Locating Multiple Optimal Solutions of Nonlinear Equation Systems Based on Multiobjective Optimization

Wu Song, Yong Wang, *Member, IEEE*, Han-Xiong Li, *Fellow, IEEE*, and Zixing Cai, *Senior Member*, *IEEE*

***Abstract*—Nonlinear equation systems may have multiple optimal solutions. The main task of solving nonlinear equation systems is to simultaneously locate these optimal solutions in a single run. When solving nonlinear equation systems by evolutionary algorithms, usually a nonlinear equation system should be transformed into a kind of optimization problem. At present, various transformation techniques have been proposed. This paper presents a simple and generic transformation technique based on multiobjective optimization for nonlinear equation systems. Unlike the previous work, our transformation technique transforms a nonlinear equation system into a biobjective optimization problem which can be decomposed into two parts. The advantages of our transformation technique are twofold: 1) all the optimal solutions of a nonlinear equation system are the Pareto optimal solutions of the transformed problem, which are mapped into diverse points in the objective space, and 2) multiobjective evolutionary algorithms can be directly applied to handle the transformed problem. In order to verify the effectiveness of our transformation technique, it has been integrated with nondominated sorting genetic algorithm II to solve nonlinear equation systems. The experimental results have demonstrated that, overall, our transformation technique outperforms another state-of-the-art multiobjective optimization based transformation technique and four single-objective optimization based approaches on a set of test instances. The influence of the types of Pareto front on the performance of our transformation technique has been investigated empirically. Moreover, the limitation of our transformation technique has also been identified and discussed in this paper.**

***Index Terms*—Nonlinear equation systems, multiple optimal solutions, transformation technique, multiobjective optimization, evolutionary algorithms.**

I. Introduction

Nonlinear equation systems (NESs) arise in many science and engineering areas such as chemical processes [1], robotics [2], electronic circuits [3], engineered materials [4], and physics [5]. A NES can be stated as follows:

$$\begin{cases} e_1(\vec{x}) = 0 \\ e_2(\vec{x}) = 0 \\ \quad\vdots \\ e_M(\vec{x}) = 0 \end{cases} \tag{1}$$

where $\vec{x} = (x_1, ..., x_D) \in S$ is the decision vector consisting of D decision variables, S is the decision space defined by the parametric constraints:

$$L_i \le x_i \le U_i, \quad 1 \le i \le D \tag{2}$$

L_i and U_i are the lower and upper bounds of x_i, respectively, $e_i(\vec{x})$ ($i \in \{1, \dots, M\}$) is the *i*th equation, and M is the number of equations. Usually, there is at least one nonlinear equation in a NES. If $\forall i \in \{1, \dots, M\}$, $e_i(\vec{x}^*) = 0$, then $\vec{x}^*$ is called an optimal solution of a NES.

Very often, a NES may contain multiple optimal solutions. Since all these optimal solutions are important for a given NES in the real-world applications, it is desirable to simultaneously locate them in a single run, such that the decision maker can select one final solution which matches at most his/her preference.

For solving NESs, a lot of classic methods, like Newton-type methods [6], [7], [8], have been proposed. However, these methods have some disadvantages such as they are heavily dependent on the starting point of the iterative process, easily trapped in a local optimal solution, and require derivative information [9]. More importantly, these methods aim at locating just one optimal solution rather than multiple optimal solutions when solving NESs. Evolutionary algorithms (EAs) are a class of metaheuristic algorithms inspired by nature. Due to the fact that they are insensitive to the shapes of the objective function such as non-convexity and discontinuity, and easy to implement, during the past decade, EAs have been widely applied to solve NESs [10].

Currently, one kind of the most successful EAs is multiobjective EAs (MOEAs) which are designed for dealing with multiobjective optimization problems (MOPs) [11], [12], [13], [14], [15]. Since the objectives in a MOP always conflict with each other, a MOP may have many or even infinite optimal solutions. The purpose of MOEAs is to find a set of representative optimal solutions called the Pareto optimal solutions in a single run. Recently, some researchers have demonstrated that MOEAs are not only effective for MOPs, but also can be extended to solve other kinds of optimization problems. For instance, Deb and Saha [16] used multiobjective optimization for solving multimodal

W. Song is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with the College of Electronics and Information Engineering, QiongZhou University, Sanya 572022, China.

Y. Wang (Corresponding Author) is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China. (e-mail: ywang@csu.edu.cn).

H.-X. Li is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China.

Z. Cai is with the School of Information Science and Engineering, Central South University, Changsha 410083, China.

optimization problems. Bui *et al*. [17] investigated the use of MOEAs for dynamic optimization problems. Cai and Wang [18] incorporated multiobjective optimization into constrained optimization problems with the aim of balancing the objective function and constraints.

As pointed out previously, when solving NESs by EAs, it is expected to locate multiple optimal solutions in a single run. Obviously, this is similar to the solution of MOPs by MOEAs. Therefore, a question arises naturally is whether a NES can be transformed into a MOP and, as a result, MOEAs can be used to solve the transformed problem. Motivated by this consideration, a simple yet generic transformation technique called MONES has been proposed in this paper. In MONES, a NES is transformed into a MOP with two objectives (i.e., a biobjective optimization problem). The transformed problem consists of two parts: the first part is the location function which is used to determine the location of the images of the optimal solutions of a NES in the objective space, and the second part is the system function which can reflect the basic characteristics of a NES. MONES has the following features:

- No prior knowledge (such as the number of the optimal solutions of a NES) is required.
- All the optimal solutions of a NES are the Pareto optimal solutions of the transformed problem.
- The images of all the optimal solutions of a NES are located on the line segment defined by "y=1-x" in the objective space of the transformed problem.
- The current MOEAs can be applied to solve the transformed problem in a straightforward manner. Therefore, multiple optimal solutions of a NES could be located simultaneously in a single run.
- If a NES contains infinite optimal solutions, it is a natural way for the current MOEAs to find a number of representative optimal solutions, the images of which may be evenly distributed along the Pareto front in the objective space of the transformed problem.

The rest of this paper is organized as follows. Section II introduces multiobjective optimization problems and the related concepts. Section III briefly reviews the related work. In Section IV, MONES is presented in detail. Moreover, the differences between MONES and another multiobjective optimization based transformation technique called CA have been analyzed. In Section V, MONES and CA are integrated with NSGA-II [14] to solve a set of test instances. The performance of MONES has been compared with that of CA and four single-objective optimization based approaches. The influence of the types of Pareto front on the performance and the limitation of MONES have also been studied in this section. Finally, Section VI concludes this paper.

II. Multiobjective Optimization Problems and The Related Concepts

Since in this paper, a NES is converted into a MOP with two objectives, next we will introduce MOPs and the related concepts. A MOP can be formulated as follows:

$$\text{minimize } \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})) \quad (3)$$

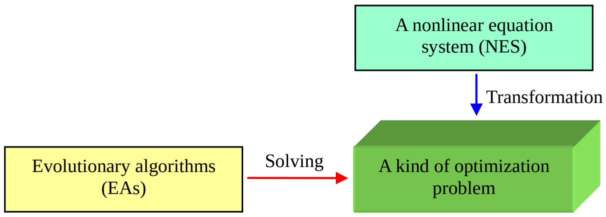


Fig. 1. The schematic graph to illustrate the principle of solving a NES by EAs

where $\vec{x} = (x_1, \ldots, x_D) \in X \subset \Re^D$ is the decision vector containing D decision variables, X is the decision space, $\vec{f}(\vec{x}) \in Y \subset \Re^M$ is the objective vector containing M objectives, and Y is the objective space.

For a MOP, the comparison between two decision vectors is based on the concept of Pareto dominance. Let $\vec{a} = (a_1, \ldots, a_D)$ and $\vec{b} = (b_1, \ldots, b_D)$ be two decision vectors, $\vec{a}$ is said to Pareto dominate $\vec{b}$ (denoted as $\vec{a} \prec \vec{b}$), if $f_i(\vec{a}) \le f_i(\vec{b})$ for $i = 1, \ldots, M$, and $\exists j \in \{1, \ldots, M\}$, $f_j(\vec{a}) < f_j(\vec{b})$. If a decision vector $\vec{x}^*$ is not Pareto dominated by any other decision vector in the decision space, then it is called a Pareto optimal solution. The set of all the Pareto optimal solutions is called the Pareto set (denoted as *PS*). Note that each solution in the Pareto set is also called a nondominated solution. The Pareto front (denoted as *PF*) is the image of the Pareto set in the objective space:

$$PF = \{\vec{f}(\vec{x}) \mid \vec{x} \in PS\} \quad (4)$$

A MOP may have many or even infinite Pareto optimal solutions. In general, it is impractical to obtain all the Pareto optimal solutions under this condition. A suitable way is to find a good approximation of the Pareto set/front in the decision/objective space, which is the essential task of the current MOEAs.

III. Previous Work

Recent years have witnessed significant progress in the development of EAs for NESs, and a considerable number of methods have been proposed. As shown in Fig. 1, the principle of the current methods can be summarized as follows: firstly a NES is transformed in to a kind of optimization problem, and then EAs are utilized to solve the transformed problem.

At present, numerous transformation techniques have been introduced. Based on the characteristics of the transformation techniques, in this paper we classify the current methods into three categories: single-objective optimization based methods, constrained optimization based methods, and multiobjective optimization based methods. Next, we will briefly review them in turn. It is necessary to point out that each kind of methods corresponds to a kind of transformation techniques.

A. *Single-objective Optimization Based Methods*

In this kind of methods, a NES is usually transformed into one of the following single-objective optimization problems:

$$\text{minimize } \sum_{i=1}^{M} |e_i(\vec{x})| \tag{5}$$

or

$$\text{minimize } \sum_{i=1}^{M} e_i^2(\vec{x}) \tag{6}$$

After the above transformation, many studies have been conducted to optimize equation (5) or equation (6) by EAs. Wu and Kang [19] presented a parallel elite-subspace EA. Mhetre *et al*. [20] illustrated how genetic algorithm can be used to find the solution of a NES. Noriega *et al*. [21] applied evolution strategy to calculate the position of all the elements of the direct position problem, which can be modeled as a NES. Wang [22] introduced an immune genetic algorithm. Xie *et al*. [23] proposed an improved bat algorithm based on differential operator and Lévy flights trajectory. Wu *et al*. [24] combined Metropolis rule with social emotional optimization algorithm. Abdollahi *et al*. [25] employed the imperialist competitive algorithm and made a simple modification. Zhou *et al*. [26] proposed a glowworm swarm optimization algorithm with leader mechanism. Recently, particle swarm optimization (PSO) [27] has attracted a lot of research effort for solving NESs. For example, MO *et al*. [28] incorporated conjugate direction method into PSO. Jaberipour *et al*. [29] improved the position and velocity update of PSO. Voglis *et al*. [30] proposed a method named PSO with deliberate loss of information.

It is noteworthy that the above work is usually biased towards finding a single optimal solution of a NES in one run, since no additional diversity mechanisms have been added. In order to locate multiple optimal solutions simultaneously, several attempts have been made. Brits *et al*. [31] adapted the standard PSO and implemented a niching strategy to guide the swarm towards different candidate solutions. Hirsch *et al*. [32] made use of a continuous global optimization heuristic called C-GRASP to solve NESs. Moreover, once an optimal solution has been found, C-GRASP will restart to optimize a modified objective function, with the aim of creating an area of repulsion around the optimal solutions that have been found. The modified objective function can be formulated as follows:

$$\text{minimize } \sum_{i=1}^{M} e_i^2(\vec{x}) + \beta \sum_{j=1}^{k} e^{-\|\vec{x}-\vec{x}_j\|} \chi_\rho(\|\vec{x}-\vec{x}_j\|) \tag{7}$$

where

$$\chi_\rho(\delta) = \begin{cases} 1, & \text{if } \delta \le \rho \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

$\vec{x}_j$ is the jth optimal solution that has been found, β is a large constant, and ρ is a small constant. Pourjafari and Mojallali [33] proposed a novel approach which is composed of a two-phase root-finder and a fitness alteration technique. The two-phase root-finder adopts invasive weed optimization as the search engine and is used to detect some optimal solutions of a NES. In addition, like [32] the fitness alteration technique aims at creating a repulsion area around the earlier located optimal solutions.

