Complex System Modeling and Simulation

Volume 1 | Issue 1 Article 2

2021

Nonlinear Equations Solving with Intelligent Optimization Algorithms: A Survey

Wenyin Gong
School of Computer Science, China University of Geosciences, Wuhan 430074, China

Zuowen Liao
Beibu Gulf Ocean Development Research Center, Beibu Gulf University, Qinzhou 535000, China

Xianyan Mi
College of Economics and Management, Beibu Gulf University, Qinzhou 535000, China

Ling Wang
Department of Automation, Tsinghua University, Beijing 100084, China

Recommended Citation

Wenyin Gong, Zuowen Liao, Xianyan Mi, Ling Wang, Yuanyuan Guo. Nonlinear Equations Solving with Intelligent Optimization Algorithms: A Survey. Complex System Modeling and Simulation 2021, 1(1): 15-32.

COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 02/06 pp 15–32
Volume 1, Number 1, March 2021
DOI: 10.23919/CSMS.2021.0002

Nonlinear Equations Solving with Intelligent Optimization Algorithms: A Survey

Wenyin Gong, Zuowen Liao*, Xianyan Mi*, Ling Wang, and Yuanyuan Guo

Abstract: Nonlinear Equations (NEs), which may usually have multiple roots, are ubiquitous in diverse fields. One of the main purposes of solving NEs is to locate as many roots as possible simultaneously in a single run, however, it is a difficult and challenging task in numerical computation. In recent years, Intelligent Optimization Algorithms (IOAs) have shown to be particularly effective in solving NEs. This paper provides a comprehensive survey on IOAs that have been exploited to locate multiple roots of NEs. This paper first revisits the fundamental definition of NEs and reviews the most recent development of the transformation techniques. Then, solving NEs with IOAs is reviewed, followed by the benchmark functions and the performance comparison of several state-of-the-art algorithms. Finally, this paper points out the challenges and some possible open issues for solving NEs.

Key words: Nonlinear Equations (NEs); Intelligent Optimization Algorithms (IOA); multiple roots location; transformation techniques; diversity preservation

1 Introduction

Nonlinear Equations (NEs)[1] are involved in physics[2], economics[3], complex system[4], power system[5], mechanical manufacturing[6], and many other fields[7]. In 1998, 18 challenging mathematical problems for the 21st century were listed by Fields Medalist Steve Smale[8]. Among these 18 problems, there are three ones related to the NEs. Therefore, it is very important to solve NEs.

In the literature, various methods were proposed for solving NEs, which can be briefly classified into two groups: i.e., numerical methods and Intelligent Optimization Algorithms (IOAs).

(1) *Numerical methods:* These type of methods[9] are commonly used to solve NEs, such as Newton method[10], Quasi-Newton method[11], and tensor method[12]. However, they have some possible defects, such as the sensitivity to the initial guess, requirement of derivative information, and expensive computational budget. More important, most of them can only locate only one root in a single run.

(2) *IOAs:* The emergence of artificial intelligence has led to the great achievement of IOAs[13]. In the literature, there are various IOAs presented, such as Genetic Algorithms (GAs)[14], Particle Swarm Optimization (PSO)[15], Differential Evolution (DE)[16], Covariance Matrix Adaptation based Evolution Strategy (CMA-ES)[17]. These methods have some distinguished advantages, for example: (1) they are not sensitive to the non-convexity and discontinuity of the objective

• Wenyin Gong is with School of Computer Science, China University of Geosciences, Wuhan 430074, China. E-mail: wygong@cug.edu.cn.
• Zuowen Liao is with Beibu Gulf Ocean Development Research Center, Beibu Gulf University, Qinzhou 535000, China. E-mail: liaozuowen@bbgu.edu.cn.
• Xianyan Mi is with College of Economics and Management, Beibu Gulf University, Qinzhou 535000, China. E-mail: 563292753@qq.com.
• Ling Wang is with Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: wangling@tsinghua.edu.cn.
• Yuanyuan Guo is with Cvent Company, Fredericton, E3B 5H8, Canada. E-mail: yy2guo@gmail.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2021-02-16; accepted: 2021-02-25

function; (2) they can easily converge to the near-optimal solution for the real-world optimization problems; (3) most of them are easily to be implemented; and (4) they are the population-based methods, which can possibly find multiple different solutions in the final population. Therefore, the use of IOAs for the NEs obtains increasing attention recently[7].

Although various IOAs are used to deal with NEs, most of them only consider to find one root[18–20]. However, in many real-world scenarios, most of NEs have multiple roots, such as the multiple steady states in the chemical engineering[21]. It is important to find different roots of NEs as many as possible to provide more effective decision making. Locating multiple roots, especially in a single run, is really useful. For example:

- Different roots may help to explain the properties of the problem under study, such as the distribution of the roots in the search region. Obviously, it can provide much richer information about the problem domain than a single-optimum method.
- Multiple high-quality roots are able to provide multiple choices for decision-makers. In many practical applications, decision-makers want to have multiple optimal solutions to make the best decision under different conditions or states. If one solution is not appropriate, another alternative can be taken immediately[22]. Therefore, locating multiple different roots is a step toward providing a robust decision.

Recently, the use of IOAs for locating different roots of NEs gets more consideration, such as the MONES method with bi-objective transformation[23], the RADE method with repulsion technique[24], and the TPEA method with multi-objective optimization and niching technique[25]. Although Wu et al.[26] summarized several methods for solving NEs using GAs in 2014, little attention is paid to other IOAs for locating multiple roots of NEs. It is the rapid development of IOAs for solving NEs that motivates us to carry out this comprehensive survey of the latest research outcomes in this field.

This paper presents an updated survey on IOAs for locating multiple roots of NEs in one run. It differs from the previous surveys in the following aspects:

- We concentrate on giving an updated survey of state-of-the-art methods that apply IOAs to locate multiple roots of NEs.
- We emphasize more on the relationship between IOAs and population diversity, together with other local search methods.
- We provide more detailed standard test functions for NEs, performance metrics, and performance comparison of several advanced IOAs.

The rest of the paper is organized as follows. Section 2 describes the general formulation of NEs and introduces various transformation techniques in detail. In Section 3, we revisit the representative IOAs to solve NEs. Section 4 shows the test functions, the performance metrics, and the experimental results, followed by the discussions and open issues in Section 5. Finally, Section 6 concludes the paper.

2 Problem Formulation

In this section, the general formulation of NEs is first given. Then, different transformation techniques to convert the NEs into an optimization problem are introduced.

2.1 Nonlinear equations

Without loss of generality, the formulation of NEs is as follows:

$$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}, \ldots, f_n(\boldsymbol{x}))^{\mathrm{T}} = \boldsymbol{0} \tag{1}$$

where n is the number of equations, $\boldsymbol{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$ is a decision vector with m dimensions, where $\boldsymbol{x} \in \mathcal{S}$ is a search space. Each variable is usually bounded as the following:

$$x_j^L \leqslant x_j \leqslant x_j^U,$$

where $j = 1, \ldots, m$, x_j^L is the lower bound, and x_j^U is the upper bound of x_j. Note that, a root $\boldsymbol{x}^* \in \mathcal{S}$ satisfies $\forall i \in \{1, n\}$, $f_i(\boldsymbol{x}^*) = 0$.

Most of NEs have various roots. For example,

$$\begin{cases} f_1(x) = 4x_1^3 - 3x_1 - \cos(x_2) = 0; \\ f_2(x) = \sin(x_1^2) - |x_2|) = 0, \end{cases}$$

where $x_1, x_2 \in [-\pi, \pi]$. This problem is plotted in Fig. 1. From Fig. 1, it can be seen that there are six

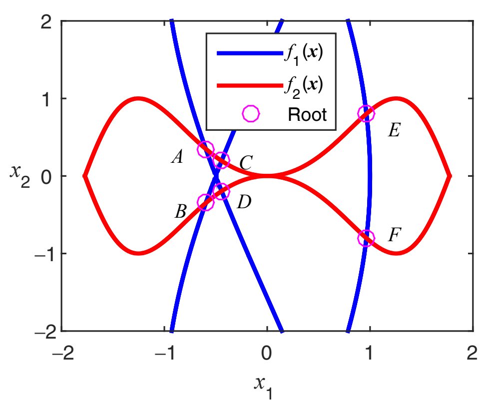

Fig. 1 An example of a NEs problem with six roots, where "o" indicates the root of this problem.

https://dc.tsinghuajournals.com/complex-system-modeling-and-simulation/vol1/iss1/2
DOI: 10.23919/CSMS.2021.0002

roots $A-F$. As mentioned above, it is useful to find different roots as many as possible.

2.2 Transformation techniques

Generally, to solve NEs with the optimization methods, an NEs problem is transformed into an optimization problem. In the literature, different transformation techniques have been developed, i.e., (1) Single-Objective Transformation (SOT); (2) Repulsion-Based Transformation (RBT); (3) Constrained Single-Objective Transformation (CSOT); (4) NonLinear Programming Transformation (NLPT); and (5) Multi-Objective Transformation (MOT). These techniques are reviewed as follows.

(1) *SOT:* An NEs problem is transformed into a single-objective optimization problem as follows[27–30]:

$$\min F(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(\boldsymbol{x}) \tag{2}$$

or

$$\min F(\boldsymbol{x}) = \sum_{i=1}^{n} |f_i(\boldsymbol{x})| \tag{3}$$

This transformation technique is simple and straightforward. The optimal solution of Eq. (2) or (3) is corresponding to a root of Eq. (1). However, without additional diversity preservation technique, the optimization algorithm using the single-objective transformation usually locates one root in one run.

