Applied Soft Computing Journal 92 (2020) 106321

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Application of modified pigeon-inspired optimization algorithm and constraint-objective sorting rule on multi-objective optimal power flow problem

Gonggui Chen [a,b], Jie Qian [a,b], Zhizhong Zhang [c,*], Shuaiyong Li [a]

[a] Key Laboratory of Industrial Internet of Things & Networked Control, Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
[b] Chongqing Key Laboratory of Complex Systems and Bionic Control, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[c] Key Laboratory of Communication Network and Testing Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

ARTICLE INFO

Article history:
Received 25 November 2019
Received in revised form 6 March 2020
Accepted 13 April 2020
Available online 21 April 2020

Keywords:
Modified pigeon-inspired optimization algorithm
Optimal power flow problem
Constraint-objective sorting rule
Penalty function method

ABSTRACT

To solve the non-differentiable optimal power flow (OPF) problems with multiple contradictory objectives, a modified pigeon-inspired optimization algorithm (MPIO) is put forward in this paper. Combining with the common-used penalty function method (PFM), the MPIO-PFM algorithm is proposed and applied to optimize the active power loss, emission and fuel cost (with valve-point loadings) of power system. Eight simulation trials carried out on MATLAB software validate MPIO-PFM algorithm can obtain superior Pareto Frontier (PF) comparing with the typical NSGA-II algorithm. Nevertheless, some Pareto solutions obtained by MPIO-PFM algorithm cannot satisfy all system constraints due to the difficulty in choosing the proper penalty coefficients. Thus, an innovative approach named as constraint-objective sorting rule (COSR) is presented in this paper. The bi-objective and tri-objective trials implemented on IEEE 30-node, 57-node and 118-node systems demonstrate that the Pareto optimal set (POS) obtained by MPIO-COSR algorithm realizes zero-violation of various system constraints. Furthermore, the generational-distance and hyper-volume indexes quantitatively illustrate that in contrast to NSGA-II and MPIO-PFM methods, the MPIO-COSR algorithm can determine the evenly-distributed PFs with satisfactory-diversity. The intelligent MPIO-COSR algorithm provides an effective way to handle the non-convex MOOPF problems.

1. Introduction

Unlike the single-objective optimization, the multi-objective optimization (MOP) problem takes more than one conflicting goals into account concurrently. It is impossible to make each goal achieve the best state at the same time. The mathematical model of MOP problem, which usually has great computational complexity and obvious non-linear characteristics, is strictly limited by various constraints [1,2].

The MOP problems are very common in practical fields such as power systems [3,4]. Electricity with self-evident importance is the most basic energy in modern society. Research shows that the multi-objective optimal power flow (MOOPF) problems have undeniable significance to achieve the safety and economy operation of power system. Essentially, the MOOPF is a non-linear minimization problem with high-dimensional feature [5–8].

1.1. Algorithm review and selection

Intelligent algorithms have been successfully applied on handling the nonlinear MOOPF problems with the gradually maturing of computer technology. For instance, the novel quasi-oppositional modified Jaya algorithm [9], the hybrid DA-PSO optimization algorithm [10] and the multi-objective firefly algorithm [11] are capable to deal with the MOOPF problems effectively. However, there is still much room in the bi-objective MOOPF optimizations on large-scale systems and the tri-objective ones.

The basic and improved pigeon-inspired optimization algorithms (PIO) with fast-convergence and strong-robustness have been applied to various fields such as the fuzzy production scheduling problems [12] and large-scale traveling salesman problems [13]. However, the shortcoming of the basic PIO algorithm that is easy to be trapped into the local optimums cannot be ignored.

[*] Corresponding author.
E-mail address: zhangzztx@163.com (Z. Zhang).

https://doi.org/10.1016/j.asoc.2020.106321
1568-4946/© 2020 Elsevier B.V. All rights reserved.

To effectively solve the complex practical problems, the PIO algorithm needs to be combined with some appropriate improvement strategies. In Ref. [14], the modified PIO algorithm considering the hierarchical learning behavior is proposed to handle the UAV distributed flocking problem among obstacles. In Ref. [15], the cooperative PIO algorithm with distance threshold is put forward to improve the population-diversity in handling the multilevel image thresholding problems. And in Ref. [16], the performance of improved PIO algorithm for the air quality prediction problems has been optimized by integrating the particle swarm optimization (PSO) algorithm. However, it can be seen from these literatures that the PIO algorithm is usually plagued by the local convergence due to the poor diversity. To pertinently improve the population-diversity and avoid the premature-convergence, the adaptive-adjusted map factor (R_{mapnew}) and nonlinearly-adjusted coefficient (ω_{div}) are integrated into the proposed modified PIO (MPIO) algorithm. The applicability and superiority of MPIO algorithm to deal with the MOOPF problems are verified by eight simulation experiments implemented in this paper.

1.2. Major contributions

In this paper, the MPIO algorithm and constraint-objective sorting rule (COSR) as two major contributions are proposed.

First, the MPIO algorithm which adopts the adaptive-adjusted R_{mapnew}, nonlinearly-adjusted ω_{div} and novel landmark searching model is put forward to solve the MOOPF problems.

Furthermore, the MOOPF problems are severely restricted by equality constraints and inequality ones [11,17,18]. The control variables can be limited within a valid range in the initialization phase. But the state variables may not meet all inequality constraints. The penalty function method (PFM), a common technique to handle the constraints of state variables [11,19], is integrated with the MPIO method to generate the novel MPIO-PFM algorithm.

Comparing with the non-dominated sorting genetic algorithm-II (NSGA-II) which is often used as the comparison benchmark to evaluate the quality of novel algorithms, the proposed MPIO-PFM has greater potential in exploring superior Pareto frontier (PF). However, experimental results indicate the MPIO-PFM algorithm cannot guarantee each solution of obtained Pareto optimal set (POS) achieves the zero system-constraints violation. The presented COSR rule, which takes the constraint-violation value as the screening factor of POS set, can effectively overcome the above shortcoming of PFM method.

By integrating the MPIO algorithm and COSR sorting strategy, an effective MPIO-COSR algorithm is put forward in this paper. Five bi-objective and three tri-objective MOOPF trials which aim to optimize the active power loss, total emission and fuel cost are carried out to verify the applicability and superiority of MPIO-COSR algorithm. In contrast to MPIO-PFM and NSGA-II algorithms, the suggested MPIO-COSR algorithm can not only find the well-distributed POS, but also realize the zero constraint-violation of each power flow solution.

1.3. Structural arrangement

The rest part of this article is set as follows. Section 2 gives the mathematical model of MOOPF problems including objective functions and system restrictions. The constraint processing methods of state variables and the proposed COSR strategy for seeking the evenly-distributed PF are described in Section 3. The basic PIO algorithm and the novel MPIO-COSR algorithm are introduced in Section 4. Section 4 summarizes the application of MPIO-COSR algorithm on MOOPF Problems as well. Section 5 shows the results of eight MOOPF trials on three different standard systems. Section 6 measures the optimization performance of MPIO-COSR algorithm according to the convergence analysis, two quantitative evaluation indexes and computational complexity. Eventually, Section 7 gives the conclusions of this paper.

2. Mathematization of MOOPF problems

In essence, the MOOPF problem is a minimization mathematical model whose objective functions and system constraints can be defined as follows [5,20,21].

$$\text{minimize}\, F_{obj} = (f_1(s,c), f_2(s,c), \ldots, f_i(s,c), \ldots, f_M(s,c)) \tag{1}$$

$$E_k(s,c) = 0,\ k = 1,2,\ldots,h \tag{2}$$

$$I_p(s,c) \le 0,\ p = 1,2,\ldots,g \tag{3}$$

where $f_i(s,c)$ is the ith objective function. E_k and I_p represent the kth equality constraint (EC) and the pth inequality constraint (IC). $M(M{\ge}2)$ is the amount of goals which are optimized synchronously while h and g are, respectively, the numbers of ECs and ICs. The s and c vectors indicate the sets of state variables and control ones.

2.1. System restrictions

The constraints of electric system are classified into ECs and ICs. The best compromise (BC) solution adopted by decision-makers should meet all system restrictions.

2.1.1. ECs

The ECs, the active and reactive power balance equations intrinsically, are defined as Eq. (4) and (5). The detail descriptions of relevant parameters are clarified in Ref. [20,22].

$$P_{Gi} - P_{Di} - V_i \sum_{j\in N_i} V_j (G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)) = 0,\ i \in N \tag{4}$$

$$Q_{Gi} - Q_{Di} - V_i \sum_{j\in N_i} V_j (G_{ij}\sin(\delta_i - \delta_j) - B_{ij}\cos(\delta_i - \delta_j)) = 0,\ i \in N_{PQ} \tag{5}$$

2.1.2. ICs

The ICs, the validity range of system variables, are defined as follows. Among them, the ICs of state variables are shown in (6)–(9) and the ICs of control variables are shown in (10)–(13).

- Generator active power at slack node (P_{G1})

$$P_{G1}^{\max} \ge P_{G1} \ge P_{G1}^{\min} \tag{6}$$

- Load node voltage (V_L)

$$V_{Lq}^{\max} \ge V_{Lq} \ge V_{Lq}^{\min},\ q = 1,2,\ldots,N_{PQ} \tag{7}$$

- Generator reactive power (Q_G)

$$Q_{Gw}^{\max} \ge Q_{Gw} \ge Q_{Gw}^{\min},\ w = 1,2,\ldots,N_G \tag{8}$$

- Apparent power of transmission line (S)

$$S_e^{\max} - S_e \ge 0,\ e = 1,2,\ldots,N_L \tag{9}$$

- Generator active power output at PV node (P_G)

$$P_{Gz}^{\max} \ge P_{Gz} \ge P_{Gz}^{\min},\ z = 2,3,\ldots,N_G \tag{10}$$

- Generator node voltage (V_G)

$$V_{Gu}^{\max} \ge V_{Gu} \ge V_{Gu}^{\min},\ u = 1,2,\ldots,N_G \tag{11}$$

- Tap ratios of transformer (T)

$$T_d^{\max} \geq T_d \geq T_d^{\min}, d = 1, 2, \ldots, N_T \tag{12}$$

- Reactive power injection (Q_C)

$$Q_{Ct}^{\max} \geq Q_{Ct} \geq Q_{Ct}^{\min}, t = 1, 2, \ldots, N_C \tag{13}$$

where N_{PQ}, N_G, N_L, N_T and N_C are the numbers of PQ nodes, generators, transmission branches, transformers and shunt compensators, respectively.

2.2. Objective functions

The active power loss F_{ap}, the basic fuel cost F_{bf}, the fuel cost with valve-point loadings F_{fv} and the emission F_{em} [20,23–25], are studied in this paper.

2.2.1. Active power loss (F_{ap}/MW)

$$F_{ap} = \sum_{k=1}^{N_L} con_{(k)}[V_i^2 + V_j^2 - 2V_iV_j\cos(\delta_i - \delta_j)] \tag{14}$$

where $con_{(k)}$ represents the conductance of the kth branch that links the ith node to the jth one. The V_i and δ_i are the voltage magnitude and angle of the ith node.

2.2.2. Basic fuel cost (F_{bf}/\$/h)

$$F_{bf} = \sum_{i=1}^{N_G}(a_i + b_iP_{Gi} + c_iP_{Gi}^2) \tag{15}$$

where a_i, b_i and c_i indicate the cost coefficients of the ith generator.

2.2.3. Fuel cost with valve-point loadings (F_{fv}/\$/h)

$$F_{fv} = \sum_{i=1}^{N_G}(a_i + b_iP_{Gi} + c_iP_{Gi}^2 + \left|d_i \times \sin(e_i \times (P_{Gi}^{\min} - P_{Gi}))\right|) \tag{16}$$

where d_i and e_i represent the cost coefficients. $P_{Gi}^{\min}$ indicates the lower active power at the ith generator node.

2.2.4. Emission (F_{em}/ton/h)

$$F_{em} = \sum_{i=1}^{N_G}[\alpha_iP_{Gi}^2 + \beta_iP_{Gi} + \gamma_i + \eta_i\exp(\lambda_iP_{Gi})] \tag{17}$$

where α_i, β_i, γ_i, η_i and λ_i depict the emission coefficients of the ith generator.

3. Constraint handling strategies

Proper constraint handling strategies are critical to successfully handle the strictly-constrained MOOPF problems.

3.1. Equality constraints processing

The two ECs (4) and (5) are considered as the ending condition of Newton–Raphson method. In other words, the end of power flow calculation procedure means that the ECs are satisfied [2,17].

3.2. Inequality constraints processing

The processing of ICs are mainly divided into two aspects: the processing of ICs on control variables and the processing of ICs on state variables.

3.2.1. Control variables processing

It is necessary to adopt the effective handling methods of system constraints when solving the MOOPF problems. The P-dimensional control variables, also known as the independent variables of MOOPF problems, can be limited to $[c^{\min}, c^{\max}]$ in the initialization phase. The unqualified control variables which violate the ICs can be adjusted according to Formula (18).

$$c_i = \begin{cases} c_i^{\min}, & c_i < c_i^{\min} \\ c_i^{\max}, & c_i > c_i^{\max} \end{cases} \tag{18}$$

3.2.2. State variables processing

Two approaches of dealing with the ICs on state variables are studied in this paper. One is the common-used PFM method, and the other is the constraint-priority dominant strategy.

(1) Penalty function method

The modified objective functions with PFM method are defined as (19). The details and application of PFM method can refer to literatures [11,22,26].

$$F_{obj-\text{mod}} = F_{obj} + penalty \tag{19}$$

$$penalty = \zeta_V\sum_{i=1}^{N_{PQ}}(V_{Li} - V_{Li}^{\text{lim}}) + \zeta_Q\sum_{i=1}^{N_G}(Q_{Gi} - Q_{Gi}^{\text{lim}}) + \zeta_P(P_{G1} - P_{G1}^{\text{lim}}) + \zeta_S\sum_{i=1}^{N_L}(S_i - S_i^{\text{lim}}) \tag{20}$$

where ζ_V, ζ_Q, ζ_P and ζ_S are the penalty coefficients used to adjust the violation limits of state variables.

The X^{lim}, which represents the boundary value of corresponding variables, can be determined according to Formula (21). For example, the voltage boundary value of the ith load node (V_{Li}^{lim}) is defined as Formula (22). It should be noted that V_{Li}^{lim} = V_{Li} when V_{Li} satisfies Formula (7), which means the valid V_{Li} will not be punished based on Eq. (20).

$$X^{\text{lim}} = \begin{cases} X^{\min} & if\ X < X^{\min} \\ X^{\max} & if\ X > X^{\max} \end{cases} \tag{21}$$

$$V_{Li}^{\text{lim}} = \begin{cases} V_{Li}^{\min} & if\ V_{Li} < V_{Li}^{\min} \\ V_{Li} & if\ V_{Li}^{\min} \leq V_{Li} \leq V_{Li}^{\max} \\ V_{Li}^{\max} & if\ V_{Li} > V_{Li}^{\max} \end{cases} \tag{22}$$

In summary, the objective function of these solutions which violate system constraints will be punished based on PFM method. However, the effectiveness of PFM method is greatly influenced by penalty coefficients. Improper penalty coefficients may result in the failure to obtain the feasible POS set with zero constraint-violation.

