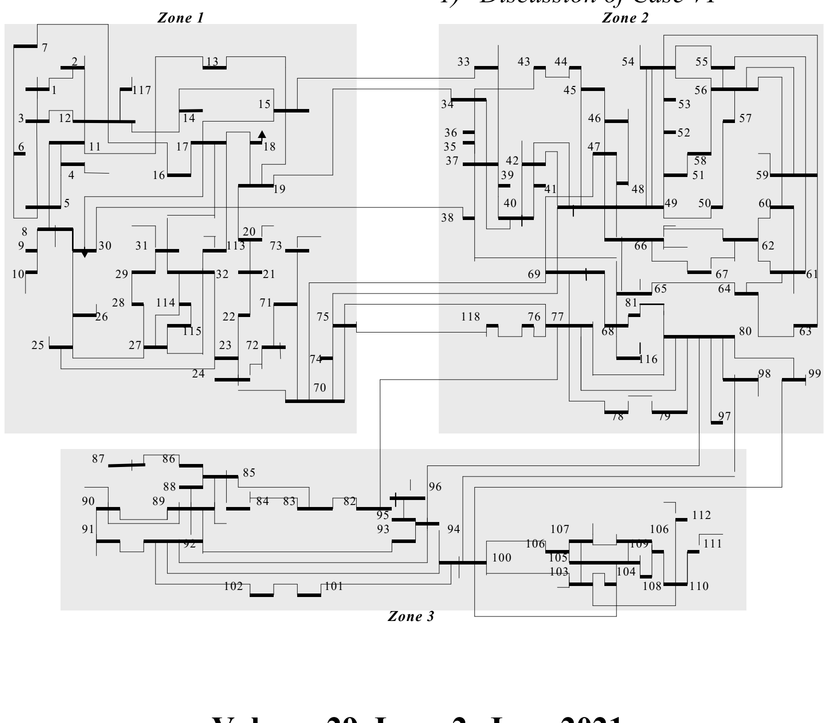


Fig. 14. The single line diagram of the System C

TABLE XIV Comparis on of the BCS for Case VI

Variables	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA
VG1(p.u.)	0.9986	0.9897	0.9989	1.0016	1.0008
VG4(p.u.)	1.0154	1.0139	1.0107	1.0125	1.0168
VG6(p.u.)	1.0042	0.9952	1.0097	1.0012	0.999
VG8(p.u.)	0.9933	0.9749	0.9828	0.9916	0.995
VG10(p.u.)	1.0138	1.0162	1.0147	1.0067	1.0194
VG12(p.u.)	1.0033	1.0133	1.0071	1.006	1.0029
VG15(p.u.)	1.0013	1.0046	1.0011	0.9998	1.0061
VG18(p.u.)	1.0052	1.0275	0.9959	0.999	1.0067
VG19 (p.u.)	1.0072	1.0156	1.0017	1.0077	1.0089
VG24(p.u.)	1.0343	0.9954	0.9921	1.0243	1.0198
VG25(p.u.)	1.0565	1.0024	1.058	1.0421	1.0411
VG26(p.u.)	1.0652	0.9941	1.0708	1.0373	1.0436
VG27(p.u.)	1.0095	1.0171	1.0117	1.0033	1.0065
VG31(p.u.)	0.9991	1.0030	1.0003	1.007	1.0037
VG32(p.u.)	1.0013	0.9894	1.0046	1.0066	1.0052
VG34(p.u.)	1.0129	1.0113	1.0101	1.0107	1.0114
VG36(p.u.)	1.0072	1.0026	1.0013	1.0076	1.0031
VG40(p.u.)	1.0022	1.0034	1.0092	1.001	1.0007
VG42(p.u.)	1.0098	1.0328	1.0092	1.0192	1.0071
VG46(p.u.)	1.0145	1.0110	1.0256	1.0246	1.0114
VG49(p.u.)	1.0235	1.0035	1.0129	1.0312	1.023
VG54(p.u.)	1.0128	1.0219	1.0157	1.0107	1.019
VG55(p.u.)	1.0073	1.0208	1.0146	1.004	1.0154
VG56(p.u.)	1.0085	1.0112	1.0157	1.0048	1.0155
VG59(p.u.)	1.0091	1.0335	1.0191	1.0228	1.0273
VG61(p.u.)	1.0034	1.0018	0.9988	1.0016	1.0017
VG62(p.u.)	0.9995	0.9912	0.9839	1.003	0.9969
VG65(p.u.)	1.0113	1.0050	1.0012	1.0094	1.0171
VG66(p.u.)	1.0331	1.0210	1.0089	1.0395	1.0316
VG69(p.u.)	1.0405	1.0302	1.0183	1.0457	1.0407
VG70(p.u.)	1.0051	1.0040	1.0016	0.9984	1.0026
VG72(p.u.)	1.0394	0.9794	1.0171	1.0163	1.0032
VG73(p.u.)	0.9842	1.0050	1.0076	1.0011	0.9956
VG74(p.u.)	1.0018	1.0030	0.9973	0.993	0.9979
VG76(p.u.)	0.9963	1.0044	1.0025	1.0092	0.9872
VG77(p.u.)	1.0061	1.0048	1.0119	1.0047	1.0051
VG80(p.u.)	1.0121	1.0211	1.0125	1.0186	1.0171
VG85(p.u.)	1.0062	1.0040	1.0068	1.0173	1.0138
VG87(p.u.)	1.0694	1.0243	0.9987	0.9877	1.0063
VG89(p.u.)	1.045	1.0272	1.0328	1.0401	1.0347
VG90(p.u.)	1.0449	1.0112	0.999	1.0116	1.0193
VG91(p.u.)	1.0463	1.0100	1.0126	1.0139	1.0166
VG92(p.u.)	1.0194	1.0179	1.018	1.0182	1.0186
VG99(p.u.)	1.0273	0.9899	1.0307	1.0163	1.0163
VG100(p.u.)	1.025	1.0126	1.0312	1.0226	1.0212
VG103(p.u.)	1.0223	1.0047	1.0203	1.0142	1.0194
VG104(p.u.)	1.0216	0.9896	1.0233	1.0116	1.0202
VG105(p.u.)	1.0073	1.0005	1.0092	1.009	1.0073
VG107(p.u.)	1.0021	1.0101	0.9732	0.9889	0.9982
VG110(p.u.)	0.9988	0.9798	1.0156	0.9903	0.9953
VG111(p.u.)	1.01	1.0082	1.0275	0.9956	0.9953
VG112(p.u.)	0.9793	0.9600	0.9929	0.9706	0.9838
VG113(p.u.)	1.0101	1.0042	1.0024	1.0085	1.0077
VG116(p.u.)	0.9928	1.0108	0.9926	1.0007	0.9976
T8	0.9624	0.9420	0.9665	0.9588	0.9786
T32	1.0035	1.0321	0.9852	1.0371	1.0517
T36	0.9521	1.0020	0.9551	0.9723	0.9738
T51	0.9698	0.9812	0.96	0.9738	0.9611
T93	1.0774	0.9804	0.9823	0.9723	0.9713
T95	0.9558	0.9853	1.0188	0.999	0.9889
T102	0.9754	1.0176	1.0682	1.0232	1.0326
T107	1.0358	1.0473	0.986	0.9552	0.9449
T127	0.9957	0.9378	1.0333	0.9531	0.9655
C34(p.u.)	0.0265	0.0522	0.0832	0.0149	0.0177
C44(p.u.)	0.0396	0.2417	0.096	0.0956	0.1343
C45(p.u.)	0.2174	0.0631	0.1201	0.0501	0.1015
C46(p.u.)	0.2845	0.1418	0.2254	0.1004	0.2808
C48(p.u.)	0.0121	0.1548	0.1531	0.1906	0.1426
C74(p.u.)	0.0155	0.0854	0.1523	0.1841	0.1635
C79(p.u.)	0.299	0.1180	0.1844	0.2562	0.2755
C82(p.u.)	0.2553	0.1929	0.257	0.0181	0.1889
C83(p.u.)	0.2903	0.2945	0.1488	0.1432	0.1408