(2) *RBT:* The repulsion technique[31] is an effective method for locating different roots of NEs. The RBT method is usually based on SOT and repulsion techniques. Its basic idea is as follows: if a root $\boldsymbol{x}^{\mathrm{r}}$ is found and saved in an archive $\mathcal{A}_{\mathrm{r}}$, a repulsion region will be generated around it by a predefined repulsive radius. Then, the fitness value of any individual falling into the repulsion region is modified, and the algorithm is driven to search for other promising regions.

Figure 2 illustrates the principle of the repulsion technique. Figure 2a shows the original curve of the function: $f(x) = \cos^2(x/\pi), x \in [-10, 10]$. It can be seen that there are two roots, i.e., $x_1 = -4.93$ and $x_2 = 4.93$. If these two roots are found, the fitness values nearby are modified according to the repulsion technique. As shown in Fig. 2b, the individuals close to the two roots (falling into the repulsion area) will be penalized with very large fitness values.

With RBT, a generic form for transforming NEs into an optimization problem is as follows:

$$\min \ R(\boldsymbol{x}) = F(\boldsymbol{x}) \odot P(\boldsymbol{x}) \tag{4}$$

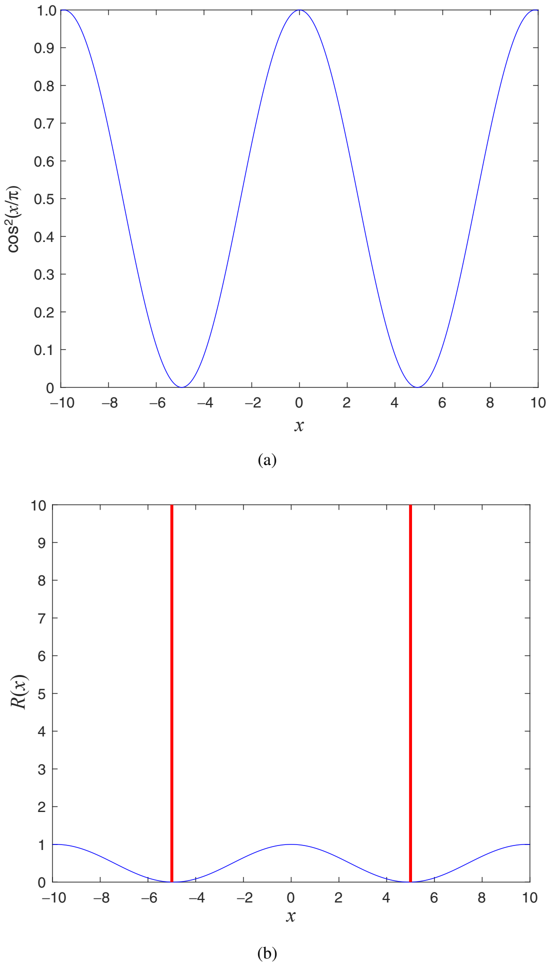

Fig. 2 Illustration of the repulsion technique. (a) Original NEs problem with two roots and (b) neighborhood around the found roots is penalized according to the repulsion technique.

where $F(\boldsymbol{x})$ is a transformed system function of NEs as shown in Eq. (2) or (3), $P(\boldsymbol{x})$ is the penalty function to penalize the solution $\boldsymbol{x}$, that is nearby the found roots, and $\odot$ is one of arithmetic operators.

Herein, some representative RBT methods in the literature are introduced.

- In Refs. [31] and [32], the following RBT method was proposed:

$$\min \ R(\boldsymbol{x}) = F(\boldsymbol{x}) \times \frac{\prod_{j=1}^{m} x_j^{|\mathcal{A}_{\mathrm{r}}|}}{\prod_{i=1}^{|\mathcal{A}_{\mathrm{r}}|} \prod_{j=1}^{m} |x_j - x_j^{\mathrm{r},i}|} \tag{5}$$

where $\boldsymbol{x}^{\mathrm{r},i}$ is the i-th root saved in the archive $\mathcal{A}_{\mathrm{r}}$, $x_j^{\mathrm{r},i}$ is its j-th variable, and $|\mathcal{A}_{\mathrm{r}}|$ is the size of the current archive $\mathcal{A}_{\mathrm{r}}$.

- Hirsch et al.[33] presented an additive repulsion technique,

$$\min \ R(\boldsymbol{x}) = F(\boldsymbol{x}) + \beta \sum_{i=1}^{|\mathcal{A}_{\mathrm{r}}|} \exp(-\delta_i)\chi_{\rho}(\delta_i) \tag{6}$$

where $\delta_i = ||\boldsymbol{x} - \boldsymbol{x}^{r,i}||$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}^{r,i}$, ρ is a small positive parameter to control the radius of the repulsion regions, β is a large constant to adjust the penalty scale, and $\chi_\rho(\delta_i)$ is the characteristic function,

$$\chi_\rho(\delta_i) = \begin{cases} 1, \text{ if } \delta_i \leqslant \rho; \\ 0, \text{ otherwise.} \end{cases}$$

- In Ref. [34], an NEs problem is formulated as

$$\min\ R(\boldsymbol{x}) = \frac{F(\boldsymbol{x})}{\prod_{i=1}^{|\mathcal{A}_r|} \arctan(\delta_i)} \tag{7}$$

- Similar to the method in Ref. [34], Pourjafari and Mojallali[35] proposed a multiplicative repulsion function as follows:

$$\min\ R(\boldsymbol{x}) = (F(\boldsymbol{x}) + \epsilon) \prod_{i=1}^{|\mathcal{A}_r|} |\coth(\rho\delta_i)| \tag{8}$$

where ϵ is a very small positive constant, and ρ is also the repulsion radius.

- Ramadas et al.[36, 37] also proposed a multiplicative repulsion technique based on the error function,

$$\min\ R(\boldsymbol{x}) = (F(\boldsymbol{x}) + \epsilon) \prod_{i=1}^{|\mathcal{A}_r|} \zeta_\rho(\gamma, \delta_i) \tag{9}$$

where γ is a parameter to measure the degree of repulsion, and $\zeta_\rho(\gamma, \delta_i)$ is an error function erf( ),

$$\zeta_\rho(\gamma, \delta_i) = \begin{cases} |\text{erf}(\gamma\delta_i)|^{-1}, \text{ if } \delta_i \leqslant \rho; \\ 1, \qquad\qquad \text{otherwise.} \end{cases}$$

Combined with RBT, the algorithm is capable of finding different roots of NEs. However, the optimal setting of the repulsion radius is problem-dependent and usually difficult.

(3) *CSOT:* In Ref. [38], a COST is presented to formulate the NEs as

$$\min\ F(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(\boldsymbol{x}) \tag{10}$$

subject to

$$f_i(\boldsymbol{x}) \geqslant 0, i = 1, \ldots, n.$$

Pourrajabian et al.[39] also introduced a COST for the NEs,

$$\min\ F(\boldsymbol{x}) = \sum_{i=1}^{n} |f_i(\boldsymbol{x})| \tag{11}$$

subject to

$$f_i(\boldsymbol{x}) = 0, i = 1, \ldots, n.$$

Solving the constrained single-objective optimization as formulated in Eq. (10) or (11), which is usually more difficult than the single-objective optimization, the constraint-handling technique is required. Again, similar to the SOT, it needs additional diversity preservation technique, so as to locate different roots of NEs when solving Eq. (10) or (11).

(4) *NLPT:* Similar to the transformation technique[40], in Ref. [41], by introducing additional parameter $\varepsilon \in \mathbf{R}^n$, an NEs problem is transformed as a nonlinear programming problem as

$$\min \quad \| \varepsilon_i - \varepsilon_i^* \|_\infty \tag{12}$$

subject to

$$\begin{cases} |f_i(\boldsymbol{x})| - \varepsilon_i \leqslant 0; \\ \varepsilon_i \geqslant 0, \end{cases}$$

where $i = 1, \ldots, n$, and $\varepsilon^* \in \mathbf{R}^n$ is the desired system precision. Therefore, solving NEs is equivalent to deal with the nonlinear system of inequities with precision $\varepsilon_i^* = 0$.

(5) *MOT:* In the Evolutionary Computation (EC) community, Evolutionary Multi-objective Optimization (EMO) is very active[42, 43]. The main purpose of EMO is to find a set of non-dominated solutions with good convergence and coverage, that is, the obtained non-dominated solutions approach the true Pareto front with good distribution. Locating multiple roots of NEs is similar to obtain the non-dominated solutions of EMO. Therefore, in the literature, several multi-objective transformation techniques are proposed to convert an NEs problem into a Multi-objective Optimization Problem (MOP).

- In 2008, Grosan and Abraham[44] proposed a transformation method, namely CA, which treats each equation as an objective function. This method is simple and easy to implement, and the specific transformation method is as follows:

$$\begin{cases} \min \quad F_1(\boldsymbol{x}) = |f_1(\boldsymbol{x})|; \\ \qquad\qquad \vdots \\ \min \quad F_n(\boldsymbol{x}) = |f_n(\boldsymbol{x})| \end{cases} \tag{13}$$

However, the CA technique has two possible drawbacks: (1) the n-objective functions in Eq. (16) cannot guarantee the conflict between two objectives; and (2) when $n > 3$, Eq. (16) is a many-objective optimization problem, which is much more difficult to be solved effectively[45]. Therefore, with the increase in the number of equations, the performance of the algorithm for solving Eq. (16) will be dramatically affected.

- To remedy the defects of CA, Song et al.[23] designed a bi-objective optimization method, namely MONES, which divides an NEs problem into the location function and system function. The transformed bi-objective optimization problem is as

$$\begin{cases} \min\ F_1(\boldsymbol{x}) = x_1 + \sum_{i=1}^{n} |f_i(\boldsymbol{x})|; \\ \min\ F_2(\boldsymbol{x}) = 1 - x_1 + n \times \max_{i=1}^{n} |f_i(\boldsymbol{x})| \end{cases} \quad (14)$$

In MONES, the first decision variable of the NEs (i.e., x_1) is used to ensure the conflict between the two objective functions. Since there are only two objective functions in MONES, it can be handled easily by most of EMO algorithms, such as NSGA-II[42]. However, if some roots have the same value in x_1 (e.g., as shown in Fig. 1, roots "*A*" and "*B*" have the same value in x_1), some of them may be lost during the run.