(2) Constraint-priority dominant strategy

As the traditional way to deal with the unqualified state variables of MOOPF problems, the PFM method has two obvious shortcomings. First, choosing an appropriate penalty coefficient requires many repeated simulation experiments, which will inevitably increase the computational complexity. Then, the PFM method may find some power flow solutions which violate system constraints when dealing with the high-dimensional MOOPF problems.

The suggested constraint-priority dominant strategy effectively overcomes the imperfections of PFM method. Determining the dominant relationship between two different power flow solutions is the basis of using the proposed COSR strategy. In this paper, the constraint violation (CV) value is given the highest-priority. In other word, these solutions with zero-violation or smaller-violation will be assigned higher adoption priorities.

There are three dominant relationships between two different solution sets ($W = (w_1, w_2, \dots, w_M)$, $V = (v_1, v_2, \dots, v_M)$). That is, W dominates V, V dominates W or they do not dominate each other. For those solutions with the same CV value, the adoption-priority is determined based on the values of these goals to be studied. The CV value of W can be calculated according to Eq. (23).

$$CV(W) = Constr\_V_{(W)} + Constr\_Q_{(W)} + Constr\_P_{(W)} + Constr\_S_{(W)} \tag{23}$$

where $Constr\_V_{(W)}$, $Constr\_Q_{(W)}$, $Constr\_P_{(W)}$ and $Constr\_S_{(W)}$, respectively, represent the absolute violation-value of W solution on node voltage, generator reactive power output, generator active power output at slack node and apparent power of branches.

Based on the constraint-priority dominant strategy, the judgment can be made that the W solution surpasses the V solution when condition (24) or (25) is satisfied. In other words, the W solution is a non-inferior solution in contrast to the V one.

$$CV(W) < CV(V) \tag{24}$$

$$\begin{cases} CV(W) = CV(V) \\ f_i(s_W, c_W) \le f_i(s_V, c_V), \forall i \in \{1, 2, \dots, M\} \\ f_j(s_W, c_W) < f_j(s_V, c_V), \exists j \in \{1, 2, \dots, M\} \end{cases} \tag{25}$$

3.3. Proposed COSR rule

To obtain the uniformly-distributed PFs, the non-inferior COSR rule which concurrently takes the rank indicator (R_{ank}) and the crowding distance indicator (C_{dis}) into account is adopted.

3.3.1. R_{ank} index

Inspired by the sorting strategy proposed by Kalyanmoy Deb [27–29], the R_{ank} index of each solution with constraints-priority can be determined.

To sustain the diversity of POS, the external archive set (EAS) which is conducive to store the elite solutions is employed. A hybrid solution set (HSS) is composed of the parent solution set (PSS) and the EAS set which both contain N alternative solutions. In order to improve the optimization efficiency, the deletion operation of repeated individuals, as one of the innovative measures in this paper, is performed on the HSS population. The R_{ank} index of remaining HSS population with size of $N_n (N < N_n \le 2N)$ can be defined as follows.

(i) Based on the constraint-priority dominant strategy shown in (24) and (25), these solutions which are not dominated by the other solutions in HSS are marked as $R_{ank} = 1$.

(ii) Regardless of the solutions with $R_{ank} = 1$, the current non-inferior solutions are found based on the same dominant rule and marked as $R_{ank} = 2$.

(iii) Repeat the previous steps to make the N_n solutions in HSS population have their corresponding R_{ank} index.

3.3.2. C_{dis} index

In order to select a favorable POS set composed of N solutions from the HSS set, the R_{ank} index is prioritized. Then, the C_{dis} index is considered for the different solutions with the same R_{ank}. The key of calculating the C_{dis} value can be generalized as Eq. (26).

$$C_{dis}(i) = \sum_{j=1}^{N} \frac{f_j(s, c_{i-1}) - f_j(s, c_{i+1})}{f_j^{\max} - f_j^{\min}} \tag{26}$$

where $f_j^{\max}$ and $f_j^{\min}$, respectively, are the maximum and minimum values of the jth goal. The $f_j(s, c_{i-1})$ and $f_j(s, c_{i+1})$ depict the jth objective values on the $(i-1)$th individual and the $(i+1)$th one.

Considering the R_{ank} and C_{dis} indexes comprehensively, the adoption-priority of each solution can be determined. When condition (27) or (28) is satisfied, it can be concluded that the ith solution predominates the jth one, that is to say, the jth solution ranks behind the ith one.

$$R_{ank}(i) < R_{ank}(j) \tag{27}$$

$$\begin{cases} R_{ank}(i) = R_{ank}(j) \\ C_{dis}(i) > C_{dis}(j) \end{cases} \tag{28}$$

In general, the smaller R_{ank} index means the smaller constraint-violation or objective-function values, while the larger C_{dis} index means the better solution-diversity. Consequently, the N top-ranked solutions in HSS population are the ultimate POS obtained by the proposed COSR strategy.

Different from the classical way of solving the many-objective optimizations by assigning different priority-factors [30–32], the COSR method proposed in this paper can more objectively determine the feasible POS set of MOOPF problems and meet the various needs of decision-makers. Furthermore, the presented COSR strategy overcomes the deficiency that the traditional method is not suitable for the multi-objective cases with unknown priorities.

In addition, the relationship between the COSR strategy and the published COFS strategy [17], which are both effective to select the qualified POS set of MOOPF problems, are clarified. The COSR and COFS strategies have different evaluation criteria for these power flow solutions with the same R_{ank} index. The COFS strategy prefers the solution with a larger fuzzy dominant fitness ($Fudf$) value, while the COSR strategy prefers the solution with a larger C_{dis} value. It can be seen from Ref. [17] that the $Fudf$ index of COFS strategy is calculated based on the $(2N\text{-}1)$ candidate solutions while the C_{dis} index of COSR strategy can be obtained based on only two adjacent solutions. Therefore, although the two mentioned selection strategies are both effective for MOOPF problems, the proposed COSR method with lower computational complexity is more advantageous on the operational efficiency.

4. Multi-objective algorithms

The basic PIO algorithm generates a random solution set in the initial stage and searches the optimal solution by iterative operation [33–35]. The unique characteristic is that PIO algorithm will perform the map searching and landmark searching to locate the optimal solution po_{best}.

4.1. Basic PIO algorithm

The map searching model updates the position and speed of pigeon population during the each iteration based on the geomagnetic field. The speed sp_i and position po_i of the ith pigeon are modeled mathematically as Formulas (29)–(30) [13,33,34,36].

$$sp_i(t+1) = sp_i(t) * e^{-R_{map}t} + \xi_1 * (po_{best} - po_i(t)) \tag{29}$$

$$po_i(t+1) = po_i(t) + sp_i(t+1) \tag{30}$$

where R_{map} is the map factor and ξ_1 indicates a random number which is limited between 0 and 1.

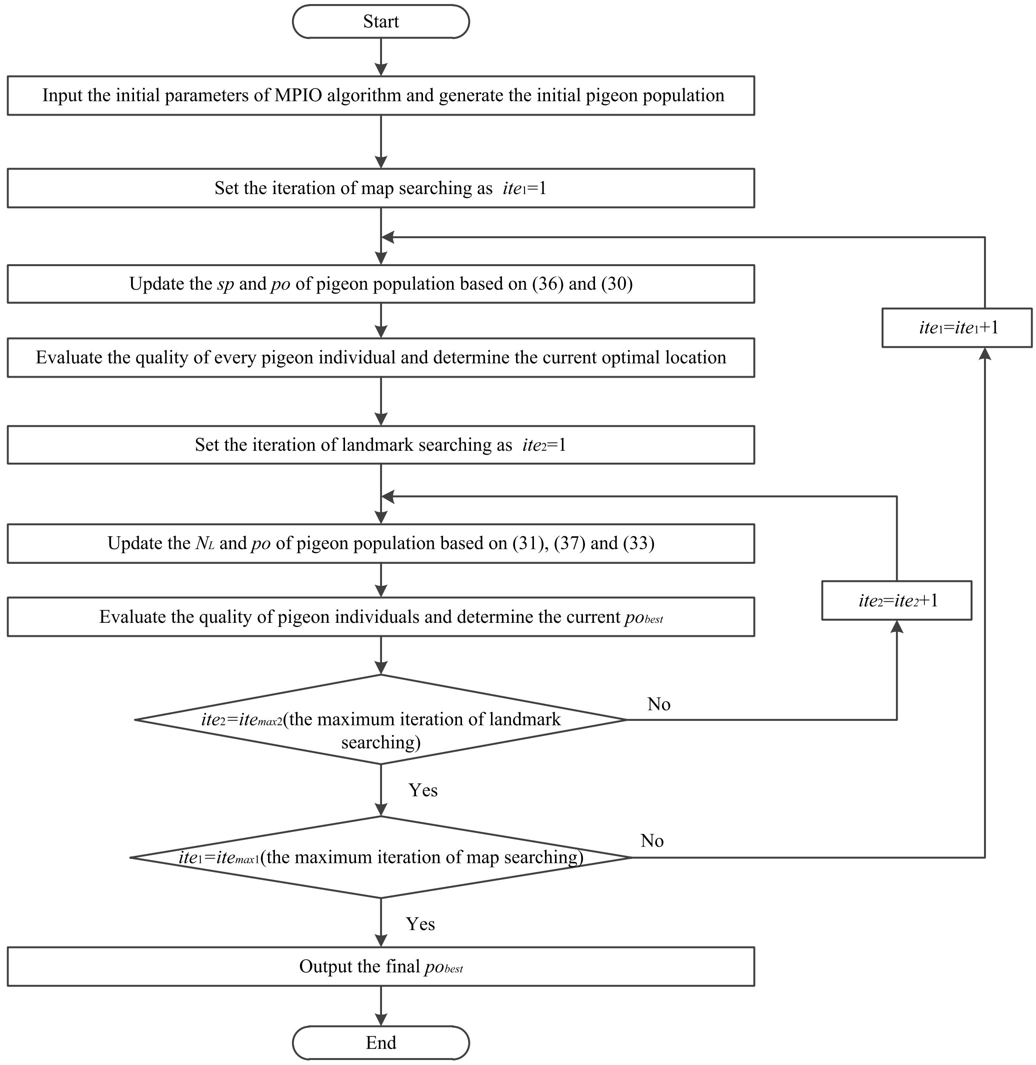
Start
Input the initial parameters of MPIO algorithm and generate the initial pigeon population
Set the iteration of map searching as ite1=1
Update the sp and po of pigeon population based on (36) and (30)
Evaluate the quality of every pigeon individual and determine the current optimal location
Set the iteration of landmark searching as ite2=1
Update the NL and po of pigeon population based on (31), (37) and (33)
Evaluate the quality of pigeon individuals and determine the current pobest
ite2=itemax2(the maximum iteration of landmark searching)
No
ite2=ite2+1
Yes
ite1=itemax1(the maximum iteration of map searching)
No
ite1=ite1+1
Yes
Output the final pobest
End

Fig. 1. Flow chart of MPIO algorithm.

The landmark searching model makes the pigeons fly freely based on the guidance of nearby pigeons or experience of themselves. The landmark searching model can be described as Formulas (31)∼(33) [13,35].

$$N_l(t+1) = N_l(t)/2 \tag{31}$$

$$po_l = (\sum_{i=1}^{N_l(t)} po_i(t) * fitness(po_i(t)))/(N_l(t) * \sum fitness(po_i(t))) \tag{32}$$

$$po_i(t+1) = po_i(t) + \xi_2 * (po_l - po_i(t)) \tag{33}$$

where N_l is the number of pigeons involved in the landmark searching and ξ_2 ($\xi_2 \in (0, 1)$) is a random number. The $fitness(po_i(t))$ depicts the quality evaluation value of the ith pigeon.

4.2. Proposed MPIO-COSR algorithm

The suggested MPIO algorithm integrating the COSR strategy extends the application field from single-objective optimizations to multi-objective ones. The standard PIO algorithm is improved from the following three aspects.

4.2.1. Adaptive-adjusted R_{mapnew}

The adaptive-adjusted R_{mapnew} is more beneficial to balance the optimization-accuracy and convergence-speed than a fixed R_{map} value. In initial state, the small R_{mapnew} means the fast-convergence and enhanced search-ability. The gradual increase of R_{mapnew} to $R_{map}(\max)$ facilitates the completion of map searching. The updated map factor with the adaptive-adjusted R_{mapnew} is modified as (34). The effective range of R_{mapnew} is set to $(R_{map}(\min), R_{map}(\max))$.

$$R_{mapnew}(t) = (R_{map}(\min) - R_{map}(\max)) * (t - ite_{\max 1})/(1 - ite_{\max 1}) + R_{map}(\max) \tag{34}$$

4.2.2. Nonlinearly-adjusted ω_{div}

The speed update model of PIO algorithm is improved by a nonlinearly-adjusted coefficient ω_{div}, which is propitious to enhance the solution variability. The ω_{div} coefficient is defined as (35).

$$\omega_{div}(t+1) = \omega_{div}^{\max} - \xi_3(\omega_{div}^{\max} - \omega_{div}^{\min}) + \xi_4(\omega_{div}(t) - \xi_5 * (\omega_{div}^{\max} + \omega_{div}^{\min})) \tag{35}$$

where $\omega_{div}^{\max}$ and $\omega_{div}^{\min}$ are the maximum and minimum of ω_{div}. The ξ_3 and ξ_4 are two random numbers within (0,1) while $\xi_5 = 0.5$.

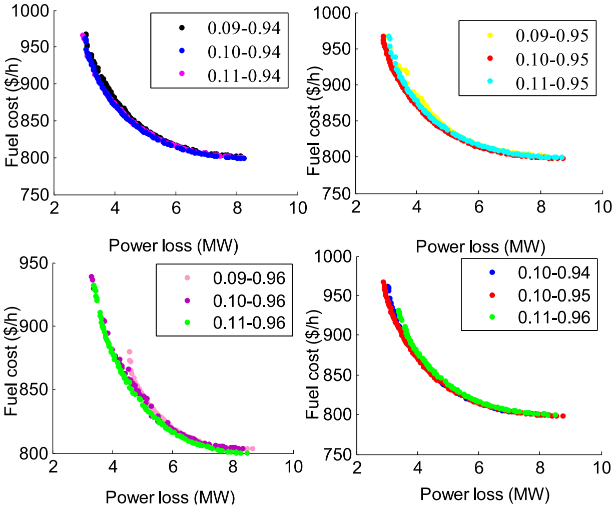

Fig. 2. PFs with different R_{mapnew}.

