Finding Multiple Roots of Nonlinear Equation Systems via A Repulsion-based Adaptive Differential Evolution

Wenyin Gong, Yong Wang, *Member, IEEE*, Zhihua Cai, and Ling Wang

***Abstract*—Finding multiple roots of nonlinear equation systems in a single run is one of the most important challenges in numerical computation. We tackle this challenging task by combining the strengths of the repulsion technique, diversity preservation mechanism, and adaptive parameter control. Firstly, the repulsion technique motivates the population to find new roots by repulsing the regions surrounding the previously found roots. However, to find as many roots as possible, algorithm designers need to address a key issue: how to maintain the diversity of the population. To this end, the diversity preservation mechanism is integrated into our approach, which consists of the neighborhood mutation and the crowding selection. In addition, we further improve the performance by incorporating the adaptive parameter control, with the aim of enhancing the search ability and remedying the trial-and-error tuning of the parameters of differential evolution for different problems. By assembling the above three aspects together, we propose a repulsion-based adaptive differential evolution, called RADE, for finding multiple roots of nonlinear equation systems in a single run. To evaluate the performance of RADE, thirty nonlinear equation systems with diverse features are chosen from the literature as the test suite. Experimental results reveal that RADE is able to find multiple roots simultaneously in a single run on all the test problems. Moreover, RADE is capable of providing better results than the compared methods in terms of both root rate and success rate.**

***Index Terms*—Nonlinear equation systems, repulsion technique, diversity preservation mechanism, adaptive parameter control, differential evolution.**

I. Introduction

NUMEROUS real-world applications can be modeled as nonlinear equation systems (NESs) [1], [2]. In 1998, Fields Medalist Steve Smale listed 18 challenging problems for the twenty-first century including "Does P = NP?" [3]. Among them, the following three challenging problems are related to NESs:

- ***Problem 4***: Integer zeros of a polynomial
- ***Problem 8***: Introduction of dynamics into economic theory
- ***Problem 17***: Solving polynomial equations

W. Gong and Z. Cai are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China. (Email: wygong@cug.edu.cn; zhcai@cug.edu.cn)

Y. Wang is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK. (Email: ywang@csu.edu.cn)

L. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China. (Email: wangling@tsinghua.edu.cn)

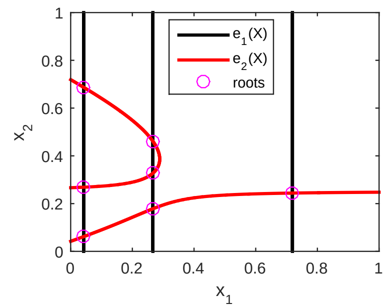


Fig. 1. Illustration of the eighth test problem (F08) in the supplementary material and its seven roots.

Therefore, finding the roots of NESs is not only of great significance for solving practical problems, but also one of the core problems of mathematics.

Without loss of generality, a NES can be defined as follows:

$$\begin{cases} e_1(\mathbf{x}) & = & 0 \\ & \vdots & \\ e_m(\mathbf{x}) & = & 0 \end{cases} \tag{1}$$

where m is the number of equations, $\mathbf{x} = (x_1, \dots, x_n)$ is an n-dimensional decision vector, n is the number of decision variables, $\mathbf{x} \in \mathbb{S}$, and $\mathbb{S} \subseteq \mathbb{R}^n$ denotes the search space. Usually

$$\mathbb{S} = \prod_{j=1}^{n} [\underline{x}_j, \overline{x}_j] \tag{2}$$

where $j = 1, \dots, n$, and $\underline{x}_j$ and $\overline{x}_j$ are the lower and upper bounds of x_j, respectively.

If $\forall i \in \{1, \dots, m\}$, $e_i(\mathbf{x}^*) = 0$, then $\mathbf{x}^*$ is said to be a root (or an optimal solution) of a NES. A NES may have multiple roots, especially when $n \geq m$. For example, taking the eighth test problem (F08) in the supplementary material into account, Fig. 1 depicts its seven roots. In F08, there are two nonlinear equations ($e_1(\mathbf{x})$ and $e_2(\mathbf{x})$) and two decision variables (x_1 and x_2), i.e., $n = m = 2$. In real-world applications, each root may be equally important; therefore, the main task of solving NESs is to simultaneously find these roots in a single run, which is nondeterministic polynomial-time hard and thus poses a grand challenge in numerical computation [4].

The *repulsion* technique (also known as the *polarization* technique) [5], [6], [7] has great potential to guide an optimization algorithm toward new roots by creating repulsion areas around the roots previously identified. However, to find as many roots as possible, the diversity and search ability of the optimization algorithm also play crucial roles other than the repulsion technique. On the one hand, after finding a root, if the diversity of the optimization algorithm is poor and if all the solutions of the optimization algorithm already gather in a small region, it is very hard to find new roots in the subsequent search. On the other hand, the optimization algorithm with weak search ability may not be capable of probing unexplored yet promising regions. Therefore, the repulsion technique should be equipped with an optimization algorithm with good diversity and strong search ability when solving NESs, which has not yet been systematically investigated in current research.

Based on this analysis, this paper makes the first attempt to combine the repulsion technique, diversity preservation mechanism, and adaptive parameter control in an effective way, with the purpose of developing a powerful method for finding multiple roots of NESs. We call the proposed method a repulsion-based adaptive differential evolution (RADE). In RADE, differential evolution (DE) [8] serves as the optimization algorithm, and the aim of the repulsion technique is to assist DE in finding multiple roots of NESs simultaneously in a single run. In addition, the diversity preservation mechanism is employed to diversify the population of DE, in which each individual implements mutation and crossover with its neighbors to generate a trial vector, and the trial vector is compared with the nearest individual in the parent population for survival. Moreover, the adaptive parameter control is used to automatically adapt the control parameters of DE to appropriate values, thereby enhancing the search ability for solving different NESs.

The main contributions of this paper can be summarized as follows:

- As aforementioned, three aspects are elaborated in RADE to solve NESs, i.e., the repulsion technique—avoiding repeated search in the regions previously explored; the diversity preservation mechanism—maintaining the diversity of the population and increasing the possibility to find new roots; and the adaptive parameter control—enhancing the search ability without trial-and-error parameter settings. As a result, RADE can be considered as an effective alternative for solving NESs.
- Thirty NESs were chosen from the literature as the test suite, which can be used as the benchmark test problems to evaluate the performance of different methods. Experimental results verify that RADE can not only consistently find multiple roots of different NESs in a single run, but also yield better results than the compared methods with respect to both root rate and success rate.
- The effectiveness of the three important aspects of RADE has been experimentally investigated. Furthermore, we have empirically discussed the influence of other repulsion techniques, other diversity preservation mechanisms, other adaptive parameter controls, other mutation operators, and different parameter settings on the performance of RADE.

The rest of this paper is organized as follows. Section II introduces the background, including the repulsion techniques and the classical DE. Section III discusses the related work and motivation. In Section IV, the proposed RADE is described in detail. The experimental studies and comprehensive discussions are given in Section V. Finally, Section VI concludes this paper.

II. Background

A. Repulsion Techniques for NESs

When solving a NES by an optimization algorithm, it is usually transformed into a *minimization* problem as follows:

$$\text{minimize} \quad f(\mathbf{x}) = \sum_{i=1}^{m} e_i^2(\mathbf{x}) \tag{3}$$

or

$$\text{minimize} \quad f(\mathbf{x}) = \sum_{i=1}^{m} |e_i(\mathbf{x})| \tag{4}$$

Afterward, to detect the roots of a NES is equivalent to finding the global minimizers of the transformed optimization problem in (3) or (4).

In the NES literature, several repulsion techniques have been developed to find multiple roots [5]. The basic idea behind the repulsion technique is that a repulsion function is designed based on $f(\mathbf{x})$ to create repulsion areas around the roots previously found so that these roots will not be the global minimizers anymore. Therefore, the repulsion technique aims at motivating the optimization algorithm to explore new search regions and to search for new roots. Next, we briefly introduce two representative (i.e., *multiplicative* and *additive*) techniques which will be utilized in this paper.

- *Multiplicative repulsion technique*: Pourjafari and Mojallali [7] presented a multiplicative repulsion function as follows:

$$\text{minimize} \quad R(\mathbf{x}) = \big(f(\mathbf{x}) + \epsilon\big) \prod_{j=1}^{K} |\coth(\alpha\delta_j)| \tag{5}$$

where

$$\delta_j = \| \mathbf{x} - \mathbf{x}_j^* \| \tag{6}$$

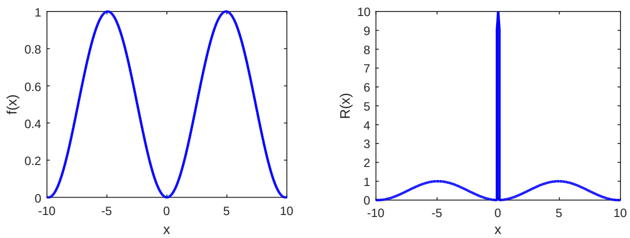
K is the number of roots that have been found, $\mathbf{x}_j^*$ is the jth root, δ_j is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_j^*$, $\alpha \geq 1$ is a density factor to adjust the radius of the repulsion regions, and ϵ is a very small positive constant or equal to 0. In this paper, ϵ = 1E-10 and $\alpha = 10$ as suggested in [7].

In [9], another multiplicative repulsion function based on the error function "erf" is presented:

$$\text{minimize} \quad R(\mathbf{x}) = \big(f(\mathbf{x}) + \epsilon\big) \prod_{j=1}^{K} \zeta_{\rho'}(\gamma, \delta_j) \tag{7}$$

where

$$\zeta_{\rho'}(\gamma, \delta_j) = \begin{cases} |\text{erf}(\gamma\delta_j)|^{-1}, & \text{if } \delta_j \leq \rho' \\ 1, & \text{otherwise} \end{cases} \tag{8}$$



(a) (b)

Fig. 2. Comparison between the original objective function $f(x)$ and the repulsion function $R(x)$: (a) the original objective function $f(x) = \sin^2(x/\pi), x \in [-10, 10]$; (b) the repulsion function $R(x)$. If one root $x = 0$ was obtained, then $R(x)$ creates a repulsion area around $x = 0$ to penalize the solutions in this repulsion area. As shown in Fig. 2(b), $x = 0$ is no longer a global minimizer because $R(x) \gg 0$ around $x = 0$.

$\gamma > 0$ scales the penalty, and ρ' adjusts the radius of the repulsion regions. In this paper, $\gamma = 0.1$ and $\rho' = 0.1 \min_{i=1}^{n}(\overline{x}_i - \underline{x}_i)$ as suggested in [9].