• In Ref. [46], a complex multi-objective transformation technique was developed, where an NEs problem is formulated as shown in Eq. (15) at the bottom of this page, where $R(\boldsymbol{x}) = \sum_{i=1}^{n} |f_i(\boldsymbol{x})|$ is the system function, and C is used to control the shape of the Pareto front. The problem formulated in Eq. (15) suffers from the "curse-of-dimensionality" with the increase of m.

• To retain the advantages of MONES and remedy its drawback in the loss of roots, Gong et al.[47] introduced the weighted bi-objective transformation technique (namely WeB) as follows:

$$\begin{cases} \min F_1(\boldsymbol{x}) = \dfrac{\sum_{i=1}^{m} w_i \times x_i}{\sum_{i=1}^{m} w_i} + \sum_{j=1}^{n} |f_j(\boldsymbol{x})|; \\ \min F_2(\boldsymbol{x}) = 1 - \dfrac{\sum_{i=1}^{m} w_i \times x_i}{\sum_{i=1}^{m} w_i} + \sum_{j=1}^{n} |f_j(\boldsymbol{x})| \end{cases} \quad (16)$$

where w_i is randomly generated weight for each variable x_i before the run and $i = 1, \ldots, m$.

• The indices of the m equations in Ref. [48, Eq. (1)] are randomly divided into two subsets S_1 and S_2. Let $\mathcal{I} = \{1, \ldots, m\}$ be the indices of NEs, $S_1 \cup S_2 = \mathcal{I}$ and $S_1 \cap S_2 = \phi$. In this way, the following constrained bi-objective transformation technique is presented:

$$\begin{cases} \min F_1(\boldsymbol{x}) = \sum_{i=1}^{|S_1|} |f_{S_1(i)}(\boldsymbol{x})|, \text{ s.t. } f_{S_1(i)}(\boldsymbol{x}) = 0; \\ \min F_2(\boldsymbol{x}) = \sum_{i=1}^{|S_2|} |f_{S_2(i)}(\boldsymbol{x})|, \text{ s.t. } f_{S_2(i)}(\boldsymbol{x}) = 0 \end{cases} \quad (17)$$

where the equality constraints are usually converted into the inequality ones.

• In Refs. [25,49], an NEs problem was transformed into a bi-objective optimization problem as

$$\begin{cases} \min\ F_1(\boldsymbol{x}_i) = \sum_{k=1}^{m} |f_k(\boldsymbol{x}_i)|; \\ \min\ F_2(\boldsymbol{x}_i) = \sum_{j=1}^{Np} K(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{cases} \quad (18)$$

where Np is the population size, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\dfrac{\| \boldsymbol{x}_i - \boldsymbol{x}_j \|}{2\delta^2}\right)$, and $\delta = 2$ is used in Refs. [25,49]. Actually, $F_2(\boldsymbol{x}_i)$ is the diversity contribution of $\boldsymbol{x}_i$ to the population. However, it is worth noting that the two objective functions in Eq. (18) cannot make sure to be conflict.

(6) *Remarks:* Table 1 summarizes the advantages and disadvantages of the above-mentioned transformation techniques. Generally,

• The SOT technique is simple, straightforward, and more popular, however, it requires additional diversity preservation technique to locate different roots of NEs.

• The RBT technique is effective to locate multiple roots of NEs. Different $P(\boldsymbol{x})$ functions have different features to penalize the solutions around the found roots. Especially, the repulsion radius should be properly set for different NEs problems.

• The CSOT and NLPT techniques are scarce due to their difficulty to be solved efficiently.

• The MOT technique obtains increasing attention recently, however, it needs to carefully design the objective functions to make sure both the conflict among different functions, it also needs to ensure the one-to-one mapping between the roots and the Pareto solutions.

After transforming an NEs problem into an optimization problem, different optimization methods can be used to deal with it. In Section 3, we review the IOAs for locating multiple roots of NEs.

3 IOAs for Multiple Roots Finding of NEs

With the purpose of locating various roots of NEs, during the last few years, different IOAs with the diversity techniques are used to solve NEs. These methods can be roughly divided into five categories: (1) repulsion-based

$$\begin{cases} \min f_1(\boldsymbol{x}) = \dfrac{x_1}{m} + \dfrac{x_2}{m-1} + \cdots + \dfrac{x_{m-1}}{2} + \dfrac{x_m}{1} + C \times R(\boldsymbol{x}) \times \ln(m+2); \\ \min f_2(\boldsymbol{x}) = \dfrac{x_1}{m} + \dfrac{x_2}{m-1} + \cdots + \dfrac{x_{m-1}}{2} + (1 - x_m) + C \times R(\boldsymbol{x}) \times \ln(m+1); \\ \min f_3(\boldsymbol{x}) = \dfrac{x_1}{m} + \dfrac{x_2}{m-1} + \cdots + \dfrac{x_{m-2}}{3} + (1 - x_{m-1}) + C \times R(\boldsymbol{x}) \times \ln(m); \\ \vdots \\ \min f_m(\boldsymbol{x}) = \dfrac{x_1}{m} + (1 - x_2) + C \times R(\boldsymbol{x}) \times \ln(3); \\ \min f_{m+1}(\boldsymbol{x}) = (1 - x_1) + C \times R(\boldsymbol{x}) \times \ln(2) \end{cases} \quad (15)$$

Table 1 Summary of the advantages and disadvantages of different transformation techniques.

Technique	Reference	Advantage	Disadvantage
SOT	[9, 27, 28] [29, 50] [51, 52] [30, 53]	• Simplicity • Ease of implementation	• Requirement of additional diversity technique to find multiple roots
RBT	[31, 32] [33, 34] [35–37]	• Simplicity • Ease of implementation • Locating multiple roots	• Requirement of proper setting of repulsion radius
CSOT	[38, 39]	• Simplicity	• Requirement of additional diversity technique to find multiple roots • Requirement of constraint-handling technique
NLPT	[41]	• Simplicity	• Requirement of additional diversity technique to find multiple roots • Requirement of NLP-handling technique
MOT	[44]	• Simplicity • Ease of implementation • Locating multiple roots	• No guarantee on the conflict between two objective functions • "Curse-of-dimensionality" at $n > 3$
	[23]	• Simplicity • Ease of implementation • Bi-objective transformation • Conflict between two objective functions • Locating multiple roots	• Loss of roots that have the same value in x_1 • Performance degeneration at large search range
	[46]	• One-to-one mapping between the roots and Pareto solutions • Conflict between two objective functions • Locating multiple roots	• Complexity • "Curse-of-dimensionality" at $m > 2$
	[47]	• Approximate one-to-one mapping between the roots and Pareto solutions • Advantages in Ref. [23]	• Performance degeneration at large search range
	[48]	• Bi-objective transformation • Locating multiple roots	• No guarantee on the conflict between two objective functions • Requirement of constraint-handling technique
	[25, 49]	• Simplicity • Ease of implementation • Bi-objective transformation • Locating multiple roots	• No guarantee on the conflict between two objective functions

methods, (2) niching-based methods, (3) clustering-based methods, (4) multi-objective optimization based methods, and (5) other methods. Generally, the methods in (1) – (3) and (5) solve the transformed single-objective optimization problem with addition diversity techniques, while the methods in (4) is used to solve the transformed MOP.

3.1 Repulsion-based methods

As mentioned in Section 2.2, the repulsion techniques are capable of driving the algorithm to escape from the regions of the found roots. Therefore, it can promote the algorithm to locate other roots of NEs. In the literature, several IOAs combined with different repulsion techniques are presented for multiple roots finding.

In Ref. [33], a Continuous Greedy Randomized Adaptive Search Procedure (C-GRASP) with repulsion was presented, where a construction phase and a local search phase are implemented. The multi-start procedure was used to find multiple roots. C-GRASP performed well on four NEs problems.

Henderson et al.[34] used a simulated annealing with the polarization technique (i.e., repulsion technique) to determine multiple roots, where the polarization technique can create repulsive regions around the found roots and guide the algorithm to explore other promising areas.The proposed method was evaluated on 5 benchmark problems and the double retrograde vaporization.

In Ref. [35], a two-phase root-finder with the repulsion technique was introduced for multiple roots location. In the global phase, the Invasive Weed Optimization (IWO) is used to detect the approximate locations of roots. In the exact phase, a clustering technique is first used to seperate the locations of roots, then IWO is also used to each cluster to locate the roots. The method is shown to perform well on the tested problems.

In Ref. [54], a Biased Random-Key GA (BRKGA) with the repulsion technique[33] was proposed to find all roots of NEs. The promising results were obtained by applying BRKGA to 7 NEs problems.

Ramadas et al.[36] used an Improved Harmony Search (I-HS)[55] with the repulsion technique shown in Eq. (9) to obtain multiple roots of NEs, where the I-HS is embedded into a repulsion algorithm. Therefore, the population re-initialization is used in I-HS to enrich the diversity.

Gong et al.[24] presented a novel DE, namely RADE, that combined the repulsion, niching, and parameter adaptation to locate multiple roots of NEs. RADE was evaluated on 30 NEs problems and promising results were obtained.