C105(p.u.)	0.1491	0.1223	0.0727	0.1942	0.0773
C107(p.u.)	0.1828	0.2412	0.0736	0.0352	0.1477
C110(p.u.)	0.126	0.2946	0.0338	0.0504	0.2812
Ploss (MW)	126.1035	132.0426	128.4577	124.5649	**124.3505**
Vd (p.u.)	0.410095	0.325706	0.4098	0.380205	**0.349527**

We all know that the size of the system will directly affect the complexity of the problem. System C has 75 reactive optimization control variables, which is much higher than the variable dimensions of system A and system B. Therefore, the MOORPD problem in this case is more complicated. However, the Pareto front presented in Figure 15 shows that the proposed NCSR-DBICA algorithm can still find a well distributed and stable Pareto front. Although the distribution of this algorithm is not as extensive as MOICA-FS algorithm, it is clear that the NCSR-DBICA algorithm can find better BCS, BP and BV. At the same time, it can be seen that the distribution of the Pareto front of NSGA-III and NSGA-II is obviously poor, and the shortcoming of MOPSO-CD which is easy to fall into local optimum is more obvious in system C. The data in Tables 14-16 also illustrates the superiority of NCSR-DBICA in large-node systems. Table 14 shows that the BCS obtained by NCSR-DBICA is superior to the other four algorithms. And it can be seen in Table 15, comparing the published NGBWCA [43], GSA [44] and CLPSO [39] algorithms, the BP value of the NCSR-DBICA algorithm is the best. The BV value of Table 16 can also explain the optimization ability of NCSR-DBICA algorithm. Not only BV is smaller than BV of NGBWCA [43], WCA [43] and OGSA [45], but the P_{loss} value corresponding to BV is also much smaller than that of other three published algorithms.