- *Additive repulsion technique*: In [5], an additive repulsion function is formulated as follows:

$$\text{minimize} \quad R(\mathbf{x}) = f(\mathbf{x}) + \beta \sum_{j=1}^{K} \mathrm{e}^{-\delta_j} \chi_\rho(\delta_j) \tag{9}$$

where

$$\chi_\rho(\delta_j) = \begin{cases} 1, & \text{if } \delta_j \leq \rho \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$\chi_\rho(\delta_j)$ is the characteristic function, ρ is a small constant to adjust the radius of the repulsion regions, and β is a large constant to control the penalty scale. In this paper, $\rho = 0.01$ and $\beta = 1000$ as suggested in [5].

Fig. 2 gives an example to explain the principle of the repulsion technique. In this example, the repulsion technique in (9) with $\rho = 0.1$ and $\beta = 10$ is used. The original objective function is: $f(x) = \sin^2(x/\pi), x \in [-10, 10]$. It is clear from Fig. 2(a) that there are three roots with $f(x) = 0$, i.e., $x = -10$, $x = 0$, and $x = 10$. Suppose that one root (e.g., $x = 0$) was obtained. Afterward, the original objective function is modified according to the repulsion function. As a result, $x = 0$ is not a global minimizer anymore. As shown in Fig. 2(b), the solutions in the repulsion area (i.e., the area around $x = 0$) are penalized while other solutions outside the repulsion area are not affected, which signifies that the repulsion area is unpromising.

There are also other similar repulsion techniques presented in the literature (see, for example, [6], [10], [11], and [12]). In this paper, we only use the above-mentioned three repulsion techniques; other repulsion techniques are not described to save space. Note that different repulsion techniques have different features, Section V-E1 will empirically evaluate the influence of the above-mentioned three repulsion techniques.

B. Differential Evolution (DE)

DE is a simple yet powerful evolutionary algorithm (EA) for numerical optimization [8]. Similar to other EAs, DE contains four main steps, i.e., *initialization*, *mutation*, *crossover*, and *selection*. Initially, the population is randomly generated in the search space. After initialization, DE generates the trial vectors by making use of the mutation and crossover. Then, the trial vectors are evaluated by the fitness function. Finally, in the selection, each trial vector is compared with its corresponding parent, and replaces the parent only if it has an equal or better fitness value. DE repeats the mutation, crossover, and selection until a predefined termination criterion is satisfied. More details of DE can be found in [13]. Originally, DE was developed for single-objective optimization problems. Recent advances have successfully adapted DE to other kinds of optimization problems, such as dynamic optimization [14], constrained optimization [15], multiobjective optimization [16], and multimodal optimization [17].

III. Related Work and Motivation

A. Related Work

As reviewed in [4], [18], and [19], several different classical methods have been proposed to enclose multiple roots of NESs, which can be classified as:

- *Newton and quasi-Newton type methods*: These methods may obtain super-linear convergence when the initial guess is close to one of the roots, e.g., [20] and [21].
- *Homotopy continuation (embedding) methods*: In these methods, a hard problem is firstly transformed into a much simpler one, and then this simpler problem gradually deforms into the original one, e.g., [22] and [23].
- *Interval-Newton methods*: The classical Newton-like iterative methods are applied to the interval variables in the interval-Newton methods, e.g., [24] and [25].
- *Deterministic branch-and-bound methods*: These methods transform a NES into a global optimization problem, and then the transformed optimization problem is solved by the branch-and-bound methods, e.g., [18] and [26].

Besides the classical methods, there are also some stochastic methods to find multiple roots for NESs. Generally, these methods can be classified into three categories:

- *Clustering-based methods*: In the literature, the clustering techniques are used to gather similar solutions into different clusters to locate different roots of NESs. For example, a clustering method is used at the exact search phase in [7]. In addition, the clustering-based Minfider is used to find multiple roots in [27] and the fuzzy clustering means is employed in [28].
- *Multiobjective optimization-based methods*: Similar to NESs, multiobjective optimization problems also have multiple Pareto optimal solutions. Recognizing this similarity, some researchers have tried to locate multiple roots of NESs based on multiobjective optimization. For example, a NES is transformed into an m-objective optimization problem in [29]. Additionally, a bi-objective transformation technique (MONES) is presented in [30] and Qin *et al.* [31] presented a $(n + 1)$-objective transformation technique for NESs.
- *Repulsion-based methods*: As mentioned in Section II-A, the repulsion techniques are able to construct the areas

of repulsion around the found roots. Based on the repulsion techniques, different methods are designed to find multiple roots of NESs, such as continuous GRASP [5], two-phase root-finder using IWO [7], improved harmony search algorithm [9], multi-start simulated annealing [10], Nelder-Mead based repulsion algorithm [12], and biased random-key GA [32].

B. Motivation

As pointed out in [4] and [19], the classical methods have some weaknesses. For example, the performance of the Newton and quasi-Newton type methods is highly sensitive to the initial guess; the embedding methods cannot directly deal with variable bounds and inequality constraints; the interval-Newton methods are computationally expensive; and the deterministic branch-and-bound methods need the specification of bounded intervals. More importantly, the classical methods only focus on one of the roots in a single run.

With respect to the clustering-based methods, it is difficult to determinate the number of clusters if no priori information is available for the problem at hand. Moreover, the data set to be clustered and the clustering period are also problem-dependent [28].

For the multiobjective optimization-based methods, the approaches in [29] and [31] may suffer from the "curse of dimensionality" due to the fact that the number of objectives in them is related to m or n. With the drastic increase of m or n, they will transform a NES into a *many*-objective optimization problem, which is a grand challenge in the field of evolutionary computation [33]. Additionally, MONES [30] has the following drawback: when several roots of a NES have the same values in the first decision variable x_1 (see, for example, F08 in Fig. 1), MONES may lose some of these roots.

As far as the repulsion-based methods are concerned, most of them need to restart many times so as to obtain different roots [5], [7], [9], [10], [32]. Under this condition, some useful information discovered previously may be lost in a new run. For example, some solutions in [7] and [32] may lie within the attraction basins of certain roots in the current run. However, such useful information is neglected unreasonably in a new run because of the population re-initialization, thus leading to inefficiency.

The reason why the restart is frequently used in the repulsion-based methods seems straightforward: if without restart, the diversity of the population will gradually decrease after one or several roots have been identified due to the lack of diversity preservation. Therefore, it is believed that if the diversity of the population can be maintained, we can eliminate the restart while keeping the advantage of the repulsion technique. Motivated by the above considerations, we present a repulsion-based adaptive DE in this paper, named RADE. In RADE, the repulsion technique is exploited to assist DE in locating different roots. In addition, the diversity preservation mechanism, consisting of the neighborhood mutation and the crowding selection, is used to diversify the population. In principle, after a new root has been found, the repulsion function should be modified according to the information of this root, which means that the repulsion technique will construct a sequence of minimization problems based on the sequentially modified repulsion function. Note that to locate a root of a specific minimization problem, the search ability of an optimization algorithm is vital. To this end, the adaptive parameter control is implemented to enhance the search ability of DE and to avoid the trial-and-error tuning of DE's parameters. The synergy of the repulsion technique, the diversity preservation mechanism, and the adaptive parameter control provides an effective way to find multiple roots of NESs simultaneously in a single run: when some roots have been detected, the population will continue to search new regions and find undetected roots because of the repulsion property, good diversity, and strong search ability. To the best of our knowledge, it is the first attempt to integrate these three aspects together to solve NESs.

Algorithm 1: Archive updating

Input: Individual $\mathbf{x}$, archive $\mathcal{A}$, accuracy level θ, distance radius τ_d, and maximum archive size $\max_s$

Output: The updated $\mathcal{A}$ and the archive size s

if $f(\mathbf{x}) < \theta$ **then** // $\mathbf{x}$ is a root
    if $s = 0$ **then** // $\mathcal{A}$ is empty
        $\mathcal{A} = \mathcal{A} \cup \mathbf{x}$;
        $s = s + 1$;
    else
        Find the closest solution $\mathbf{x}' \in \mathcal{A}$ to $\mathbf{x}$ in the decision space;
        if $\parallel \mathbf{x} - \mathbf{x}' \parallel < \tau_d$ **then** // $\mathbf{x}$ and $\mathbf{x}'$ are too close
            if $f(\mathbf{x}) < f(\mathbf{x}')$ **then**
                $\mathbf{x}' = \mathbf{x}$;
        else
            if $s < \max_s$ **then**
                $\mathcal{A} = \mathcal{A} \cup \mathbf{x}$;
                $s = s + 1$;
            else // $\mathcal{A}$ is full
                if $f(\mathbf{x}) < f(\mathbf{x}')$ **then**
                    $\mathbf{x}' = \mathbf{x}$;

IV. Proposed Approach

In this section, a repulsion-based adaptive DE (RADE) is proposed to find multiple roots of NESs. In RADE, DE serves as the search engine. The details of RADE are described in the following subsections.

A. Repulsion in RADE

After an initial population is generated, RADE will search for the first root of a NES by minimizing $f(\mathbf{x})$ in (3) or (4). Once the first root has been located, RADE will change to optimize $R(\mathbf{x})$ in (5), (7), or (9) during the rest of evolution. Therefore, for each individual, its fitness function $F(\mathbf{x})$ can be defined as follows:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{if no root has been found} \\ R(\mathbf{x}), & \text{otherwise} \end{cases} \tag{11}$$

It can be seen from (5), (7), and (9) that when computing $R(\mathbf{x})$, the information of all the roots previously found should be used. Therefore, we store all the obtained roots into a predefined archive $\mathcal{A}$. In RADE, if a new root is identified, we will update $\mathcal{A}$ according to Algorithm 1. If this new root can be added into $\mathcal{A}$, then we will implement the following processes: 1) modifying $R(\mathbf{x})$ by incorporating the information of this new root; and 2) re-evaluating all the individuals in the population based on the modified $R(\mathbf{x})$. In Algorithm 1, θ is a very small positive number to judge whether an individual is a root or not, τ_d is a distance radius to avoid redundant roots in $\mathcal{A}$, and $\max_s$ is the maximum archive size[1].

Over the course of evolution, $R(\mathbf{x})$ will be sequentially modified with the information of the newly found roots in $\mathcal{A}$, and any individual lying within one of the repulsion areas determined by $R(\mathbf{x})$ will be penalized as introduced in Section II-A and Fig. 2.

B. Diversity Preservation Mechanism

Solving multimodal optimization problems has attracted increasing interest in the field of evolutionary computation [34]. Similar to NESs, multimodal optimization problems also involve multiple optimal solutions. As pointed out by Qu *et al.* [35], diversity is one of the major issues to locate multiple optimal solutions of multimodal optimization problems in a single run. It is interesting to note that when transforming a NES into (3) or (4), either (3) or (4) is essentially equivalent to a multimodal optimization problem. Therefore, the excellent diversity techniques designed for multimodal optimization problems can be readily generalized to deal with NESs. Along this line, we introduce a diversity preservation mechanism inspired by [35], which integrates the neighborhood mutation and the crowding selection with DE.