The repulsive radius plays an important role in the repulsion technique. However, previously-mentioned repulsion-based methods used fixed value of the repulsive radius, which is highly dependent on test problems and is not conducive to be extended to general cases. To alleviate this drawback, Liao et al.[56] proposed a generic framework with Evolutionary Algorithm (EA) and dynamic repulsion radius, i.e., DREA. In the process of iteration, the repulsive radius is adjusted dynamically with the number of iterations. In DREA, different repulsion techniques and different EAs can be integrated into this framework. Finally, the population re-initialization is used to maintain the population diversity, which greatly improves the probability of locating the roots.

3.2 Niching-based methods

Niching techniques modify the search behavior of classical EAs to gather different sub-populations within a single population to find multiple solutions[22]. They are able to maintain the population diversity during the search. Various niching techniques are used to solve Multi-Modal Optimization Problems (MMOPs). Due to the similarity between the solving of MMOPs and multiple roots finding of NEs, combining the niching techniques with IOAs is a promising way to locate different roots of NEs.

In Ref. [57], a niching PSO was presented, where a neighborhood best (nbest) technique is proposed to update the particles' velocity. The solutions in the same neighborhood, measured by the Euclidean distance, form a niche to make the algorithm find multiple roots. The proposed nbest PSO yielded good results on some simple NEs with a small number of roots.

Zhou and Jiang[58] proposed a Hybrid Niching GA (HNGA). In HNGA, the deterministic crowding method created a niche evolutionary environment to maintain the population diversity and to make the algorithm converge to multiple roots simultaneously. The Quasi-Newton method was used as the local search for an accurate search that further improves convergence and accuracy.

In Ref. [24], the niching technique was applied in RADE to maintain the population diversity. In this way, it can locate various roots of NEs.

Liao et al.[50] proposed memetic niching based EA, referred to as MENI-EA, to find multiple roots of NEs. In Ref. [50], two diversity mechanisms of the neighborhood mutation and crowding technique were added to EA to enhance the population diversity. To refine the fitness of individuals that satisfied corresponding conditions, the numerical method for NEs was incorporated into EA to obtain high-quality roots. In MENI-EA, numerical method together with diversity-preserving mechanism were integrated into EA to yield promising performance.

He et al.[51] presented a fuzzy neighborhood-based DE with orientation, namely FNODE. In Ref. [51], FNODE proposed the fuzzy neighborhood technique that introduced the probabilistic selection operation to form the final neighborhood. Moreover, the orientation-based mutation was designed, where the orientation information of the neighborhood individual migration is combined into the mutation operation to generate the offspring. The improved fuzzy neighborhood technology and the orientation-based mutation can balance between exploration and exploitation in FNODE, which improve the ability to solve NEs.

In Ref. [59], a hybrid swarm intelligence with improved ring topology for NEs was presented, where an enhanced ring topology was designed to make use of the knowledge of the neighborhood. Moreover, artificial bee colony was hybridized with DE and the crowding technique to improve the search efficiency. Finally, the individual re-initialization mechanism was used to enrich the population diversity. Experimental results indicated that the proposed method obtained the competitive results compared with oter related methods.

3.3 Clustering-based methods

Similar to the niching techniques, the clustering techniques can partition a population into different groups and preserve the population diversity. Thus, several researchers combined the clustering techniques with IOAs to find multiple roots of NEs.

Tsoulos and Stavrakoudis[52] proposed a global search algorithm based on clustering to identify multiple roots of NEs. The global search algorithms, multistart and minfinder, were used to solve the transformation problem, where the minfinder is a clustering algorithm, whose main purpose is to locate all the optimal solutions of the function in the decision space.

Sacco and Henderson[53] combined Fuzzy Clustering Means (FCM) to design an optimization algorithm to solve NEs. In the first stage, the Luus-Jaakola method is used to search the whole decision space to obtain a large number of candidate solutions. In the second stage, to locate multiple roots, the candidate solutions are clustered by FCM. In the last stage, the Nelder-Mead (N-M) local search algorithm is employed to search in each cluster to accelerate the population convergence.

Liao et al.[60] designed a Decomposition-based DE with Re-initialization (DDE/R), where a parameter-free decomposition based on the clustering technique is used to divide the population into different sub-populations to locate multiple roots. Then, a sub-population control strategy is employed to enhance the search ability. In addition, the subpopulation re-initialization mechanism is used to improve the population diversity.

Wu et al.[30] presented a clustering-based DE with different crowding factors for NEs. In Ref. [30], a one-step K-means clustering method was combined with the niching technique to guide the population towards multiple roots.

3.4 Multi-objective optimization based methods

Multi-objective technique is one of effective algorithms to deal with MOPs. The main goal is to find a set of Pareto optimal solutions, which is analogue to find different roots of NEs. In recent years, there are several achievements by using a multi-objective technique to solve NEs.

Grosan and Abraham[44] made the first attempt to solve NEs with the multi-objective-based method, where each equation was treated as an objective function, and transformed the NEs into an MOP, as shown in Eq. (16). Then, NSGA-II[42] was applied to solve the transformation problem. The experiments showed that the proposed method achieved satisfactory results compared with other methods.

In Ref. [23], the MONES algorithm was developed to locate multiple roots of NEs, where the NEs problem was transformed into a bi-objective optimization problem, as show in Eq. (14), and NSGA-II was used to solve the transformation problem. The algorithm was shown to perform well on 7 problems.

Qin et al.[46] transformed NEs into an n-objective optimization problem, as show in Eq. (15). Then, the HypE algorithm[61] was employed to solve the transformed problem. The proposed method was also tested on 7 problems presented in Ref. [23] and showed acceptable performance.

To overcome the loss of roots of MONES, in Ref. [47], Gong et al. presented a weighted bi-objective transformation technique (namely A-WeB) to enhance the performance of MONES in solving NEs. In A-WeB, the weights in objective function are randomly generated from 0 to 1. In the optimization process, SHADE[62] and NSGA-II[42] are combined to generate offspring through mutation and crossover operators. Moreover, the parameters are adjusted adaptively, which improves the search efficiency. A-WeB was extensively evaluated on 38 NEs problems and yielded highly competitive performance compared with other methods.

Naidu and Ojha[48] presented a Hybrid Cooperative Multi-objective Optimization IWO (HCMOIWO) for NEs with the transformed problem, shown in Eq. (17). In HCMOIWO, the population is divided into two sub-populations of equal size, each subpopulation corresponds to an objective function, and each subpopulation is searched based on IWO and space transformation search. Then, the next generation individuals are selected by non-dominated sorting from the combined sub-populations. The non-inferior individuals are saved in the pre-given archive. Finally, the individuals in the archive will be obtained.

Gao et al.[25] designed a Two-Phase EA (TPEA) to solve NEs. In TPEA, NEs problem was firstly transformed into the problem, shown in Eq. (18). In the first stage, the niching technique and the diversity index based on the Gaussian kernel function were used together to maintain the population diversity. Subsequently, NCDE[63] and NSGA-II[42] were carried out alternately to produce high-quality candidate solutions. In the second stage, TPEA devised an effective method to identify the promising region (the region where the optimal solution may exist) and finally find the roots of NEs through DE as a local search algorithm.

Gao et al.[49] combined a diversity indicator, multi-objective optimization technique, and clustering technique (namely MOPEA) to solve NEs. MOPEA firstly designed a diversity indicator to preserve population diversity. Then, the K-means clustering-based selection strategy partitioned population into different subregions and identified the promising solutions. Finally, the local search refined the solutions to obtain the high-quality roots.

3.5 Other algorithms

In addition to the above types of IOAs for multiple roots finding of NEs, researchers also proposed other different methods to solve NEs. Some representatives are reviewed as follows:

• *GA*: In Ref. [27], GA was used to obtain the efficient initial guesses, then, these solutions were refined by the Newton method. In Ref. [38], a Vasconcelos GA (VGA) was presented to optimize the transformed problem, as shown in Eq. (10). Rovira et al.[64] presented a methodology to sort out equations of NEs, and solved it by means of genetic-based machine learning and GA. In Ref. [65], GA was integrated into Gauss-Legendre numerical integration to deal with NEs. Wang[66] utilized Immune Genetic Algorithm (IGA) to identify the roots. IGA adopts the individual distance comparison method based on fitness value, which not only preserves excellent individuals, but also reduces the selection of similar individuals. In Ref. [39], GA with the augmented Lagrangian function was used to solve the transformed problem, as shown in Eq. (11). In Ref. [67], GA with symmetric and harmonious individuals was introduced for solving NEs, where the Newton's method was used to accelerate the convergence. Joshi and Krishna[68] adopted GA to solve NEs derived from different applications.

• *PSO*: Mo et al.[69] put forward Conjugate Direction Particle Swarm Optimization (CDPSO) to locate the roots of NEs. In CDPSO, conjugate direction method was introduced into PSO to improve the defect, so that PSO was easy to fall into local optima in solving high-dimensional problems. In Ref. [20], an improved PSO method was presented to avoid trapping into the local optima, where new velocity and position update strategies were proposed. In Ref. [70], Voglis et al. presented a PSO with Deliberate Loss of Information (PSO-DLI), which allows the undisrupted move of particles. To solve NEs, the transformation technique shown in Eq. (3) was used. To enhance the effectiveness and robustness, in Ref. [18], a chaotic quantum behaved PSO was proposed, where different chaotic maps were used.

• *DE*: Ramadas and Fernandes[71] presented a DE with weighted combined mutation to solve NEs with Eq. (3) transformation. In Ref. [72], a hybrid DE with IWO (DEIWO) was proposed. In DEIWO, IWO is mainly used to the exploration, while DE is used to the exploitation. In Ref. [73], DE was hybridized with monarch butterfly optimization for solving NEs with the merit function of Eq. (2).

• Firefly Algorithm (FA): Wang and Zhou[74] combined FA with pattern search strategy to solve NEs. Numerical results showed that the proposed method had global convergence reliability and had advantages in solving higher dimension problems. Ariyaratne et al.[75] proposed a modified firefly algorithm to locate multiple roots of NEs. Ariyaratne et al.[76] presented an enhanced firefly algorithm for NEs. The proposed method adopted a self-tuning framework to modify the parameter and identified multiple roots in a single run.