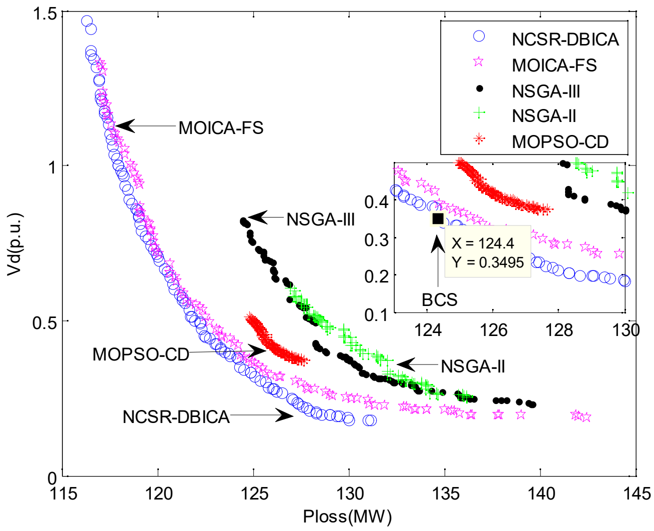


Fig. 15. Pareto front obtained by NCSR-DBICA and other four algorithms in Case VI

TABLE XV BP for Case VI

Algorithms	Ploss (MW)	Vd (p.u.)
MOPSO-CD	124.8260	0.512675
NSGA-II	126.9328	0.603065
NSGA-III	124.4407	0.826786
MOICA-FS	117.0452	1.33008
NCSR-DBICA	**116.3312**	1.465746
NGBWCA[43]	121.47	1.452
GSA[44]	127.7603	--
CLPSO[39]	130.96	--

TABLE XVI BV for Case VI

Algorithms	Ploss (MW)	Vd (p.u.)
MOPSO-CD	127.6464	0.37358
NSGA-II	136.1902	0.254913
NSGA-III	139.6224	0.232104
MOICA-FS	142.4927	0.188804
NCSR-DBICA	131.1776	**0.179212**
NGBWCA[43]	152.31	0.3194
WCA[43]	165.71	0.3752
OGSA[45]	157.72	0.3666

D. *Performance Evaluation*

The performance evaluation indicators of multi-objective algorithms can be roughly divided into three categories: the evaluation of the degree of approximation of the results and the true Pareto front, the convergence evaluation of the algorithm and the diversity evaluation of the solution. And the best one Pareto front obtained from the five algorithms is regarded as the true Pareto front. Thus, the GD and HV are selected to evaluation the performance of NCSR-DBICA, MOICA-FS, NSGA-III, NSGA-II and MOPSO-CD. In addition, to analyze the stability and convergence of the algorithm, the 30 or 20 experiment results obtained by the algorithm and their convergence of the Pareto front at different iterations will be analyzed.

1) *Statistical Analysis*

The GD is used to describe the distance between the non-dominated solution obtained by the algorithm and the true Pareto front. In our research, the best one Pareto front will be selected as the true Pareto front in each case. For GD, the smaller the value is, the closer to the true Pareto front. For the HV evaluation method, it was first proposed by Zitzler, which represents the volume formed by the individual in the solution set and the reference point [29, 46]. And this indicator can simultaneously evaluate the convergence and distribution of the solution set. It's worth noting that the larger the HV value is, the better the Pareto front is.

In order to analyze the performance of the simulation algorithms, a statistical analysis of 30 or 20 experiments is performed using the above two indicators, and the box plots will be used to illustrate the statistical results of the GD and HV indicators. Figures 16-17 show the box plots of GD and HV for five algorithms. Furthermore, Table 17 shows the statistical data of the NCSR-DBICA, MOICA-FS, NSGA-III, NSGA-II and MOPSO-CD algorithms for test Cases I-VI.

TABLE XVII The mean and standard deviation of GD and HV for five algorithms

Algorithms	Indictor		Case I	Case II	Case III	Case IV	Case V	Case VI
MOPSO-CD	GD	Mean	0.4240	0.4084	0.1303	0.6044	1.1305	0.6202
		Std	0.2405	0.2357	0.0338	0.3954	0.0991	0.1915
	HV	Mean	0.9429	0.0002	0.0268	1.1676	0.0362	22.393
		Std	0.2594	8.7E-05	0.0007	0.2815	0.0053	3.1509
NSGA-II	GD	Mean	0.0453	0.5186	0.1658	0.6120	0.8759	0.3773

		Std	0.0366	0.1026	0.0345	0.2283	0.1946	0.2110
	HV	Mean	1.3636	0.0002	0.0235	2.4793	0.0135	25.526
		Std	0.0770	5.1E-05	0.0009	0.2914	0.0085	2.2551
	GD	Mean	0.0705	0.4307	0.0946	0.8675	0.8157	0.3010
NSGA-III		Std	0.0735	0.1232	0.0271	0.2305	0.2102	0.0785
	HV	Mean	1.0652	0.0004	0.0315	2.8371	0.0168	28.234
		Std	0.0360	0.0001	0.0007	0.5931	0.0090	2.6867
	GD	Mean	0.0426	0.3547	0.0397	0.9053	0.3576	0.3312
MOICA-FS		Std	0.0270	0.0989	0.0198	0.1261	0.2213	0.1503
	HV	Mean	1.3475	0.0003	0.0303	2.5303	0.0117	31.609
		Std	0.0187	5.7E-05	0.0005	0.3176	0.0094	2.8191
	GD	Mean	0.0337	0.1209	0.0234	0.1967	0.1398	0.0945
NCSR-DBICA		Std	0.0137	0.0517	0.0079	0.0938	0.0790	0.0251
	HV	Mean	1.9843	0.0007	0.0385	5.1232	0.0478	38.995
		Std	0.0189	3.2E-05	0.0003	0.1520	0.0016	0.1563