Suppose that the population $\mathcal{P}$ of DE consists of NP individuals: $\mathbf{x}_1, \ldots, \mathbf{x}_{NP}$. In contrast to the traditional DE mutation, the neighborhood mutation only selects several similar individuals measured by Euclidean distance in the decision space to generate a mutant vector for each individual $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})$ $(i = 1, \ldots, NP)$:

1) For each individual $\mathbf{x}_i$, select ℓ individuals with the smallest Euclidean distances to $\mathbf{x}_i$ in $\mathcal{P}$ to form the subpopulation $subpop_i$;
2) Generate a mutation vector $\mathbf{v}_i = (v_{i,1}, \ldots, v_{i,n})$:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \tag{12}$$

where $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, and $\mathbf{x}_{r3}$ are randomly chosen from $subpop_i$, and F is the scaling factor of DE.

It is obvious that the neighborhood mutation produces a mutant vector within a local area, which enables $\mathcal{P}$ to distribute around the attraction basins of different roots.

After the crossover, a trial vector $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,n})$ is created:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j < CR \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{13}$$

where $rand_j$ is a uniformly distributed random number from $[0, 1]$, j_{rand} is a random integer uniformly generated from $\{1, \cdots, n\}$, and CR is the crossover rate of DE.

In the crowding selection, the trial vector $\mathbf{u}_i$ compares with the closest individual (denoted as $\mathbf{u}'_i$) in $\mathcal{P}$ for survival:

$$\mathbf{u}'_i = \begin{cases} \mathbf{u}_i, & \text{if } F(\mathbf{u}_i) \le F(\mathbf{u}'_i) \\ \mathbf{u}'_i, & \text{otherwise} \end{cases} \tag{14}$$

The crowding selection is able to facilitate multiple convergence since it is carried out between the most similar two individuals.

Overall, the neighborhood mutation and the crowding selection make $\mathcal{P}$ converge toward different roots of a NES by diversifying it.

C. Adaptive Parameter Control

DE has three parameters (i.e., the population size NP, the scaling factor F, and the crossover rate CR) that need to be set by the users. It is noteworthy that the search ability of DE is strongly dependent on the parameter settings [36]. To enhance the search ability and alleviate the tedious tasks of parameter tuning for different problems, F and CR are adaptively controlled based on their successful experience in RADE. We do not develop a new parameter adaptation technique while using the method proposed in SHADE [37], because of its superior performance on IEEE CEC2005 and IEEE CEC2013 benchmarks [37], [38].

In SHADE [37], each individual $\mathbf{x}_i$ $(i = 1, \ldots, NP)$ in the population is associated with a pair of F and CR, denoted as F_i and CR_i. In addition, SHADE also maintains a historical memory with H_m entries, i.e., $M_{F,j}$ and $M_{CR,j}$ $(j = 1, \ldots, H_m)$, for F and CR. Initially, the elements of $M_{F,j}$ and $M_{CR,j}$ $(j = 1, \ldots, H_m)$ are set to 0.5.

At each generation, F_i and CR_i are generated for $\mathbf{x}_i$ as follows:

$$F_i = \text{randc}(M_{F,h_i}, 0.1) \tag{15}$$

$$CR_i = \text{randn}(M_{CR,h_i}, 0.1) \tag{16}$$

where $\text{randc}(\cdot,\cdot)$ is the Cauchy random number, $\text{randn}(\cdot,\cdot)$ is the Gaussian random number, and h_i is an integer randomly selected from $\{1, \ldots, H_m\}$.

During the evolution, if the trial vector is able to successfully replace the parent, then the corresponding F_i and CR_i will be recorded in $\mathbf{S_F}$ and $\mathbf{S_{CR}}$, respectively. At the end of each generation, M_F and M_{CR} are updated as follows:

$$M_{F,k} = \begin{cases} \text{mean}_{\text{WL}}(\mathbf{S_F}) & \text{if } \mathbf{S_F} \ne \emptyset \\ M_{F,k} & \text{otherwise} \end{cases} \tag{17}$$

$$M_{CR,k} = \begin{cases} \text{mean}_{\text{WA}}(\mathbf{S_{CR}}) & \text{if } \mathbf{S_{CR}} \ne \emptyset \\ M_{CR,k} & \text{otherwise} \end{cases} \tag{18}$$

where $\text{mean}_{\text{WL}}(\mathbf{S_F})$ is the weighted Lehmer mean of $\mathbf{S_F}$ and $\text{mean}_{\text{WA}}(\mathbf{S_{CR}})$ is the weighted arithmetic mean of $\mathbf{S_{CR}}$. In (17) and (18), k is an index between 1 and H_m. k is set to 1 at the beginning of evolution and increases by 1 after one generation. If $k > H_m$, then it is reset to 1.

[1]Note that the parameter $\max_s$ is not mandatory. In this paper, it is used in order not to save too many solutions into $\mathcal{A}$.

Algorithm 2: RADE

Input: Control parameters: NP, H_m, and Max_FEs
Output: The final archive $\mathcal{A}$
1 Randomly generate an initial population $\mathcal{P}$, which contains NP individuals: $\mathbf{x}_1, \ldots, \mathbf{x}_{NP}$;
2 Calculate $f(\mathbf{x}_i)$ $(i = 1, \ldots, NP)$ with (3);
3 Set all values in the historical memory $M_{CR,j}$ and $M_{F,j}$ $(j = 1, \ldots, H_m)$ to 0.5;
4 $k = 1$;
5 Set the archive $\mathcal{A} = \emptyset$ and $FEs = NP$;
6 **while** $FEs < Max\_FEs$ **do**
7   **for** $i = 1$ *to* NP **do**
8   Update $\mathcal{A}$ with $\mathbf{x}_i$ using Algorithm 1;
9   Set $\mathbf{S_{CR}} = \emptyset$ and $\mathbf{S_F} = \emptyset$;
10   **for** $i = 1$ *to* NP **do**
11   Implement (15) and (16) to produce F_i and CR_i for $\mathbf{x}_i$;
12   Generate the trial vector $\mathbf{u}_i$ via the neighborhood mutation and crossover in (12) and (13) ;
13   Calculate $f(\mathbf{u}_i)$ with (3);
14         $FEs = FEs + 1$;
15   Find the closest solution $\mathbf{u}'_i$ in the decision space to $\mathbf{u}_i$ in $\mathcal{P}$;
16   Calculate $F(\mathbf{u}'_i)$ and $F(\mathbf{u}_i)$ with (14);
17   **if** $F(\mathbf{u}_i) \leq F(\mathbf{u}'_i)$ **then**
18             $\mathbf{u}'_i = \mathbf{u}_i$;
19             $f(\mathbf{u}'_i) = f(\mathbf{u}_i)$;
20             $\mathbf{S_F} \leftarrow F_i$, $\mathbf{S_{CR}} \leftarrow CR_i$;
21   **if** $\mathbf{S_F} \neq \emptyset$ *and* $\mathbf{S_{CR}} \neq \emptyset$ **then**
22   Update $M_{F,k}$ and $M_{CR,k}$ based on $\mathbf{S_F}$ and $\mathbf{S_{CR}}$ (see (17) and (18));
23         $k = k + 1$;
24   **if** $k > H_m$ **then**
25             $k = 1$;

We make a simple modification to the parameter adaptation in SHADE [37]: the weighted means $\text{mean}_{\text{WL}}(\mathbf{S_F})$ and $\text{mean}_{\text{WA}}(\mathbf{S_{CR}})$ are replaced with the means $\text{mean}_{\text{L}}(\mathbf{S_F})$ and $\text{mean}_{\text{A}}(\mathbf{S_{CR}})$ presented in JADE [39][2], respectively. The reason is that the fitness-based weights calculated in [37] are not suitable for NESs with multiple roots. For example, a parent $\mathbf{x}_i$ with $f(\mathbf{x}_i) = 0$ generates a different trial vector $\mathbf{u}_i$ with $f(\mathbf{u}_i) = 0$. Hence, $\Delta = f(\mathbf{u}_i) - f(\mathbf{x}_i) = 0$. In this way, the influence of F and CR will be ignored if the weighted means of SHADE [37] are used. However, in fact, they are successful parameters since they generate a new root $\mathbf{u}_i$.

D. The Framework of RADE

By incorporating the above-mentioned repulsion technique, diversity preservation mechanism, and adaptive parameter control, the framework of RADE is given in Algorithm 2:

- In line 16, the repulsion technique is applied when the archive is not empty. Note that, since $f(\mathbf{x})$ has been computed in (3), we do not need to re-computer $f(\mathbf{x})$ in (14).
- In line 12, the neighborhood mutation is used.
- In lines 15, 17, and 18, the crowding selection occurs.
- In lines 3, 9, 11, and 20-22, the adaptive parameter control is implemented.

Initially, the population $\mathcal{P}$ containing NP individuals is randomly generated and all the individuals in $\mathcal{P}$ are evaluated by $f(\mathbf{x})$ in (3). At each iteration, the archive $\mathcal{A}$ is updated with the individuals in $\mathcal{P}$ by making use of Algorithm 1. Thereafter, NP trial vectors are generated via the neighborhood mutation and crossover of DE, and evaluated by $f(\mathbf{x})$ in (3). Subsequently, the crowding selection is applied to each trial vector and its closest individual in $\mathcal{P}$. If $\mathcal{A} \neq \emptyset$, the comparison between them is based on the repulsion function $R(\mathbf{x})$; otherwise $f(\mathbf{x})$. After the crowding selection, the parameters CR and F of DE will be updated adaptively. Finally, $\mathcal{A}$ will be output if the maximum number of fitness evaluations (i.e., Max_FEs) is reached.

Remakr 1: From Algorithm 2, we can see that there are two major differences between RADE and NCDE [35]:

1) The repulsion technique is used in RADE to find multiple roots of NESs. It is able to avoid convergence to the previously found roots. Since the repulsion technique is employed, an external archive $\mathcal{A}$ is used to store the roots that have already been found.
2) The adaptive parameter control is implemented in RADE, which can enhance the search ability and eliminate the trial-and-error parameter tuning of DE.

Remakr 2: In [9], an improved harmony search (I-HS) is proposed to detect multiple roots of a NES in a single run. Unfortunately, several algorithmic parameters in [9] should be set properly. It is worth noting that RADE is similar to the work in [9], because both of them employ EAs and the repulsion technique to find multiple roots of a NES. However, there are four significant differences between them:

1) This paper combines the neighborhood mutation, the crowding selection, and the adaptive parameter control together to form an effective optimizer for NESs.
2) In [9], the population re-initialization is used to maintain the diversity, whereas in RADE the neighborhood mutation and the crowding selection are adopted to preserve the diversity. As we mentioned before, a possible drawback of the population re-initialization is the loss of some useful information obtained in the previous population.
3) They have different frameworks: as shown in [9, Algorithm 1], the optimization algorithm is embedded into the repulsion technique; while in RADE, the repulsion technique is embedded into the optimization algorithm.
4) In this paper, comprehensive experiments have been conducted on 30 NESs with diverse features to evaluate the performance of RADE.