After integrating R_{mapnew} and ω_{div}, the *sp* term of MPIO algorithm is modified as Formula (36).

$$sp_i(t+1) = \omega_{div}(t+1)*sp_i(t)*e^{-R_{mapnew}t} + \xi_1*(po_{best} - po_i(t)) \quad (36)$$

4.2.3. Novel landmark searching model

The original PIO algorithm will perform the landmark searching after completing the map searching. However, the MPIO algorithm proposed in this paper performs the landmark searching after each map searching. The novel landmark searching model helps to explore a higher-quality solution near the current optimal solution and reduce the restriction of local optimum.

To successfully solve the MOOPF problems, the Formula (32) of single-objective PIO algorithm is modified to Formula (37) which can handle the multi-objective problems.

$$po_l = (\sum_{i=1}^{N_l(t)} po_i(t)*sat(po_i(t)))/(N_l(t)*\sum sat(po_i(t))) \quad (37)$$

For MOOPF problems, the po_i represents the P-dimensional control variables of the ith power flow solution. The $sat(po_i)$, which is the quality evaluation value of the ith pigeon, indicates the total satisfaction value of the ith solution defined as Formula (38). Generally speaking, the larger the *sat* value is, the better the performance of the corresponding power flow solution has. The BC solution of MOOPF problem is actually the one with the highest *sat* value of final POS set [20,37].

$$sat(i) = \frac{\sum_{j=1}^{M} Fs_j(i)}{\sum_{i=1}^{N}\sum_{j=1}^{M} Fs_j(i)} \quad (38)$$

where $Fs_j(i)$ indicates the satisfaction function of the ith solution on the jth objective which can be defined as (39). The $f_j^{\max}$ is the maximum of the jth goal and $f_j^{\min}$ is the minimum one.

$$Fs_j(i) = \begin{cases} 1 & f_j \le f_j^{\min} \\ \dfrac{f_j^{\max} - f_j}{f_j^{\max} - f_j^{\min}} & f_j^{\min} < f_j < f_j^{\max} \\ 0 & f_j \ge f_j^{\max} \end{cases}, \quad i = 1, 2, \ldots, N \quad j = 1, 2, \ldots, M \quad (39)$$

Based on the above improvements, the novel MPIO algorithm is generated and its flow chart is shown in Fig. 1.

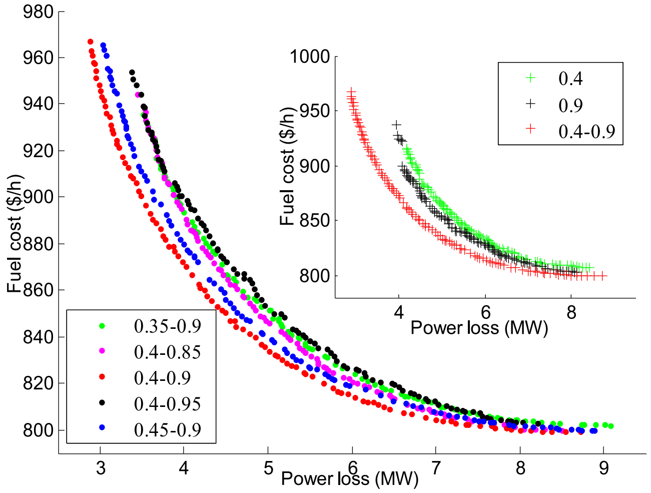

Fig. 3. PFs with different ω_{div}.

4.3. Application of MPIO-COSR algorithm on MOOPF problems

For MOOPF problems, the each pigeon indicates a P-dimensional control variables set of power system. It is worth pointing out that $P = 24$ in IEEE 30-node system, $P = 33$ in IEEE 57-node system and $P = 128$ in IEEE 118-node system. The larger dimensions caused by the complex structure mean that it is more difficult to solve the MOOPF problems.

To validate the great advantages of MPIO-COSR algorithm, eight MOOPF trials have been carried out in this paper. In detail, the pseudo code of MPIO-COSR algorithm on MOOPF problem is summarized in Table 1.

5. Trials and results

In contrast to NSGA-II and MPIO-PFM algorithms, the proposed MPIO-COSR algorithm can not only find the uniformly-distributed PF, but also realize that each non-inferior solution of POS to meet all system restrictions.

In order to verify the effectiveness of MPIO-COSR algorithm, eight MOOPF trials shown in Table 2 are simulated in this paper.

Table 1
Pseudo code of MPIO-COSR algorithm.

```
input: the initial parameters of MPIO-COSR algorithm and the $N$ pigeon individuals
($N$ randomly generated $P$-dimension sets of control variables)
begin
  $ite_1 = 1$
  while $ite_1 < ite_{max1}$
  for $i = 1,2,\ldots,N$
  update the $sp$ and $po$ of $i$th power flow solution based on (36) and (30)
  calculate the power flow to clarify the objective function value $f(s,c)$ and the constraints violation value $CV$
  end
  determine the current POS set based on the non-inferior COSR rule and the current $po_{best}$ solution based on $sat$ value;
    $ite_2 = 1$
    $N_l$=$N$
    while $ite_2 < ite_{max2}$
    $N_l$=$max(ceil(N_l/2),10)$
    for $i = 1,2,\ldots, N_l$
      update the $po$ of $i$th power flow solution based on (37) and (33)
      calculate the power flow
    end
    determine the current POS set and the current $po_{best}$ solution
    $ite_2 = ite_2$+1;
    end while
  $ite_1 = ite_1$+1
  end while
end
output: the control variables set of the $po_{best}$ solution
```

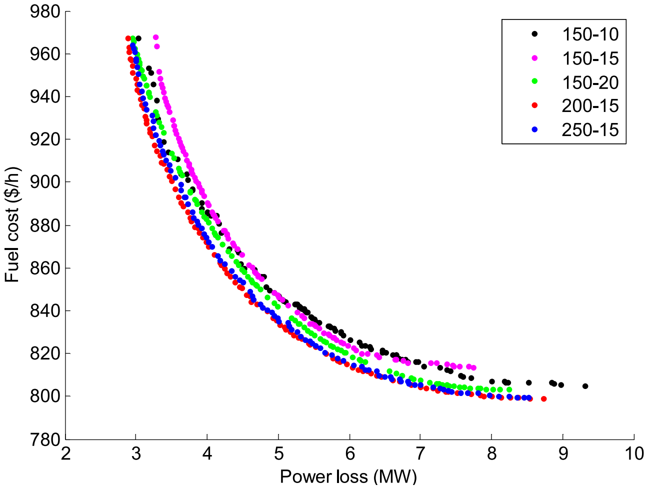

Fig. 4. PFs with different $ite_{\max}$.

All testing cases are conducted on the MATLAB 2014a software in a PC with Intel(R) Core(TM) i5–7500 CPU @ 3.40 GHz with 8 GB RAM.

5.1. Algorithm parameters

To optimize the performance of MPIO-COSR algorithm in handling the MOOPF problems, it is critical to set the reasonable ranges of two added R_{mapnew} and ω_{div} coefficients. A bi-objective case considering both F_{bf} and F_{ap} is taken as an example to determine a relatively-optimal combination of parameters.

Firstly, the effect of R_{mapnew} on optimization quality is discussed. Fig. 2 shows the PFs based on different range of R_{mapnew}, indicating that the more desirable PF is achieved when the R_{mapnew} range is set to (0.10, 0.95). Meanwhile, the influence of different ω_{div} ranges on the optimization performance is discussed as well. Fig. 3 shows the obtained PFs based on different ω_{div}, which clearly states that the more superior PF can be achieved when the valid range of ω_{div} is set to (0.4, 0.9). Besides, Fig. 3 also indicates that the diversity and distribution-uniformity of obtained PFs are significantly worse when the ω_{div} coefficient is set to a fixed value of 0.4 or 0.9, which proves the effectiveness of introducing the nonlinearly adjusted ω_{div} coefficient.

Consequently, the valid R_{mapnew} and ω_{div} are respectively limited within (0.10, 0.95) and (0.4, 0.9). To obtain the high-quality PF, it is also important to set a suitable $ite_{\max}$. Fig. 4 shows the PFs with different $ite_{\max}$, indicating that there is no significant difference in the solution-diversity of obtained PFs. However, the $ite_{max1} = 200$ and $ite_{max2} = 15$ achieves a relatively-best PF according to the Pareto dominance rule. Then, based on the R_{mapnew}, ω_{div} and $ite_{\max}$ set above, the availability of MPIO-COSR algorithm in solving the MOOPF problems with different population sizes is studied. Fig. 5 shows the obtained PFs when the population size is set as 30, 50, 75 and 100, which demonstrates that the proposed MPIO-COSR algorithm is capable to find the well-distributed PFs in different population sizes. Undoubtedly, the larger N is, the more difficult it is to solve the MOOPF problems. The N is set as 100 for all MOOPF simulation experiments studied in this paper to comprehensively evaluate the performance of the novel MPIO-COSR algorithm.

It should be noted that the $ite_{\max}$ of five trials on IEEE 30-node system is set as $ite_{max1} = 200$ with $ite_{max2} = 15$ while the ite_{max1} of other three trials on IEEE 57-node and 118-node systems is increased appropriately. Besides, the detail parameters of three involved algorithms are shown in Table 3.

5.2. Trials on IEEE 30-node system

Three bi-objective and two tri-objective cases are performed on IEEE 30-node system. The structure and details such as the effective range of control variables are provided in literatures [20,22,38].

5.2.1. Trial 1 : F_{bf} and F_{ap}

In Trial 1, the F_{bf} and F_{ap} are optimized simultaneously by NSGA-II, MPIO-PFM and MPIO-COSR approaches. The PFs of Trial 1 are shown in Fig. 6 and the control variables of BC solutions

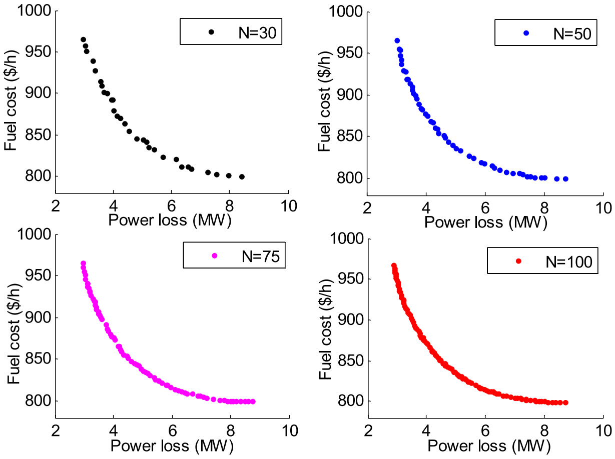

Fig. 5. PFs with different population sizes.

Table 2
Objective combination.

Combination	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8
F_{ap}	✓	✓		✓	✓		✓	
F_{bf}	✓		✓	✓		✓	✓	✓
F_{fv}		✓			✓			
F_{em}			✓	✓	✓	✓	✓	✓
System	IEEE 30	IEEE 30	IEEE 30	IEEE 30	IEEE 30	IEEE 57	IEEE 57	IEEE 118

Table 3
Algorithm parameters.

Parameters	Trial 1∼Trial 5		Trial 6∼Trial 7		Trial 8	
	NSGA-II	MPIO-PFM/ MPIO-COSR	NSGA-II	MPIO-PFM/ MPIO-COSR	NSGA-II	MPIO-PFM/ MPIO-COSR
N	100	100	100	100	100	100
Size of EAS	100	100	100	100	100	100
ite_{max1}	–	200	–	350	–	500
ite_{max2}	–	15	–	15	–	15
$ite_{max-NSGA}$	300	–	500	–	500	–
ω_{div}^{max}		0.90		0.90		0.90
ω_{div}^{min}		0.40		0.40		0.40
R_{map}(max)	–	0.95	–	0.95	–	0.95
R_{map}(min)	–	0.10	–	0.10	–	0.10
Mutation index/percentage	20/1	–	20/1	–	20/1	–
Crossover index/percentage	20/0.1	–	20/0.1	–	20/0.1	–

are shown in Table 4. It is not difficult to find that the BC solution with 5.1085 MW of F_{ap} and 831.5576 $/h of F_{bf}, which is achieved by the suggested MPIO-COSR algorithm, dominates the BC solutions obtained by NSGA-II and MPIO-PFM methods. In 30 independent trials for Trial 1, the number of feasible solutions that do not violate any system restriction is shown in Fig. 7. Fig. 7 visually indicates that the NSGA-II and MPIO-COSR algorithms can guarantee all obtained solutions satisfy the system constraints while the PF of MPIO-COSR algorithm realizes higher standard.

Besides, Table 5 gives the comparison results in other published literatures of Trial 1. It indicates that the BC solution of the novel MPIO-COSR algorithm dominates the ones of MOHS, NSGA-III, MOPSO and NHBA-CPFD algorithms. In addition, MPIO-COSR algorithm is not inferior to the other algorithms shown in Table 5. Consequently, MPIO-COSR algorithm has great potential and advantages in determining the better BC solutions.

5.2.2. Trial 2: F_{ap} and F_{fv}

In Trial 2, the performance of three involved algorithms for optimizing F_{ap} and F_{fv} is considered. The PFs of Trial 2 are illustrated in Fig. 8 and the control variables of BC solutions are given in Table 4. Table 4 clearly states that the BC solution obtained by MPIO-COSR algorithm with 5.6459 MW of F_{ap} and 865.9780 $/h of F_{fv} dominates both the BC solution obtained by NSGA-II with 5.7754 MW of F_{ap} and 868.6378 $/h of F_{fv} and the BC solution obtained by MPIO-PFM with 5.6621 MW of F_{ap} and 867.3608 $/h of F_{fv}. Besides, the BC solution of MPIO-COSR algorithm is also more superior to the BC of NHBA method composed by 5.6761 MW of F_{ap} and 869.9526 $/h of F_{fv}.

Compared with NSGA-II algorithm, MPIO-PFM algorithm has clear superiority in exploring better PFs. However, Fig. 9 shows that there is some solutions obtained by MPIO-PFM algorithm do not satisfy all inequality constraints. The presented MPIO-COSR algorithm is capable to overcome the above shortcoming effectively.

Table 4
The control variables of BC solutions for Trial 1 and Trial 2.