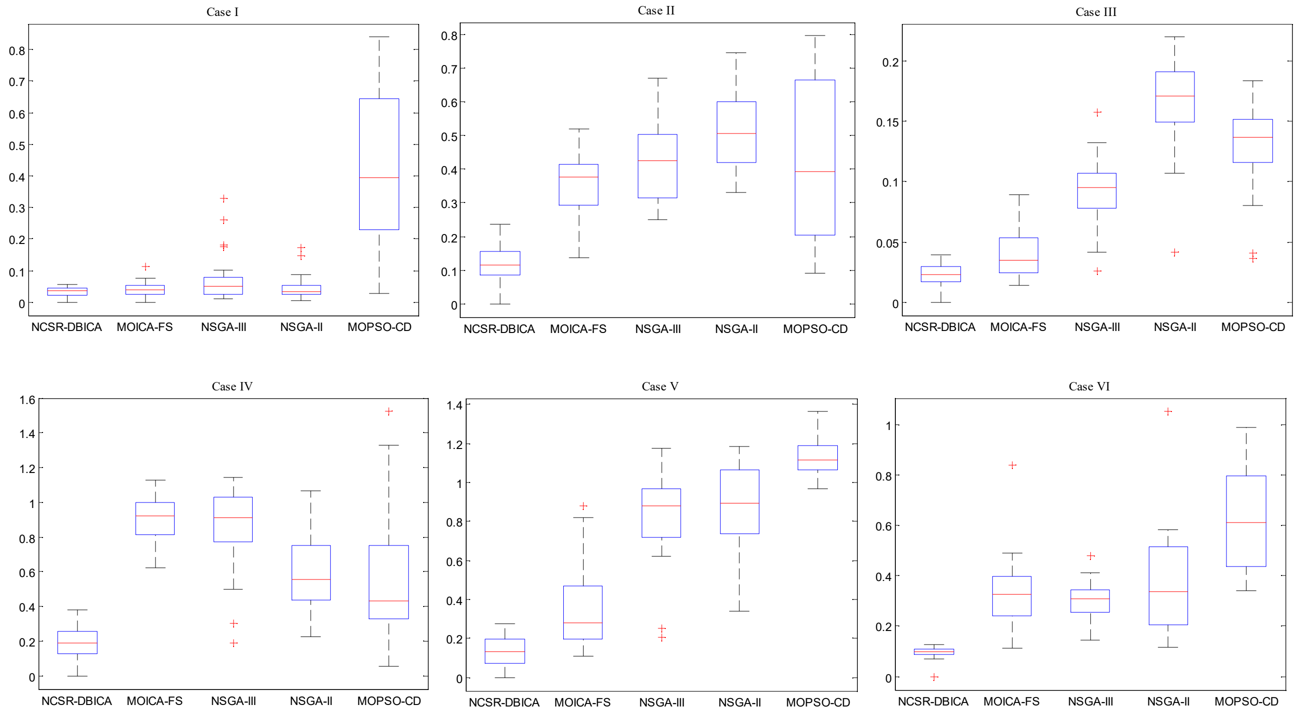


Fig. 16. Box plots of GD for NCSR-DBICA and other four algorithms.