V. Experimental Studies and Discussions

This section dedicates to the experimental studies and discussions, including studying the principle of RADE, comparing RADE with five well-established methods, discussing the effect of different components and the parameter settings in RADE, and comparing RADE with other reported results.

[2]For the sake of brevity, we omit the detailed descriptions of $\text{mean}_{\text{WL}}(\mathbf{S_F})$, $\text{mean}_{\text{WA}}(\mathbf{S_{CR}})$, $\text{mean}_{\text{L}}(\mathbf{S_F})$, and $\text{mean}_{\text{A}}(\mathbf{S_{CR}})$ in SHADE [37] and JADE [39]. More details can be found in the corresponding references.

TABLE I
BRIEF INFORMATION OF THE 30 NESS, WHERE n IS THE NUMBER OF DECISION VARIABLES, LE AND NE ARE THE NUMBER OF LINEAR AND NONLINEAR EQUATIONS, RESPECTIVELY, NOR IS THE NUMBER OF THE KNOWN ROOTS OF A NES, AND Max_FEs IS THE MAXIMUM NUMBER OF FITNESS EVALUATIONS.

Prob.	n	LE	NE	NOR	Max_FEs
F01	20	0	2	2	50,000
F02	2	1	1	11	50,000
F03	2	0	2	15	50,000
F04	2	0	2	13	50,000
F05	10	0	10	1	50,000
F06	2	1	1	8	50,000
F07	2	0	2	2	50,000
F08	2	0	2	7	50,000
F09	5	4	1	3	100,000
F10	3	0	3	2	50,000
F11	2	0	2	4	50,000
F12	2	0	2	10	50,000
F13	3	0	3	12	50,000
F14	2	0	2	9	50,000
F15	2	0	2	2	50,000
F16	2	0	2	13	50,000
F17	8	1	7	16	100,000
F18	2	0	2	6	50,000
F19	20	19	1	2	200,000
F20	3	0	3	7	50,000
F21	2	0	2	4	50,000
F22	2	0	2	6	50,000
F23	3	0	3	16	500,000
F24	3	0	3	8	100,000
F25	3	0	3	2	100,000
F26	2	0	2	2	50,000
F27	2	0	2	3	50,000
F28	2	0	2	2	50,000
F29	3	0	3	5	50,000
F30	2	0	2	4	50,000

A. Test Problems and Performance Criteria

To comprehensively evaluate the performance of different methods, we chose 30 NESs (denoted as F01-F30) from the literature. These test problems have different features, and some of them come from real-world applications, such as the interval arithmetic problem (F05) [29], multiple steady states problem (F08) [4], robot kinematics problem (F17) [40], and molecular conformation (F23) [41]. Table I briefly describes them and their detailed information has been provided in the supplementary material.

As pointed out previously, the transformed optimization problems of NESs in (3) and (4) are similar to multimodal optimization problems, because both of them are required to find multiple roots/optimal solutions. Therefore, we borrowed two performance criteria in [42] for multimodal optimization to compare the performance of different methods for NESs:

- Root rate (RR): It computes the average ratio of roots found over multiple runs:

$$RR = \frac{\sum_{i=1}^{N_r} NOR_i}{NOR \cdot N_r} \tag{19}$$

  where N_r is the number of runs, NOR_i is the number of roots found in the ith run, and NOR is the number of the known roots of a NES. In this paper, we made use of the archive $\mathcal{A}$ to compute NOR_i in each run based on Algorithm 1. The three parameters in Algorithm 1 were set as follows and kept unchanged:
  - Accuracy level: $\theta = $ 1E-06, if $n \leq 5$; otherwise, $\theta =$ 1E-04.
  - Distance radius: $\tau_d = 0.001$, if $n \leq 5$; otherwise, $\tau_d = 0.01$.
  - Maximum archive size: $\max_s = NP$.
- Success rate (SR): It measures the ratio of successful runs:

$$SR = \frac{N_{sr}}{N_r} \tag{20}$$

  where N_{sr} is the number of successful runs. A successful run is defined as a run where all the known roots of a NES are found.

TABLE II
PARAMETER SETTINGS OF THE SIX COMPARED METHODS.

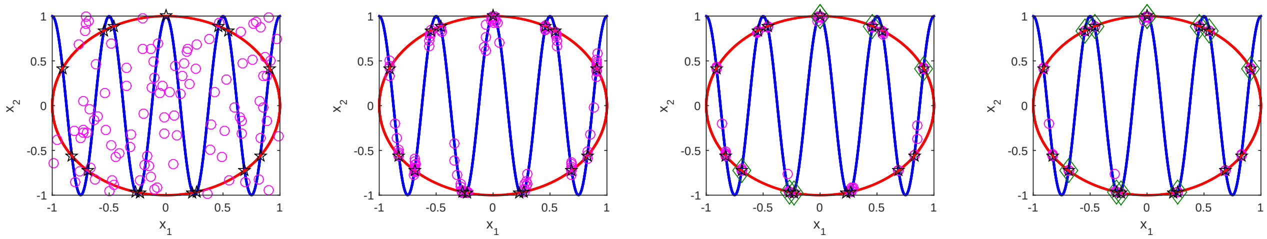
RADE	$NP = 100, H_m = 200$
NCDE	$NP = 100, F = 0.9, CR = 0.1$
R-JADE	$NP = 100, \mu_{CR} = 0.5, \mu_F = 0.5$
R-CLPSO	$NP = 100, m = 7, c = 2.0$
I-HS	$NP = 10$, HMCR $= 0.95$, PAR$_{\min} = 0.35$, PAR$_{\max} = 0.99$, BW$_{\min} = 10^{-6}$, BW$_{\max} = 5$
MONES	$NP = 100, H_m = 200$

To test the statistical differences among different methods, based on the RR and SR values, we conducted the multiple-problem Wilcoxon's test and the Friedman's test via KEEL software [43]. In the multiple-problem Wilcoxon's test, $p < 0.05$ means that there is a significant difference between the two compared methods.

B. Methods for Comparison and Their Parameter Settings

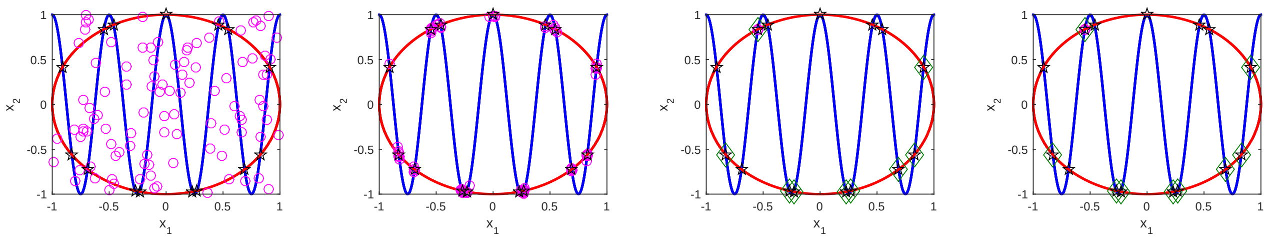
RADE is compared with five different methods:

- NCDE: This method is presented in [35]. It is chosen based on two considerations: 1) it obtains very promising results for multimodal optimization problems; and 2) similar to NCDE, the neighborhood mutation and the crowding selection are also adopted in RADE. For NCDE, (3) is considered as the objective function. Through comparing with NCDE, the influence of the repulsion technique and adaptive parameter control in RADE can be evaluated.
- R-JADE, R-CLPSO, and I-HS: They are repulsion-based methods. In R-JADE and R-CLPSO, JADE proposed in [39] and CLPSO proposed in [44] are considered as the optimization algorithms, respectively. The reason why the repulsion technique is integrated with JADE and CLPSO is the following: JADE and CLPSO exhibit outstanding performance among different DE variants and particle swarm optimization (PSO) variants, respectively. In [9], an improved harmony search (I-HS) proposed in [45] is combined with the repulsion technique to solve NESs. By comparing with R-JADE, R-CLPSO, and I-HS, we can assess the influence of different optimization algorithms on solving NESs.
- MONES: It is a recent multiobjective optimization-based method [30], which is able to locate multiple roots of



(a) gen=1 (b) gen=100 (c) gen=300 (d) gen=500

Fig. 3. Evolution of RADE-WoR over a typical run on F03. Circles denote the individuals in the population, pentagrams denote the known roots, and diamonds denote the found roots in the archive (the same below).



(a) gen=1 (b) gen=100 (c) gen=300 (d) gen=500

Fig. 4. Evolution of RADE-WoD over a typical run on F03.

NESs. Note that MONES originally uses NSGA-II [46] as the search engine. In this paper, SHADE replaces the crossover and mutation operators as well as the algorithmic parameters in NSGA-II. It is because DE operators are able to achieve better results than the orignal operators in NSGA-II for multiobjective optimization problems [16].

For the six compared methods, the parameter settings are given in Table II[3]. All the parameter settings were kept the same in our experiments unless a change was mentioned. Max_FEs is given in the last column of Table I. It is necessary to emphasize that Max_FEs was set according to the difficulties of different test problems. Each test problem was optimized over 100 independent runs. For fair comparison, all of the compared methods started with the same initial population in each of 100 runs.

Note that the repulsion technique in (5) was used in R-JADE, R-CLPSO, I-HS, and RADE. The effect of other repulsion techniques introduced in Section II-A is discussed in Section V-E1.

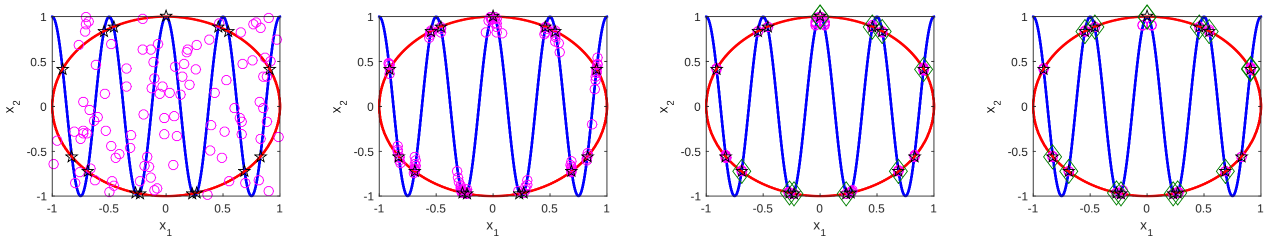
C. Proof-of-Principle Results

Firstly, we are interested in analyzing the working principle of RADE. For this purpose, we chose the third test problem F03 as an instance and considered three variants of RADE: 1) RADE-WoR, in which the repulsion technique was removed from RADE and $F(\mathbf{x}) = f(\mathbf{x})$ in (14) during the evolution; 2) RADE-WoD, where we eliminated the diversity preservation mechanism from RADE and the classic DE/rand/1/bin was utilized; and 3) RADE-WoA, in which the adaptive parameter control of RADE was not used and F and CR were respectively fixed to 0.9 and 0.1 according to the suggestion in [35]. For F03, since $Max\_FEs = 50,000$ and $NP = 100$, the maximum number of generation is equal to 500. As shown in Table I, F03 contains two decision variables; thus it is easy to monitor the evolutionary status of a method. When solving F03, RADE and its three variants used the same initial population to ensure a fair comparison.