Despite the above three approaches have the capability to find multiple optimal solutions of a NES, some problem-dependent parameters have been introduced, such as the number of individuals in each niching and the clustering radius. In addition, a certain level of prior knowledge is required by them, such as the number of the optimal solutions.

B. *Constrained Optimization Based Methods*

Kuri-Morales [34] transformed a NES into the following constrained optimization problem:

$$\begin{cases} \text{minimize } \sum_{i=1}^{M} |e_i(\vec{x})| \\ \text{subject to } e_i(\vec{x}) \ge 0, \quad i = 1, \ldots, M \end{cases} \tag{9}$$

After the above transformation, a penalty function is designed to handle constraints and a rugged genetic algorithm is employed to search the optimal solution.

In contrast, a complex NES is converted into the following constrained optimization problem by Pourrajabian *et al*. in [35]:

$$\begin{cases} \text{minimize } \sum_{i=1}^{M} |e_i(\vec{x})| \\ \text{subject to } e_i(\vec{x}) = 0, \quad i = 1, \ldots, M \end{cases} \tag{10}$$

Subsequently, genetic algorithm, coupled with augmented Lagrangian function, has been used to find the optimal solution.

Similar to the first kind of methods, this kind of methods also cannot guarantee to obtain multiple optimal solutions of a NES in a single run, if no extra mechanisms have been incorporated to increase the diversity of the population.

C. *Multiobjective Optimization Based Methods*

Grosans and Abraham [36] proposed a very interesting piece of work (abbreviated as CA in this paper), which converts a NES into a MOP by considering each equation as an objective. The transformed problem can be expressed as follows:

$$\begin{cases} \text{minimize } |e_1(\vec{x})| \\ \text{minimize } |e_2(\vec{x})| \\ \quad\vdots \\ \text{minimize } |e_M(\vec{x})| \end{cases} \tag{11}$$

Then, this transformed problem is solved via a multiobjective optimization based iterative evolutionary model. According to the empirical study, CA is promising for dealing with NESs, and outperforms Effati and Nazemi's method [37] and some classical methods.

Note that in the community of evolutionary multiobjective optimization, MOPs with more than three objectives are termed many-objective optimization problems, which have posed a grand challenge to the current MOEAs [38]. In CA, the number of objectives in the transformed problem is equal to the number of equations in a NES and, consequently, the solution of the transformed problem by MOEAs will become a very challenging task as the number of equations increases.

It is clear from the above introduction that the transformation techniques have a significant effect on the solution of NESs by EAs. There are three kinds of transformation techniques. The principle of the first kind of transformation techniques is easy to understand. Compared with the first kind of transformation techniques, the second kind of transformation techniques requires additional constraint-handling scheme due to the appearance of

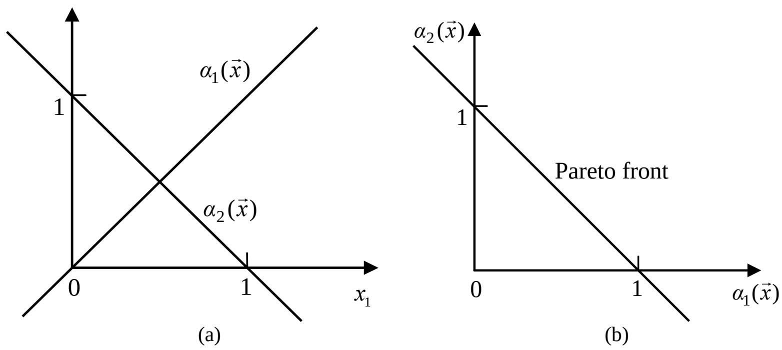


Fig. 2. The relationship between $\alpha_1(\vec{x})$ and $\alpha_2(\vec{x})$. (a) The relationship in the decision space. (b) The relationship in the objective space.

constraints. Moreover, as pointed out previously, diversity mechanisms which drive a population towards different areas in the search space are worthwhile elaborated in these two kinds of transformation techniques, so as to locate multiple optimal solutions of a NES in one run. Nevertheless, the diversity mechanisms may lead to some problem-dependent parameters. The idea of the third kind of transformation techniques is interesting. However, it may suffer from the "curse of dimensionality" (i.e., many-objective).

The above discussion motivates us to propose a more effective transformation technique. Our work in this study falls in the third kind. However, unlike the previous work, a NES is transformed into a MOP with two objectives.

IV. A Generic Transformation Technique

A. MONES

This paper presents a generic transformation technique based on multiobjective optimization for NESs called MONES, which converts a NES into a biobjective optimization problem. Inspired by [39], the transformed problem is composed of two parts: the first part is the location function which includes the location information of the images of the optimal solutions of a NES in the objective space, and the other part is the system function which includes the basic information of a NES.

The location function can be formulated as follows:

$$\begin{cases}\text{minimize } \alpha_1(\vec{x}) = x_1 \\ \text{minimize } \alpha_2(\vec{x}) = 1 - x_1\end{cases} \tag{12}$$

where $\vec{x} = (x_1,...,x_D) \in S$ is the decision vector and x_1 is the first decision variable of a NES.

The relationship between $\alpha_1(\vec{x})$ and $\alpha_2(\vec{x})$ has been depicted in Fig. 2. As shown in Fig. 2(a), with respect to x_1, $\alpha_1(\vec{x})$ is a strictly monotone increasing function and $\alpha_2(\vec{x})$ is a strictly monotone decreasing function. Therefore, as the two objectives of the location function, $\alpha_1(\vec{x})$ and $\alpha_2(\vec{x})$ conflict with each other. According to the concepts introduced in Section II, it can be easily deduced that each decision vector in the decision space of a NES is a Pareto optimal solution of the location function. Moreover, as shown in Fig. 2(b), the Pareto front of the location function is a line segment defined by "*y*=1-*x*" in the objective space.

The system function has the following form:

$$\begin{cases}\text{minimize } \beta_1(\vec{x}) = \sum_{i=1}^{M} |e_i(\vec{x})| \\ \text{minimize } \beta_2(\vec{x}) = M * \max(|e_1(\vec{x})|,...,|e_M(\vec{x})|)\end{cases} \tag{13}$$

where $\vec{x} = (x_1,...,x_D) \in S$ is the decision vector, and $e_1(\vec{x})$,..., $e_M(\vec{x})$ are the M equations of a NES.

In the system function, $\beta_1(\vec{x})$ is the sum of the absolute values of all the equations, and $\beta_2(\vec{x})$ is the maximum absolute value of all the equations multiplied by the number of equations M. Since the maximum absolute value is one element of the absolute values of all the equations, M is used to make $\beta_1(\vec{x})$ and $\beta_2(\vec{x})$ have the similar scale. There are two properties for $\beta_1(\vec{x})$ and $\beta_2(\vec{x})$: 1) both $\beta_1(\vec{x})$ and $\beta_2(\vec{x})$ are nonnegative, and 2) for an optimal solution $\vec{x}^*$ of a NES, $\beta_1(\vec{x}^*) = \beta_2(\vec{x}^*) = 0$.

By combining the location function with the system function, the transformed biobjective optimization problem can be obtained:

$$\begin{cases}\text{minimize } f_1(\vec{x}) = \alpha_1(\vec{x}) + \beta_1(\vec{x}) \\ \qquad\qquad = x_1 + \sum_{i=1}^{M} |e_i(\vec{x})| \\ \text{minimize } f_2(\vec{x}) = \alpha_2(\vec{x}) + \beta_2(\vec{x}) \\ \qquad\qquad = 1 - x_1 + M * \max(|e_1(\vec{x})|,...,|e_M(\vec{x})|)\end{cases} \tag{14}$$

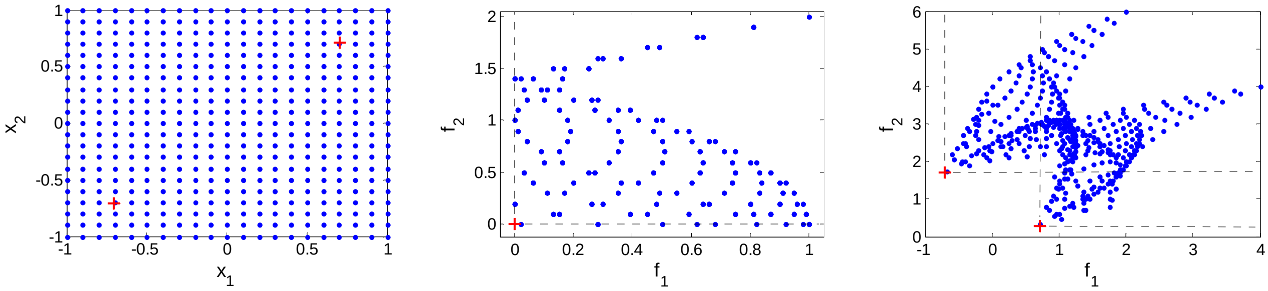
Theorem 1: All the optimal solutions of a NES are the Pareto optimal solutions of the transformed problem.

Proof: Let $\vec{x}^*$ be one of the optimal solutions and $\vec{x}'$ be a decision vector in the decision space of a NES. According to the property of the location function, $\vec{x}^*$ cannot be Pareto dominated by $\vec{x}'$ in terms of equation (12). Furthermore, since $\beta_1(\vec{x}^*) = \beta_2(\vec{x}^*) = 0$ and $\beta_1(\vec{x}'), \beta_2(\vec{x}') \geq 0$, $\vec{x}^*$ also cannot be Pareto dominated by $\vec{x}'$ in terms of equation (14). Therefore, $\vec{x}^*$ is a Pareto optimal solution of the transformed problem.

Theorem 1 reflects the relationship between a NES and the transformed problem.

Theorem 2: The images of all the optimal solutions of a NES are located on the line segment defined by "*y*=1-*x*" in the objective space.

Proof: Let $\vec{x}^*$ be one of the optimal solutions of a NES. According to Theorem 1, $\vec{x}^*$ is a Pareto optimal solution of the transformed problem. Since $\beta_1(\vec{x}^*) = \beta_2(\vec{x}^*) = 0$, equation



(a) (b) (c)

Fig. 3. The difference between CA and MONES in the objective space. (a) 443 solutions in the decision space: "." represents the uniformly chosen solutions and "+" represents the optimal solutions. (b) The images of these 443 solutions in the objective space defined by CA. (c) The images of these 443 solutions in the objective space defined by MONES.

(14) is equivalent to equation (12) under this condition. As a result, $\vec{x}^*$ is also a Pareto optimal solution of the location function. As shown in Fig. 2(b), the image of $\vec{x}^*$ is located on the line segment defined by "*y*=1-*x*" in the objective space.

Theorem 2 verifies that the location information of the images of all the optimal solutions of a NES in the objective space is determined by the location function.

Remark 1: After the above transformation, MOEAs can be directly applied to solve the transformed problem (i.e., equation (14)). When using MOEAs to solve the transformed problem, the goal is to find diverse Pareto optimal solutions in a single run. Consequently, multiple optimal solutions of a NES could be located concurrently according to Theorem 1. In addition, a NES may contain infinite optimal solutions. In this case, MOEAs can also be applied to produce a good approximation to the Pareto set in a single run, the image of which may be evenly located on the Pareto front in the objective space.

B. The Differences between CA and MONES

It is worth noting that both CA [36] and MONES belong to the third kind of transformation techniques introduced in Section III-C. However, there are three major differences between CA and MONES which have been summarized as follows.