Variables	Trial 1			Trial 2			
	NSGA-II	MPIO-PFM	MPIO-COSR	NSGA-II	MPIO-PFM	MPIO-COSR	NHBA [39]
P_{G2} (MW)	52.6103	53.6818	47.7971	46.8927	46.5133	43.8208	52.3984
P_{G5} (MW)	29.8274	32.8854	30.8934	33.6199	32.7137	29.8216	31.9677
P_{G8} (MW)	34.3296	35.0000	35.0000	34.9469	34.6203	34.0807	34.6347
P_{G11} (MW)	27.7208	24.8533	29.1416	24.4436	22.7827	25.6364	19.6335
P_{G13} (MW)	25.9714	22.8184	23.6042	15.7963	19.0848	21.4943	20.2407
V_{G1} (p.u.)	1.0787	1.1000	1.1000	1.0954	1.1000	1.1000	1.0992
V_{G2} (p.u.)	1.0680	1.0893	1.0874	1.0847	1.0866	1.0912	1.0922
V_{G5} (p.u.)	1.0478	1.0710	1.0614	1.0632	1.0649	1.0703	1.0697
V_{G8} (p.u.)	1.0645	1.0836	1.0721	1.0741	1.0780	1.0773	1.0804
V_{G11} (p.u.)	1.0899	1.0738	1.0914	1.0150	1.1000	1.1000	1.0972
V_{G13} (p.u.)	1.0870	1.0952	1.1000	1.0397	1.0839	1.0958	1.0707
T_{11} (p.u.)	1.0494	1.0391	1.0172	1.0346	1.0122	1.0492	1.0257
T_{12} (p.u.)	0.9132	0.9023	0.9060	1.0187	0.9716	0.9010	0.9754
T_{15} (p.u.)	0.9814	1.0045	0.9835	1.0566	1.0067	1.0021	0.9818
T_{36} (p.u.)	0.9803	0.9940	0.9613	1.0285	0.9929	0.9792	0.9868
Q_{C10} (p.u.)	0.0059	0.0132	0.0282	0.0319	0.0061	0.0217	0.0276
Q_{C12} (p.u.)	0.0152	0.0192	0.0401	0.0217	0.0155	0.0039	0.0277
Q_{C15} (p.u.)	0.0305	0.0384	0.0469	0.0455	0.0427	0.0341	0.0242
Q_{C17} (p.u.)	0.0103	0.0339	0.0351	0.0500	0.0256	0.0390	0.0256
Q_{C20} (p.u.)	0.0363	0.0240	0.0252	0.0177	0.0211	0.0321	0.0397
Q_{C21} (p.u.)	0.0409	0.0119	0.0427	0.0351	0.0189	0.0481	0.0310
Q_{C23} (p.u.)	0.0288	0.0300	0.0256	0.0176	0.0429	0.0226	0.0315
Q_{C24} (p.u.)	0.0120	0.0359	0.0316	0.0146	0.0089	0.0235	0.0269
Q_{C29} (p.u.)	0.0249	0.0317	0.0181	0.0397	0.0445	0.0340	0.0205
F_{bf} ($/h)	833.5363	832.2274	**831.5576**	–	–	–	–
F_{fv} ($/h)		–	–	868.6378	867.3608	**865.9780**	868.9526
F_{ap} (MW)	5.3483	5.1270	**5.1085**	5.7754	5.6621	**5.6459**	5.6761

Table 5
Comparison results of Trial 1.

Algorithms	Size	ite_{max}	F_{ap} (MW)	F_{bf} ($/h)
NSGA-II	100	300	5.3483	833.5363
MPIO-PFM	100	200–15	5.1270	832.2274
MPIO-COSR	100	200–15	**5.1085**	**831.5576**
MOEA/D [37]	100	500	4.9099	835.36
MOABC/D [40]	100	30000 function evaluations	5.2451	827.636
MOHS [41]	20	1000	5.3143	832.6709
ESDE-MC [42]	20	200	5.2270	827.1592
NSGA-III [20]	100	300	5.1775	836.8076
MOPSO [20]	100	300	5.2310	852.8083
MOMICA [43]	120	500	4.5603	848.0544
NHBA-CPFD [39]	100	300	5.1096	831.8513

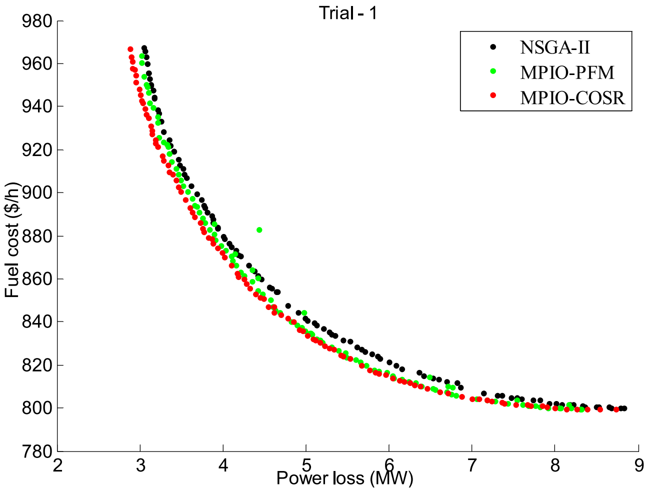

Fig. 6. PFs of Trial 1.

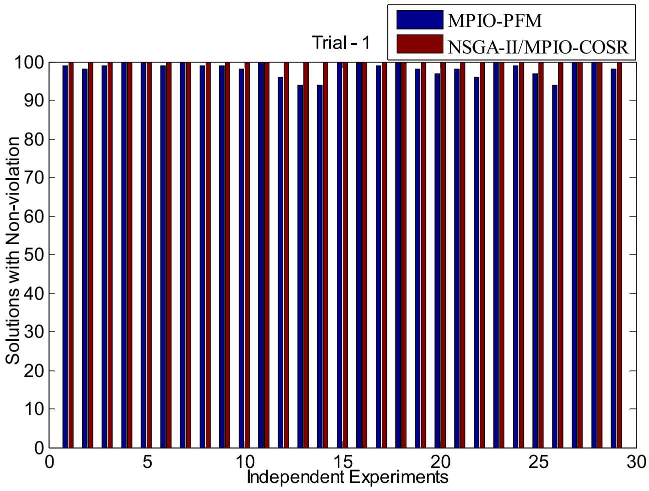

Fig. 7. The number of feasible solutions for Trial 1.

5.2.3. Trial 3: F_{em} and F_{bf}

The MPIO-COSR and two comparison algorithms are tested for optimizing the F_{em} and F_{bf} concurrently in Trial 3. Fig. 10 and Table 6 give the obtained PFs and the control variables of BC solutions, respectively. Table 6 suggests that based on the values of F_{em} and F_{bf}, the BC solution achieved by MPIO-COSR algorithm

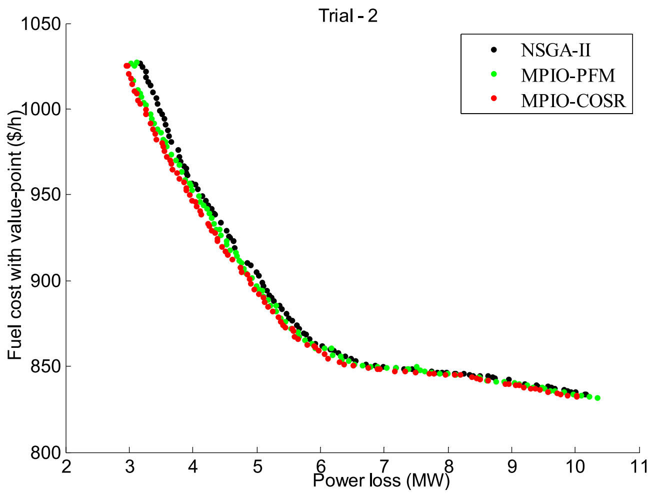

Fig. 8. PFs of Trial 2.

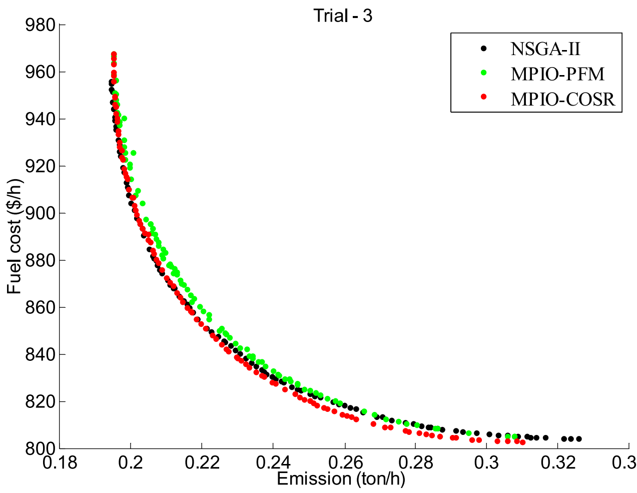

Fig. 10. PFs of Trial 3.

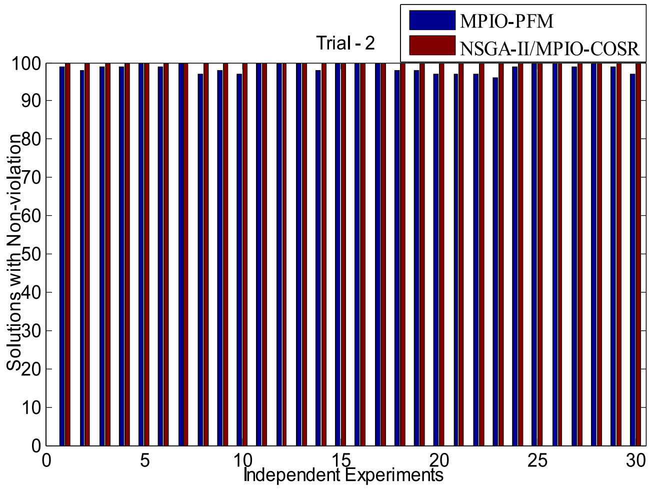

Fig. 9. The number of feasible solutions for Trial 2.

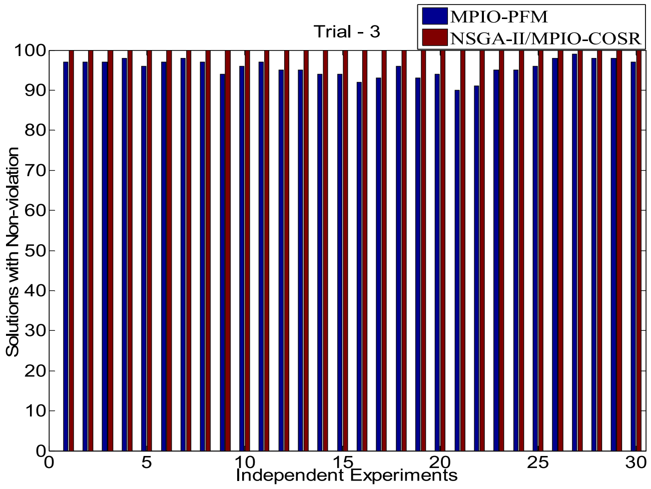

Fig. 11. The number of feasible solutions for Trial 3.

with 0.2351 ton/h of F_{em} and 832.4655 \$/h of F_{bf} can dominate the BC solutions achieved by NSGA-II and MPIO-PFM methods. Besides, Fig. 11 shows the number of available solutions which do not offend any system constraint. It turns out that the MPIO-COSR algorithm can obtain a more preponderant PF and achieve the zero-violation of all non-inferior solutions.

In addition, Table 7 gives the comparison results of Trial 3 from other published results. It intuitively states MPIO-COSR algorithm is more advantageous than most algorithms shown in Table 7 such as ESDE, MOEA/D, BSA and NHBA algorithms. Based on the obtained BC solutions, the presented MPIO-COSR algorithm and these algorithms such as ESDE-EC, MODFA do not dominate each other. In general, the dominant position of MPIO-COSR algorithm in seeking satisfactory BC solutions cannot be ignored.

5.2.4. Trial 4: F_{em}, F_{bf} and F_{ap}

In contrast to bi-objective optimization, the tri-objective one is a more complex problem with higher-dimension, which can measure the performance of MPIO-COSR method more comprehensively. In Trial 4, the F_{em}, F_{bf} and F_{ap} are optimized simultaneously. The PFs of Trial 4 are illustrated in Fig. 12, which clearly shows that three mentioned algorithms can obtain evenly-distributed PFs when the MPIO-COSR algorithm obtains a better one.

Table 6 shows the control variables of BC solutions found by three algorithms and it indicates that the BC solution obtained by MPIO-COSR algorithm with 0.2126 ton/h of F_{em}, 4.3177 MW of F_{ap} and 863.9503 \$/h of F_{bf} can dominate the ones achieved by NSGA-&#x2161; and MPIO-PFM methods. Compared with the MOFA-PFA algorithm in Ref [11], MPIO-COSR algorithm achieves the smaller F_{bf} and F_{em} when the F_{ap} is not much different.

Fig. 13 visually states that based on the number of feasible solutions, the NSGA-II and MPIO-COSR algorithms surpass MPIO-PFM method. Furthermore, the MPIO-COSR algorithm is superior to NSGA-II method in the high-quality of BC solutions and the uniform-distribution of PFs.

5.2.5. Trial 5: F_{em}, F_{ap} and F_{fv}

The performance of three involved algorithms for optimizing the F_{em}, F_{ap} and F_{fv} synchronously is studied in Trial 5. The PFs of Trial 5 are illustrated in Fig. 14 and the control variables of BC solutions are given in Table 8.

Table 8 indicates that the BC solution obtained by MPIO-COSR algorithm with 0.2234 ton/h of F_{em}, 4.8659 MW of F_{ap} and 910.6654 \$/h of F_{fv} can dominate the ones achieved by NSGA-II and MPIO-PFM methods. The comparing results of Trial 5 are also shown in Table 8. Although MPIO-COSR algorithm cannot directly dominate MOFA-PFA and MOFA-CPA algorithms, it achieves the relatively-better BC solution with smaller F_{fv} and F_{em} values among the three stated goals.

Table 6
The control variables of BC solutions for Trial 3 and Trial 4.