In addition, other IOAs, such as glowworm swarm optimization[77], SA[78], bat algorithm[79], social emotion optimization[80], imperialist competition algorithm[81], cuckoo search algorithm[82], and continuous variable neighborhood search[83], were also used to solve NEs.

4 Performance Comparison of IOAs for NEs

This section presents the comparison of some representative IOAs for solving NEs, where 30 benchmark problems with two indicators are used to compare the performance of different IOAs.

4.1 Test problems

In recent years, 30 NEs with different characteristics[24] are used to verify the performance of different IOAs for locating multiple roots of NEs. The basic features of these test functions are briefly given in Table 2, where m is the number of decision variables; *Range* is the feasible region of the decision vector; *LE* represents the number of the linear equations; *NE* is the number of the nonlinear equations; *NoR* is the number of the roots; and max_{NFE} is the maximal number of function evaluations. More details of these problems can be found in Ref. [24].

In Table 2, the test functions F01, F05, and F19 have

Table 2 Brief information of the test problems.

Problem	*m*	*Range*	*LE*	*NE*	*NoR*	max_{NFE}
F01	20	$[-1, 1]^n$	0	2	2	50 000
F02	2	$[-1, 1]^n$	1	1	11	50 000
F03	2	$[-1, 1]^n$	0	2	15	50 000
F04	2	$[-10, 10]^n$	0	0	13	50 000
F05	10	$[-2, 2]^n$	0	10	1	50 000
F06	2	$[-1, 1]^n$	1	1	8	50 000
F07	2	$[-1, 1], [-10, 0]$	0	2	2	50 000
F08	2	$[0, 1]^n$	0	2	7	50 000
F09	5	$[-10, 10]^n$	4	1	3	100 000
F10	3	$[-5, 5], [-1, 3], [-5, 5]$	0	3	2	50 000
F11	2	$[-1, 1], [-10, 10]$	0	2	4	50 000
F12	2	$[-1, 2]^n$	0	2	10	50 000
F13	3	$[-0.6, 0.6], [-0.6, 0.6], [-5, 5]$	0	3	12	50 000
F14	2	$[-5, 5]^n$	0	2	9	50 000
F15	2	$[0.25, 1], [1.5, 2\pi]$	0	2	2	50 000
F16	2	$[0, 2\pi]^n$	0	2	13	50 000
F17	8	$[-1, 1]^n$	1	7	16	100 000
F18	2	$[-2, 2]^n$	0	2	6	50 000
F19	20	$[-2, 2]^n$	19	1	2	200 000
F20	3	$[-1, 1]^n$	0	3	7	50 000
F21	2	$[-2, 2]^n$	0	2	4	50 000
F22	2	$[-2, 2]^n$	0	2	6	50 000
F23	3	$[-20, 20]^n$	0	3	16	500 000
F24	3	$[0, 1]^n$	0	3	8	100 000
F25	3	$[-3, 3]^n$	0	3	2	50 000
F26	2	$[-1, -0.1], [-2, 2]$	0	2	2	50 000
F27	2	$[-5, 1.5], [0, 5]$	0	2	3	50 000
F28	2	$[0, 2], [10, 30]$	0	2	2	50 000
F29	3	$[0, 2], [-10, 10], [-1, 1]$	0	3	5	50 000
F30	2	$[-2, 2], [0, 1.1]$	0	2	4	50 000

higher objective function dimensions, which mainly investigate the exploitation ability of the algorithm. Attention should be paid to the population convergence when designing the algorithm; the test functions F02, F03, F04, F12, F13, F16, F17, and F23 have many roots, which mainly require the exploration ability of the algorithm, and pay attention to the population diversity when designing the algorithm. Other functions mainly test the comprehensive ability of the algorithm.

4.2 Performance metrics

Generally, two performance metrics in Refs. [24, 56] are applied to assess the performance of different algorithms.

- Root Ratio (RR): It calculates the average ratio of the found roots over multiple runs,

$$RR = \frac{\sum_{i=1}^{N_r} N_{i,f}}{NoR \times N_r} \tag{19}$$

where N_r is the number of runs; $N_{i,f}$ is the number of the roots obtained in the i-th run; NoR is the number of the known roots of an NE. Each algorithm conducts over 30 independent runs for fair comparison.

- Success Rate (SR): SR computes the percentage of all roots successfully found in multiple run,

$$SR = \frac{N_{r,s}}{N_r} \tag{20}$$

where $N_{r,s}$ is the number of successful runs.

4.3 Comparison of different IOAs

Tables 3 and 4 give the experimental results (RR and SR) of 8 representative IOAs on 30 NEs. These IOAs are DR-JADE[56], RADE[24], KSDE[30], FONDE[51], DDE/R[60], MONES[23], A-WeB[47], and TPEA[49].

From Tables 3 and 4, we can have a understanding on the performance of different IOAs for NEs, which is helpful to grasp the performance of IOAs in this field. Thus, we can improve the performance of existing algorithms and propose more effective new algorithms

Table 3 Comparison of root ratio with respect to different intelligent optimization algorithms.

Problem	DR-JADE	RADE	KSDE	FONDE	DDE/R	MONES	A-WeB	TPEA
F01	1.00	1.00	1.00	1.00	0.93	0.94	1.00	0.92
F02	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F03	1.00	1.00	1.00	1.00	1.00	0.96	0.62	1.00
F04	1.00	0.99	0.98	0.96	1.00	0.93	1.00	1.00
F05	0.81	0.99	0.99	1.00	1.00	1.00	0.95	1.00
F06	0.72	0.90	0.98	0.99	1.00	0.13	1.00	1.00
F07	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
F08	1.00	0.99	0.90	1.00	1.00	0.43	0.94	1.00
F09	1.00	0.99	0.99	1.00	0.97	0.88	0.83	1.00
F10	0.99	0.97	1.00	1.00	1.00	0.96	0.89	1.00
F11	1.00	0.63	0.91	1.00	1.00	0.00	0.88	1.00
F12	0.77	0.98	0.99	0.86	1.00	0.43	0.97	0.94
F13	1.00	0.99	1.00	0.73	0.72	0.54	1.00	0.91
F14	0.99	0.94	0.99	1.00	1.00	0.99	0.66	1.00
F15	0.84	1.00	1.00	1.00	1.00	0.98	0.94	1.00
F16	1.00	0.79	0.93	1.00	1.00	0.43	0.62	1.00
F17	0.16	1.00	1.00	0.98	1.00	0.16	0.94	0.97
F18	1.00	1.00	1.00	1.00	1.00	0.50	0.62	1.00
F19	1.00	1.00	1.00	0.93	0.90	0.30	0.95	0.72
F20	1.00	0.99	1.00	1.00	1.00	0.54	0.99	1.00
F21	0.60	1.00	1.00	1.00	1.00	0.53	1.00	1.00
F22	1.00	0.89	0.92	1.00	1.00	0.53	0.09	1.00
F23	1.00	1.00	1.00	0.91	0.94	0.14	1.00	0.97
F24	1.00	1.00	1.00	1.00	1.00	0.64	1.00	1.00
F25	1.00	0.99	1.00	1.00	1.00	0.31	1.00	1.00
F26	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00
F27	1.00	0.99	1.00	1.00	1.00	1.00	0.93	1.00
F28	0.98	1.00	1.00	1.00	1.00	0.87	1.00	1.00
F29	0.87	0.56	0.50	0.99	1.00	0.82	0.01	1.00
F30	1.00	0.83	1.00	1.00	1.00	1.00	0.03	1.00
Avg.	0.92	0.95	0.97	0.98	0.98	0.66	0.83	0.98

in the future study.

DR-JADE and RADE are repulsion-based methods, and their RR and SR are $\{0.92, 0.76\}$ and $\{0.95, 0.82\}$, respectively. FONDE is a niching-based method, and its RR and SR is $\{0.98, 0.89\}$. KSDE and DDE/R are clustering-based methods, and their RR and SR are $\{0.97, 0.88\}$ and $\{0.99, 0.93\}$, respectively. MONES, A-WeB, and TPEA are multi-objective based methods, and their RR and SR are $\{0.66, 0.42\}$, $\{0.83, 0.66\}$, and $\{0.98, 0.93\}$, respectively. Apparently, DDE/R and TPEA obtain the best RR and SR. DR-JADE, RADE, KSDE, and FONDE show good optimization performance, because they can establish a good balance of exploration and exploitation. MONES and A-WeB obtain the worst results. To this end, the experimental results demonstrate that the algorithm can obtain better results only by keeping the good balance of diversity and convergence.

Additionally, Table 5 summarizes the advantages and disadvantages of eight IOAs. From Table 5, we can understand the advantages and disadvantages of different types of methods in locating multiple roots of NEs. To effectively and efficiently locate as many roots as possible, it is vital to balance the exploration and the exploitation for solving NEs[84].

5 Discussion and Open Issue

According to the above survey of IOAs for multiple roots finding of NEs and the empirical comparison among several IOAs, this section provides the discussions and points out some possible open issues.

5.1 On diversity preservation

With the purpose of finding as many roots as possible of NEs by means of IOAs, one of essential issues is to preserve the population diversity. The algorithm needs

Table 4 Comparison of success rate with respect to different intelligent optimization algorithms.