Fig. 3-Fig. 6 provide the evolution of RADE and its three variants over a typical run on F03. From Fig. 3-Fig. 6, we can give the following comments:

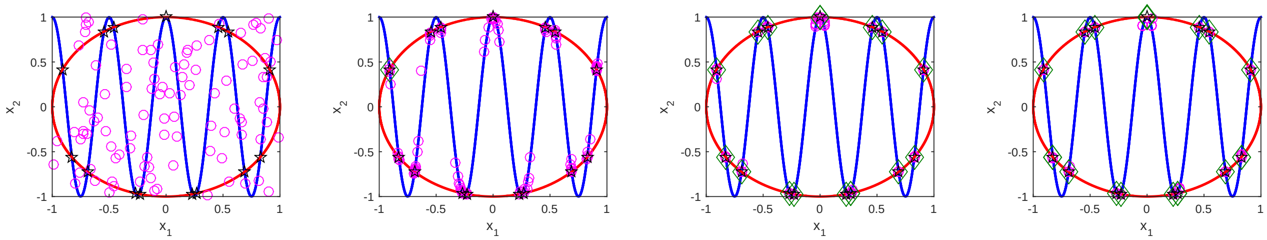
- RADE-WoR loses some of the roots. Although some individuals already surround the attraction basins of some roots, they cannot be stored into the archive due to the low precision. Moreover, as shown in Fig. 3, if two roots are very close to each other, RADE-WoR is likely to miss one of them. The reasons of the above phenomena are twofold: 1) due to the lack of the repulsion technique, if some roots have been found, the individuals near these roots will continue to search in the attraction basins of these roots, which inevitably wastes a lot of computational resource and results in low precision of other individuals; and 2) if two roots are very close and if one of them has been identified, the individuals around the unfound root may be replaced with other individuals surrounding the found root.
- It is clear from Fig. 4 that RADE-WoD also fails to locate all the roots. Actually, we can observe that during the evolution, RADE-WoD is able to find some of the roots. It is because in the early and middle stages of evolution, the diversity of the population is good and the repulsion technique can guide the population to find

[3] As suggested in [35], the neighborhood size ℓ was dynamically controlled as $\ell = 5 + \lfloor 5 \cdot (max\_gen - gen)/max\_gen \rfloor$ for RADE and NCDE, where gen and max_gen are the current and maximum number of generations, respectively.



(a) gen=1 (b) gen=100 (c) gen=300 (d) gen=500

Fig. 5. Evolution of RADE-WoA over a typical run on F03.



(a) gen=1 (b) gen=100 (c) gen=300 (d) gen=500

Fig. 6. Evolution of RADE over a typical run on F03.

TABLE III
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR THE FOUR COMPARED METHODS.

RADE VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
RADE-WoR	303.5	161.5	1.49E-01	357.0	108.0	**9.30E-03**
RADE-WoD	433.5	31.5	**7.00E-07**	426.0	39.0	**2.84E-06**
RADE-WoA	460.0	5.0	**1.86E-08**	454.5	10.5	**9.13E-08**

TABLE IV
RANKINGS OF THE FOUR COMPARED METHODS BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
RADE	**1.5833**	**1.5833**
RADE-WoR	1.7833	1.9500
RADE-WoD	3.3833	3.2667
RADE-WoA	3.2500	3.2000

some roots. However, at the end of evolution, all the individuals converge to one of the roots as shown in Fig. 4. It is not difficult to understand since the diversity of the population gradually decreases. Note that if all the individuals already get stuck in one of the roots, it is very difficult for them to jump out from the search region even though the repulsion technique is available, owing to the poor diversity.

- As depicted in Fig. 5, again, it is a very challenging task for RADE-WoA to find all the roots. This can be explained as follows: the search ability of RADE-WoA is weak because of the unsuitable parameter settings for F and CR, which results in the low precision of some individuals.
- From Fig. 6, RADE succeeds in finding all the roots of F03. The success of RADE can be attributed to the facts that: 1) the repulsion technique has the capability to avoid searching around the roots found previously and, as a result, increases the possibility to detect more roots; 2) the diversity preservation mechanism can maintain the diversity of the population, which is definitely beneficial for finding multiple roots during the evolution; and 3) the adaptive parameter control is able to pursue suitable parameters, thus enhancing the performance of RADE.

To further investigate the effectiveness of the three aspects of RADE, we applied RADE and its three variants to solve the 30 NESs. Table S-R-I in the supplemental material summarizes the detailed results. Tables III and IV report the statistical test results based on the multiple-problem Wilcoxon's test and the Friedman's test, respectively. From Table III, RADE achieves higher R^+ values than R^- values in all the cases. Moreover, RADE performs significantly better than RADE-WoD and RADE-WoA because the p-values are less than 0.05. According to the Friedman's test in Table IV, RADE ranks the first in both RR and SR criteria.

Clearly, the above results verify the motivation of this paper—the repulsion technique, the diversity preservation mechanism, and the adaptive parameter control are three indispensable elements for RADE to effectively find multiple roots of NESs.

D. Comparison with NCDE, R-JADE, R-CLPSO, I-HS, and MONES

The performance of RADE is compared with that of the five methods introduced in Section V-B. The detailed results in terms of RR and SR are respectively reported in Table S-R-II and Table S-R-III in the supplementary material. It can be seen from Table S-R-II and Table S-R-III that RADE provides the highest average RR value, i.e., 0.9491 and the highest average SR value, i.e., 0.8247. Furthermore, RADE

TABLE V
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR THE SIX COMPARED METHODS.

RADE VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
NCDE	445.0	20.0	**3.73E-08**	439.5	25.5	**1.82E-07**
R-JADE	422.5	42.5	**5.11E-06**	411.5	53.5	**8.38E-05**
R-CLPSO	424.5	40.5	**3.68E-06**	423.0	42.0	**4.70E-06**
I-HS	387.5	77.5	**9.12E-04**	373.0	92.0	**1.67E-03**
MONES	427.0	38.0	**2.38E-06**	407.0	58.0	**4.70E-05**

TABLE VI
RANKINGS OF THE SIX COMPARED METHODS BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
RADE	**1.9000**	**2.0333**
NCDE	4.3833	4.1500
R-JADE	3.4500	3.5500
R-CLPSO	3.7333	3.8000
I-HS	3.3500	3.4167
MONES	4.1833	4.0500

successfully solves 13 out of 30 test problems over 100 runs. In contrast, NCDE, R-JADE, R-CLPSO, I-HS, and MONES successfully solve five, eight, eight, nine, and five test problems over 100 runs, respectively.

The statistical test results obtained by the multiple-problem Wilcoxon's test are reported in Table V. Additionally, the rankings of the six compared methods derived from the Friedman's test are presented in Table VI. From Table V, RADE consistently provides significantly better results than NCDE, R-JADE, R-CLPSO, MONES, and I-HS because the p-values are less than 0.05 in all the cases. In addition, RADE has the best ranking as shown in Table VI. Therefore, we can draw the following conclusions:

- Comparing RADE with NCDE, we can conclude that the repulsion technique and the adaptive parameter control significantly improve the performance of NCDE.
- Comparing RADE with R-JADE, R-CLPSO, and I-HS, we can conclude that although the repulsion technique can be exploited to find different roots of NESs, the performance of the repulsion-based methods is remarkably influenced by the optimization algorithms.

The above comparison indicates that, on the whole, the cooperation of the repulsion technique, the diversity preservation mechanism, and the adaptive parameter control is really effective. As a consequence, RADE is able to yield improved performance compared with the five competitors.

E. Discussions on Different Components

This subsection discusses the effectiveness of different components in RADE.

1) On Other Repulsion Techniques: In Section II-A, three representative repulsion techniques were introduced. In the previous experiments, the repulsion technique in (5) was used. To investigate the effectiveness of other repulsion techniques, we replaced (5) with (7) and (9) in RADE. As a result, two RADE variants are obtained, called RADE-1 and RADE-2.

The detailed results in terms of RR and SR are provided in Table S-R-IV in the supplementary material. The statistical test results are reported in Tables VII and VIII. From Table VIII, we can see that RADE with the repulsion technique in (5) performs the best, followed by RADE-1. However, there is no significant difference between RADE and each of RADE-1 and RADE-2 according to the multiple-problem Wilcoxon's test in Table VII since $p > 0.05$.

TABLE VII
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR RADE WITH DIFFERENT REPULSION TECHNIQUES.

RADE VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
RADE-1	266.0	199.0	≥ 0.2	270.5	194.5	≥ 0.2
RADE-2	319.5	145.5	≥ 0.2	322.5	142.5	≥ 0.2

TABLE VIII
RANKINGS OF RADE WITH DIFFERENT REPULSION TECHNIQUES BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
RADE	**1.8333**	**1.8333**
RADE-1	1.9667	1.9833
RADE-2	2.2000	2.1833

TABLE IX
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR THE SIX COMPARED ALGORITHMS.

RADE-3 VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
RADE	157.0	308.0	≥ 0.2	140.0	325.0	≥ 0.2
NCDE	438.5	26.5	**6.56E-06**	446.5	18.5	**3.34E-05**
R-JADE	393.5	71.5	**1.51E-02**	396.0	69.0	**4.97E-02**
R-CLPSO	396.5	68.5	**1.23E-02**	392.0	73.0	6.74E-02
MONES	426.0	39.0	**2.17E-04**	401.5	63.5	**1.82E-03**

2) On Other Diversity Preservation Mechanisms: In RADE, the crowding selection was used as the selection operator. To study the effectiveness of other diversity preservation mechanisms, the speciation selection [35] was applied to replace the crowding selection in RADE. The resulting method is referred to as RADE-3. In RADE-3, all other parameter settings were kept the same with RADE. In the supplementary material, Table S-R-V reports the RR and SR values of RADE-3. In addition, Table IX compares RADE-3 with RADE, NCDE, R-JADE, and MONES according to the multiple-problem Wilcoxon's test, by integrating the experimental results in Tables S-R-II and S-R-III.