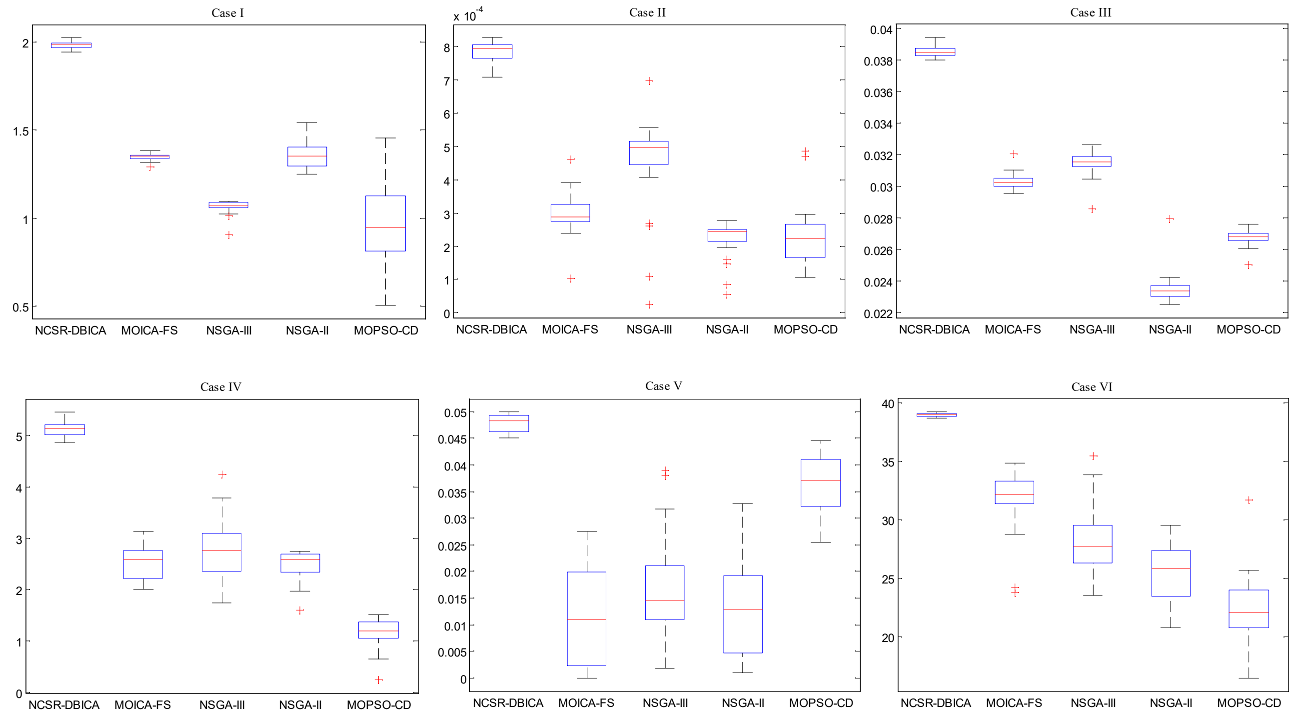


Fig. 17. Box plots of HV for NCSR-DBICA and other four algorithm

Figure 16 shows six box plots of the GD for five algorithms in six cases. By analyzing the box plot of Case I, it can be seen that the average value of NCSR-DBICA is lower than the other four algorithms, which proves that the Pareto front of the NCSR-DBICA is closest to the true Pareto front among the five algorithms. In addition, NCSR-DBICA does not have outliers, MOICA-FS, NSGA-III and NSGA-II have 1, 4 and 2 outliers, respectively, which proves the stability of the NCSR-DBICA algorithm. Similarly, the superiority of the NCSR-DBICA algorithm can also be derived by analyzing Cases II-VI. Besides, through the HV box plots of the five algorithms of Figure 17, it can be inferred that the convergence and distribution of NCSR-DBICA are the best in all six cases. In addition, it can also be seen form Table 17 that the mean and standard deviation of GD and HV for the NCSR-DBICA algorithm are both optimal. The mean CPU time of five algorithms for test systems of Cases I-VI are shown in Table 18.

TABLE XVIII The mean CPU time of five algorithms for Cases I-VI

Test Case	Mean CPU Time (sec)/Tmax				
	MOPSO-CD	NSGA-II	NSGA-III	MOICA-FS	NCSR-DBICA
Case I	339.44/500	336.46/500	345.86/500	396.25/500	338.42/500
Case II	328.02/500	325.43/500	360.88/500	393.43/500	337.46/500
Case III	310.89/500	297.72/500	326.18/500	376.60/500	312.26/500
Case IV	521.00/500	492.58/500	515.09/500	517.66/500	492.35/500
Case V	510.09/500	503.10/500	519.90/500	524.27/500	495.32/500
Case VI	2043.0/600	2011.8/600	2156.5/600	2263.9/600	1956.6/600

2) Stability and Convergence Analysis

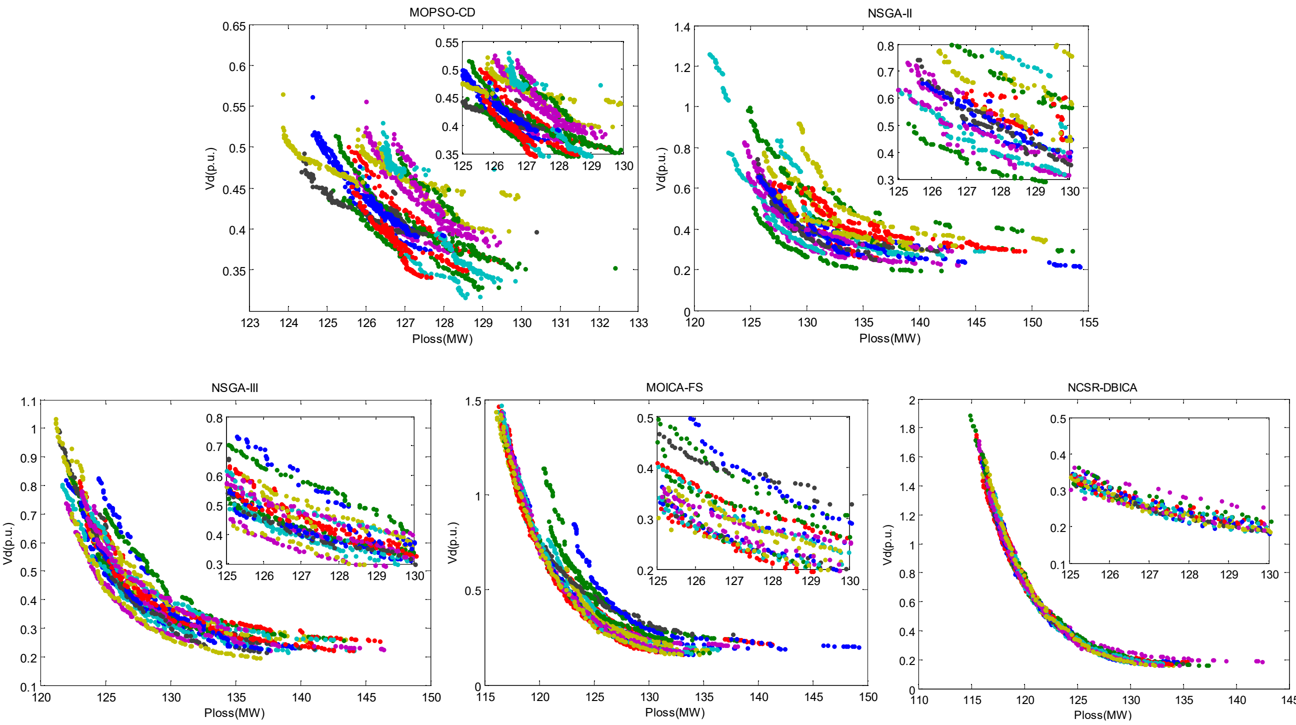
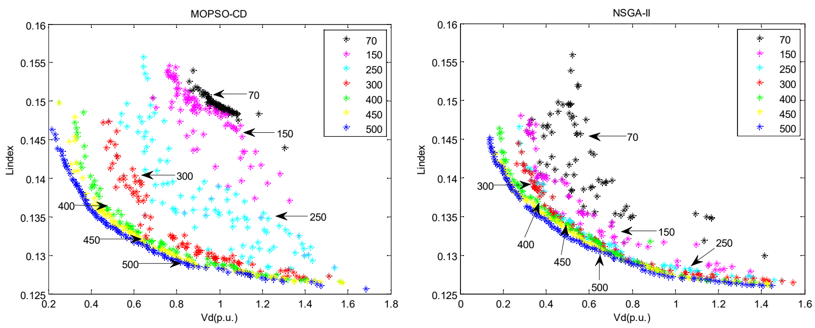