Problem	DR-JADE	RADE	KSDE	FONDE	DDE/R	MONES	A-WeB	TPEA
F01	1.00	1.00	1.00	1.00	0.86	0.93	1.00	0.82
F02	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F03	1.00	1.00	1.00	1.00	1.00	0.58	0.36	1.00
F04	1.00	0.90	0.80	0.52	1.00	0.67	1.00	1.00
F05	0.00	0.95	0.96	1.00	1.00	1.00	0.58	1.00
F06	0.00	0.31	0.86	0.98	1.00	0.00	1.00	1.00
F07	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00
F08	1.00	0.93	0.40	1.00	1.00	0.00	0.60	1.00
F09	1.00	0.98	0.93	1.00	0.93	0.66	0.12	1.00
F10	0.93	0.91	1.00	1.00	1.00	0.93	0.68	1.00
F11	1.00	0.00	0.43	1.00	1.00	0.00	0.28	1.00
F12	0.00	0.89	0.93	0.28	1.00	0.00	0.76	0.40
F13	1.00	0.94	1.00	0.02	0.00	0.00	1.00	0.36
F14	0.96	0.43	0.93	1.00	1.00	0.92	0.00	1.00
F15	0.00	1.00	1.00	1.00	1.00	0.96	0.66	1.00
F16	1.00	0.69	0.86	1.00	1.00	0.00	0.24	1.00
F17	0.00	1.00	1.00	0.76	1.00	0.00	0.70	0.82
F18	1.00	1.00	1.00	1.00	1.00	0.00	0.98	1.00
F19	1.00	1.00	1.00	0.86	0.80	0.07	1.00	0.64
F20	1.00	0.99	1.00	1.00	1.00	0.00	0.14	1.00
F21	0.00	1.00	1.00	1.00	1.00	0.01	1.00	1.00
F22	1.00	0.19	0.43	1.00	1.00	0.00	0.00	1.00
F23	1.00	1.00	1.00	0.28	0.33	0.00	1.00	0.80
F24	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00
F25	1.00	0.99	1.00	1.00	1.00	0.07	1.00	1.00
F26	1.00	1.00	1.00	1.00	1.00	1.00	0.88	1.00
F27	1.00	0.97	1.00	1.00	1.00	1.00	0.66	1.00
F28	0.93	1.00	1.00	1.00	1.00	0.78	1.00	1.00
F29	0.00	0.00	0.00	0.98	1.00	0.13	0.00	1.00
F30	1.00	0.67	1.00	1.00	1.00	1.00	0.02	1.00
Avg.	0.76	0.82	0.88	0.89	0.93	0.42	0.66	0.93

to maintain proper diversity during the whole evolution process in order to explore new search regions, and hence to locate new roots. In the EC community, there are different techniques presented to maintain the population diversity, for example:

• *Niching*: The niching techniques[22, 85, 86], which were originally designed to handle the MMOPs, were developed to divide the population into different niches (or sub-populations). Different optimal solutions are expected to lie in the regions of different niches. Meanwhile, each niche evolves independently to track a peak and the population can find multiple peaks finally. Therefore, the niching techniques can maintain the population diversity. However, different niching techniques usually introduce new parameters, which may be sensitive to be set properly[22]. Thus, when the niching techniques are combined with IOAs for locating various roots of NEs, the niching parameter settings need further study.

• *Clustering*: The clustering techniques are able to classify the data into different clusters[87]. Similar to the niching techniques, the clustering techniques can also be used to divide the population into different sub-populations, and hence to maintain the population diversity. However, the proper setting of the number of clusters is difficult. One possible way is to use the advanced clustering techniques, such as the density-based clustering[88], that can automatically detect the number of clusters.

• *Multi-objective optimization*: The main purpose of EMO[89] is to obtain a non-dominated Pareto set with good convergence and coverage. To achieve this goal, the population usually maintains a proper diversity with different techniques, such as the crowding used in NSGA-II[42], the clustering used in SPEA2[90], the decomposition used in MOEA/D[43]. When an NEs

Table 5 Summary of the advantages and disadvantages of the 8 IOAs.

Method	Algorithm	Advantage	Disadvantage
Repulsion-based	DR-JADE	• Preserving population diversity	• Lose several roots
	RADE	• Parameter adaptive	• Low search efficient
Niching-based	FNODE	• High search efficient	• Easy to trap into local optimum
	DDE/R	• For locating multiple roots of NEs, niching-based provides better results in both of *RR* and *SR*.	• Premature convergence
Clustering-based	KSDE	• Partition population into multiple sub-populations	• Parameter setting depends on different problems
		• Simple and effective	• Performance is easily affected by parameters
Multi-objective-based	MONES	• Using multi-objective transformation technique	• MONES and A-WeB need to balance the exploration and exploitation.
	A-WeB	• TPEA obtains better results in both of *RR* and *SR*	• Low robustness
	TPEA	• Maintain population diversity	

problem is formuated as an MOP, different EMO algorithms can be used to solve it. These type of methods are efficient when the NEs problems have infinite roots[23,47]. However, if the roots are sparsely and unevenly distributed in the search space, the performance of EMO-based methods may be reduced.

• *Re-initialization*: The re-initialization of some solutions and the whole population can enrich the population diversity directly. Due to the distinguished feature of NEs, when a solution is treated as a root, its objective function value of Eq. (2) or (3) approximates 0. In this way, the found roots can be saved in an *external* archive, and then these solutions in the population can be re-initialized to maintain the diversity. The promising results obtained in DR-JADE[56] and DDE/R[60] verified the effectiveness of the re-initialization techniques.

5.2 On root quality

In addition to the population diversity, the quality of roots is another issue that needs to be considered. For example, an NEs problem is transformed into an optimization problem, as shown in Eq. (2), if $F(\boldsymbol{x}) \leqslant \epsilon$, $\boldsymbol{x}$ is treated as a root. ϵ is a very small positive value, which is used to measure the quality of the obtained roots. The performance of the algorithm is influenced by the setting of ϵ[50]. Therefore, to obtain high quality roots of NEs, the exploitation of the algorithm needs to be strengthened.

5.3 Open issues

Although some achievements have been made in solving NEs by using IOAs, there are still several problems to be solved, mainly reflected in the following aspects:

(1) *Theoretical property*: The traditional numerical methods have a theoretical basis for solving NEs, but the theoretical study of IOAs has been lagged behind its application. Such theoretical study can guide for the design of effective IOAs for NEs.

(2) *High-dimensional problems*: Recently, most of the researches are focused on low-dimensional problems, and there is not enough studies on high-dimensional NEs. Therefore, the use of IOAs for high-dimensional NEs needs further consideration.

(3) *Constrained problems*: Many real-world applications are highly constrained. However, most of the existing IOAs are designed for the unconstrained NEs problems, and there is a lack of systematic research on how to deal with constraints for the constrained NEs problems. Therefore, coupled to efficient constraint-handling techniques[91], solving the constrained NEs problems with IOAs is another future direction.

(4) *Transformation techniques*: Most studies artificially convert NEs into single-objective or multi-objective optimization problems, yet without considering the characteristics of equations. Different transformation techniques that consider the features of NEs should be further studied, such as the technique in Ref. [64]. Additionally, in Ref. [92], Song et al. investigated the NEs knowledge and combined the variable reduction strategy into IOA to solve NEs, thus achieving the competitive results in terms of RR and SR.

(5) *New test problems with a large number of roots*: Recent studies on locating multiple roots of NEs focus on a small number of roots. However, if the number of roots is large, the performance of current IOAs deteriorates dramatically[60]. Therefore, developing new test NEs problems with a large number of roots is required to further investigate the performance of IOAs. Besides, how to design the evaluation indicators to evaluate the effectiveness of the algorithm is also interesting. In addition, designing enhanced IOAs to deal with the

problems with a large number of roots is an important challenge.

(6) *Hybrid algorithm*: Recently, there exist many kinds of literature hybridizing IOAs with local search to solve NEs. There are still few methods to combine IOAs with machine learning to locate multiple roots of NEs. Also, combining IOAs with the numerical methods is also a possible way to design enhanced NEs solvers.

(7) *Real-world applications*: In Ref. [7], Mehta and Grosan listed various real-world applications of NEs, such as the mixed-variable problems[93, 94]. The use of IOAs for the real-world NEs problems needs further attention.

6 Conclusion

IOAs are powerful optimization methods for global optimization. Combined with the diversity preservation techniques, IOAs can locate different roots of NEs. In this paper, we have revisited the transformation techniques and reviewed recent developments of IOAs in solving NEs. Besides, the test function and evaluation indicators of NEs are described in detail. Moreover, we select eight representative IOAs for comparison and analyze the advantages and disadvantages of these algorithms. Finally, we have discussed several open research issues. We hope that these issues will help to re-stimulate the interests and research efforts in this field.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China (No. 62076225), the Natural Science Foundation of Guangxi Province (No. 2020JJA170038), and the High-Level Talents Research Project of Beibu Gulf (No. 2020KYQD06).

References

[1] J. Pintér, Solving nonlinear equation systems via global partition and search: Some experimental results, *Computing*, vol. 43, no. 4, pp. 309–323, 1990.

[2] M. Kastner, Phase transitions and configuration space topology, *Rev. Mod. Phys.*, vol. 80, pp. 167–187, 2008.

[3] F. Facchinei and C. Kanzow, Generalized nash equilibrium problems, *Annals of Oporation Research*, vol. 5, no. 3, pp. 173–210, 2007.

[4] D. Guo, Z. Nie, and L. Yan, The application of noise-tolerant ZD design formula to robots' kinematic control via time-varying nonlinear equations solving, *IEEE Transactions on Systems, Man, and Cybernetics*: *Systems*, vol. 48, no. 12, pp. 2188–2197, 2017.

[5] H. D. Chiang and T. Wang, Novel homotopy theory for nonlinear networks and systems and its applications to electrical grids, *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1051–1060, 2017.

[6] G. Li, P. Niu, W. Zhang, and Y. Zhang, Control of discrete chaotic systems based on echo state network modeling with an adaptive noise canceler, *Knowledge-Based Systems*, vol. 35, pp. 35–40, 2012.