Variables	Trial 3			Trial 4			
	NSGA-II	MPIO-PFM	MPIO-COSR	NSGA-II	MPIO-PFM	MPIO-COSR	MOFA-PFA [11]
P_{G2} (MW)	59.4325	59.3460	59.6975	68.7162	64.5985	68.0089	57.890
P_{G5} (MW)	27.1332	32.8673	30.8573	38.6107	39.9652	34.3260	36.290
P_{G8} (MW)	33.8130	26.6984	32.9071	29.7645	31.7062	34.4931	35.000
P_{G11} (MW)	24.8982	22.2277	22.2523	24.1976	25.0777	27.0861	29.271
P_{G13} (MW)	25.6253	26.3319	25.6386	29.4938	28.4935	30.7136	40.000
V_{G1} (p.u.)	1.0229	1.0684	1.0888	1.0995	1.0984	1.1000	1.0985
V_{G2} (p.u.)	1.0004	1.0548	1.0800	1.0931	1.0946	1.0893	1.0869
V_{G5} (p.u.)	0.9655	1.0325	1.0556	1.0565	1.0698	1.0701	1.0625
V_{G8} (p.u.)	0.9808	1.0312	1.0577	1.0802	1.0801	1.0747	1.0767
V_{G11} (p.u.)	1.0645	1.0771	1.0860	1.0567	1.0334	1.0748	1.0857
V_{G13} (p.u.)	0.9926	1.0532	1.0555	1.0719	1.0515	1.0877	1.0386
T_{11} (p.u.)	0.9774	0.9791	1.0271	1.0271	1.0052	1.0752	1.0860
T_{12} (p.u.)	1.0787	1.0724	0.9985	0.9370	1.0647	0.9193	0.9930
T_{15} (p.u.)	0.9541	0.9903	0.9900	1.0393	1.0493	1.0156	1.0520
T_{36} (p.u.)	0.9372	0.9720	0.9604	0.9709	0.9986	0.9787	1.0770
Q_{C10} (p.u.)	0.0121	0.0327	0.0149	0.0191	0.0320	0.0383	0.0140
Q_{C12} (p.u.)	0.0312	0.0304	0.0152	0.0317	0.0201	0.0274	0.0220
Q_{C15} (p.u.)	0.0139	0.0100	0.0334	0.0445	0.0365	0.0485	0.0080
Q_{C17} (p.u.)	0.0357	0.0384	0.0336	0.0364	0.0185	0.0494	0.0250
Q_{C20} (p.u.)	0.0073	0.0428	0.0274	0.0385	0.0448	0.0427	0.0390
Q_{C21} (p.u.)	0.0169	0.0450	0.0349	0.0140	0.0318	0.0473	0.0270
Q_{C23} (p.u.)	0.0037	0.0157	0.0269	0.0174	0.0158	0.0218	0.0100
Q_{C24} (p.u.)	0.0455	0.0247	0.0440	0.0417	0.0478	0.0457	0.0170
Q_{C29} (p.u.)	0.0239	0.0142	0.0381	0.0079	0.0126	0.0415	0.0500
F_{bf} ($/h)	833.2605	833.1703	**832.4655**	866.0169	866.0601	**863.9503**	879.91
F_{ap} (MW)	–	–	–	4.5583	4.4474	**4.3177**	4.2179
F_{em} (ton/h)	0.2367	0.2397	**0.2351**	0.2165	0.2160	**0.2126**	0.2165

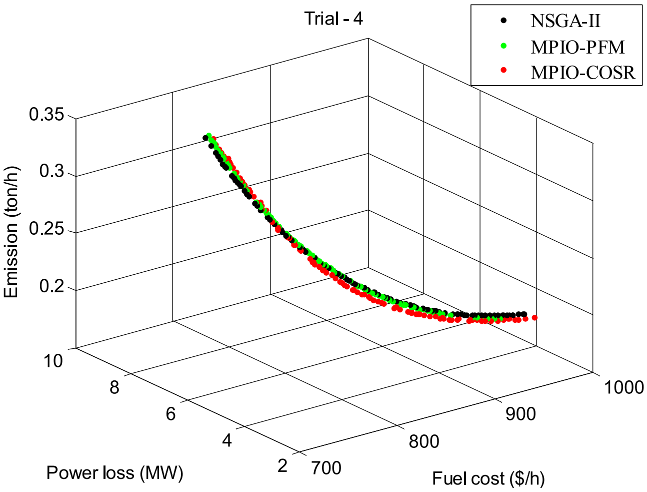

Fig. 12. PFs of Trial 4.

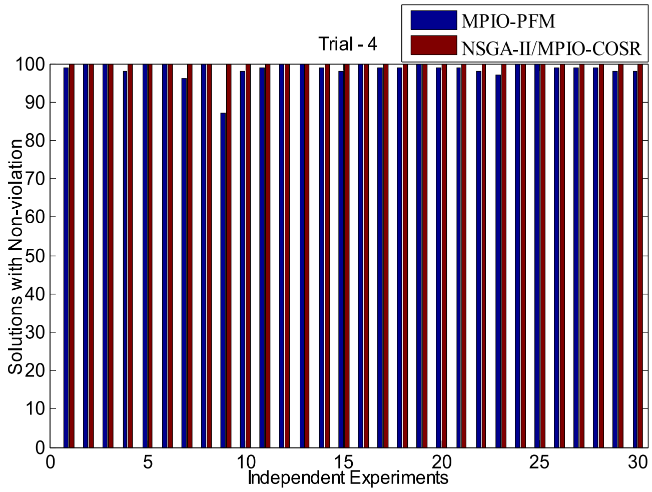

Fig. 13. The number of feasible solutions for Trial 4.

In 30 independent trials for Trial 5, the number of solutions that achieve none-violation is shown in Fig. 15. Fig. 14 and Fig. 15 indicate that although the NSGA-II and MPIO-COSR algorithms both guarantee every obtained solution achieve zero constraint-violation, the MPIO-COSR algorithm is more advantageous in handling the high-dimensional MOOPF problems.

5.3. Trials on IEEE 57-node system

To extend the study of MOOPF problems, the bi-objective and tri-objective optimizations are performed on the IEEE 57-node system with larger-scale and higher-dimension. The structure and details of IEEE 57-node system are provided in [20,22].

5.3.1. Trial 6: F_{em} and F_{bf}

In Trial 6, the NSGA-II, MPIO-PFM and MPIO-COSR algorithms are tested for optimizing the F_{em} and F_{bf} concurrently. Fig. 16 and Table 9 give the obtained PFs and the control variables of BC solutions, respectively.

Fig. 16 clearly states that all three algorithms can obtain the PFs with relative uniform-distribution while the MPIO-COSR algorithm can achieve the best one. Table 9 indicates that the BC solution obtained by MPIO-COSR algorithm with 1.2314 ton/h of F_{em} and 43131.2743 $/h of F_{bf} can dominate both the BC solution obtained by NSGA-II algorithm with 1.2466 ton/h of F_{em} and 43351.1353 $/h of F_{bf} and the BC solution obtained by MPIO-PFM algorithm with 1.2386 ton/h of F_{em} and 43205.8477 $/h of F_{bf}. Besides, MPIO-COSR algorithm also surpasses the published MODFA algorithm composed of 1.2679 ton/h of F_{em} and 43174.5740 $/h of F_{bf}.

Table 7
Comparison results of Trial 3.

Algorithms	F_{bf} ($/h)	F_{em} (ton/h)
NSGA-II	833.2605	0.2367
MPIO-PFM	833.1703	0.2397
MPIO-COSR	**832.4655**	**0.2351**
ESDE [42]	833.4743	0.2540
ESDE-EC [42]	831.0943	0.2510
ESDE-MC [42]	830.7185	0.2483
MOEA/D [37]	833.72	0.2438
BSA [44]	835.0199	0.2425
AGSO [45]	843.5473	0.2539
MODFA [20]	831.6652	0.2432
NSGA-III [20]	832.5323	0.2483
MOPSO [20]	833.7139	0.2492
NHBA [39]	832.6471	0.2375

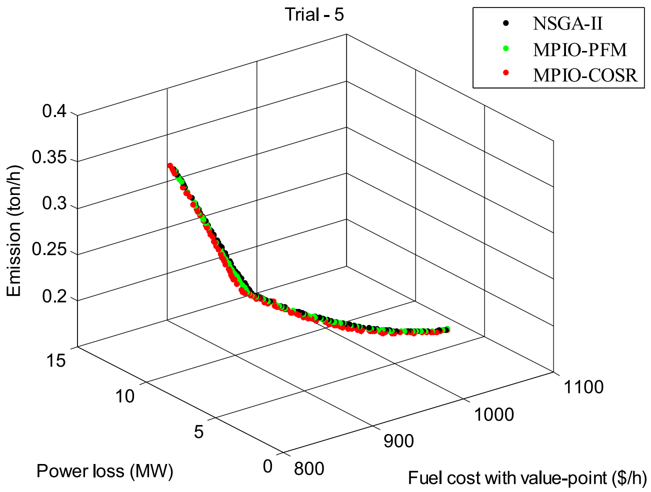

Fig. 14. PFs of Trial 5.

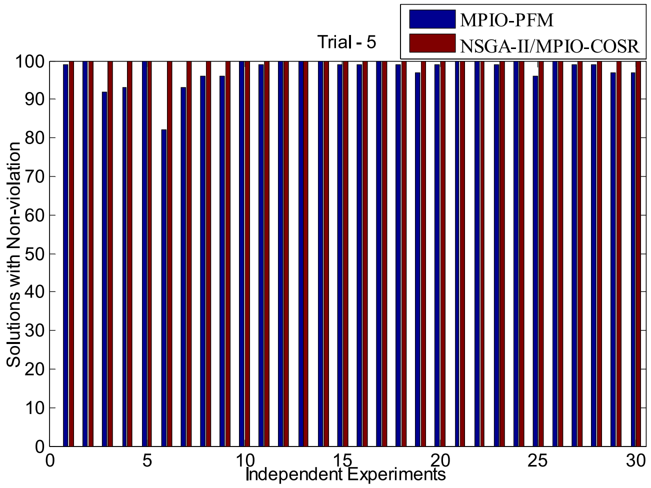

Fig. 15. The number of feasible solutions for Trial 5.

Besides, Fig. 17 gives the number of feasible solutions of Trial 6. In contrast to NSGA-II and MPIO-PFM methods, the proposed MPIO-COSR algorithm has obvious advantages to seek the superior POS that satisfies all inequality constraints in the complex, large-scale power systems.

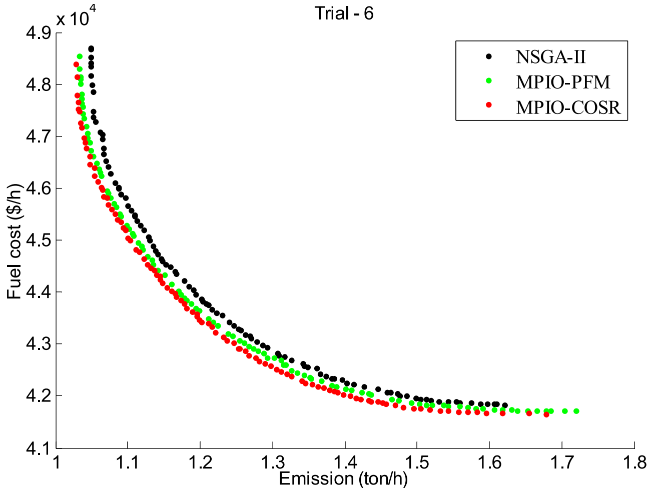

Fig. 16. PFs of Trial 6.

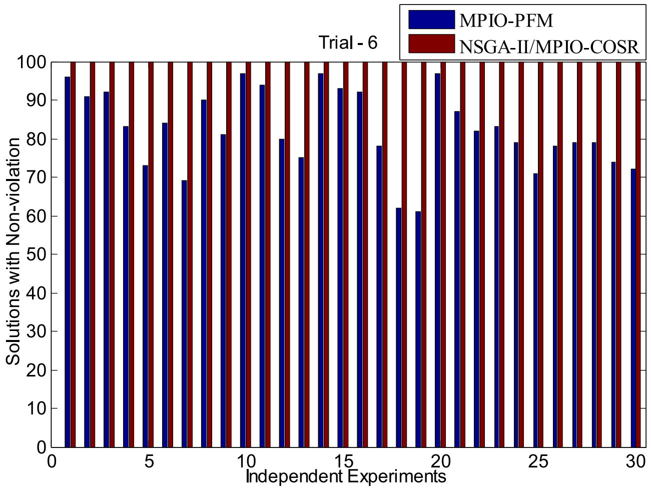

Fig. 17. The number of feasible solutions for Trial 6.

5.3.2. Trial 7: F_{bf}, F_{em} and F_{ap}

The Trial 7 aims to optimize F_{bf}, F_{em} and F_{ap} at the same time on the IEEE 57-node system. Fig. 18 gives the PFs obtained by NSGA-II and MPIO-COSR methods. Meanwhile, Fig. 19 gives the PFs obtained by MPIO-PFM and MPIO-COSR methods. It clearly states that the PFs of NSGA-II and MPIO-PFM are much more distributed than the one of MPIO-COSR algorithm. The number of feasible solutions with zero system-contravention for Trial 7 is shown in Fig. 20 while the control variables of BC solutions are tabulated in Table 9. It can be seen from Table 9 that the BC solutions obtained by MPIO-COSR algorithm includes 42133.3305 $/h of F_{bf}, 1.4360 ton/h of F_{em} and 11.7711 MW of F_{ap} and it can dominate the BC solutions obtained by NSGA-II and MPIO-PFM algorithms. Fig. 20 clearly indicates that the proposed MPIO-COSR algorithm has great advantages to achieve more zero-violation Pareto solutions especially in tri-objective optimization of larger-scale IEEE 57-node system.

5.4. Trials on IEEE 118-node system

In Trial 8, the F_{bf} and F_{em} are optimized simultaneously on the IEEE 118-node system. The details of IEEE 118-node system are provided in [17,20]. The PFs obtained by NSGA-II and

Table 8
The control variables of BC solutions for Trial 5.

Variables	NSGA-II	MPIO-PFM	MPIO-COSR	MOFA-PFA [11]	MOFA-CPA [11]
P_{G2} (MW)	64.0830	65.2618	66.4521	59.497	61.750
P_{G5} (MW)	36.2881	36.1904	30.2492	31.945	30.736
P_{G8} (MW)	28.8384	29.6200	34.5049	34.633	35.000
P_{G11} (MW)	21.9934	21.9466	25.3537	30.000	29.457
P_{G13} (MW)	28.7908	28.1961	26.7809	27.781	27.389
V_{G1} (p.u.)	1.0998	1.0983	1.0992	1.0920	1.0965
V_{G2} (p.u.)	1.0900	1.0899	1.0901	1.0848	1.0904
V_{G5} (p.u.)	1.0717	1.0734	1.0679	1.0642	1.0687
V_{G8} (p.u.)	1.0742	1.0755	1.0754	1.0744	1.0743
V_{G11} (p.u.)	1.0839	1.0829	1.0948	1.0901	1.0892
V_{G13} (p.u.)	1.0767	1.0751	1.0967	1.0413	1.0517
T_{11} (p.u.)	1.0483	1.0459	0.9918	1.0730	1.0760
T_{12} (p.u.)	0.9451	0.9458	0.9783	1.0030	0.9400
T_{15} (p.u.)	0.9806	0.9815	0.9704	1.0400	0.9650
T_{36} (p.u.)	1.0054	1.0065	0.9723	1.0200	0.9890
Q_{C10} (p.u.)	0.0397	0.0394	0.0401	0.0160	0.0460
Q_{C12} (p.u.)	0.0436	0.0441	0.0067	0.0110	0.0020
Q_{C15} (p.u.)	0.0490	0.0489	0.0291	0.0000	0.0470
Q_{C17} (p.u.)	0.0328	0.0322	0.0163	0.0210	0.0410
Q_{C20} (p.u.)	0.0251	0.0255	0.0317	0.0040	0.0340
Q_{C21} (p.u.)	0.0211	0.0211	0.0420	0.0310	0.0290
Q_{C23} (p.u.)	0.0365	0.0361	0.0284	0.0130	0.0060
Q_{C24} (p.u.)	0.0264	0.0257	0.0239	0.0360	0.0240
Q_{C29} (p.u.)	0.0443	0.0428	0.0063	0.0420	0.0220
$F_{f\upsilon}$ ($/h)	914.3344	915.4396	**910.6654**	918.57	916.59
F_{ap} (MW)	4.9074	4.8839	**4.8659**	4.7949	4.7780
F_{em} (ton/h)	0.2259	0.2250	**0.2234**	0.2323	0.2322

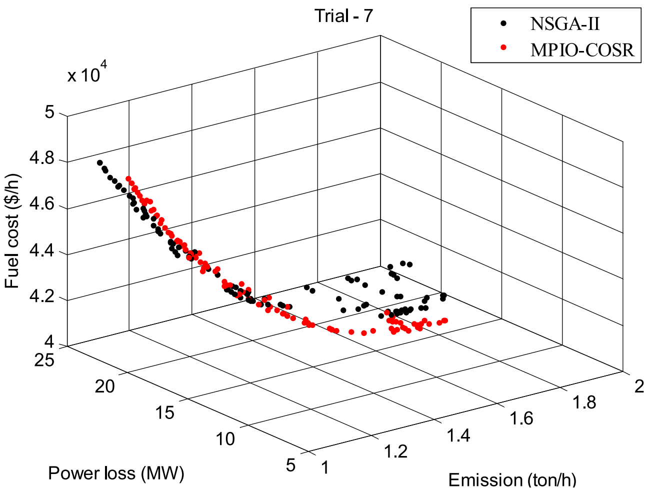

Fig. 18. PFs of NSGA-II and MPIO-COSR for Trial 7.