1) The first difference is that CA transforms a NES into a MOP with M objectives, while MONES transforms a NES into a MOP with two objectives. As mentioned previously, CA may suffer from the "curse of dimensionality", because the number of objectives is equal to the number of equations in a NES. Clearly, MONES is a possible way to overcome this shortcoming since only two objectives are needed to consider.

2) Due to the fact that the transformed objectives in CA and MONES are distinct, there is a difference between CA and MONES in the objective space. To better understand the second difference, we employ the following NES as an example:

$$\begin{cases} e_1(\vec{x}) = x_1^2 + x_2^2 - 1 = 0 \\ e_2(\vec{x}) = x_1 - x_2 = 0 \\ -1 \le x_1, x_2 \le 1 \end{cases} \tag{15}$$

This NES includes two decision variables and two equations. The shapes of the first and second equations are a circle and a line segment, respectively. There are two optimal solutions of this NES in the decision space, which are the intersection points of the circle and the line segment: $\vec{x}_1^* = (0.707, 0.707)$ and $\vec{x}_2^* = (-0.707, -0.707)$.

Based on equation (11), CA will transform this NES into the following MOP:

$$\begin{cases} \text{minimize } f_1(\vec{x}) = | x_1^2 + x_2^2 - 1 | \\ \text{minimize } f_2(\vec{x}) = | x_1 - x_2 | \end{cases} \tag{16}$$

In contrast, the following MOP will be taken into account by MONES based on equation (14):

$$\begin{cases} \text{minimize } f_1(\vec{x}) = x_1 + | x_1^2 + x_2^2 - 1 | + | x_1 - x_2 | \\ \text{minimize } f_2(\vec{x}) = 1 - x_1 + 2 * \max(| x_1^2 + x_2^2 - 1 |, | x_1 - x_2 |) \end{cases} \tag{17}$$

Fig. 3 is utilized to explain the second difference between CA and MONES. In Fig. 3(a), we choose 441 solutions uniformly sampled from the decision space and the two optimal solutions. Fig. 3 (b) and Fig. 3(c) present the images of the above 443 solutions in the objective space defined by CA and MONES, respectively.

As can be observed from Fig. 3(b) and Fig. 3(c), both the two optimal solutions of the given NES are the Pareto optimal solutions, regardless of the transformation techniques. However, it is interesting to note that for CA, the images of these two optimal solutions are the same in the objective space (i.e., many-to-one mapping as shown in Fig. 3(b)). This mapping might have a side effect on concurrently keeping multiple optimal solutions. It is because when using MOEAs to solve a MOP, the selection process is implemented based only on the information in the objective space and, as a result, some of the optimal solutions found in the decision space might be neglected unreasonably. On the contrary, multiple optimal solutions can survive at the same time in MONES, since the images of the optimal solutions of the given NES are different in the objective space (i.e., one-to-one mapping as shown in Fig. 3(c)).

3) In order to search for multiple optimal solutions of a NES in a single run, the individual which is close to one of the optimal solutions is promising and should survive into the next population. Indeed, there is a significant difference

TABLE I
THE INFORMATION OF SIX INDIVIDUALS

Individual	Position		The objective function values in CA		The objective function values in MONES	
	x_1	x_2	f_1	f_2	f_1	f_2
A	-0.95	-0.75	0.4650	0.2000	-0.2850	2.8800
B	-0.60	-0.70	0.1500	0.1000	-0.3500	1.9000
C	0.45	-0.80	0.1575	1.2500	1.8575	3.0500
D	0.50	-0.90	0.0600	1.4000	1.9600	3.3000
E	0.75	0.40	0.2775	0.3500	1.3775	0.9500
F	0.80	0.20	0.3200	0.6000	1.7200	1.4000

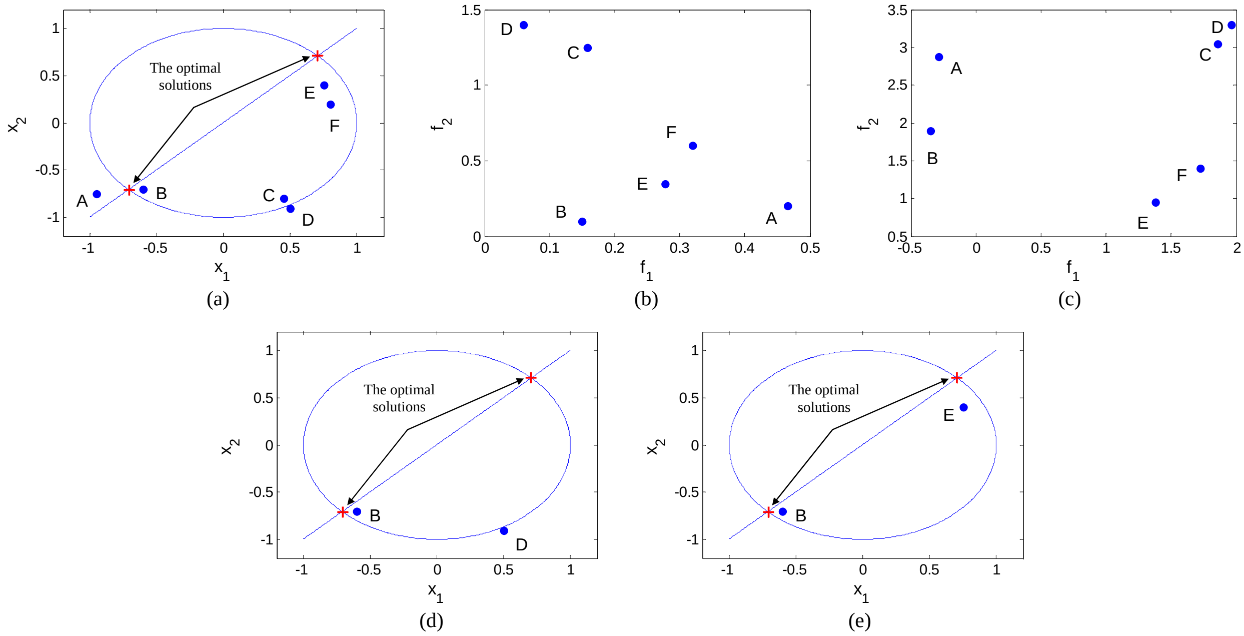


Fig. 4. The difference between CA and MONES in maintaining the potential individuals. (a) The six individuals of Table I in the decision space. (b) The images of these six individuals in the objective space defined by CA. (c) The images of these six individuals in the objective space defined by MONES. (d) The nondominated individuals in the decision space with CA. (e) The nondominated individuals in the decision space with MONES.

between CA and MONES in maintaining such potential individuals.

To investigate the third difference, equation (15) is still adopted as an example. Suppose that the population contains six individuals, the information of which has been summarized in Table I. Based on the information in Table I, Fig. 4(a) depicts these six individuals in the decision space. In addition, Fig. 4(b) and Fig. 4(c) present the images of these six individuals in the objective space defined by CA and MONES, respectively. From the concepts introduced in Section II, it can be deduced that individuals B and D are the nondominated individuals in Fig. 4(b) and individuals B and E are the nondominated individuals in Fig. 4(c). These two groups of the nondominated individuals in the decision space have been shown in Fig. 4(d) and Fig. 4(e), respectively.

If only two individuals can be preserved into the next population, then CA will prefer individuals B and D, and MONES will place more emphasis on individuals B and E. Obviously, CA is not capable of maintaining the potential individuals under this condition, since individual E which is close to one of the optimal solutions has been removed. As a consequence, CA is very likely to miss some optimal solutions in the end. Compared with CA, MONES seems to be more powerful.

V. EXPERIMENTAL STUDY

A. *Test Instances*

In this paper, seven test instances (denoted as F1-F7) with different characteristics are used to investigate the effectiveness of MONES. The first five test instances are designed in this paper and the remaining two test instances are real-world applications from the neurophysiology [40] and the economics modeling [41], respectively. The details of these seven test instances have been reported in Table II, including the number of the decision variables, the decision space, the number of the linear equations, the number of the nonlinear equations, and the number of the optimal solutions. These test instances can be categorized into three groups, according to the number of the optimal solutions:

1) F1 and F2 have two optimal solutions. Note that F1 has been introduced in Section IV-B. F2 includes 20 decision variables and is designed to evaluate the performance of an algorithm in a high-dimensional decision space. In principle, F2 can be regarded as a generalized implementation of F1. The optimal solutions of F1 and F2 are the same in the $x_1 - x_2$ space.

TABLE II
DETAILS OF THE SEVEN TEST INSTANCES

Test instance		Number of the decision variables	The decision space	Number of the linear equations	Number of the nonlinear equations	Number of the optimal solutions
F1	$x_1^2+x_2^2-1=0$ $x_1-x_2=0$	2	$[-1,1]^2$	1	1	2
F2	$\sum_{i=1}^{D}x_i^2-1=0$ $\|x_1-x_2\|+\sum_{i=3}^{D}x_i^2=0$	20	$[-1,1]^{20}$	0	2	2
F3	$x_1-\sin(5*\pi*x_2)=0$ $x_1-x_2=0$	2	$[-1,1]^2$	1	1	11
F4	$x_1-\cos(4*\pi*x_2)=0$ $x_1^2+x_2^2=1$	2	$[-1,1]^2$	0	2	15
F5	$x_1+x_2+x_3-1=0$ $x_1-x_2^3=0$	3	$[-1,1]^3$	1	1	infinite
F6	$x_1^2+x_3^2=1$ $x_2^2+x_4^2=1$ $x_5x_3^3+x_6x_4^3=0$ $x_5x_1^3+x_6x_2^3=0$ $x_5x_1x_3^2+x_6x_4^2x_2=0$ $x_5x_1^2x_3+x_6x_2^2x_4=0$	6	$[-1,1]^6$	0	6	infinite
F7	$(x_k+\sum_{i=1}^{D-k-1}x_ix_{i+k})x_D-c_k=0,\ 1\le k\le D-1$ $\sum_{l=1}^{D-1}x_l+1=0$	20	$[-1,1]^{20}$	1	19	infinite

2) F3 and F4 have more than two optimal solutions. Concretely, F3 has 11 optimal solutions and F4 has 15 optimal solutions. In these two test instances, some optimal solutions are very close to each other, which makes them very difficult for an algorithm to locate all the optimal solutions in a single run.

3) F5, F6, and F7 have infinite optimal solutions. In F7, $\forall k\in\{1,\ldots,D-1\}$, $c_k=0$. For these three test instances, it is impossible to obtain all the optimal solutions in a single run. Therefore, one has to find a set of representative optimal solutions in one run, which can well approximate the whole optimal set. In addition, F6 and F7 can be exploited to show the limitations of CA, since they include six and 20 equations, respectively. Under this condition, CA will have six and 20 objectives (i.e., many-objective), respectively.

B. Using MOEAs to Solve the Transformed Problem

It is necessary to emphasize that the primary focus of this paper is the transformation technique. Since CA and MONES belong to the same kind of transformation techniques, the performance comparison is mainly conducted between them in this paper. Essentially, the current MOEAs can be applied to solve the MOPs transformed by CA and MONES in a straightforward way. In this paper, one of the most popular MOEAs, nondominated sorting genetic algorithm II (NSGA-II) [14], has been chosen as an example. The implementation of NSGA-II for solving the transformed problem has been introduced as follow.