[7] D. Mehta and C. Grosan, A collection of challenging optimization problems in science, engineering and economics, in *Proc. of 2015 IEEE Congress on Evolutionary Computation* (*CEC*), Sendai, Japan, 2015, pp. 2697–2704.

[8] S. Smale, Mathematical problems for the next century, *Math. Intell.*, vol. 20, no. 2, pp. 7–15, 1998.

[9] C. T. Kelley, Numerical methods for nonlinear equations, *Acta Numerica*, vol. 27, pp. 207–287, 2018.

[10] H. Ramos and M. Monteiro, A new approach based on the newton's method to solve systems of nonlinear equations, *Journal of Computational and Applied Mathematics*, vol. 318, pp. 3–13, 2017.

[11] H. Liu, Y. Zhou, and Y. Li, A Quasi-Newton population migration algorithm for solving systems of nonlinear equations, *Journal of Computers*, vol. 6, no. 1, pp. 36–42, 2011.

[12] A. Bouaricha and R. B. Schnabel, Tensor methods for large sparse systems of nonlinear equations, *Mathematical Programming*, vol. 82, no. 3, pp. 377–400, 1998.

[13] Y. Gao, P. Zhao, L. S. Turng, and H. Zhou, *Intelligent Optimization Algorithms*. John Wiley & Sons, 2013, pp. 283–292.

[14] T. Bäck, *Evolutionary Algorithms in Theory and Practice*: *Evolution Strategies*, *Evolutionary Programming*, *Genetic Algorithms*. Oxford, UK: Oxford University Press, 1996.

[15] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of IEEE International Conference on Neural Networks IV*, Perth, Australia, 1995, pp. 1942–1948.

[16] R. Storn and K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[17] N. Hansen, S. D. Müller, and P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[18] O. E. Turgut, M. S. Turgut, and M. T. Coban, Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, *Computers & Mathematics with Applications*, vol. 68, no. 4, pp. 508–530, 2014.

[19] M. Abdollahi, A. Bouyer, and D. Abdollahi, Improved cuckoo optimization algorithm for solving systems of nonlinear equations, *The Journal of Supercomputing*, vol. 72, no. 3, pp. 1246–1269, 2016.

[20] M. Jaberipour, E. Khorram, and B. Karimi, Particle swarm algorithm for solving systems of nonlinear equations, *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 566–576, 2011.

[21] C. A. Floudas, Recent advances in global optimization for process synthesis, design and control: Enclosure of all

solutions, *Computers & Chemical Engineering*, vol. 23, pp. 963–973, 1999.

[22] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, Seeking multiple solutions: An updated survey on niching methods and their applications, *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 518–538, 2017.

[23] W. Song, Y. Wang, H. X. Li, and Z. Cai, Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization, *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 414–431, 2015.

[24] W. Gong, Y. Wang, Z. Cai, and L. Wang, Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution, *IEEE Transactions on Systems, Man, and Cybernetics*: *Systems*, vol. 50, no. 4, pp. 1499–1513, 2020.

[25] W. Gao, G. Li, Q. Zhang, Y. Luo, and Z. Wang, Solving nonlinear equation systems by a two-phase evolutionary algorithm, *IEEE Transactions on Systems*, *Man*, *and Cybernetics*: *Systems*, doi: 10.1109/TSMC.2019.2957324.

[26] L. Wu, H. Ren, and W. Bi, Review of the genetic algorithm for nonlinear equations, *Electronic Science and Technology*, vol. 4, pp. 173–178, 2014.

[27] C. Karr, B. Weck, and L. Freeman, Solutions to systems of nonlinear equations via a genetic algorithm, *Engineering Applications of Artificial Intelligence*, vol. 11, no. 3, pp. 369–375, 1998.

[28] M. D. Stuber, V. Kumar, and P. I. Barton, Nonsmooth exclusion test for finding all solutions of nonlinear equations, *Bit Numerical Mathematics*, vol. 50, no. 4, pp. 885–917, 2010.

[29] J. A. Koupaei and S. Hosseini, A new hybrid algorithm based on chaotic maps for solving systems of nonlinear equations, *Chaos, Solitons* & *Fractals*, vol. 81, pp. 233–245, 2015.

[30] J. Wu, W. Gong, and L. Wang, A clustering-based differential evolution with different crowding factors for nonlinear equations system, *Applied Soft Computing*, vol. 98, p. 106733, 2021.

[31] N. Henderson, L. Freitas, and G. Platt, Prediction of critical points: A new methodology using global optimization, *Aiche Journal*, vol. 50, no. 6, pp. 1300–1314, 2004.

[32] L. Freitas, G. Platt, and N. Henderson, Novel approach for the calculation of critical points in binary mixtures using global optimization, *Fluid Phase Equilibria*, vol. 225, pp. 29–37, 2004.

[33] M. J. Hirsch, P. M. Pardalos, and M. G. Resende, Solving systems of nonlinear equations with continuous GRASP, *Nonlinear Analysis*: *Real World Applications*, vol. 10, no. 4, pp. 2000–2006, 2009.

[34] N. Henderson, W. F. Sacco, and G. M. Platt, Finding more than one root of nonlinear equations via a polarization technique: An application to double retrograde vaporization, *Chemical Engineering Research and Design*, vol. 88, nos. 5&6, pp. 551–561, 2010.

[35] E. Pourjafari and H. Mojallali, Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering, *Swarm and Evolutionary Computation*, vol. 4, pp. 33–43, 2012.

[36] G. C. Ramadas, E. M. Fernandes, and A. A. Rocha, Multiple roots of systems of equations by repulsion merit functions, in *Proc. of Computational Science and Its Applications–ICCSA 2014*, Banff, Canada, 2014, pp. 126–139.

[37] G. C. V. Ramadas, A. M. A. C. Rocha, and E. M. G. P. Fernandes, Testing nelder-mead based repulsion algorithms for multiple roots of nonlinear systems via a two-level factorial design of experiments, *PLos One*, vol. 10, no. 4, p. e0121844, 2015.

[38] A. F. Kuri-Morales, Solution of simultaneous non-linear equations using genetic algorithms, *WSEAS Transactions on Systems*, vol. 2, no. 1, pp. 44–51, 2003.

[39] A. Pourrajabian, R. Ebrahimi, M. Mirzaei, and M. Shams, Applying genetic algorithms for solving nonlinear algebraic equations, *Applied Mathematics and Computation*, vol. 219, no. 24, pp. 11483–11494, 2013.

[40] C. Maranas and C. Floudas, Finding all solutions of nonlinearly constrained systems of equations, *Journal of Global Optimization*, vol. 7, no. 2, pp. 143–182, 1995.

[41] A. A. Mousa and I. M. El-Desoky, GENLS: Co-evolutionary algorithm for nonlinear system of equations, *Applied Mathematics and Computation*, vol. 197, no. 2, pp. 633–642, 2008.

[42] 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.

[43] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. on Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[44] C. Grosan and A. Abraham, A new approach for solving nonlinear equations systems, *IEEE Transactions on Systems, Man and Cybernetics, Part A*: *Systems and Humans*, vol. 38, no. 3, pp. 698–714, 2008.

[45] B. Li, J. Li, T. Ke, and Y. Xin, Many-objective evolutionary algorithms: A survey, *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[46] S. Qin, S. Zeng, W. Dong, and X. Li, Nonlinear equation systems solved by many-objective hype, in *Proc. of 2015 IEEE Congress on Evolutionary Computation*, Sendai, Japan, 2015, pp. 2691–2696.

[47] W. Gong, Y. Wang, Z. Cai, and S. Yang, A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems, *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 697–713, 2017.

[48] Y. R. Naidu and A. K. Ojha, Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations, *IEEE Transactions on Systems*, *Man*, *and Cybernetics*: *Systems*, vol. 48, no. 6, pp. 821–832, 2018.

[49] W. Gao, Y. Luo, J. Xu, and S. Zhu, Evolutionary algorithm with multiobjective optimization technique for solving nonlinear equation systems, *Information Sciences*, vol. 541, pp. 345–361, 2020.

[50] Z. Liao, W. Gong, and L. Wang, Memetic niching-based

evolutionary algorithms for solving nonlinear equation system, *Expert Systems with Applications*, vol. 149, p. 113261, 2020.

[51] W. He, W. Gong, L. Wang, X. Yan, and C. Hu, Fuzzy neighborhood-based differential evolution with orientation for nonlinear equation systems, *Knowledge-Based Systems*, vol. 182, p. 104796, 2019.

[52] I. Tsoulos and A. Stavrakoudis, On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods, *Nonlinear Analysis*: *Real World Applications*, vol. 11, no. 4, pp. 2465–2471, 2010.

[53] W. Sacco and N. Henderson, Finding all solutions of nonlinear systems using a hybrid metaheuristic with fuzzy clustering means, *Applied Soft Computing*, vol. 11, no. 8, pp. 5424–5432, 2011.

[54] R. M. A. Silva, M. G. C. Resende, and P. M. Pardalos, Finding multiple roots of a box-constrained system of nonlinear equations with a biased random-key genetic algorithm, *Journal of Global Optimization*, vol. 60, no. 2, pp. 289–306, 2014.

[55] M. Mahdavi, M. Fesanghary, and E. Damangir, An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[56] Z. Liao, W. Gong, X. Yan, L. Wang, and C. Hu, Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms, *IEEE Transactions on Systems*, *Man*, *and Cybernetics*: *Systems*, vol. 50, no. 4, pp. 1590–1601, 2020.

[57] R. Brits, A. P. Engelbrecht, and F. van den Bergh, Solving systems of unconstrained equations using particle swarm optimization, in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, Tunisia, 2002, pp. 1–6.