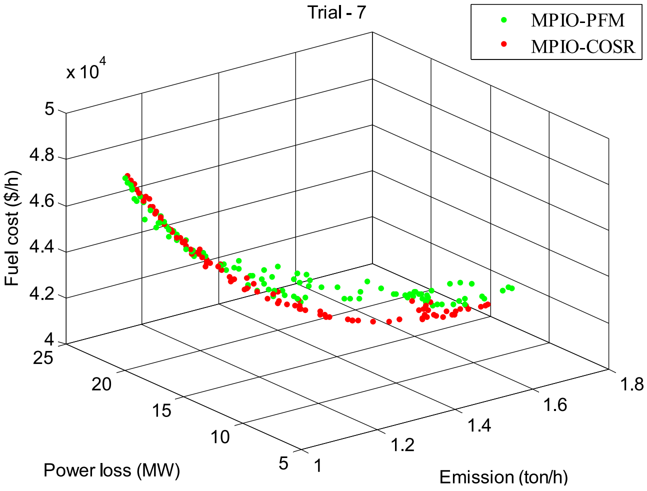

Fig. 19. PFs of MPIO-PFM and MPIO-COSR for Trial 7.

MPIO-COSR algorithm is shown in Fig. 21. It is worth noting that the MPIO-PFM method cannot obtain the qualified PF which achieves zero constraints-violation due to the high-dimension and complex-structure of IEEE 118-node system.

Fig. 21 also indicates the NSGA-II method finds 1.85 ton/h of minimal F_{em} and 56827.89 $/h of minimal F_{bf}. The suggested MPIO-COSR method finds 1.83 ton/h of minimal F_{em} and 56885.22 $/h of minimal F_{bf}. Besides, Table 10 intuitively states that the BC solution of MPIO-COSR algorithm which includes 58697.61 $/h of F_{bf} and 2.56 ton/h of F_{em} is in a dominant position than that of traditional NSGA-II algorithm.

6. Comprehensive analysis

The performance of MPIO-COSR algorithm in solving the MOOPF problems is analyzed based on the convergence, computational complexity, generational distance (GD) and hypervolume (HV) evaluation indexes.

6.1. Convergence analysis

Trial 1 and Trial 6, respectively, carried out on IEEE 30-node and 57-node systems are adopted to demonstrate the advantage of MPIO-COSR algorithm in fast-convergence.

Fig. 22 gives the PFs in iteration process of Trial 1. The NSGA-II, MPIO-PFM and MPIO-COSR algorithms, respectively, realize zero constraints-violation at the 59th, 97-15th and 33-15th iteration. Besides, Fig. 22 also indicates NSGA-II method achieves the relatively uniformly-distributed PF around the 150th iteration while

Table 9
The control variables of BC solutions for Trial 6 and Trial 7.

Variables	Trial 6				Trial 7		
	NSGA-II	MPIO-PFM	MPIO-COSR	MODFA [20]	NSGA-II	MPIO-PFM	MPIO-COSR
P_{G2} (MW)	98.6492	99.6650	100.0000	99.9703	96.3769	67.7035	88.7122
P_{G3} (MW)	86.8695	91.7559	85.5144	88.2975	53.7609	67.0055	72.3392
P_{G6} (MW)	99.9425	100.0000	99.7250	99.9135	77.7030	91.5160	99.6767
P_{G8} (MW)	362.0020	362.0245	348.8135	343.6324	360.5024	306.2448	366.8565
P_{G9} (MW)	100.0000	99.9343	98.8729	99.9138	99.9403	93.6950	99.6824
P_{G12} (MW)	297.9466	299.1797	306.1420	310.8878	406.8049	410.0000	366.3118
V_{G1} (p.u.)	0.9809	1.0118	1.0793	1.0600	1.0007	1.0485	1.0964
V_{G2} (p.u.)	0.9789	1.0092	1.0768	1.0544	0.9850	1.0402	1.0963
V_{G3} (p.u.)	0.9824	1.0137	1.0716	1.0467	0.9675	1.0498	1.0963
V_{G6} (p.u.)	1.0278	1.0216	1.0903	1.0500	0.9752	1.0605	1.0954
V_{G8} (p.u.)	1.0658	1.0150	1.0879	1.0558	0.9885	1.0555	1.0964
V_{G9} (p.u.)	1.0433	1.0098	1.0651	1.0433	0.9871	1.0380	1.0923
V_{G12} (p.u.)	1.0080	0.9967	1.0460	1.0332	0.9929	1.0429	1.0852
T_{19} (p.u.)	0.9498	0.9212	1.0312	0.9916	1.0154	0.9822	1.0899
T_{20} (p.u.)	0.9329	1.0102	1.0520	0.9805	0.9194	1.0046	0.9706
T_{31} (p.u.)	1.0263	1.0939	0.9862	0.9972	1.0891	1.0638	1.0295
T_{35} (p.u.)	1.0399	0.9523	0.9439	0.9693	1.0387	1.0609	1.0961
T_{36} (p.u.)	1.0159	1.0884	1.0043	0.9646	1.0096	1.0053	0.9578
T_{37} (p.u.)	1.0754	1.0571	1.0161	0.9788	1.0275	1.0083	1.0249
T_{41} (p.u.)	0.9481	0.9632	1.0041	0.9570	0.9081	1.0082	1.0245
T_{46} (p.u.)	1.0483	0.9715	1.0109	0.9741	1.0386	1.0029	0.9303
T_{54} (p.u.)	0.9091	1.0519	0.9165	1.0310	0.9122	0.9442	1.0035
T_{58} (p.u.)	0.9077	0.9055	0.9596	0.9523	0.9025	0.9615	1.0094
T_{59} (p.u.)	0.9580	0.9330	0.9588	0.9452	0.9000	0.9271	0.9971
T_{65} (p.u.)	0.9964	0.9520	0.9773	1.0045	0.9226	0.9425	1.0076
T_{66} (p.u.)	0.9909	1.0150	0.9573	0.9344	0.9032	0.9000	0.9473
T_{71} (p.u.)	0.9488	0.9042	0.9922	0.9481	0.9007	0.9405	0.9786
T_{73} (p.u.)	1.0334	1.0711	1.0562	0.9621	1.0725	1.0098	1.0282
T_{76} (p.u.)	1.0567	0.9418	0.9812	0.9587	0.9871	1.0601	1.0444
T_{80} (p.u.)	1.0292	1.0203	1.0582	0.9703	0.9765	0.9898	1.0236
Q_{C18} (p.u.)	0.0108	0.1988	0.2283	0.1896	0.1584	0.0680	0.0992
Q_{C25} (p.u.)	0.2006	0.1539	0.1166	0.1191	0.1829	0.1041	0.1625
Q_{C53} (p.u.)	0.2091	0.0821	0.1542	0.0331	0.0660	0.1703	0.1389
F_{bf} ($/h)	43351.1353	43205.8477	**43131.2743**	43174.5740	42187.9327	43133.9896	**42133.3305**
F_{em} (ton/h)	1.2466	1.2386	**1.2314**	1.2679	1.5765	1.5027	**1.4360**
F_{ap} (MW)	–	–	–	–	13.4586	11.7899	**11.7711**

Table 10
The special solutions for Trial 8.

Algorithms	BC		Boundary solution-1		Boundary solution-2	
	F_{bf}	F_{em}	minimal F_{bf}	F_{em}	F_{bf}	minimal F_{em}
NSGA-II	58804.23	2.58	**56827.89**	4.01	61811.41	1.85
MPIO-COSR	**58697.61**	**2.56**	56885.22	3.94	61344.67	**1.83**

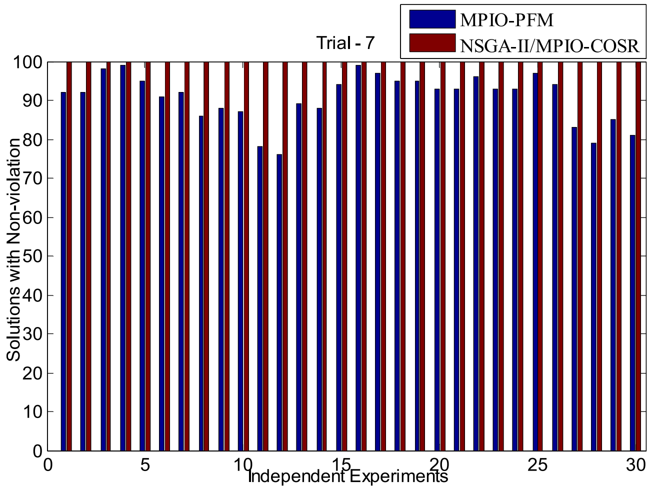

Fig. 20. The number of feasible solutions for Trial 7.

MPIO-COSR algorithm can achieve ideal PF around the 75-15th iteration. However, it is more difficult for MPIO-PFM method to find the satisfactory PF than NSGA-II and MPIO-COSR methods.

Furthermore, Fig. 23 shows the PFs in iteration process of Trial 6 which is implemented on the IEEE 57-node system. It can be seen from Fig. 23 that the NSGA-II, MPIO-PFM and MPIO-COSR algorithms find the qualified PF at the 166th, 174-15th and 126-15th iteration in turn. Fig. 23 also states that the proposed MPIO-COSR algorithm finds the uniformly-distributed PF at the 180-15th iteration, which outperforms the NSGA-II and MPIO-PFM algorithms.

In summary, the MPIO-COSR algorithm is more advantageous in the optimization-performance and fast-convergence.

6.2. GD Index

To measure the distance between the obtained PF and the real one, the index of GD defined as Eq. (40) is employed [20,39,46]. It is worthy to note that GD=0 indicates all generated non-inferior solutions are completely consistent with the real PF. Therefore, a smaller value of GD index represents the better convergence to

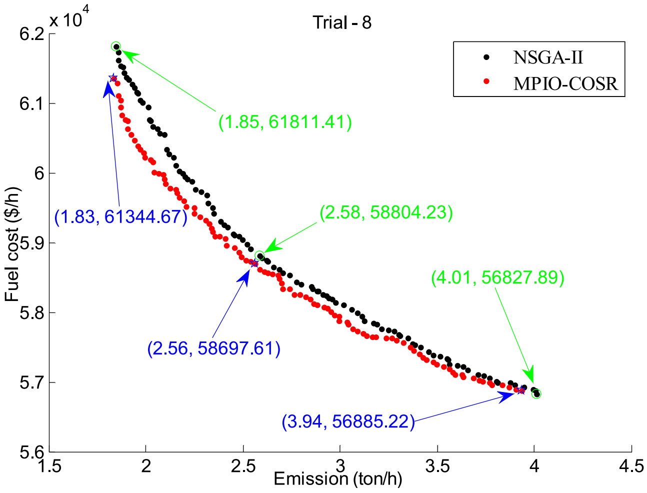

Fig. 21. PFs of Trial 8.

true PF.

$$GD = \frac{\sqrt{\sum_{i=1}^{N} Euc_i^2}}{N} \tag{40}$$

where Euc_i is the Euclidean distance between the ith non-inferior solution and the nearest solution of the reference PF. In this paper, the reference PF is actually the obtained PF corresponding to the final BC solution.

Table 11
The average and deviation values of GD index.

GD index		NSGA-II	MPIO-PFM	MPIO-COSR
Trial 1	Average	0.0777	0.0792	0.0738
	Standard deviation	0.0149	0.0158	0.0146
Trial 2	Average	0.0887	0.0797	0.0754
	Standard deviation	0.0185	0.0154	0.0146
Trial 3	Average	0.0825	0.0854	0.0669
	Standard deviation	0.0179	0.0194	0.0129
Trial 6	Average	0.4872	0.4886	0.4614
	Standard deviation	0.0944	0.1200	0.0902
Trial 8	Average	0.5466	–	0.4039
	Standard deviation	0.1500	–	0.0931

As a common statistical tool, the boxplot technology is used to analyze the average, standard deviation and outlier-distribution of GD indicator for five bi-objectives optimization trials (Trial 1∼Trial 3, Trial 6 and Trial 8).

Fig. 24 shows the boxplots of GD criteria and it clearly shows that the MPIO-COSR algorithm achieves the closer boxplots in 30 separate studies. In detail, Table 11 gives the mean and deviation values of all bi-objective cases. The minimal average and deviation values of GD index quantitatively show that MPIO-COSR method is superior to NSGA-II and MPIO-PFM methods in the convergence and distribution of obtained PFs.