Step 1 $G=0$; // G is the generation number

Step 2 Randomly generate an initial population P_G of size N from the decision space;

Step 3 Evaluate each individual in P_G based on equation (11) or equation (14);

Step 4 Implement the binary tournament selection, simulated binary crossover, and polynomial mutation to generate the offspring population Q_G;

Step 5 Evaluate each individual in Q_G based on equation (11) or equation (14);

Step 6 $H_G=P_G\cup Q_G$;

Step 7 Divide H_G into several nondomination levels (denoted as $ND_1, ND_2,\ldots$) according to a fast nondominated sorting;

Step 8 $P_{G+1}=\phi$ and $i=1$;

Step 9 **While** $|P_{G+1}|<N$

Step 10 $P_{G+1}=P_{G+1}\cup ND_i$ and $i=i+1$;

Step 11 **End While**

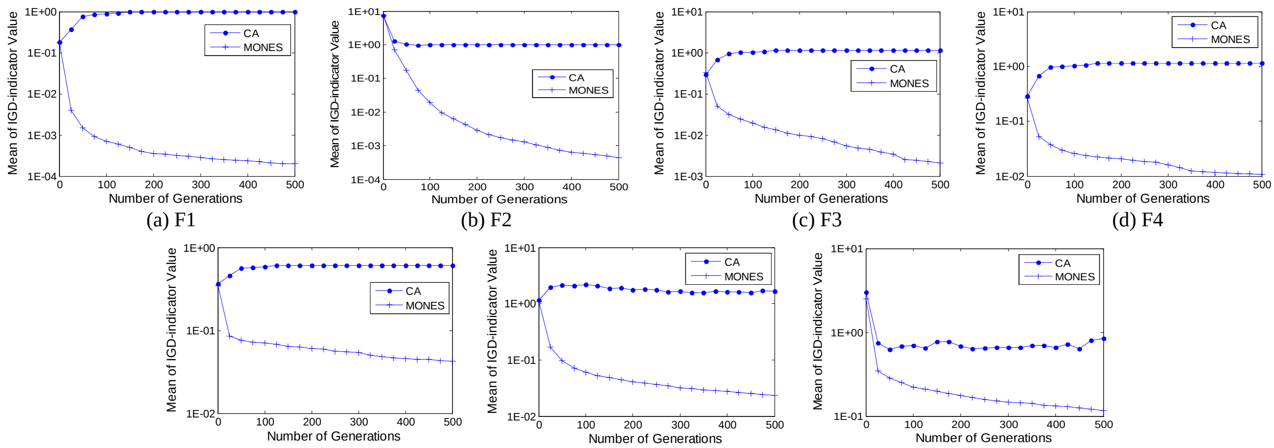
Step 12 Let $P_{G+1}=P_{G+1}\setminus ND_{i-1}$, delete $(|ND_{i-1}|+|P_{G+1}|-N)$ individuals with the smallest crowding-distance values in ND_{i-1}, and let $P_{G+1}=P_{G+1}\cup ND_{i-1}$;

Step 13 If the stopping criterion is satisfied, stop and output the final population, otherwise $G=G+1$ and go to step 4.

In the above procedure, $|P_{G+1}|$ and $|ND_{i-1}|$ are the numbers of individuals in P_{G+1} and ND_{i-1}, respectively. The details of the fast nondominated sorting and the crowding-

TABLE III
EXPERIMENTAL RESULTS OF CA AND MONES OVER 30 INDEPENDENT RUNS ON SEVEN TEST INSTANCES IN TERMS OF TWO PERFORMANCE INDICATORS. THE BETTER MEAN IGD-INDICATOR VALUE AND MEAN NOF-INDICATOR VALUE FOR EACH TEST INSTANCE ARE HIGHLIGHTED IN BOLDFACE.

Test instance	Status	IGD		NOF	
		CA	MONES	CA	MONES
F1	Best	9.97E-01	1.47E-04	1.00E+00	2.00E+00
	Mean	9.99E-01	**2.01E-04**	1.00E+00	**2.00E+00**
	Worst	1.00E+00	3.77E-04	1.00E+00	2.00E+00
	Std Dev	4.09E-04	4.74E-05	0.00E+00	0.00E+00
F2	Best	9.90E-01	2.06E-04	1.00E+00	2.00E+00
	Mean	9.99E-01	**4.44E-04**	1.00E+00	**2.00E+00**
	Worst	1.00E+00	9.25E-04	1.00E+00	2.00E+00
	Std Dev	1.98E-03	1.95E-04	0.00E+00	0.00E+00
F3	Best	5.51E-01	1.11E-03	1.00E+00	1.10E+01
	Mean	1.16E+00	**2.12E-03**	1.00E+00	**1.10E+01**
	Worst	1.30E+00	4.45E-03	1.00E+00	1.10E+01
	Std Dev	2.26E-01	7.48E-04	0.00E+00	0.00E+00
F4	Best	7.36E-01	2.82E-03	1.00E+00	1.50E+01
	Mean	7.83E-01	**1.06E-02**	1.00E+00	**1.41E+01**
	Worst	1.18E+00	2.90E-02	1.00E+00	1.10E+01
	Std Dev	1.10E-01	7.52E-03	0.00E+00	1.16E+00
F5	Best	3.14E-01	1.63E-02	1.00E+00	5.00E+01
	Mean	6.11E-01	**4.24E-02**	1.00E+00	**3.73E+01**
	Worst	7.07E-01	2.12E-01	1.00E+00	2.10E+01
	Std Dev	1.12E-01	3.80E-02	0.00E+00	7.05E+00
F6	Best	6.55E-01	1.28E-02	0.00E+00	8.10E+01
	Mean	1.66E+00	**2.31E-02**	0.00E+00	**7.31E+01**
	Worst	3.11E+00	5.13E-02	0.00E+00	6.60E+01
	Std Dev	6.03E-01	9.45E-03	0.00E+00	3.81E+00
F7	Best	2.62E-01	4.46E-02	8.00E+00	3.80E+01
	Mean	8.41E-01	**1.15E-01**	1.80E+00	**2.02E+01**
	Worst	3.45E+00	2.37E-01	0.00E+00	0.00E+00
	Std Dev	6.59E-01	6.05E-02	1.81E+00	1.15E+01



(a) F1 (b) F2 (c) F3 (d) F4
(e) F5 (f) F6 (g) F7

Fig. 5. The mean of IGD-indicator values in CA and MONES during the evolution over 30 independent runs.

distance calculation refer to [14]. In our experiments, we used the same parameter settings as in [14]: the crossover probability $p_c = 0.9$, the mutation probability $p_m = 1/D$, and the distribution indexes for crossover and mutation were $\eta_c = 20$ and $\eta_m = 20$, respectively.

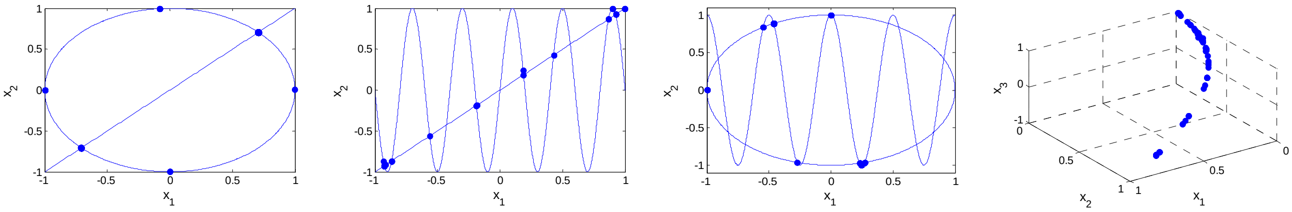
C. Performance Indicators

In this paper, two performance indicators are employed to investigate the capability of an algorithm to concurrently locate multiple optimal solutions of a NES.

1) The inverted generational distance (IGD) [42]: Let IP be a set of the images of the individuals of a population in the objective space, and IP^* be a set of the images of all the optimal solutions of a NES in the objective space: $IP^* = \{\vec{v}_1, \ldots, \vec{v}_{|IP^*|}\}$. The IGD indicator is computed as

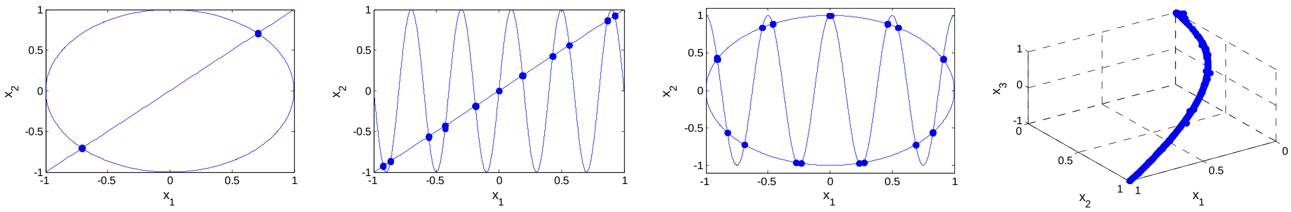
$$IGD(IP, IP^*) = \frac{\sum_{i=1}^{|IP^*|} d(\vec{v}_i, IP)}{|IP^*|} \tag{18}$$

where $d(\vec{v}_i, IP)$ is the minimum Euclidean distance between $\vec{v}_i$ and the points in IP. Note that if a NES (such as F5, F6, or F7) has infinite optimal solutions, IP^* is a set of uniformly distributed points in the objective space along the Pareto front. In this paper, the number of points in IP^* (i.e., $|IP^*|$) is set to 100 for F5, F6, and F7. Besides, to make the comparison



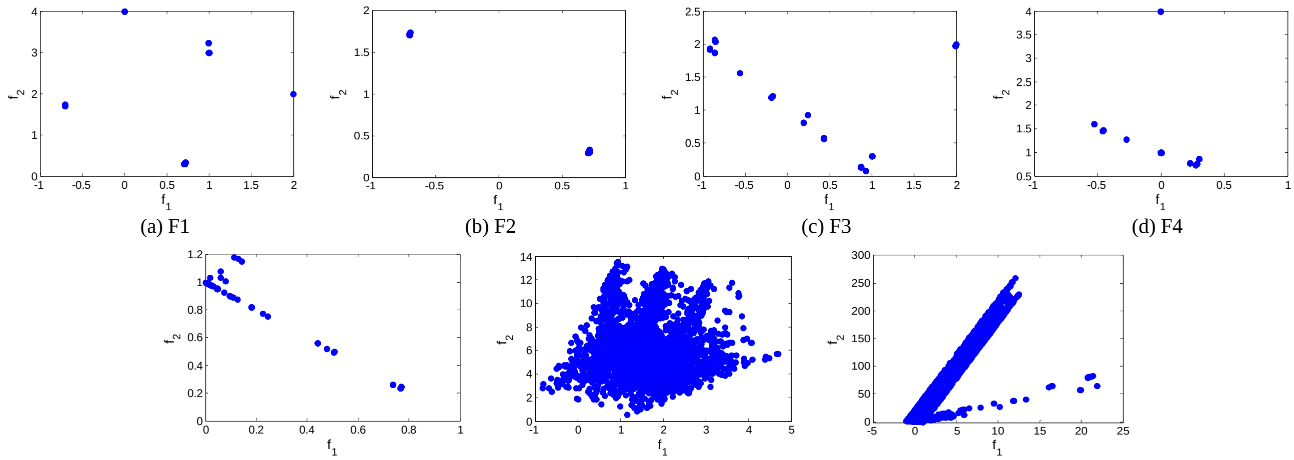
(a) F1 (b) F3 (c) F4 (d) F5

Fig. 6. Plots of the 30 final populations in the decision space obtained by CA.