Table S-R-V indicates that RADE-3 is slightly worse than RADE in terms of the average RR and SR. However, from Table IX, it can be seen that RADE-3 obtains significantly better performance than NCDE, R-JADE, and MONES. It also significantly outperforms R-CLPSO in terms of RR at $p = 0.05$ and in terms of SR at $p = 0.1$, respectively. Hence, one can conclude that other advanced diversity preservation mechanism could also be applicable to the RADE framework.

3) On Other Adaptive Parameter Controls: In the DE literature, there are also other techniques for adaptive parameter control, e.g., jDE [47] and JADE [39]. The effectiveness of the techniques in jDE and JADE was investigated by incorporating them into RADE. The resultant methods are denoted as RADE-4 (RADE with jDE's adaptive parameter

TABLE X
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR RADE WITH DIFFERENT ADAPTIVE PARAMETER CONTROLS.

RADE VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
RADE-4	195.0	270.0	≥ 0.2	245.5	219.5	≥ 0.2
RADE-5	356.0	109.0	≥ 0.2	340.5	124.5	≥ 0.2

TABLE XI
RANKINGS OF RADE WITH DIFFERENT ADAPTIVE PARAMETER CONTROLS BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
RADE	**1.8500**	**1.8500**
RADE-4	1.9167	2.0000
RADE-5	2.2333	2.1500

TABLE XII
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR RADE WITH DIFFERENT MUTATION OPERATORS.

RADE VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
RADE-6	459.5	5.5	**8.94E-08**	454.0	11.0	**1.64E-06**
RADE-7	437.5	27.5	**2.52E-03**	426.0	39.0	**4.32E-02**
RADE-8	461.0	4.0	**1.30E-08**	457.5	7.5	**1.64E-07**
RADE-9	426.5	38.5	**2.98E-04**	435.0	30.0	**9.41E-04**
SaDE	324.0	141.0	≥ 0.2	322.0	143.0	≥ 0.2

TABLE XIII
RANKINGS OF RADE WITH DIFFERENT MUTATION OPERATORS BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
RADE	**1.7333**	**1.9000**
RADE-6	5.4000	4.9667
RADE-7	3.1500	3.2167
RADE-8	4.8833	4.9167
RADE-9	3.6833	3.8500
SaDE	2.1500	2.1500

control) and RADE-5 (RADE with JADE's adaptive parameter control), respectively.

The detailed results are provided in Table S-R-VI in the supplementary material. Additionally, Table X and Table XI report the statistical test results based on the multiple-problem Wilcoxon's test and the Friedman's test, respectively. It is clear from Table XI that RADE yields better ranking than RADE-4 and RADE-5. However, there are no significant differences among these three methods from Table X.

Remark 3: Based on the above empirical discussions, it seems that there exist some other options for the advanced repulsion techniques, diversity preservation mechanisms, and adaptive parameter controls under our RADE framework, which verifies the robustness of our RADE framework.

4) On Different Mutation Operators: There is a variety of mutation operators in the DE community [13]. Generally, different mutation operators are suitable for different problems [48], [49]. In RADE, "DE/rand/1" was originally employed in the neighborhood mutation as shown in (12). We replaced "DE/rand/1" with other widely used mutation operators, with the aim of evaluating the effectiveness of different mutation operators for solving NESs. To this end, we tested four RADE variants: 1) RADE-6, i.e., RADE with "DE/rand-to-best/1"; 2) RADE-7, i.e., RADE with "DE/rand/2"; 3) RADE-8, i.e., RADE with "DE/current-to-best/2"; and 4) RADE-9, i.e., RADE with "DE/current-to-rand/1"[4]. In addition, since the adaptation technique for mutation operators has been proved to be effective for improving the performance of DE in single-objective optimization problems [48], SaDE, a representative adaptation technique, was also assessed in this section. Note that for the four RADE variants and SaDE, the neighborhood mutation and the crowding selection were also adopted. The parameter settings were kept unchanged for the four RADE variants. For SaDE, the parameter settings were the same as in the original paper [48].

Tables S-R-VII and S-R-VIII in the supplementary material report the RR and SR values, respectively. The statistical test results derived from the multiple-problem Wilcoxon's and Friedman's tests are respectively given in Tables XII and XIII. Tables S-R-VII, S-R-VIII, XII, and XIII reveal that:

- Mutation operators have a significant effect on the performance of RADE. Overall, "DE/rand/1" accomplishes the best results, followed by "DE/rand/2" and "DE/current-to-rand/1". Whereas, the mutation operators with the best solution drastically degenerate the performance of RADE.
- SaDE provides the second best results in terms of both RR and SR, which means that mutation operator adaptation could be a good choice when solving different NESs.

TABLE XIV
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR RADE WITH DIFFERENT NP VALUES.

$NP = 100$ VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
$NP = 50$	323.0	142.0	5.06E-02	309.5	155.5	1.17E-01
$NP = 80$	263.5	201.5	≥ 0.2	233.5	231.5	≥ 0.2
$NP = 120$	286.5	178.5	≥ 0.2	246.0	219.0	≥ 0.2
$NP = 150$	260.5	204.5	≥ 0.2	267.5	197.5	≥ 0.2
$NP = 200$	254.5	210.5	≥ 0.2	280.5	184.5	≥ 0.2

TABLE XV
RANKINGS OF RADE WITH DIFFERENT NP VALUES BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
$NP = 50$	4.3833	4.2667
$NP = 80$	3.5333	3.4167
$NP = 100$	3.3833	3.3500
$NP = 120$	**3.1500**	**3.2167**
$NP = 150$	3.2667	3.2833
$NP = 200$	3.2833	3.4667

F. Discussions on Different Parameter Settings

This subsection discusses the influence of different parameter settings of NP, H_m, and α on the performance of RADE.

1) Effect of Different Population Sizes: In the previous sections, the population size NP was fixed to 100, which has been widely adopted in the DE literature [37], [39], [47],

[4] "best" in "DE/rand-to-best/1" and "DE/current-to-best/2" means the best individual in the current population based on $F(\mathbf{x})$ in (14). For RADE-6, RADE-7, and RADE-8, the binomial crossover in DE was used, however for RADE-9, the crossover was not used.

TABLE XVI
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR RADE WITH DIFFERENT H_m VALUES.

$H_m = 800$ VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
$H_m = 5$	360.0	105.0	≥ 0.2	340.5	124.5	≥ 0.2
$H_m = 10$	372.0	93.0	≥ 0.2	354.5	110.5	≥ 0.2
$H_m = 30$	358.5	106.5	≥ 0.2	322.5	142.5	≥ 0.2
$H_m = 50$	356.5	108.5	≥ 0.2	335.5	129.5	≥ 0.2
$H_m = 100$	377.5	87.5	≥ 0.2	338.5	126.5	≥ 0.2
$H_m = 200$	354.0	111.0	≥ 0.2	314.5	150.5	≥ 0.2
$H_m = 300$	355.5	109.5	≥ 0.2	343.5	121.5	≥ 0.2
$H_m = 400$	311.0	154.0	≥ 0.2	306.0	159.0	≥ 0.2
$H_m = 500$	262.0	203.0	≥ 0.2	235.5	229.5	≥ 0.2
$H_m = 1000$	322.0	143.0	≥ 0.2	317.5	147.5	≥ 0.2

TABLE XVII
RANKINGS OF RADE WITH DIFFERENT H_m VALUES BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
$H_m = 5$	7.6333	7.2000
$H_m = 10$	7.3167	7.1500
$H_m = 30$	7.1333	6.6500
$H_m = 50$	6.6500	6.5833
$H_m = 100$	6.3833	6.0333
$H_m = 200$	5.7667	5.7500
$H_m = 300$	6.1667	6.1500
$H_m = 400$	5.0333	5.5833
$H_m = 500$	4.4833	**4.7000**
$H_m = 800$	**4.4500**	4.8167
$H_m = 1000$	4.9833	5.3833

[49]. We empirically investigated the effect of population size by testing different NP values: 50, 80, 120, 150, and 200. All other parameter settings in RADE were kept the same as in Table II. Tables S-R-IX and S-R-X in the supplementary material report the detailed results of RR and SR, respectively. In addition, Tables XIV and XV respectively show the statistical results obtained by the multiple-problem Wilcoxon's test and the Friedman's test.

From Table XV, we can observe that $NP = 120$ is able to provide the best performance on the whole. The results in Table XIV indicate that there are not significant differences among RADE with different population sizes. However, by carefully looking at the results in Tables S-R-IX and S-R-X, we find that RADE with a smaller population size provides better results on F04, F09, F16, and F19; while on F02, F03, F12, F13, F17, and F23, the better results are obtained by RADE with a larger population size. This phenomenon could motivate us to study the dynamic or adaptive control of population size to further improve the performance of RADE. However, it is out of the scope of this work. We will leave it to our future work. In general, $NP \in [80, 120]$ can yield promising results.

2) Influence of Different H_m Values: In RADE, the parameter H_m controls the size of history memory for adaptively updating F and CR in Section IV-C. In the previous experiments, $H_m = 200$ was used. We tested the following H_m values: 5, 10, 30, 50, 100, 300, 400, 500, 800, and 1000. All other parameter settings were kept unchanged. The detailed results of RR and SR are respectively reported in Tables S-R-XI and S-R-XII in the supplementary material. Furthermore, the statistical test results obtained by the multiple-problem Wilcoxon's test are shown in Table XVI and the rankings resulting from the Friedman's test are given in Table XVII.

TABLE XVIII
RESULTS OBTAINED BY THE MULTIPLE-PROBLEM WILCOXON'S TEST FOR RADE WITH DIFFERENT α VALUES.

$\alpha = 1$ VS	RR			SR		
	R^+	R^-	p-value	R^+	R^-	p-value
$\alpha = 2$	296.0	169.0	≥ 0.2	267.5	197.5	≥ 0.2
$\alpha = 5$	298.0	167.0	≥ 0.2	277.0	188.0	≥ 0.2
$\alpha = 10$	279.5	185.5	≥ 0.2	241.0	224.0	≥ 0.2
$\alpha = 20$	342.0	123.0	≥ 0.2	311.5	153.5	≥ 0.2
$\alpha = 50$	326.5	138.5	≥ 0.2	299.5	165.5	≥ 0.2
$\alpha = 100$	324.0	141.0	≥ 0.2	267.0	198.0	≥ 0.2

TABLE XIX
RANKINGS OF RADE WITH DIFFERENT α VALUES BY THE FRIEDMAN'S TEST.