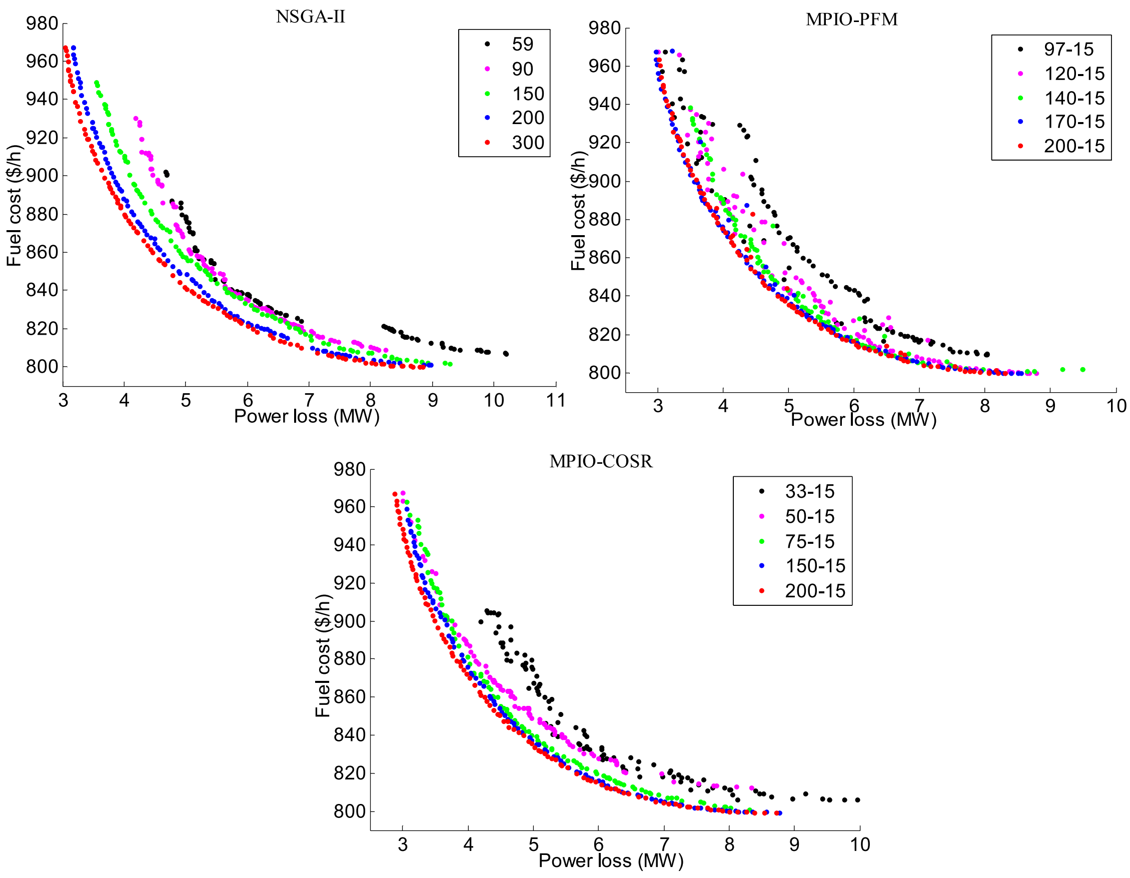

Fig. 22. Iterative process of Trial 1.

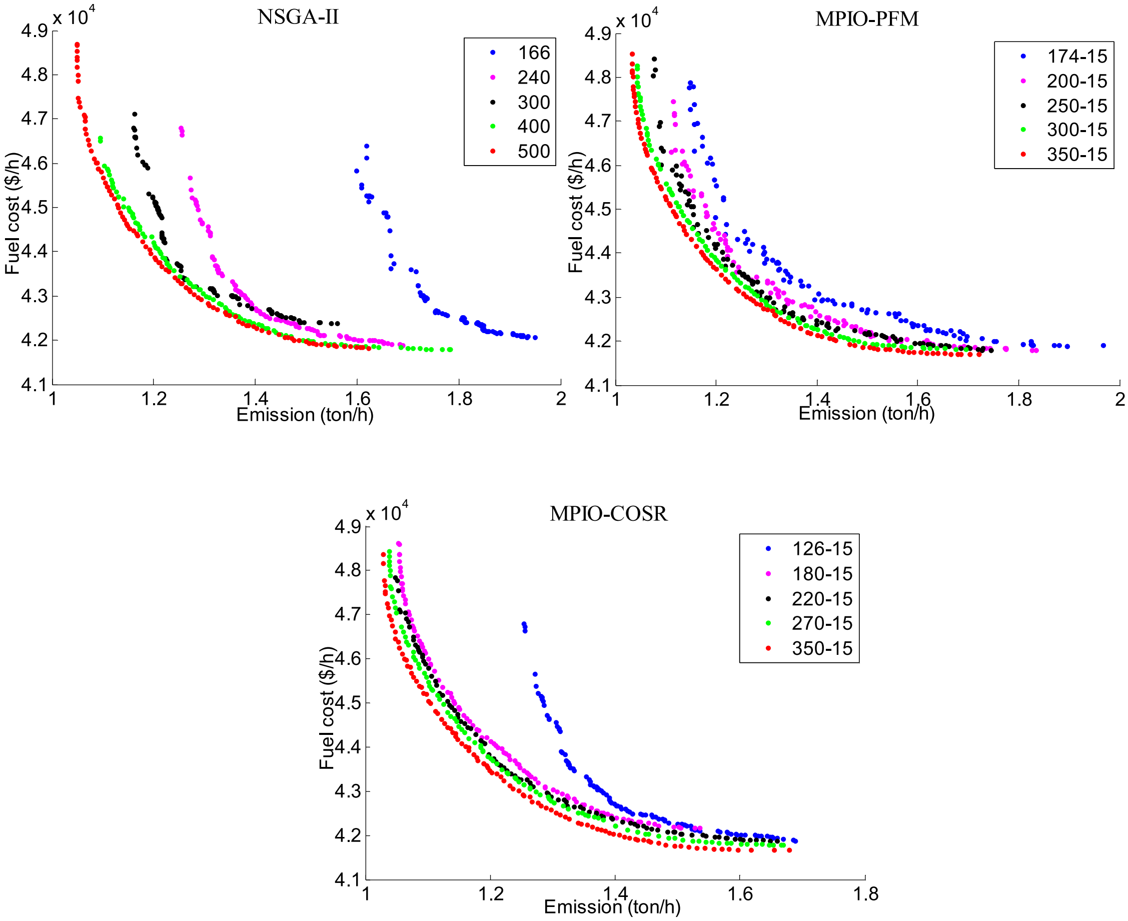

Fig. 23. Iterative process of Trial 6.

6.3. HV Index

Besides the GD index, HV index as the widely-used diversity metric is also adopted in this paper. The HV metric is defined as Eq. (41) [20]. The HV value is actually the volume of space covered by obtained PF. The larger HV value usually means that the diversity of the corresponding PF is more preferable.

$$HV = volume\left(\bigcup_{i=1}^{N} v_i\right) \tag{41}$$

Fig. 25 shows the boxplots of HV index while Table 12 gives the mean and deviation values of five involved bi-objective cases. It indicates that MPIO-COSR algorithm gets the larger average value of HV index on these trials of IEEE 30-node system. Although the HV index of suggested MPIO-COSR algorithm is not the best in the more complex Trial 6 and Trial 8, there is not much difference with the maximum value. What is more notable is that the MPIO-COSR algorithm obtains the smallest deviation value of HV metric in all five experiments, which undoubtedly proves the superiority of MPIO-COSR algorithm in operational stability.

6.4. Computational complexity

Finally, the computational complexity of three algorithms represented by the average running time of 30 independent trials is discussed. The mean CPU time of eight involved trials are summarized in Table 13. It clearly suggests that the MPIO-PFM and MPIO-COSR algorithms need more CPU time due to the searching process of landmark comparing with NSGA-II method. In the future, the improvement of pigeon-inspired algorithm will focus on improving the search efficiency and reducing the running time.

Table 12
The average and deviation values of HV index.

HV index		NSGA-II	MPIO-PFM	MPIO-COSR
Trial 1	Average	820.4107	965.0256	987.0964
	Standard deviation	26.1843	37.6850	14.6959
Trial 2	Average	1212.3191	1072.5836	1237.9271
	Standard deviation	39.8316	31.7375	13.5627
Trial 3	Average	14.3574	15.9851	17.7299
	Standard deviation	0.7071	0.4996	0.2134
Trial 6	Average	5722.3041	6983.9397	6286.3466
	Standard deviation	1056.8106	170.5493	112.1408
Trial 8	Average	7362.7959	–	7356.2989
	Standard deviation	661.9132	–	409.5993

7. Conclusion

A novel MPIO algorithm and an innovative COSR strategy are proposed in this paper to handle the MOOPF problems with multiple objectives and strict constraints. Eight simulation cases of optimizing the active power loss, the emission and the fuel cost (with valve-point loadings) are performed on three different scale power systems. Numerous results powerfully demonstrate that the presented MPIO-COSR algorithm is more conducive than NSGA-II and MPIO-PFM methods in seeking the superior PFs with evenly-distribution and the qualified BC solutions with zero constraint-violation. The preferable advantages

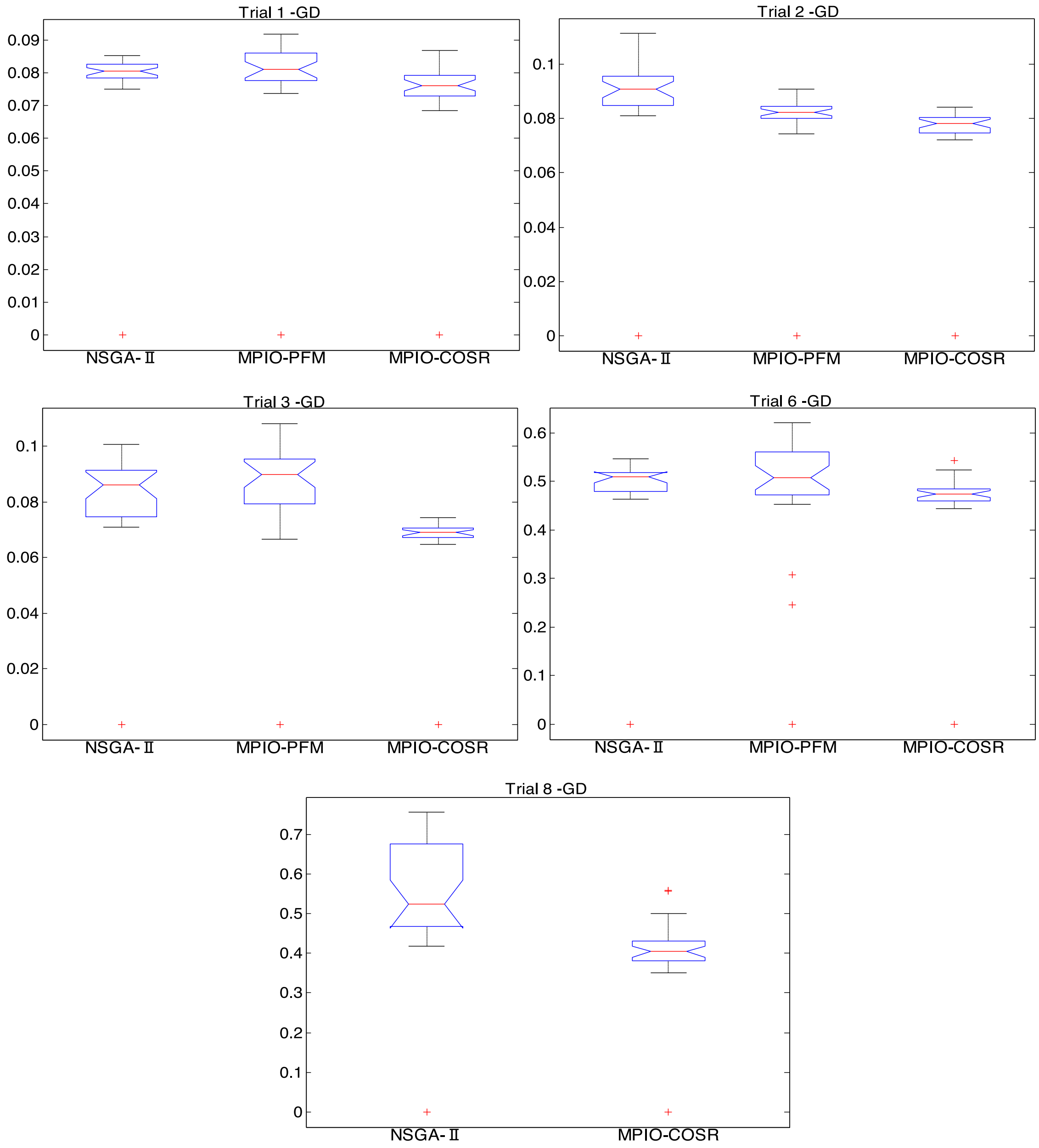

Fig. 24. GD index of five bi-objective trials.

Table 13
The mean running time.

Algorithm	The mean CPU time (second)							
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8
NSGA-II	204.7	212.7	203.5	280.8	307.7	506.1	518.6	1740.5
MPIO-PFM	221.6	232.2	220.1	297.4	319.8	540.8	551.0	1814.7
MPIO-COSR	215.5	219.6	214.6	287.6	311.7	512.6	526.5	1788.9

of proposed MPIO-COSR algorithm in high-dimensional optimization are verified by three trials carried out on the IEEE 57-node and 118-node systems. Additionally, the GD and HV evaluation indexes also validate that the MPIO-COSR algorithm is capable to obtain preferable PFs with favorable-diversity and superior-convergence, even in the complex large-scale electric systems.