Fig. 18. 20 independent experimental results of the NCSR-DBICA and other four algorithms in Case VI



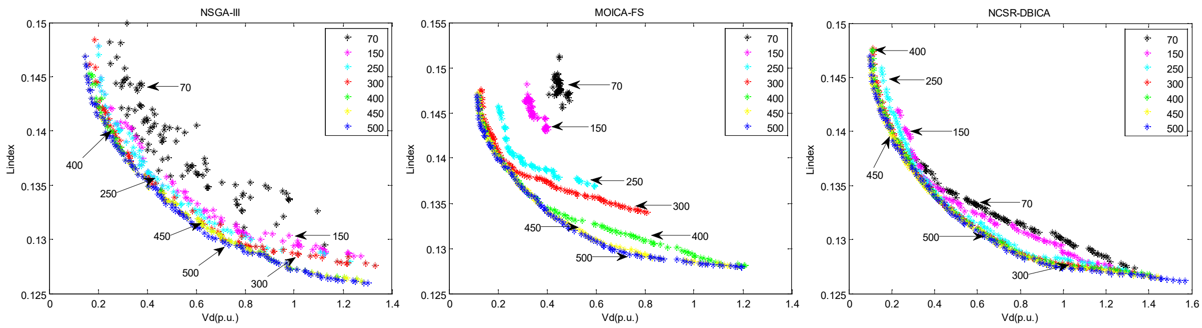


Fig. 19. Pareto fronts of the NCSR-DBICA and other four algorithms over different iterations for Case III

In this subsection, the stability and convergence of the algorithms will be further analyzed. Figure 18 shows the Pareto fronts obtained by the five algorithms in case VI. The stability of the algorithm can be visually observed through the comparison of experimental results. It can be seen from Figure 18 that the distribution of 20 independent experimental results of the MOPSO-CD is very scattered, which indicates that the stability of the algorithm is poor. The distribution of the results obtained by the NCSR-DBICA is the most concentrated among the five algorithms, which shows that the stability of the algorithm is better than that of the other four algorithms.

To analyze the convergence of the algorithm, the Pareto fronts of different iterations of MOPSO-CD, NSGA-II, NSGA-III, MOICA-FS and NCSR-DBICA are recorded and saved. Figure 19 shows the Pareto fronts of the five algorithms over different iterations of Case III. It can be seen that the MOPSO-CD and the NSGA-II can find a clear Pareto front when they evolve to the 400th generation, the NSGA-III and the MOICA-FS can find a suitable Pareto front in the evolutions to 250th generation. For NCSR-DBICA, a satisfactory Pareto front can be found in the 70th generation. In addition, the NCSR-DBICA is stable in the 300th generation, but the other four algorithms are stable at least in the 400th generation, which proves that the NCSR-DBICA has the fastest convergence speed among the five algorithms.

V. Conclusion

In this paper, the proposed NCSR-DBICA methodology with its sub-methods: the dynamic balance strategy, the constraint-based country stratification mechanism and the novel ranking method based on country satisfaction and country distance has successfully solved the MOORPD problem. The simulation results of different objective function combinations considering the active power losses, voltage deviation and voltage stability index on the IEEE 30, 57 and 118 bus test systems have proved the effectiveness of the improved algorithm. Compared with the experimental results of MOICA-FS, NSGA-III, NSGA-II and MOPSO-CD, it can be found that the BCS, BP, BV and BL of the NCSR-DBICA method are better than that of other four algorithms, especially in the IEEE118 bus test system. In addition, the performance evaluation results of the five algorithms also show that the diversity, stability and convergence of the NCSR-DBICA methodology are the best. Therefore, through all the simulation results, it can be concluded that the proposed NCSR-DBICA is effective and superior in dealing with MOORPD problem.