(a) F1 (b) F3 (c) F4 (d) F5

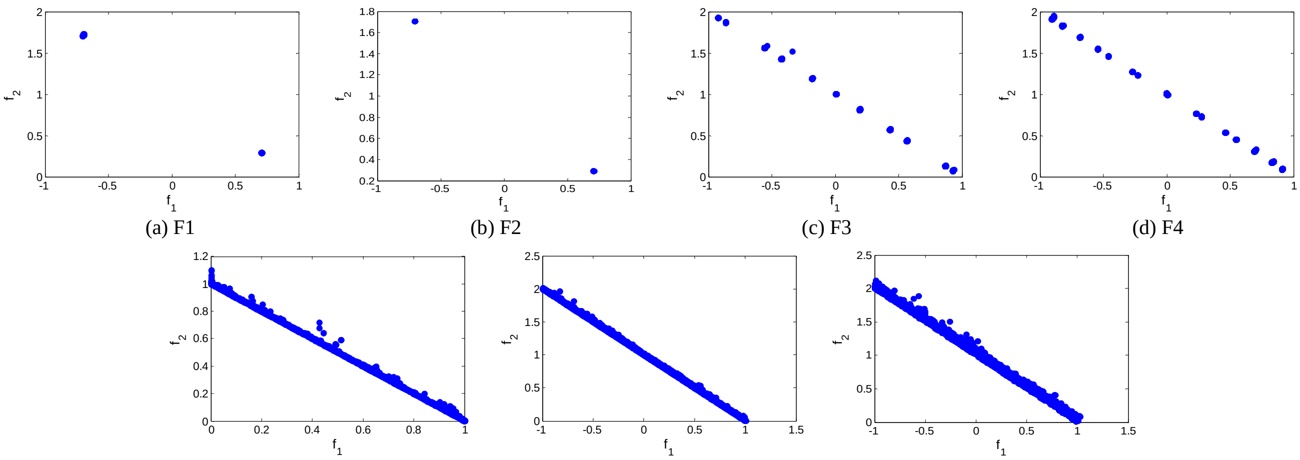
Fig. 7. Plots of the 30 final populations in the decision space obtained by MONES.



(a) F1 (b) F2 (c) F3 (d) F4

(e) F5 (f) F6 (g) F7

Fig. 8. Plots of the 30 final populations in the objective space obtained by CA.



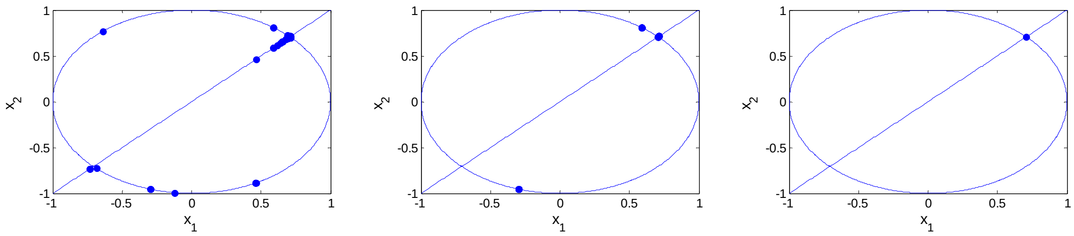
(a) F1 (b) F2 (c) F3 (d) F4

(e) F5 (f) F6 (g) F7

Fig. 9. Plots of the 30 final populations in the objective space obtained by MONES.

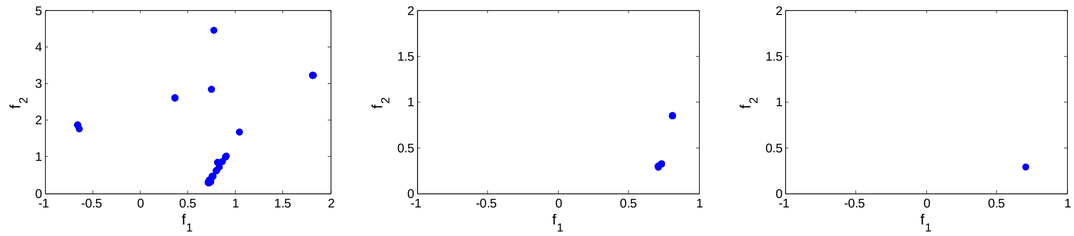
convenient, the individuals in the final populations derived from an algorithm are mapped to the objective space defined by MONES.

This indicator is able to measure both the diversity and convergence of *IP*. In an ideal case, $IGD(IP, IP^*) = 0$. If a NES contains several optimal solutions (rather than infinite



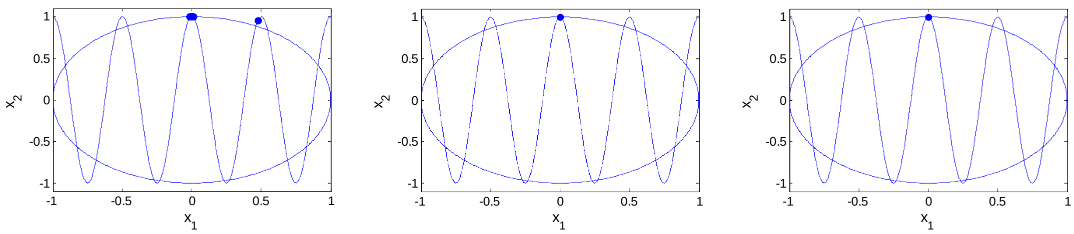
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 10. The convergence behavior in a typical run provided by CA in the decision space for F1.



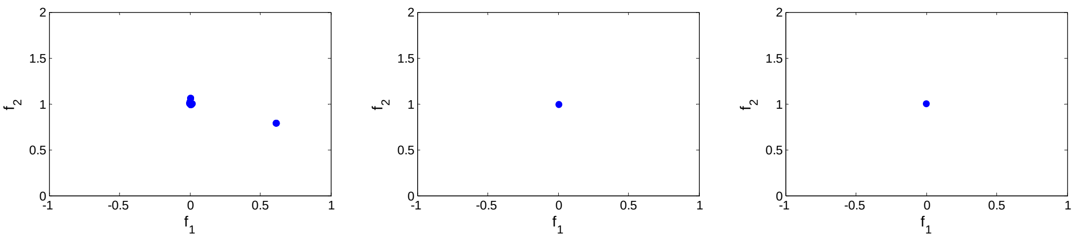
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 11. The convergence behavior in a typical run provided by CA in the objective space for F1.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 12. The convergence behavior in a typical run provided by CA in the decision space for F4.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 13. The convergence behavior in a typical run provided by CA in the objective space for F4.

optimal solutions), $IGD(IP, IP^*) = 0$ indicates all the optimal solutions have been found in a single run. In addition, if a NES contains infinite optimal solutions, $IGD(IP, IP^*) = 0$ indicates that all the points in IP are located on the Pareto front and cover the Pareto front uniformly.

2) Number of the optimal solutions found (NOF): The NOF indicator is computed in this paper according to equations (19) and (20)

$$NOF(IP, IP^*) = \sum_{i=1}^{|IP^*|} flag(\vec{v}_i) \tag{19}$$

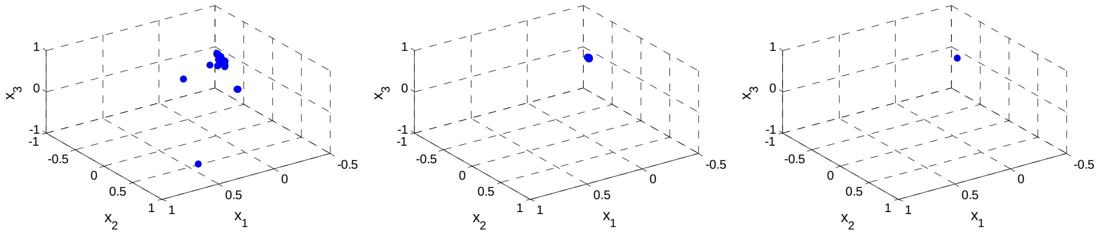
$$\begin{cases} flag(\vec{v}_i) = 1, & \text{if } d(\vec{v}_i, IP) \le \varepsilon,\ \vec{v}_i \in IP^* \\ flag(\vec{v}_i) = 0, & \text{otherwise} \end{cases} \tag{20}$$

where the parameter ε is a user-defined threshold value. In this paper, ε was set to 0.01 for F5 and 0.02 for the other six test instances, respectively. In general, the larger the NOF-indicator value, the better the performance of an algorithm.

D. Comparison between CA and MONES

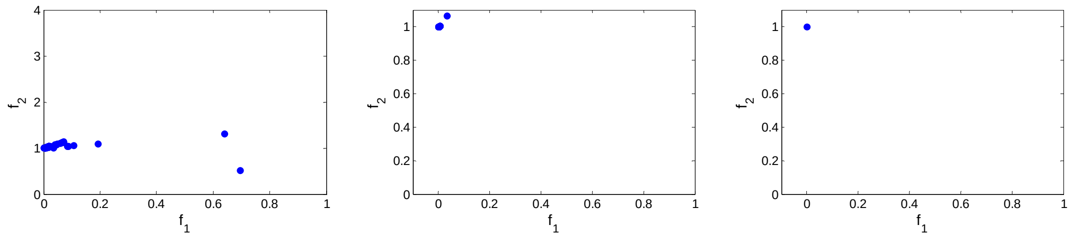
To have a fair comparison, for both CA and MONES, the population size N was set 100, 30 independent runs were executed on each test instance, and the maximum generation number was set to 500 (i.e., the maximum number of fitness evaluations was 50000). Moreover, CA and MONES were combined with the same version of NSGA-II with the same parameter settings as introduced in Section V-B.

Table III presents the best, mean, worst, and standard deviation of the IGD-indicator values and the NOF-indicator values derived from CA and MONES. With respect to the IGD indicator, we also implemented a statistical test (i.e., Wilcoxon's rank sum test at a 0.05 significance level) between CA and MONES. The statistical test reveals that MONES is significantly better than CA on all the test instances. In terms of the NOF indicator, it is clear that



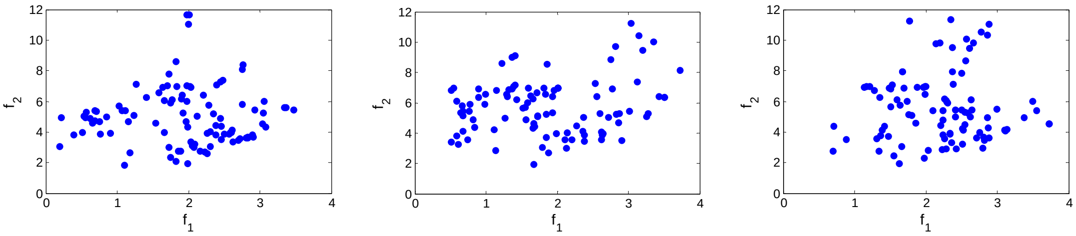
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 14. The convergence behavior in a typical run provided by CA in the decision space for F5.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 15. The convergence behavior in a typical run provided by CA in the objective space for F5.



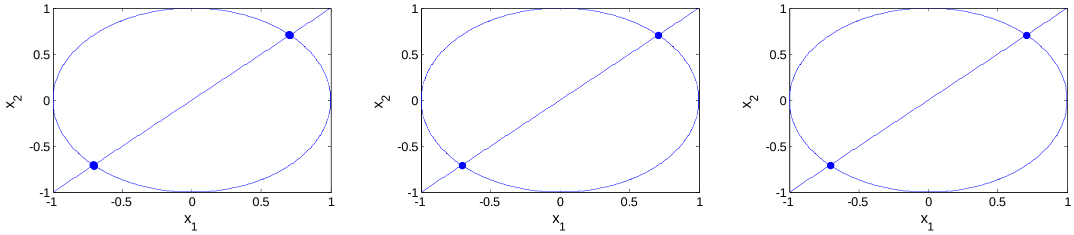
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 16. The convergence behavior in a typical run provided by CA in the objective space for F6.

MONES can successfully locate all the optimal solutions for F1-F3 over 30 runs. For F4, MONES is able to find all the optimal solutions over a majority of runs. Since $|IP^*|$ was set to 100 for F5-F7 as introduced in Section V-C, the default number of the optimal solutions in F5-F7 is 100 in this paper. For these three test instances, MONES has the capability to maintain a lot of the optimal solutions in the final populations. However, it seems to be very challenging for CA to locate multiple optimal solutions in a single run on all the test instances. Specifically, CA tends to find only one optimal solution for F1-F5, and fails in converging to any of the optimal solutions for F6.