[58] L. Zhou and C. Jiang, New method for solving nolinear equation systems, *Journal of Chinese Computer Systems*, vol. 9, no. 4, pp. 1709–1713, 2008.

[59] Z. Liao, W. Gong, and L. Wang, A hybrid swarm intelligence with improved ring topology for nonlinear equations, *Scientia Sinica Informationis*, vol. 50, no. 3, pp. 396–407, 2020.

[60] Z. Liao, W. Gong, L. Wang, X. Yan, and C. Hu, A decomposition-based differential evolution with reinitialization for nonlinear equations systems, *Knowledge-Based Systems*, vol. 191, p. 105312, 2020.

[61] 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.

[62] R. Tanabe and A. Fukunaga, Success-history based parameter adaptation for differential evolution, in *Proc. 2013 IEEE Congress on Evolutionary Computation* (*CEC*), Cancun, Mexico, 2013, pp. 71–78.

[63] B. Qu, P. Suganthan, and J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, *IEEE Trans. on Evol. Comput.*, vol. 16, no. 5, pp. 601–614, 2012.

[64] A. Rovira, M. Valdés, and J. Casanova, A new methodology to solve non-linear equation systems using genetic algorithms. Application to combined cyclegas turbine simulation, *International Journal for Numerical Methods in Engineering*, vol. 63, no. 10, pp. 1424–1435, 2005.

[65] P. S. Mhetre, Genetic algorithm for linear and nonlinear equation, *International Journal of Advanced Engineering Technology*, vol. 3, no. 2, pp. 114–118, 2012.

[66] J. Wang, Immune genetic algorithm for solving nonlinear equations, in *Proc. of 2011 International Conference on Mechatronic Science, Electric Engineering and Computer* (*MEC*), Jilin, China, 2011, pp. 2094–2097.

[67] H. Ren, L. Wu, W. Bi, and I. K. Argyros, Solving nonlinear equations system via an efficient genetic algorithm with symmetric and harmonious individuals, *Applied Mathematics and Computation*, vol. 219, no. 23, pp. 10967–10973, 2013.

[68] G. Joshi and M. B. Krishna, Solving system of non-linear equations using genetic algorithm, in *Proc. of 2014 International Conference on Advances in Computing*, *Communications and Informatics* (*ICACCI*), Delhi, India, 2014, pp. 1302–1308.

[69] Y. Mo, H. Liu, and Q. Wang, Conjugate direction particle swarm optimization solving systems of nonlinear equations, *Computers & Mathematics with Applications*, vol. 57, nos. 11&12, pp. 1877–1882, 2009.

[70] C. Voglis, K. Parsopoulos, and I. Lagaris, Particle swarm optimization with deliberate loss of information, *Soft Computing*, vol. 16, no. 8, pp. 1373–1392, 2012.

[71] G. C. V. Ramadas and E. M. G. P. Fernandes, Combined mutation differential evolution to solve systems of nonlinear equations, *AIP Conference Proceedings*, vol. 1558, no. 1, pp. 582–585, 2013.

[72] Y. Zhou, Q. Luo, and H. Chen, A novel differential evolution invasive weed optimization algorithm for solving nonlinear equations systems, *Journal of Applied Mathematics*, vol. 2013, p. 757391, 2013.

[73] A. M. Ibrahim and M. A. Tawhid, A hybridization of differential evolution and monarch butterfly optimization for solving systems of nonlinear equations, *Journal of Computational Design and Engineering*, vol. 6, no. 3, pp. 354–367, 2019.

[74] X. Wang and N. Zhou, Pattern search firefly algorithm for solving systems of nonlinear equations, in *Proceedings of 2014 7th International Symposium on Computational Intelligence and Design*, *ISCID 2014*, Hangzhou, China, 2014, pp. 228–231.

[75] M. Ariyaratne, T. Fernando, and S. Weerakoon, Solving systems of nonlinear equations using a modified firefly algorithm (MODFA), *Swarm and Evolutionary Computation*, vol. 48, pp. 72–92, 2019.

[76] A. Ariyaratne, T. Fernando, and S. Weerakoon, A self-tuning algorithm to approximate roots of systems of nonlinear equations based on the firefly algorithm, *International Journal of Swarm Intelligence*, vol. 5, no. 1, p. 60, 2020.

[77] Y. Zhou, J. Liu, and G. Zhao, Leader glowworm swarm

optimization algorithm for solving nonlinear equations systems, *Electrical Review*, vol. 88, no. 1, pp. 101–106, 2012.

[78] H. E. Oliveira and A. Petraglia, Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing, *Applied Soft Computing*, vol. 13, no. 11, pp. 4349–4357, 2013.

[79] J. Xie, Y. Zhou, and H. Chen, A novel bat algorithm based on differential operator and lévy flights trajectory, *Computational Intelligence and Neuroscience*, vol. 2013, p. 453812, 2013.

[80] J. Wu, Z. Cui, and J. Liu, Using hybrid social emotional optimization algorithm with metropolis rule to solve nonlinear equations, in *Proc. of IEEE International Conference on Cognitive Informatics & Cognitive Computing*, Banff, Canada, 2011, pp. 405–411.

[81] M. Abdollahi, A. Isazadeh, and D. Abdollahi, Imperialist competitive algorithm for solving systems of nonlinear equations, *Computers & Mathematics with Applications*, vol. 65, no. 12, pp. 1894–1908, 2013.

[82] X. Zhang, Q. Wan, and Y. Fan, Applying modified cuckoo search algorithm for solving systems of nonlinear equations, *Neural Computing and Applications*, vol. 31, pp. 553–576, 2019.

[83] J. Pei, Z. Dražić, M. Dražić, N. Mladenović, and P. M. Pardalos, Continuous variable neighborhood search (C-VNS) for solving systems of nonlinear equations, *INFORMS Journal on Computing*, vol. 31, no. 2, pp. 235–250, 2019.

[84] M. Črepinšek, S. H. Liu, and M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, *ACM Comput. Surv.*, vol. 45, no. 3, pp. 35:1–35:33, 2013.

[85] E. L. Yu and P. N. Suganthan, Ensemble of niching algorithms, *Information Sciences*, vol. 180, no. 5, pp. 2815–2833, 2010.

[86] X. Li, Niching without niching parameters: Particle swarm optimization using a ring topology, *IEEE Trans. on Evol. Comput.*, vol. 14, no. 1, pp. 150–169, 2010.

[87] A. K. Jain, M. N. Murty, and P. J. Flynn, Data clustering: A review, *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[88] A. Rodriguez and A. Laio, Clustering by fast search and find of density peaks, *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[89] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[90] E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, in *Proc. of Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems* (*EUROGEN 2001*), Athens, Greece, 2002, pp. 95–100.

[91] K. Deb, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, vol. 186, nos. 2&4, pp. 311–338, 2000.

[92] A. Song, G. Wu, and W. Pedrycz, Integrating variable reduction strategy with evolutionary algorithm for solving nonlinear equations systems, arXiv preprint arXiv: 2008.04223, 2020.

[93] F. Wang, Y. Li, A. Zhou, and K. Tang, An estimation of distribution algorithm for mixed-variable Newsvendor problems, *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 479–493, 2020.

[94] F. Wang, H. Zhang, and A. Zhou, A particle swarm optimization algorithm for mixed-variable optimization problems, *Swarm and Evolutionary Computation*, vol. 60, p. 100808, 2021.

Zuowen Liao received the BEng and MEng degrees from Wuhan Institute of Chemical Technology in 2006 and 2010, respectively. He received the PhD degree from China University of Geosciences, Wuhan, China in 2019. He is currently an associate professor at Beibu Gulf Ocean Development Research Center, Beibu Gulf University, Qinzhou, China. His current research interests include evolutionary computation, memetic computation, and computational intelligence.

Wenyin Gong received the BEng, MEng, and PhD degrees in computer science from China University of Geosciences, Wuhan, China in 2004, 2007, and 2010, respectively.

He is currently a professor at School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include evolutionary algorithms, evolutionary optimization, and their applications. He has published over 70 research papers in journals and international conferences.

He served as a referee for over 40 international journals, such as *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *IEEE Computational Intelligence Magazine*, *ACM Transactions on Intelligent Systems and Technology*, *Information Sciences*, *European Journal of Operational Research*, *Applied Soft Computing*, *Journal of Power Sources*, and so on.

Xianyan Mi received the BS degree from Guangxi University of Finance and Economics, Nanning, China in 2012. She is currently an associate professor at the College of Economics and Management, Beibu Gulf University, Qinzhou, China. Her main research interests include evolutionar computation, swarm intelligence, data mining, and operational research.

Ling Wang received the BEng degree in automation and the PhD degree in control theory and control engineering from Tsinghua University, Beijing, China in 1995 and 1999, respectively. Since 1999, he has been at the Department of Automation, Tsinghua University, where he became a full professor in 2008. His current research interests include intelligent optimization and scheduling. He has authored five academic books and more than 300 refereed papers.

He is a recipient of the National Natural Science Fund for Distinguished Young Scholars of China, the National Natural Science Award (Second Place) in 2014, the Science and Technology Award of Beijing City in 2008, the Natural Science Award (First Place in 2003 and Second Place in 2007) nominated by the Ministry of Education of China. He now is the editor-in-chief of *International Journal of Automation and Control*, and the associate editor of *IEEE Transactions on Evolutionary Computation*, *Swarm and Evolutionary Computation*, *Expert Systems with Applications*, etc.

Yuanyuan Guo received the BEng and MEng degrees in computer science from China University of Geosciences, Wuhan, China in 2003 and 2006, respectively. She received the PhD degree in computer science from University of New Brunswick, Fredericton, Canada, in 2012.

She is currently a lead data scientist at Cvent Company, Fredericton, Canada. Her research interests include various machine learning algorithms and their practical applications.