References

[1] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian and A. Habibi, "A new hybrid algorithm for optimal reactive power dispatch problem with discrete and continuous control variables", *Applied Soft Computing*, vol.22, no.5, pp.126-140, 2014.

[2] P. Subbaraj and P. N. Rajnarayanan, "Hybrid particle swarm optimization based optimal reactive power dispatch", *International Journal of Computer Applications*, vol.1, no.5, pp.79-85, 2010.

[3] G. Chen, J. Qian, Z. Zhang, and Z. Sun, "Multi-objective improved bat algorithm for optimizing fuel cost, emission and active power loss in power system," *IAENG International Journal of Computer Science,* vol. 46, no.1, pp. 118-133, 2019.

[4] Z. Chen, X. Yuan, J. I. Bin, P. Wang and H. Tian, "Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II", *Energy Conversion & Management*, vol.84, no., pp.390-404, 2014.

[5] G. Chen, S. Qiu, Z. Zhang, S. Zhi and H. Liao, "Optimal power flow using Gbest-Guided cuckoo search algorithm with feedback control strategy and constraint domination rule", *Mathematical Problems in Engineering*,2017, vol.2017, no.2, pp.1-14, 2017.

[6] L. Hongxin, L. Yinhong and C. Jinfu, "Adaptive multiple evolutionary algorithms search for multi-objective optimal reactive power dispatch", *International Transactions on Electrical Energy Systems*, vol.24, no.6, pp.780-795, 2014.

[7] G. Chen, X. Yi, Z. Zhang and H. Lei, "Solving the multi-objective optimal power flow problem using the multi-objective firefly algorithm with a constraints-prior pareto-domination approach", *Energies*, vol.11, no.12, pp.1-18, 2018.

[8] X. Yuan, B. Zhang, P. Wang, J. Liang, Y. Yuan, Y. Huang and X. Lei, "Multi-objective optimal power flow based on improved strength pareto evolutionary algorithm", *Energy*, vol.122, pp.70-82, 2017.

[9] K. Deb, M. Mohan and S. Mishra, "Evaluating the ϵ-dominance based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions", *Evolutionary Computation*, vol.13, no.4, pp.501-525, 2014.

[10] Li S, Jiao L, Zhang Y, et al. "A scheme of resource allocation for heterogeneous services in peer-to-peer networks using particle swarm optimization," IAENG International Journal of Computer Science, vol.44, no.4, pp.482-488, 2017.

[11] L. Zhihuan, L. Yinhong and D. Xianzhong, "Non-dominated sorting genetic algorithm-ii for robust multi-objective optimal reactive power dispatch", *IET Generation, Transmission & Distribution*, vol.4, no.9, pp.1000-1008, 2010.

[12] X. Yuan, B. Ji, Y. Yuan, et al, "An efficient chaos embedded hybrid approach for hydro-thermal unit commitment problem", *Energy*

Conversion & Management, vol.91, pp.225-237, 2015.

[13] G. Chen, L. Liu and S. Huang, "Enhanced GSA-Based optimization for minimization of power losses in power system", *Mathematical Problems in Engineering*, vol.2015, no.1, pp.1-13, 2015.

[14] M. Basu, "Multi-objective optimal reactive power dispatch using multi-objective differential evolution", *International Journal of Electrical Power & Energy Systems*, vol.82, pp.213-224, 2016.

[15] K. Nuaekaew, P. Artrit, N. Pholdee and S. Bureerat, "Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer", *Expert Systems with Applications*, vol.87, pp.79-89, 2017.

[16] S. M. Mohseni-Bonab, A. Rabiee, B. Mohammadi-Ivatloo, S. Jalilzadeh and S. Nojavan, "A two-point estimate method for uncertainty modeling in multi-objective optimal reactive power dispatch problem", *International Journal of Electrical Power & Energy Systems*, vol.75, pp.194-204, 2016.

[17] S. Hosseini and A. Al Khaled, "A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research", *Applied Soft Computing*, vol.24, pp.1078-1094, 2014.

[18] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition", *IEEE Congress on Evolutionary Computation*, vol.7, pp.4661-4667, 2007.

[19] Z. Sherinov and A. Ünveren, "Multi-objective imperialistic competitive algorithm with multiple non-dominated sets for the solution of global optimization problems", *Soft Computing*, no.2, pp.1-16, 2017.

[20] M. R. Maheri and M. Talezadeh, "An enhanced imperialist competitive algorithm for optimum design of skeletal structures", *Swarm and Evolutionary Computation*, vol.40, pp.24-36, 2018.

[21] M. H. Moradi, A. Zeinalzadeh, Y. Mohammadi and M. Abedini, "An efficient hybrid method for solving the optimal sitting and sizing problem of DG and shunt capacitor banks simultaneously based on imperialist competitive algorithm and genetic algorithm", *International Journal of Electrical Power & Energy Systems*, vol.54, no.1, pp.101-111, 2014.