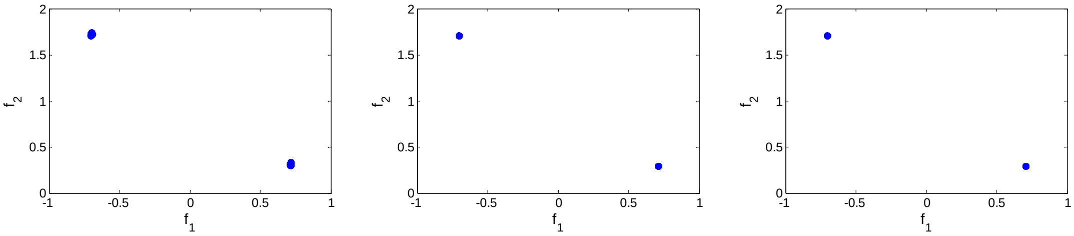
Fig. 5 shows the evolution of the mean IGD-indicator values provided by CA and MONES for all the test instances over 30 independent runs. As depicted in Fig. 5, CA is stuck in terms of the IGD indicator after about 100 generations for all the test instances. Moreover, it is interesting to see that for F1, F3, F4, F5, and F6, the IGD-indicator values in CA increase as the search proceeds before 100 generations. The above phenomenon can be explained as follows. Due to the relatively smaller decision space of F1, F3, F4, F5, and F6, the IGD-indicator value of the initial population which consists of N individuals randomly and uniformly sampled from the decision space is small (between 0.1 and 1.1 based on our observation). Since CA tends to converge to one of the optimal solutions, the population may cluster in a small area of the decision space gradually. As a result, the IGD-indicator value of the initial population is even smaller than that of the population which approximates the Pareto front, and the IGD-indicator value gradually increases before 100 generations. In addition, although very few optimal solutions can be found by CA for F2 and F7, the IGD-indicator values in CA gradually decrease at the early stage because of the relatively larger decision space (i.e., $[-1,1]^{20}$), which results in the relatively larger IGD-indicator value of the initial population. In contrast, MONES can reduce the IGD-indicator values continually during the evolution.

Figs. 6-9 plot the distribution of the 30 final populations obtained by CA and MONES in both the objective space and the decision space. Due to the fact that the number of the decision variables is larger than three for F2, F6, and F7, the distribution information of these three test instances has not been provided in the decision space. In Figs. 6 and 8, we can observe that different optimal solutions can be located by CA over 30 trials, although it may find only one optimal solution in each trial. However, Figs. 6 and 8 also imply that using CA with different starting populations gives no guarantee of finding all the optimal solutions. For example, with respect to F3 and F4, some optimal solutions still cannot be found by CA over 30 trials. Moreover, some parts of the Pareto front



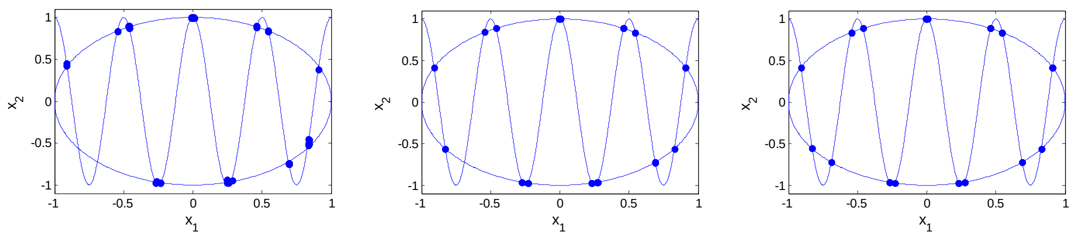
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 17. The convergence behavior in a typical run provided by MONES in the decision space for F1.



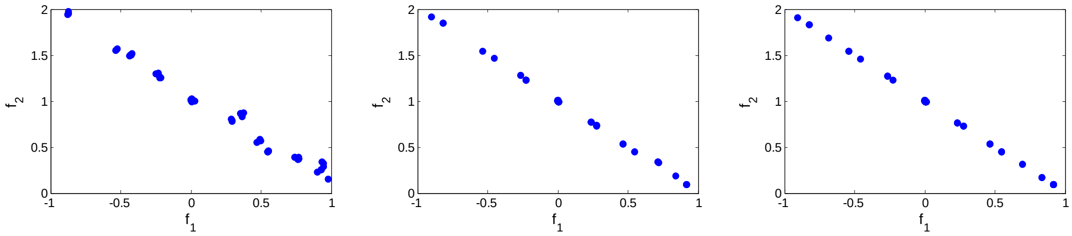
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 18. The convergence behavior in a typical run provided by MONES in the objective space for F1.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 19. The convergence behavior in a typical run provided by MONES in the decision space for F4.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

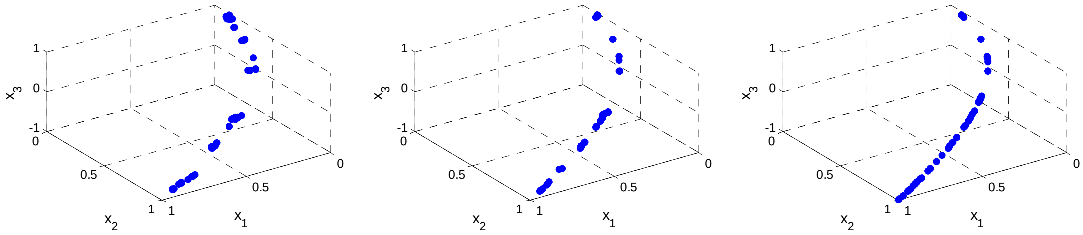
Fig. 20. The convergence behavior in a typical run provided by MONES in the objective space for F4.

are missed in the objective space for F5. It is also interesting to note that some additional solutions which are far away from the optimal solutions have been maintained by CA in the final populations, see, for example, F1, F3, and F4. This observation is consistent with our analysis in Fig. 4. It might be an inherent drawback of the transformation manner of CA. Finally, for F6 and F7, CA cannot well approximate the Pareto fronts and the distribution of the populations is irregular in the objective space. It is because the problems transformed by CA are many-objective optimization problems and the individuals in the population are frequently nondominated with each other, therefore, the behavior of CA is similar to random walk for F6 and F7. Compared with CA, MONES is much more effective to locate multiple optimal solutions for all the test instances as shown in Figs. 7 and 9.

E. The Convergence Behaviors of CA and MONES in one Run

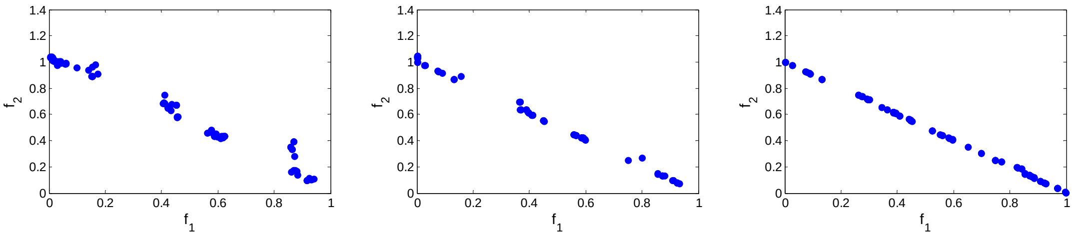
This subsection aims at further investigating the convergence behaviors of CA and MONES in a single run. Figs. 10-23 show, in both the decision space and objective space, the distribution of the final populations provided by CA and MONES over a typical run on four representative test instances, i.e., F1, F4, F5, and F6. Similar to Section V-D, the distribution information of F6 in the decision space has not been provided.

As shown in Figs. 10-15, at the early stage (about 20 iterations), some potential regions can be covered by the population of CA. However, after 100 iterations the population converges to only one of the optimal solutions. As



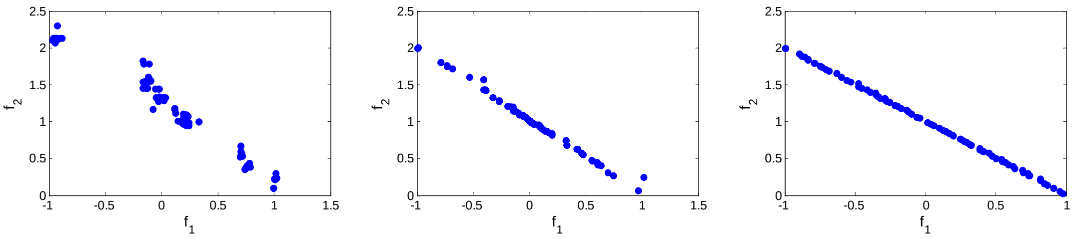
(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 21. The convergence behavior in a typical run provided by MONES in the decision space for F5.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 22. The convergence behavior in a typical run provided by MONES in the objective space for F5.



(a) 20 iterations (b) 100 iterations (c) 500 iterations

Fig. 23. The convergence behavior in a typical run provided by MONES in the objective space for F6.

analyzed in Section IV-B, the disappearance of some potential solutions might be due to two facts: 1) in CA all the optimal solutions of a NES are mapped into the same point in the objective space and, as a result, some of the optimal solutions are neglected in the selection process unreasonably; and 2) in CA some potential individuals are Pareto dominated by other individuals in the population and they are eliminated during the evolution. In addition, as shown in Fig. 16, the convergence behavior of CA is quite random for F6 because of the many-objective feature of the transformed problem.

Figs. 17-23 exhibit the convergence behavior of MONES. It is clear from Figs. 17-20 that after 100 iterations, MONES can locate all the optimal solutions for F1 and F4 except that one optimal solution is missed for F4. Moreover, for F4 the missed optimal solution can be found when the evolution terminates. For F5 and F6, MONES can yield a set of representative optimal solutions in a typical run after 500 iterations as shown in Figs. 21-23. An interesting observation is that for F4, some basins of attraction, one of which contain an optimal solution, could not be covered by the population at the early stage of search (about 20 iterations), however, during the evolution such basins of attraction can be gradually explored and the entire population is distributed around different optimal solutions in the end. This could be attributed to the fact that MONES maps the optimal solutions of a NES into different points in the objective space and at the same time NSGA-II supports the coexistence of these points.

The above experimental results, in conjunction with the experimental results in Section V-D reveal that, overall, as a transformation technique, MONES is more effective than CA for solving NESs.

F. Comparison with Four Single-objective Optimization Based Methods

In this subsection, we compared MONES with four single-objective optimization based methods. As introduced in Section III-A, Hirsch *et al.* [32] designed a modified objective function (i.e., equation (7)) to generate an area of repulsion around the optimal solutions that have been found. Moreover, C-GRASP is used as the search engine. Note that C-GRASP is a point-to-point algorithm and the implementation process of C-GRASP is quite complicated. Recognizing that the population based metaheuristic algorithm is usually more powerful than the point-to-point algorithm, we replaced C-GRASP with particle swarm optimization (PSO) [27] and differential evolution (DE) [43] respectively to make the method in [32] more effective. As a result, we obtained two methods in this paper denoted as RepulsionPSO and RepulsionDE, respectively. Following the suggestion in [32], β was set to 1000 and ρ was set to 0.01 in equation (7) for both RepulsionPSO and RepulsionDE.

TABLE IV
EXPERIMENTAL RESULTS OF REPULSIONPSO, REPULSIONDE, RPSO, CDE, AND MONES OVER 30 INDEPENDENT RUNS ON SEVEN TEST INSTANCES. "MEAN NOF" AND "STD DEV" INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE NUMBER OF THE OPTIMAL SOLUTIONS FOUND IN 30 RUNS, RESPECTIVELY. WILCOXON'S RANK SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED BETWEEN MONES AND EACH OF REPULSIONPSO, REPULSIONDE, RPSO, AND CDE.