Algorithm	Ranking (RR)	Ranking (SR)
$\alpha = 1$	**3.2667**	**3.5833**
$\alpha = 2$	3.8167	3.7333
$\alpha = 5$	4.0667	3.9167
$\alpha = 10$	3.8833	3.7500
$\alpha = 20$	4.4333	4.4667
$\alpha = 50$	4.4000	4.3833
$\alpha = 100$	4.1333	4.1667

Tables S-R-XI and S-R-XII indicate that $H_m = 800$ provides the highest average RR and SR values. For $H_m > 200$, RADE gets better results in both RR and SR criteria compared with the default setting $H_m = 200$. However, for $H_m < 200$, the performance degradation occurs as H_m decreases. The reason might be the following: most of NESs in this paper contain many roots, when some of them are found during the previous generations, the fitness function will be altered according to the repulsion function. Subsequently, the optimizer will face a new optimization problem, and the previous suitable parameters recorded in the historical memory may be not suitable for the new optimization problem. Therefore, performance seems to be improved as H_m increases. Table XVII also verifies the above observation: RADE with $H_m \geq 200$ provides better rankings than $H_m < 200$.

Table XVI shows that the performance differences among RADE with different H_m values are not significant compared with $H_m = 800$, which means that the influence of H_m is not significant in RADE. According to Tables S-R-XI, S-R-XII, and XVII, the value of H_m can be chosen from a large range, e.g., $[200, 1000]$.

3) Influence of Different α Values: The repulsion function in (5) has two parameters ϵ and α. The parameter α, which is used to adjust the radius of the repulsion areas, plays a more important role than ϵ[5]. Therefore, the influence of different α values on the performance of RADE was empirically studied. The detailed results of RR and SR are respectively shown in Tables S-R-XIII and S-R-XIV in the supplementary material. Table XVIII reports the statistical test results provided by the

[5]We also set different ϵ values (e.g., 1E-08, 1E-10, 1E-15, 1E-20, and 0) in RADE, the experimental results show that the influence of ϵ can be negligible. Interested readers can find the detailed results in Tables S-R-XV and S-R-XVI in the supplementary material.

TABLE XX
EXPERIMENTAL RESULTS OF RADE AND N-M BASED ON 12 TEST PROBLEMS PRESENTED IN [12].

Prob.	RADE		N-M [12]	
	RR	SR	RR	SR
c01	**1.0000**	**1.00**	**1.0000**	**1.00**
c02	**1.0000**	**1.00**	**1.0000**	0.97
c03	**1.0000**	**1.00**	0.9778	0.83
c04	**1.0000**	**1.00**	0.9000	0.80
c05	**1.0000**	**1.00**	**1.0000**	**1.00**
c06	**0.9969**	**0.96**	0.9692	0.70
c07	**1.0000**	**1.00**	0.9571	0.67
c08	**1.0000**	**1.00**	**1.0000**	**1.00**
c09	**0.8733**	**0.67**	0.7333	0.50
c10	0.9300	0.86	**1.0000**	**1.00**
c11	**1.0000**	**1.00**	**1.0000**	**1.00**
c12	**1.0000**	**1.00**	0.9833	0.97
Avg.	**0.9834**	**0.9575**	0.9601	0.8700

multiple-problem Wilcoxon's test for RADE with different α values. The rankings obtained by the Friedman's test are described in Table XIX.

It can be seen from Tables S-R-XIII, S-R-XIV, and XIX that RADE with $\alpha = 1$ achieves the highest average RR and SR values and the best ranking, followed by $\alpha = 2$. Although the default setting $\alpha = 10$ is not the best, it still obtains acceptable results. RADE with larger α values ($\alpha \geq 20$) performs worse than RADE with smaller α values ($\alpha \leq 10$). The reason can be explained as follows: as mentioned in [7], if α decreases, the radius of the repulsion regions generated by $|\coth(\alpha x)|$ will be enlarged. It means that the penalty term $\prod_{j=1}^{K} |\coth(\alpha\delta_j)|$ in (5) increases with the decrease of α. In this way, the method with smaller α values will pay more attention to search for new roots, due to the higher penalty on the solutions that are closer to the roots found previously.

From Table XVIII, there are no significant performance differences among RADE with different α values. Suggested by the results in Tables S-R-XIII, S-R-XIV, XVIII, and XIX, the reasonable value of α is between 1 and 10.

G. Comparison with Other Reported Results

In the previous experiments, the superior performance of RADE has been demonstrated through the 30 NESs in Table I. To further understand the performance of RADE, it was compared with other reported results in [12]. Twelve NESs collected in [12] were used in this paper, which are referred to as c01–c12. In [12], the repulsion technique is integrated with Nelder-Mead (N-M) method to find multiple roots of NESs. To make a fair comparison, RADE implemented the same number of fitness evaluations for each test problem as in [12] over 100 independent runs. In addition, the original results in [12] were converted into the RR and SR values based on (19) and (20), respectively.

The experimental results are summarized in Table XX. On six out of 12 test problems, RADE provides better results in terms of both RR and SR criteria. However, N-M outperforms RADE on only one test problem. In addition, the number of test problems which can be successfully solved by RADE and N-M over all 100 runs is nine and six, respectively. Therefore, it can be concluded that RADE is also better than N-M for solving NESs.

VI. CONCLUSION

Solving NESs is a very important area in numerical computation. Very often, NESs contain multiple roots. However, it is a very challenging task to find multiple roots of NESs simultaneously in a single run. To address this issue, in this paper, we proposed a repulsion-based adaptive differential evolution, called RADE, in which the repulsion technique, the diversity preservation mechanism, and the adaptive parameter control were combined to solve NESs effectively. The performance of RADE was evaluated by 30 NESs selected from the literature. The experimental results suggested that RADE is able to find multiple roots simultaneously in a single run on all the test problems. It can also provide very competitive performance compared with other well-established methods. In addition, we carried out extensive experiments to systematically analyze the working principle of RADE, as well as the effectiveness of different components of RADE and the influence of parameter settings. According to the experiments, we demonstrated that the repulsion technique, the diversity preservation mechanism, and the adaptive parameter control are three indispensable elements of RADE, which verifies the motivation of this paper.

For the repulsion techniques introduced in Section II-A, several parameters need to be set properly. In the future, we will attempt to design dynamic or adaptive parameter controls for the repulsion techniques, such as the dynamic or adaptive radius of the repulsion regions. Additionally, we will apply RADE to deal with complex real-world NESs.

The source code of RADE can be obtained from the authors upon request.

ACKNOWLEDGMENTS

The authors would like to thank Prof. B. Y. Qu and Prof. P. N. Suganthan for kindly providing the source code of NCDE.

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] D. Mehta and C. Grosan, "A collection of challenging optimization problems in science, engineering and economics," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 2697–2704.
[3] S. Smale, "Mathematical problems for the next century," *The Mathematical Intelligencer*, vol. 20, no. 2, pp. 7–15, Mar 1998.
[4] C. A. Floudas, "Recent advances in global optimization for process synthesis, design and control: Enclosure of all solutions," *Computers & Chemical Engineering*, vol. 23, Supplement, no. 0, pp. 963–973, 1999.
[5] 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.
[6] 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, no. 5-6, pp. 551–561, 2010.
[7] 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, no. 0, pp. 33–43, 2012.
[8] 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, Dec 1997.

[9] G. C. Ramadas, E. M. Fernandes, and A. A. Rocha, "Multiple roots of systems of equations by repulsion merit functions," in *Computational Science and Its Applications - ICCSA 2014*, ser. Lecture Notes in Computer Science, B. Murgante, S. Misra, A. Rocha, C. Torre, J. Rocha, M. Falcão, D. Taniar, B. Apduhan, and O. Gervasi, Eds., 2014, vol. 8580, pp. 126–139.

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

[11] K. Parsopoulos and M. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. on Evol. Comput.*, vol. 8, no. 3, pp. 211–224, June 2004.

[12] 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.

[13] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. on Evol. Comput.*, vol. 15, no. 1, pp. 4–31, 2011.

[14] U. Halder, S. Das, and D. Maity, "A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 881–897, 2013.

[15] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 716–727, 2015.

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

[17] Y. Wang, H.-X. Li, G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, April 2015.

[18] 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.

[19] S. K. Rahimian, F. Jalali, J. Seader, and R. White, "A new homotopy for seeking all real roots of a nonlinear equation," *Computers & Chemical Engineering*, vol. 35, no. 3, pp. 403–411, 2011.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C: The art of scientific computing (2nd ed.)*. New York: Cambridge University Press, 1992.

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

[22] K. S. Gritton, J. Seader, and W.-J. Lin, "Global homotopy continuation procedures for seeking all roots of a nonlinear equation," *Computers & Chemical Engineering*, vol. 25, no. 7C8, pp. 1003–1019, 2001.

[23] D. Mehta, "Finding all the stationary points of a potential-energy landscape via numerical polynomial-homotopy-continuation method," *Phys. Rev. E*, vol. 84, p. 025702, 2011.

[24] H. Schwandt, "Parallel interval Newton-like Schwarz methods for almost linear parabolic problems," *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 437–444, 2007.

[25] C.-Y. Chen, "A performance comparison of the zero-finding by extended interval Newton method for Peano monosplines," *Applied Mathematics and Computation*, vol. 219, no. 12, pp. 6919–6930, 2013.

[26] M. D. Stuber, V. Kum, and P. I. Barton, "Nonsmooth exclusion test for finding all solutions of nonlinear equations," *BIT*, vol. 50, no. 4, pp. 885–917, 2010.

[27] 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.

[28] 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.

[29] 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, May 2008.

[30] W. Song, Y. Wang, H.-X. Li, and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization," *IEEE Trans. on Evol. Comput.*, vol. 19, no. 3, pp. 414–431, June 2015.

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

[32] 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, 2013.

[33] W. Hu, G. G. Yen, and G. Luo, "Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–14, 2016.

[34] S. Das, S. Maity, B.-Y. Qu, and P. Suganthan, "Real-parameter evolutionary multimodal optimization ł a survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71–88, 2011.

[35] 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, Oct 2012.

[36] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 219–232, 2016.

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

[38] ——, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1952–1959.

[39] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. on Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct 2009.

[40] C. Wang, R. Luo, K. Wu, and B. Han, "A new filled function method for an unconstrained nonlinear equation," *Journal of Computational and Applied Mathematics*, vol. 235, no. 6, pp. 1689–1699, 2011.

[41] I. Z. Emiris and B. Mourrain, "Computer algebra methods for studying and computing molecular conformations," *Algorithmica*, vol. 25, no. 2-3, pp. 372–402, 1999.

[42] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC2013 special session and competition on niching methods for multimodal function optimization," Evolutionary Computation and Machine Learning Group, RMIT University, Australia, Tech. Rep., 2013.

[43] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.

[44] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. on Evol. Comput.*, vol. 10, no. 3, pp. 281–295, 2006.

[45] 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.

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

[47] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. on Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec 2006.

[48] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. on Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr 2009.

[49] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Part B – Cybernetics*, vol. 41, no. 2, pp. 397–413, 2011.