[22] R. Enayatifar, M. Yousefi, A. H. Abdullah and A. N. Darus, "MOICA: a novel multi-objective approach based on imperialist competitive algorithm", *Applied Mathematics & Computation*, vol.219, no.17, pp.8829-8841, 2013.

[23] G. Chen, J. Cao and Z. Zhang, "Application of global best imperialist competition algorithm for multi-objective reactive power optimization", *Chinese Automation Congress*, Xian, China, pp. 1240-1245, 2018.

[24] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based Nondominated sorting approach, part i: solving problems with box constraints", *IEEE Transactions on Evolutionary Computation*, vol.18, no.4, pp.577-601, 2014.

[25] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part ii: handling constraints and extending to an adaptive approach", *IEEE Transactions on Evolutionary Computation*, vol.18, no.4, pp.602-622, 2014.

[26] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol.6, no.2, pp.182-197, 2002.

[27] P. Chakraborty, S. Das, A. Abraham, V. Snásel and G. G. Roy, "On convergence of multi-objective particle swarm optimizers", *Evolutionary Computation. IEEE*, 2010.

[28] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, vol.8, no.3, pp.256-279, 2004.

[29] J. Bader and E. Zitzler, "Hype: an algorithm for fast hypervolume-based many-objective optimization", *Evolutionary Computation*, vol.19, no.1, pp.45-76, 2011.

[30] L. While, L. Bradstreet and L. Barone, "A fast way of calculating exact hypervolumes", *IEEE Transactions on Evolutionary Computation,* vol.16, no.1, pp.86-95, 2012.

[31] G. Chen, J. Cao, Z. Zhang, and Z. Sun, "Application of Imperialist Competitive Algorithm with its Enhanced Approaches for Multi-objective Optimal Reactive Power Dispatch Problem", *Engineering Letters*, vol.27, no.3, pp.579-592, 2019.

[32] S. Abbasbandy, "Improving newton-raphson method for nonlinear equations by modified adomian decomposition method", *Applied Mathematics & Computation*, vol.145, no.2-3, pp.887-893, 2003.

[33] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian and M. Gitizadeh, "Multi-objective optimal electric power planning in the power system using gaussian bare-bones imperialist competitive algorithm", *Information Sciences*, vol.294, no.10, pp.286-304, 2015.

[34] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian, M. Gharibzadeh and A. Azizi Vahed, "Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm", *Energy*, vol.78, no.78, pp.276-289, 2014.

[35] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian, H. R. Massrur and M. Gharibzadeh, "Application of imperialist competitive algorithm with its modified techniques for multi-objective optimal power flow problem: a comparative study", *Information Sciences*, vol.281, no., pp.225-247, 2014.

[36] K. Y. Lee, Y. M. Park and J. L. Ortiz, "A united approach to optimal real and reactive power dispatch", *IEEE Power Engineering Review*, vol.PAS-104, no.5, pp.1147-1153, 2010.

[37] G. Chen, L. Liu, P. Song and Y. Du, "Chaotic improved pso-based multi-objective optimization for minimization of power losses and l index in power systems", *Energy Conversion & Management*, vol.86, no.10, pp.548-560, 2014.

[38] G. Chen, L. Liu, Z. Zhang and S. Huang, "Optimal reactive power dispatch by improved gsa-based algorithm with the novel strategies to handle constraints", *Applied Soft Computing*, vol.50, pp.58-70, 2016.

[39] K. Mahadevan and P. S. Kannan, "Comprehensive learning particle swarm optimization for reactive power dispatch", *Applied Soft Computing*, vol.10, no.2, pp.641-652, 2010.

[40] R. D. Zimmerman, C. E. M. Sanchez and D. Gan, Matpower: matlab power simulation package, <http://www.pserc.cornell.edu/matpower/>.

[41] G. Chen, Z. Lu, Z. Zhang and Z. Sun, "Research on hybrid modified cuckoo search algorithm for optimal reactive power dispatch problem", *IAENG International Journal of Computer Science*, vol.45, no.2, pp.328-339, 2018.

[42] G. Chen, X. Yi, Z. Zhang and H. Wang, "Applications of multi-objective dimension-based firefly algorithm to optimize the power losses, emission, and cost in power systems", *Applied Soft Computing*, vol.68, pp.322-342, 2018.

[43] A. A. Heidari, R. A. Abbaspour and A. R. Jordehi, "Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems", *Applied Soft Computing*, vol.57, no., pp.657-671, 2017.

[44] S. Duman, Y. So Nmez, U. Gu Venc and N. Yo Ru Keren, "Optimal reactive power dispatch using a gravitational search algorithm", *IET Generation, Transmission & Distribution*, vol.6, no.6, pp.563-576,

2012.

[45] B. Shaw, V. Mukherjee and S. P. Ghoshal, "Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm", *International Journal of Electrical Power & Energy Systems*, vol.55, no.1, pp.29-40, 2013.

[46] E. Zitzler, D. Brockhoff and L. Thiele, "The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration", *International Conference on Evolutionary Multi-criterion Optimization*, pp.862-876, 2007.