Test Instance	RepulsionPSO Mean NOF±Std Dev	RepulsionDE Mean NOF±Std Dev	rpso Mean NOF±Std Dev	CDE Mean NOF±Std Dev	MONES Mean NOF±Std Dev
F1	1.73E+00±5.21E-01－	2.00E+00±0.00E+00≈	1.43E+00±6.26E-01－	2.00E+00±0.00E+00≈	2.00E+00±0.00E+00
F2	4.67E-01±5.07E-01－	0.00E+00±0.00E+00－	3.50E-01±5.57E-01－	0.00E+00±0.00E+00－	2.00E+00±0.00E+00
F3	1.63E+00±8.09E-01－	6.13E+00±1.22E+00－	1.60E+00±8.14E-01－	4.43E+00±1.72E+00－	1.10E+01±0.00E+00
F4	3.43E+00±1.33E+00－	8.07E+00±1.82E+00－	2.97E+00±1.21E+00－	8.26E+00±1.86E+00－	1.41E+01±1.16E+00
F5	2.77E+00±1.38E+00－	6.70E+00±1.62E+00－	2.63E+00±1.27E+00－	1.45E+01±3.15E+00－	3.73E+01±7.05E+00
F6	6.67E-02±2.54E-01－	1.50E+00±6.29E-01－	1.00E-01±3.05E-01－	1.95E+01±4.30E+00－	7.31E+01±3.81E+00
F7	5.57E+00±1.63E+00－	1.63E+00±6.69E-01－	9.47E+00±2.26E+00－	6.57E+00±3.09E+00－	2.02E+01±1.15E+01
－	7	6	7	6	
＋	0	0	0	0	
≈	0	1	0	1	

"－", "＋", and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of MONES, respectively

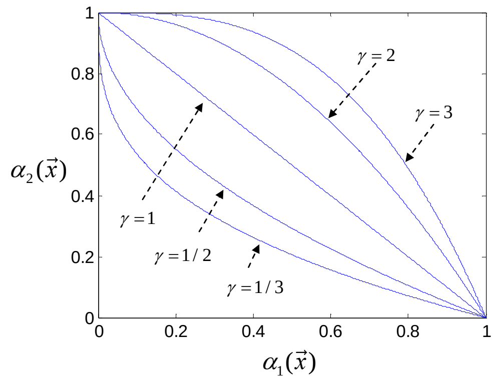


Fig. 24. The Pareto fronts with five different γ values for equation (21) with $0 \le x_1 \le 1$.

On the other hand, after transforming a NES into a single-objective optimization problems (i.e., equation (5) or equation (6)), the transformed problem is essentially similar to a multimodal optimal problems since both of them may include multiple optimal solutions. Thus, the niching methods [44] designed for multimodal optimal problems could also be applicable to locate multiple optimal solutions of a NES in a single run. In order to test the effectiveness of the niching methods for solving NESs, two well-known niching methods have been chosen. One is the niching PSO using a ring topology (denoted as rpso) [45] and the other is the crowding based DE (denoted as CDE) [46]. When applying rpso and CDE to solve a NES, equation (5) is considered as the objective function in this paper.

For the four single-objective optimization based methods, i.e., RepulsionPSO, RepulsionDE, rpso, and CDE, 30 independent runs were conducted on the seven test functions. To make the comparison fair, the population size was set to 100 and the maximum generation number was set to 500 which are the same as those in MONES. For RepulsionDE and CDE, the scaling factor and the crossover control parameter of DE were set to 0.9 and 0.1, respectively. In addition, RepulsionPSO and rpso employ the constricted PSO [47] as the search engine. Note that in RepulsionPSO, the *gbest* constricted PSO has been used. However, in rpso the *lbest* constricted PSO with a ring topology has been adopted.

The NOF-indicator values of RepulsionPSO, RepulsionDE, rpso, CDE, and MONES have been summarized in Table IV. Table IV indicates that MONES performs better than RepulsionPSO and rpso on all the test instances. In addition, MONES has an edge over RepulsionDE and CDE on six test instances and exhibits similar performance with RepulsionDE and CDE on F1. In particular, RepulsionDE and CDE fail to find any of the optimal solutions for F2.

The above comparison suggests that although the single-objective formulation of a NES has the capability to find multiple optimal solutions for some test instances by integrating with the creation of repulsion area or the niching techniques, its performance is less reliable and robust than MONES. Compared with the single-objective formulation, the main superiority of MONES is that the distribution of the optimal solutions of a NES can be controlled in the objective space of the transformed biobjective optimization problem.

G. The Influence of the Types of Pareto Front

In MONES, the Pareto front of the transformed biobjective optimization problem is determined by the location function (i.e., equation (12)), which leads to a linear Pareto front. One may be interested in the influence of other types of Pareto front on the performance of MONES. Actually, the types of Pareto front can be adapted by changing equation (12) into the following equation:

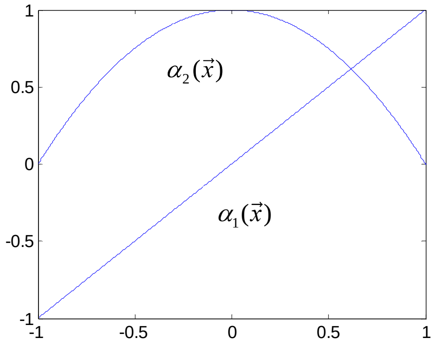


Fig. 25. The relationship between $\alpha_1(\vec{x})$ and $\alpha_2(\vec{x})$ in the objective space for $\gamma = 2$ and $-1 \le x_1 \le 1$.

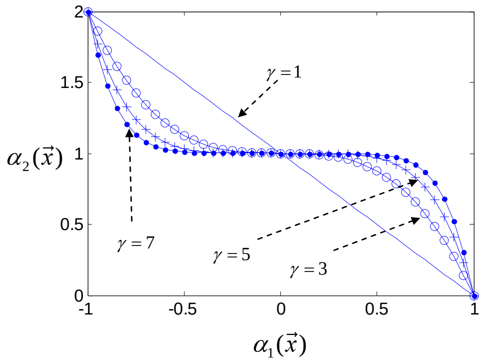


Fig. 26. The Pareto fronts in the case of γ =1, 3, 5, and 7 for equation (21) with $-1 \le x_1 \le 1$.

TABLE V
EXPERIMENTAL RESULTS OF MONES WITH FOUR DIFFERENT γ ON SEVEN TEST INSTANCES. "MEAN NOF" AND "STD DEV" INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE NUMBER OF THE OPTIMAL SOLUTIONS FOUND IN 30 RUNS, RESPECTIVELY. WILCOXON'S RANK SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED BETWEEN MONES WITH $\gamma = 1$ AND EACH OF MONES WITH $\gamma = 3$, MONES WITH $\gamma = 5$, and MONES WITH $\gamma = 7$.

Test Instance	$\gamma = 3$ Mean NOF±Std Dev	$\gamma = 5$ Mean NOF±Std Dev	$\gamma = 7$ Mean NOF±Std Dev	$\gamma = 1$ Mean NOF±Std Dev
F1	2.00E+00±0.00E+00≈	2.00E+00±0.00E+00≈	2.00E+00±0.00E+00≈	2.00E+00±0.00E+00
F2	1.77E+00±4.31E-01−	1.23E+00±4.30E-01−	1.23E+00±4.30E-01−	2.00E+00±0.00E+00
F3	8.83E+00±8.74E-01−	6.80E+00±8.87E-01−	6.13E+00±6.29E-01−	1.10E+01±0.00E+00
F4	1.04E+01±1.06E+00−	7.50E+00±1.04E+00−	6.17E+00±1.14E+00−	1.41E+01±1.16E+00
F5	3.51E+01±7.72E+00≈	2.52E+01±1.03E+01−	1.77E+01±1.16E+01−	3.73E+01±7.05E+00
F6	5.89E+01±7.05E+00−	4.07E+01±5.37E+00−	3.24E+01±5.25E+00−	7.31E+01±3.81E+00
F7	1.60E+01±1.13E+01≈	1.36E+01±1.03E+01−	1.13E+01±7.83E+00−	2.02E+01±1.15E+01
−	4	6	6	
+	0	0	0	
≈	3	1	1	

"−", "+", and "≈" denote that the performance of the corresponding algorithm is worse than, better than, and similar to that of MONES with $\gamma = 1$ respectively.

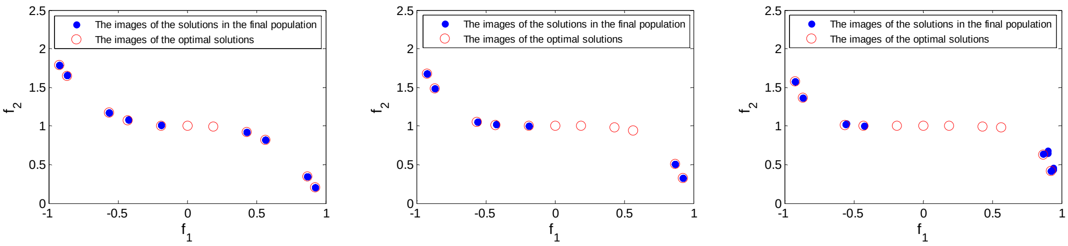
$$\begin{cases} \text{minimize} \quad \alpha_1(\vec{x}) = x_1 \\ \text{minimize} \quad \alpha_2(\vec{x}) = 1 - x_1^{\gamma} \end{cases} \tag{21}$$

where $\gamma > 0$ is a coefficient.

With respect to equation (21), if $x_1 \ge 0$, $\alpha_1(\vec{x})$ is a strictly monotone increasing function and $\alpha_2(\vec{x})$ is a strictly monotone decreasing function. Consequently, $\alpha_1(\vec{x})$ and $\alpha_2(\vec{x})$ conflict with each other, and each value of x_1 is an optimal solution of equation (21). Moreover, if $0 < \gamma \le 1$, the Pareto front is convex and if $\gamma > 1$, the Pareto front is concave [48]. Fig. 24 plots the Pareto fronts with five different α values when $0 \le x_1 \le 1$.

However, it is necessary to note that the search range of x_i $(i \in \{1, \ldots, D\})$ is [-1,1] in this paper. If $-1 \le x_1 < 0$, the following three cases will occur according to the value of γ:

1) γ is a fraction number (such as 1/2 and 17/5): in this case x_1^{γ} may be a complex number (for example, $(-0.5)^{1/2}$ and $(-0.8)^{17/5}$ are complex numbers), which inevitably causes an algorithm to have difficulty in finding the Pareto optimal solutions of the transformed problem.



(a) $\gamma = 3$ (b) $\gamma = 5$ (c) $\gamma = 7$

Fig. 27. The images of the solutions of the final population in a typical run and the images of the 11 optimal solutions in the objective space in the case of γ =3, 5, and 7 for F3.

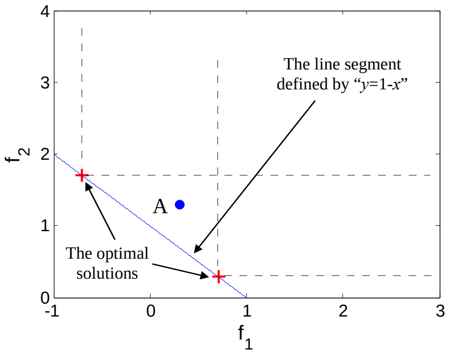


Fig. 28. An example to illustrate the limitation of MONES. "+" represents the optimal solutions and "." represents an unreasonable solution.

2) γ is a positive even number (such as 2, 4, and 6): in this case $\alpha_1(\vec{x})$ does not conflict with $\alpha_2(\vec{x})$. Fig. 25 gives an example of $\gamma = 2$. As shown in Fig. 25, both $\alpha_1(\vec{x})$ and $\alpha_2(\vec{x})$ are strictly monotone increasing functions for $-1 \le x_1 < 0$. Note that if $\alpha_1(\vec{x})$ does not conflict with $\alpha_2(\vec{x})$, Theorem 1 in Section IV-A which reflects the relationship between a NES and the transformed problem does not hold.