Consequently, the novel MPIO-COSR algorithm is competent to solve the security-constrained MOOPF problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

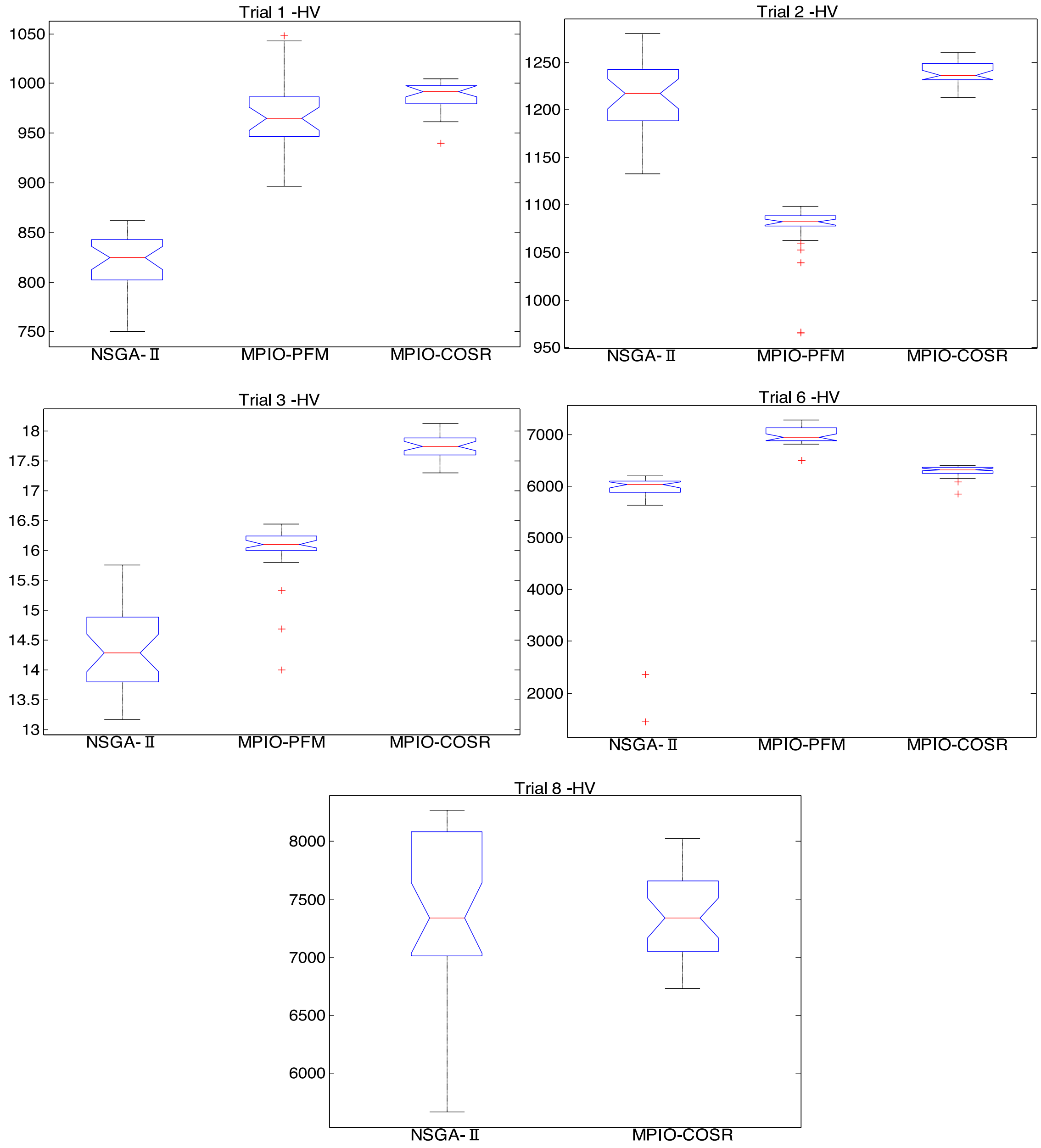

Fig. 25. HV index of five bi-objective trials.

CRediT authorship contribution statement

Gonggui Chen: Conceptualization, Methodology, Writing - review & editing. **Jie Qian:** Conceptualization, Methodology, Writing - original draft. **Zhizhong Zhang:** Validation, Data curation. **Shuaiyong Li:** Validation, Supervision.

Acknowledgments

This work is supported by the National Natural Science Foundation Project of China (No. 61703066), Natural Science Foundation Project of Chongqing, China (No. cstc2018jcyjAX0536), Innovation Team Program of Chongqing Education Committee, China (CXTDX201601019) and Chongqing University Innovation Team, China under Grant (KJTD201312).

References

[1] X. Yuan, Z. Chen, Y. Yuan, et al., Design of fuzzy sliding mode controller for hydraulic turbine regulating system via input state feedback linearization method, Energy 93 (2015) 173–187, http://dx.doi.org/10.1016/j.energy.2015.09.025.

[2] G. Chen, L. Liu, Z. Zhang, et al., Optimal reactive power dispatch by improved GSA-based algorithm with the novel strategies to handle constraints, Appl. Soft Comput. 50 (2017) 58–70, http://dx.doi.org/10.1016/j.asoc.2016.11.008.

[3] T. Cardoso Bora, V. Cocco Mariani, L. Dos Santos Coelho, Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm, Appl. Therm. Eng. 146 (2019) 688–700, http://dx.doi.org/10.1016/j.applthermaleng.2018.10.020.

[4] X. Yuan, B. Ji, Y. Yuan, et al., An efficient chaos embedded hybrid approach for hydro-thermal unit commitment problem, Energy Convers. Manage. 91 (2015) 225–237, http://dx.doi.org/10.1016/j.enconman.2014.12.021.

[5] X. Yuan, B. Zhang, P. Wang, et al., Multi-objective optimal power flow based on improved strength Pareto evolutionary algorithm, Energy 122 (2017) 70–82, http://dx.doi.org/10.1016/j.energy.2017.01.071.

[6] S. Rahmani, N. Amjady, Enhanced goal attainment method for solving multi-objective security-constrained optimal power flow considering dynamic thermal rating of lines, Appl. Soft Comput. 77 (2019) 41–49, http://dx.doi.org/10.1016/j.asoc.2019.01.014.

[7] J.B. Hmida, M.J. Morshed, J. Lee, et al., Hybrid imperialist competitive and grey wolf algorithm to solve multiobjective optimal power flow with wind

and solar units, Energies 11 (11) (2018) 2891, http://dx.doi.org/10.3390/en11112891.

[8] O.M. Bamigbola, M.M. Ali, M.O. Oke, Mathematical modeling of electric power flow and the minimization of power losses on transmission lines, Appl. Math. Comput. 241 (2014) 214–221, http://dx.doi.org/10.1016/j.amc.2014.05.039.

[9] W. W, H. Hashim, M. Norman, et al., A novel quasi-oppositional modified jaya algorithm for multi-objective optimal power flow solution, Appl. Soft Comput. 65 (2018) 360–373, http://dx.doi.org/10.1016/j.asoc.2018.01.039.

[10] S. Khunkitti, A. Siritaratiwat, S. Premrudeepreechacharn, et al., A hybrid DA-PSO optimization algorithm for multiobjective optimal power flow problems, Energies 11 (9) (2018) 2270, http://dx.doi.org/10.3390/en11092270.

[11] G. Chen, X. Yi, Z. Zhang, et al., Solving the multi-objective optimal power flow problem using the multi-objective firefly algorithm with a constraints-prior pareto-domination approach, Energies 11 (12) (2018) 3438, http://dx.doi.org/10.3390/en11123438.

[12] X. Fu, F.T.S. Chan, B. Niu, et al., A multi-objective pigeon inspired optimization algorithm for fuzzy production scheduling problem considering mould maintenance, Sci. China Inf. Sci. 62 (7) (2019) 70202, http://dx.doi.org/10.1007/s11432-018-9693-2.

[13] Y. Zhong, L. Wang, M. Lin, et al., Discrete pigeon-inspired optimization algorithm with metropolis acceptance criterion for large-scale traveling salesman problem, Swarm Evol. Comput. 48 (2019) 134–144, http://dx.doi.org/10.1016/j.swevo.2019.04.002.

[14] Q. Huaxin, D. Haibin, A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles, Inform. Sci. 509 (2020) 515–529, http://dx.doi.org/10.1016/j.ins.2018.06.061.

[15] Y. Wang, G. Zhang, X. Zhang, Multilevel image thresholding using tsallis entropy and cooperative pigeon-inspired optimization bionic algorithm, J. Bionic Eng. 16 (2019) 954–964, http://dx.doi.org/10.1007/s42235-019-0109-1.

[16] F. Jiang, J. He, T. Tian, A clustering-based ensemble approach with improved pigeon-inspired optimization and extreme learning machine for air quality prediction, Appl. Soft Comput. 85 (2019) 105827, http://dx.doi.org/10.1016/j.asoc.2019.105827.

[17] G. Chen, J. Qian, Z. Zhang, et al., Multi-objective optimal power flow based on hybrid firefly-bat algorithm and constraints-prior object-fuzzy sorting strategy, IEEE Access 7 (2019) 139726–139745, http://dx.doi.org/10.1109/ACCESS.2019.2943480.

[18] G. Xiong, S. Dongyuan, Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects, Energy 157 (2018) 424–435, http://dx.doi.org/10.1016/j.energy.2018.05.180.

[19] T. Ding, R. Bo, F. Li, et al., Exact penalty function based constraint relaxation method for optimal power flow considering wind generation uncertainty, IEEE Trans. Power Syst. 30 (3) (2015) 1546–1547, http://dx.doi.org/10.1109/TPWRS.2014.2341177.

[20] G. Chen, X. Yi, Z. Zhang, et al., Applications of multi-objective dimension-based firefly algorithm to optimize the power losses, emission, and cost in power systems, Appl. Soft Comput. 68 (2018) 322–342, http://dx.doi.org/10.1016/j.asoc.2018.04.006.

[21] S. Rahmani, N. Amjady, Improved normalised normal constraint method to solve multi-objective optimal power flow problem, IET Gener. Transm. Distrib. 12 (4) (2018) 859–872, http://dx.doi.org/10.1049/iet-gtd.2017.0289.

[22] G. Chen, L. Zhengmei, Z. Zhizhong, Improved krill herd algorithm with novel constraint handling method for solving optimal power flow problems, Energies 11 (1) (2018) 76, http://dx.doi.org/10.3390/en11010076.

[23] A.F. Attia, R.A. El Sehiemy, H.M. Hasanien, Optimal power flow solution in power systems using a novel Sine-Cosine algorithm, Int. J. Electr. Power Energy Syst. 99 (2018) 331–343, http://dx.doi.org/10.1016/j.ijepes.2018.01.024.

[24] X. Yuan, P. Wang, Y. Yuan, et al., A new quantum inspired chaotic artificial bee colony algorithm for optimal power flow problem, Energy Convers. Manage. 100 (2015) 1–9, http://dx.doi.org/10.1016/j.enconman.2015.04.051.

[25] E.P.D. Carvalho, A.D.S. Júnior, T.F. Ma, Reduced gradient method combined with augmented Lagrangian and barrier for the optimal power flow problem, Appl. Math. Comput. 200 (2) (2008) 529–536, http://dx.doi.org/10.1016/j.amc.2007.11.025.

[26] A. Panda, S. Pani, A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems, Appl. Soft Comput. 46 (2016) 344–360, http://dx.doi.org/10.1016/j.asoc.2016.04.030.

[27] K. Deb, A. Pratap, S. Agarwal, et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 2 (6) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[28] P.C. Roy, K. Deb, M.M. Islam, An efficient nondominated sorting algorithm for large number of fronts, IEEE Trans. Cybern. 49 (3) (2019) 859–869, http://dx.doi.org/10.1109/TCYB.2017.2789158.

[29] H. Li, K. Deb, Q. Zhang, et al., Comparison between MOEA/D and NSGA-III on a set of many and multi-objective benchmark problems with challenging difficulties, Swarm Evol. Comput. 46 (2019) 104–117, http://dx.doi.org/10.1016/j.swevo.2019.02.003.

[30] Y.F. Li, N. Pedroni, E. Zio, A memetic evolutionary multi-objective optimization method for environmental power unit commitment, IEEE Trans. Power Syst. 28 (3) (2013) 2660–2669, http://dx.doi.org/10.1109/TPWRS.2013.2241795.

[31] R. Chai, S. Al, T. Antonios, et al., Unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning, J. Guid. Control Dyn. 41 (7) (2018) 1521–1530, http://dx.doi.org/10.2514/1.g003189.

[32] T. Niknam, M.R. Narimani, J. Aghaei, et al., Improved particle swarm optimisation for multi-objective optimal power flow considering the cost, loss, emission and voltage stability index, IET Gener. Transm. Distrib. 6 (6) (2012) 515–527, http://dx.doi.org/10.1049/iet-gtd.2011.0851.

[33] Y.M. Deng, W. Zhu, H. Duan, Hybrid membrane computing and pigeon-inspired optimization algorithm for brushless direct current motor parameter design, Sci. China Technol. Sci. 59 (9) (2016) 1435–1441, http://dx.doi.org/10.1007/s11431-016-6048-8.

[34] Z. Cui, J. Zhang, Y. Wang, et al., A pigeon-inspired optimization algorithm for many-objective optimization problems, Sci. China Inf. Sci. 62 (7) (2019) 702127, http://dx.doi.org/10.1007/s11432-018-9729-5.

[35] H. Duan, P. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, Int. J. Intell. Comput. Cybern. 7 (1) (2014) 24–37, http://dx.doi.org/10.1108/IJICC-02-2014-0005.

[36] Y. Sun, S. Yin, H. Li, et al., GPOGC: gaussian pigeon-oriented graph clustering algorithm for social networks cluster, IEEE Access 7 (2019) 99254–99262, http://dx.doi.org/10.1109/ACCESS.2019.2926816.

[37] J. Zhang, Q. Tang, P. Li, et al., A modified MOEA/D approach to the solution of multi-objective optimal power flow problem, Appl. Soft Comput. 47 (2016) 494–514, http://dx.doi.org/10.1016/j.asoc.2016.06.022.

[38] A.A.A. Mohamed, Y.S. Mohamed, A.A.M. El-Gaafary, et al., Optimal power flow using moth swarm algorithm, Electr. Power Syst. Res. 142 (2017) 190–206, http://dx.doi.org/10.1016/j.epsr.2016.09.025.

[39] G. Chen, J. Qian, Z. Zhang, et al., Applications of novel hybrid bat algorithm with constrained pareto fuzzy dominant rule on multi-objective optimal power flow problems, IEEE Access 7 (2019) 52060–52084, http://dx.doi.org/10.1109/ACCESS.2019.2912643.

[40] M.A. Medina, S. Das, C.A.C. Coello, et al., Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow – A comparative study, Eng. Appl. Artif. Intell. 32 (2014) 10–20, http://dx.doi.org/10.1016/j.engappai.2014.01.016.

[41] S. Sivasubramani, K.S. Swarup, Multi-objective harmony search algorithm for optimal power flow problem, Int. J. Electr. Power Energy Syst. 33 (3) (2011) 745–752, http://dx.doi.org/10.1016/j.ijepes.2010.12.031.

[42] H. Pulluri, R. Naresh, V. Sharma, An enhanced self-adaptive differential evolution based solution methodology for multiobjective optimal power flow, Appl. Soft Comput. 54 (2017) 229–245, http://dx.doi.org/10.1016/j.asoc.2017.01.030.

[43] M. Ghasemi, S. Ghavidel, M.M. Ghanbarian, et al., Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm, Energy 78 (2014) 276–289, http://dx.doi.org/10.1016/j.energy.2014.10.007.

[44] A.E. Chaib, H.R.E.H. Bouchekara, R. Mehasni, et al., Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm, Int. J. Electr. Power Energy Syst. 81 (2016) 64–77, http://dx.doi.org/10.1016/j.ijepes.2016.02.004.

[45] D. Narges, T. Saeed, H. Mehrdad, et al., Adaptive group search optimization algorithm for multi-objective optimal power flow problem, Appl. Soft Comput. 38 (2016) 1012–1024, http://dx.doi.org/10.1016/j.asoc.2015.10.057.

[46] H. Ishibuchi, R. Imada, Y. Setoguchi, et al., Reference point specification in inverted generational distance for triangular linear pareto front, IEEE Trans. Evol. Comput. 22 (6) (2018) 961–975, http://dx.doi.org/10.1109/TEVC.2017.2776226.