3) γ is a positive odd number (such as 1, 3, and 5): in this case $\alpha_1(\vec{x})$ conflicts with $\alpha_2(\vec{x})$.

According to the characteristics of the above three cases, we can conclude that the third case is appropriate for MONES, since it ensures that Theorem 1 holds. Fig. 26 depicts the Pareto fronts in the case of γ =1, 3, 5, and 7 for equation (21) with $-1 \le x_1 \le 1$. From Fig. 26, it is clear that the Pareto front in the objective space is nonlinear when γ is a positive odd number larger than 1. Moreover, some segments of the Pareto front are almost horizontal or vertical with the increase of γ, which poses a great challenge to MOEAs for approximating them [49].

In order to investigate the effect of the parameter γ on the performance of MONES, we replaced equation (12) with equation (21) and tested MONES with four different γ: 1, 3, 5, and 7. It is necessary to emphasize that all the optimal solutions of a NES are located on the nonlinear Pareto front for MONES with $\gamma = 3$, MONES with $\gamma = 5$, and MONES with $\gamma = 7$. Table V summarizes the NOF-indicator values of the seven test instances. The parameter settings are the same as those introduced in Section V-D.

One of the first observations from Table V is that in terms of the NOF indicator, MONES with $\gamma = 1$ outperforms MONES with $\gamma = 3$, MONES with $\gamma = 5$, and MONES with $\gamma = 7$ on four, six, and six test instances, respectively. However, the three competitors cannot surpass MONES with $\gamma = 1$ even on one test instance. In the case of $\gamma = 3$, 5, and 7, the corresponding algorithms lose some optimal solutions located on the segments of the Pareto front which are almost horizontal or vertical. We choose F3 as an example to explain this. Fig. 27 shows the images of the solutions of the final population in a typical run and the images of the 11 optimal solutions of F3 in the objective space, in the case of $\gamma = 3$, 5, and 7. From Fig. 27, it is evident that some optimal solutions of F3, the images of which are located on the almost horizontal segments of the Pareto front, cannot be found.

In addition, as can be seen from Table V, the performance of MONES gradually degrades with the increase of γ. Fig. 27 also verifies this. In Fig. 27, the number of the optimal solutions found by MONES constantly decreases with the increase of γ. It is not difficult to understand since the larger the γ value the longer the segments of the Pareto front which are almost horizontal and vertical [49].

In summary, the value of the parameter γ in equation (21) can be set to a positive odd number. If γ is a positive odd number larger than 1, the Pareto front of the transformed problem will become nonlinear. The experimental results have demonstrated that the nonlinear Pareto fronts have an effect impact on the performance of MONES, compared against the linear Pareto front (i.e., $\gamma = 1$). In the community of evolutionary multiobjective optimization, in order to compare the performance of different MOEAs, the Pareto front usually exhibits complicated nonlinear characteristics. However, the main aim of this paper is to present a simple and effective transformation technique for solving a NES and, consequently, the linear Pareto front is recommended.

H. Limitation

To the best of our knowledge, CA is the first attempt to introduce multiobjectivity for the solution of NESs. In this paper, we try to design a different multiobjective transformation called MONES for solving NESs.

A NES may include several optimal solutions. According to Theorem 2 introduced in Section IV-A, the images of these optimal solutions will be discretely distributed on the line segment defined by "y=1-x" in the objective space of the transformed biobjective optimization problem. It is likely that some individuals in the population, which are far away from the optimal solutions, are nondominated with the optimal solutions. As a result, the final population of MONES may contain such unreasonable solutions other than the optimal solutions when the evolution halts. Fig. 28 gives an example to illustrate the above limitation of MONES. Suppose that there are two optimal solutions of a NES in Fig. 28, the images of which are discretely located on the line segment defined by "y=1-x". As shown in Fig. 28, individual A is not an optimal solution. However, individual A may be maintained in the final population of MONES since it is nondominated with the two optimal solutions.

It is necessary to point out that in this paper the above limitation of MONES has not been observed in all the experiments on the seven test functions, while it exists theoretically. To address the above limitation, a simple method is introduced as follows. Firstly, all the individuals in the final population are used to evaluate the objective function in equation (5). If the objective function value of an individual is larger than a predefined threshold value, then it should be eliminated.

The reason of the above limitation is due to the fact that the images of the optimal solutions are discrete in the objective space for some NESs. If a NES involves infinite optimal solutions and if the images of the optimal solutions are continuously distributed on the Pareto front (such as F5-F7), the above limitation of MONES does not exist, since all the solutions other than the optimal solutions will be Pareto dominated by the optimal solutions.

VI. Conclusion

Nonlinear equation systems (NESs) may have multiple optimal solutions. During the past decade, evolutionary algorithms (EAs) have attracted much attention to solve NESs. When using EAs to solve NESs, the transformation technique, the aim of which is to transform a NES into a kind of optimization problem, plays a critical role.

This paper presents a simple and generic transformation technique based on multiobjective optimization for NESs, called MONES. It transforms a NES into a biobjective optimization problem. Afterward, the transformed problem can be directly optimized by the current MOEAs. Due to the fact that CA [36] and MONES belong to the same kind of transformation techniques, the principal differences between them have been analyzed in depth and extensive experiments have been carried out to compare their performance. In order to make the comparison fair, NSGA-II [14] has been combined with CA and MONES in a straightforward way. The experimental results suggest that compared with CA, MONES exhibits a superior performance on seven test instances in terms of two performance indicators, and that MONES has the capability to simultaneously locate multiple optimal solutions of a NES in a single run. The effectiveness of MONES has been further verified by comparing with four single-objective optimization based methods. Moreover, we have experimentally studied the influence of the types of Pareto front on the performance of MONES and discussed the limitation of MONES.

It is necessary to emphasize that in order to test the effectiveness of the transformation technique, NSGA-II adopted in this paper have not been fine-tuned for the test suite. In the future, we will improve the baseline NSGA-II or propose more powerful MOEAs and combine them with MONES to solve NESs with more complex characteristics. In addition, we are also considering the possibility of generalizing MONES to solve the differential equation systems.

The source code of MONES is written in MATLAB and can be obtained from the corresponding author (Yong Wang) upon request.

References

[1] C. Patrascioiu and C. Marinoiu, "The applications of the non-linear equations systems algorithms for the heat transfer processes," in *Proceedings of the 12th WSEAS international conference on Mathematical methods, computational techniques and intelligent systems*, 2010, pp. 30-35.

[2] C. L. Collins, "Forward kinematics of planar parallel manipulators in the Clifford algebra of P^2," *Mechanism and Machine Theory*, vol. 37, no. 8, pp. 799-813, 2002.

[3] I. Amaya, J. Cruz, and R. Correa, "Solution of the mathematical model of a nonlinear direct current circuit using particle swarm optimization," *Dyna*, vol. 79, no. 172, pp. 77-84, 2012.

[4] C. Gu, T. Qu, X, Li, and Z. Han, "AC Losses in HTS tapes and devices with transport current solved through the resistivity-adaption algorithm," *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 2, 8201708, 2013.

[5] N. Henderson, W. F. Sacco, N. Barufatti, and M. M. Ali, "Calculation of critical points of thermodynamic mixtures with differential evolution algorithms," *Industrial & Engineering Chemistry Research*, vol. 49, no. 4, pp. 1872-1882, 2010.

[6] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Math. Comput.,* vol. 19, no. 92, pp. 577-593, 1965.

[7] A. Friedlander, M. A. Gomes-Ruggiero, D. N. Kozakevich, J. M. Martínez, and S. A. Santos, "Solving nonlinear systems of equations by means of Quasi-Newton methods with a nonmonotone strategy," *Optimization Methods and Software*, vol. 8, no. 1, pp. 25-51, 1997.

[8] M. D. González-Lima and F. M. de Oca, "A Newton-like method for nonlinear system of equations," *Numer. Algor.*, vol. 52, no. 3, pp. 479-506, 2009.

[9] G. C. V. Ramadas and E. M. G. P. Fernandes, "Solving nonlinear equations by a Tabu search strategy," in *Proceedings of the 11th*

International Conference on Computational and Mathematical Methods in Science and Engineering, 2011, pp. 1578-1589.

[10] 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.

[11] K. Deb, "Multi-objective optimization using evolutionary algorithms," Baffins Lane, Chichester, Wiley, 2001.

[12] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149-172, 2000.

[13] E. Zitzler, M. Laumannns, and L Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. of the EUROGEN 2001—Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problem*, 2001, pp. 95-100.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.

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

[16] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol. Comput.*, vol. 20, no. 1, pp. 27-62, 2012.

[17] L. T. Bui, H. A. Abbass, and J. Branke, "Multiobjective optimization for dynamic environments," in *Proc. CEC*, 2005, pp. 2349-2356.

[18] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658-675, 2006.

[19] Z. Wu and L. Kang, "A fast and elitist parallel evolutionary algorithm for solving systems of non-linear equations," in *Proc. CEC*, 2003, pp. 1026-1028.

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

[21] A. Noriega, J. L. Cortizo, E. Rodriguez, R. Vijande, and J. M. Sierra, "A new method to approximate the field of movements of 1-DOF linkages with lower-pairs," *Meccanica*, vol. 45, no. 5, pp. 681-692, 2010.

[22] J. Wang, "Immune genetic algorithm for solving nonlinear equations," in *Proceedings of 2011 International Conference on Mechatronic Science, Electric Engineering and Computer*, 2011, pp. 2094-2097.

[23] 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, Article ID 453812, 13 pages, http://dx.doi.org/10.1155/2013/453812

[24] J. Wu, Z. Cui, and J. Liu "Using hybrid social emotional optimization algorithm with metropolis rule to solve nonlinear equations," in *10th IEEE International Conference on Cognitive Informatics & Cognitive Computing*, 2011, pp. 405-411.

[25] 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.

[26] 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.

[27] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39-43.

[28] Y. Mo, H. Liu, and Q. Wang, "Conjugate direction particle swarm optimization solving systems of nonlinear equations," *Computers and Mathematics with Applications*, vol. 57, no. 11-12, pp. 1877-1882, 2009.

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

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

[31] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Solving systems of unconstrained equations using particle swarm optimization," in *Proceedings of International Conference on Systems, Man and Cybernetics*, 2002, pp. 6-9.

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

[33] 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.

[34] 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.

[35] A. Pourrajabian, R. Ebrahimi, M. Mirzaei, and M. Shams, "Applying genetic algorithms for solving nonlinear algebraic equations," *Appl. Math. Comput*., vol. 219, no. 24, pp. 11483-11494, 2013.

[36] C. Grosan and A. Abraham, "A new approach for solving nonlinear equation systems," *IEEE Transactions on Systems Man and Cybernetics - Part A*, vol. 38, no. 3, pp. 698-714, 2008.

[37] S. Effati and A. R. Nazemi, "A new method for solving a system of the nonlinear equations," *Appl. Math. Comput*., vol. 168, no. 2, pp. 877-894, 2005.

[38] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms for many-objective optimisation," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474-494, 2013.

[39] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284-302, 2009.

[40] J. Verschelde, P. Verlinden, and R. Cools, "Homotopies exploiting Newton polytopes for solving sparse polynomial systems," *SIAM J. Numer. Anal.*, vol. 31, no. 3, pp. 915–930, 1994.

[41] A. P. Morgan, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[42] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41-63, 2008.

[43] R. Storn and K. V. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Opt.,* vol. 11, no. 4, pp. 341–359, 1997.

[44] S. Mahfoud, "Niching methods for genetic algorithms," Doctoral dissertation, Comput. Sci., Univ. Illinois, Urbana, IL, 1995.

[45] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150-169, 2010.

[46] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. CEC*, 2004, pp. 1382-1389.

[47] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58-73, 2002.

[48] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no.3, pp. 205-230, 1999.

[49] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina, "Pareto-adaptive ε-dominance," *Evol. Comput.*, vol. 15, no. 4, pp. 493-517, 2007.