Supplementary Material for "Finding Multiple Roots of Nonlinear Equation Systems via A Repulsion-based Adaptive Differential Evolution"

Wenyin Gong, Yong Wang, *Member, IEEE*, Zhihua Cai, and Ling Wang

S-I. Thirty Test Problems

1) F01:

$$\begin{cases} \sum_{i=1}^{n} x_i^2 - 1 = 0 \\ |x_1 - x_2| + \sum_{i=3}^{n} x_i^2 = 0 \end{cases} \tag{1}$$

where $x_i \in [-1, 1]$, $i = 1, \ldots, n$, and $n = 20$. It has two roots: (-0.707107, -0.707107, 0, …, 0) and (0.707107, 0.707107, 0, …, 0).

2) F02:

$$\begin{cases} x_1 - \sin(5\pi x_2) = 0 \\ x_1 - x_2 = 0 \end{cases} \tag{2}$$

where $x_i \in [-1, 1]$, $i = 1, \ldots, n$, and $n = 2$. It has 11 roots as shown in Table S-I.

TABLE S-I
Roots of F02.

x_1	x_2
-0.924840	-0.924840
-0.866760	-0.866760
-0.562010	-0.562010
-0.428168	-0.428168
-0.187960	-0.187960
0.000000	0.000000
0.187960	0.187960
0.428168	0.428168
0.562010	0.562010
0.866760	0.866760
0.924840	0.924840

3) F03:

$$\begin{cases} x_1 - \cos(4\pi x_2) = 0 \\ x_1^2 + x_2^2 = 1 \end{cases} \tag{3}$$

where $x_i \in [-1, 1]$, $i = 1, \ldots, n$, and $n = 2$. It has 15 roots as shown in Table S-II.

4) F04:

$$\begin{cases} \cos(2x_1) - \cos(2x_2) - 0.4 = 0 \\ 2(x_2 - x_1) + \sin(2x_2) - \sin(2x_1) - 1.2 = 0 \end{cases} \tag{4}$$

where $x_i \in [-10, 10]$, $i = 1, \ldots, n$, and $n = 2$. It has 13 roots as shown in Table S-III.

W. Gong and Z. Cai are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China. (Email: wygong@cug.edu.cn; zhcai@cug.edu.cn)

Y. Wang is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK. (Email: ywang@csu.edu.cn)

L. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China. (Email: wangling@tsinghua.edu.cn)

TABLE S-II
Roots of F03.

x_1	x_2
0.416408	-0.909178
-0.561364	-0.827569
-0.724322	-0.689462
0.837812	-0.545959
0.886984	-0.461799
-0.962322	-0.271914
-0.972855	-0.231415
1.000000	0.000000
-0.972855	0.231416
-0.962322	0.271914
0.886984	0.461799
0.837812	0.545959
-0.724322	0.689462
-0.561364	0.827569
0.416408	0.909178

TABLE S-III
Roots of F04.

x_1	x_2
-9.268258	-8.931402
-8.744542	-7.164787
-6.126665	-5.789809
-5.602950	-4.023195
-2.985073	-2.648216
-2.461357	-0.881602
0.156520	0.493376
0.680236	2.259991
3.298113	3.634969
3.821828	5.401583
6.439705	6.776562
6.963421	8.543176
9.581298	9.918154

5) F05:

$$\begin{cases} x_1 - 0.25428722 - 0.18324757 x_4 x_3 x_9 = 0 \\ x_2 - 0.37842197 - 0.16275449 x_1 x_{10} x_6 = 0 \\ x_3 - 0.27162577 - 0.16955071 x_1 x_2 x_{10} = 0 \\ x_4 - 0.19807914 - 0.15585316 x_7 x_1 x_6 = 0 \\ x_5 - 0.44166728 - 0.19950920 x_7 x_6 x_3 = 0 \\ x_6 - 0.14654113 - 0.18922793 x_8 x_5 x_{10} = 0 \\ x_7 - 0.42937161 - 0.21180486 x_2 x_5 x_8 = 0 \\ x_8 - 0.07056438 - 0.17081208 x_1 x_7 x_6 = 0 \\ x_9 - 0.34504906 - 0.19612740 x_{10} x_6 x_8 = 0 \\ x_{10} - 0.42651102 - 0.21466544 x_4 x_8 x_1 = 0 \end{cases} \tag{5}$$

where $x_i \in [-2, 2]$, $i = 1, \ldots, n$, and $n = 10$. It has one root: (0.257833, 0.381097, 0.278745, 0.200669, 0.445251, 0.149184, 0.432010, 0.073403, 0.345967, 0.427326).

6) F06:

$$\begin{cases} x_1 - 0.25 = 0 \\ x_1 \sin(4\pi x_2^2) + 0.75 x_1 - 0.25 = 0 \end{cases} \tag{6}$$

where $x_i \in [-1, 1]$, $i = 1, \ldots, n$, and $n = 2$. It has eight roots as shown in Table S-IV.

TABLE S-IV
ROOTS OF F06.

x_1	x_2
0.250000	-0.854337
0.250000	-0.721185
0.250000	-0.479471
0.250000	-0.141801
0.250000	0.141801
0.250000	0.479471
0.250000	0.721185
0.250000	0.854337

7) F07:

$$\begin{cases} x_1^2 - x_2 - 2 = 0 \\ x_1 + \sin\left(\frac{\pi}{2} x_2\right) = 0 \end{cases} \tag{7}$$

where $x_1 \in [0, 1]$ and $x_2 \in [-10, 0]$. It has two roots: (0, -2) and (0.707660, -1.5).

8) F08:

$$\begin{cases} (1-R)\left[\left(\frac{D}{10(1+\beta_1)} - x_1\right) \cdot \exp\left(\frac{10x_1}{1+\frac{10x_1}{\gamma}}\right)\right] - x_1 = 0 \\ (1-R)\left[\left(\frac{D}{10} - \beta_1 x_1 - (1+\beta_2)x_2\right) \cdot \exp\left(\frac{10x_2}{1+\frac{10x_2}{\gamma}}\right)\right] + x_1 - (1+\beta_2)x_2 = 0 \end{cases} \tag{8}$$

where $x_i \in [0, 1]$, $i = 1, \ldots, n$, $n = 2$, $R = 0.96$, $D = 22$, $\gamma = 1000$, and $\beta_1 = \beta_2 = 2$. It has seven roots as shown in Table S-V.

TABLE S-V
ROOTS OF F08.

x_1	x_2
0.042100	0.061813
0.042100	0.268723
0.266600	0.178430
0.266600	0.327267
0.266600	0.461111
0.042318	0.686779
0.719074	0.244197

9) F09:

$$\begin{cases} 2x_1 + x_2 + x_3 + x_4 + x_5 - 6.0 = 0 \\ x_1 + 2x_2 + x_3 + x_4 + x_5 - 6.0 = 0 \\ x_1 + x_2 + 2x_3 + x_4 + x_5 - 6.0 = 0 \\ x_1 + x_2 + x_3 + 2x_4 + x_5 - 6.0 = 0 \\ x_1 x_2 x_3 x_4 x_5 - 1.0 = 0 \end{cases} \tag{9}$$

where $x_i \in [-10, 10]$, $i = 1, \ldots, n$, and $n = 5$. It has three roots: (1, 1, 1, 1, 1), (0.916355, 0.916355, 0.916355, 0.916355, 1.418227), and (-0.579043, -0.579043, -0.579043, -0.579043, 8.895215).

10) F10:

$$\begin{cases} 3x_1^2 + \sin(x_1 x_2) - x_3^2 + 2.0 = 0 \\ 2x_1^3 - x_2^2 - x_3 + 3.0 = 0 \\ \sin(2x_1) + \cos(x_2 x_3) + x_2 - 1.0 = 0 \end{cases} \tag{10}$$

where $x_1 \in [-5, 5]$, $x_2 \in [-1, 3]$, and $x_3 \in [-5, 5]$. It has two roots: (-0.064417, 2.090440, -1.370473) and (-0.032759, 1.264629, 1.400644).

11) F11:

$$\begin{cases} x_1^2 - |x_2| + 1 + \frac{1}{9}|x_1 - 1| = 0 \\ x_2^2 + 5x_1^2 - 7 + \frac{1}{9}|x_2| = 0 \end{cases} \tag{11}$$

where $x_1 \in [-1, 1]$ and $x_2 \in [-10, 10]$. It has four roots: (-0.814326, -1.864719), (0.861828, -1.758100), (-0.814326, 1.864719), and (0.861828, 1.758100).

12) F12:

$$\begin{cases} \sin(x_1^3) - 3x_1 x_2^2 - 1 = 0 \\ \cos(3x_1^2 x_2) - |x_2^3| + 1 = 0 \end{cases} \tag{12}$$

where $x_i \in [-2, 2]$, $i = 1, \ldots, n$, $n = 2$. It has ten roots as shown in Table S-VI..

TABLE S-VI
ROOTS OF F12.

x_1	x_2
-1.810885	-0.349092
-1.810885	0.349092
-1.502221	-0.409077
-1.502221	0.409077
-1.791302	0.301926
-1.791302	-0.301926
-0.947268	0.785020
-0.947268	-0.785020
-0.213057	1.256845
-0.213057	-1.256845

13) F13:

$$\begin{cases} 5x_1^9 - 6x_1^5 x_2^2 + x_1 x_2^4 + 2x_1 x_3 = 0 \\ -2x_1^6 x_2 + 2x_1^2 x_2^3 + 2x_2 x_3 = 0 \\ x_1^2 + x_2^2 - 0.265625 = 0 \end{cases} \tag{13}$$

where $x_1 \in [-0.6, 6]$, $x_2 \in [-0.6, 0.6]$, and $x_3 \in [-5, 5]$. It has 12 roots as shown in Table S-VII.

TABLE S-VII
ROOTS OF F13.

x_1	x_2	x_3
0.279855	0.432789	-0.014189
0.279855	-0.432789	-0.014189
-0.279855	0.432789	-0.014189
-0.279855	-0.432789	-0.014189
0.466980	0.218070	0.000000
-0.466980	0.218070	0.000000
0.466980	-0.218070	0.000000
-0.466980	-0.218070	0.000000
0.000000	0.515388	0.000000
0.000000	-0.515388	0.000000
0.515388	0.000000	-0.012446
-0.515388	0.000000	-0.012446

14) F14:

$$\begin{cases} 4x_1^3 + 4x_1 x_2 + 2x_2^2 - 42x_1 - 14 = 0 \\ 4x_2^3 + 2x_1^2 + 4x_1 x_2 - 26x_2 - 22 = 0 \end{cases} \tag{14}$$

where $x_i \in [-5, 5]$, $i = 1, \ldots, n$, and $n = 2$. It has nine roots as shown in Table S-VIII.

15) F15:

$$\begin{cases} 0.5\sin(x_1 x_2) - \frac{0.25}{\pi} x_2 - 0.5x_1 = 0 \\ \left(1 - \frac{0.25}{\pi}\right)\left[\exp(2x_1) - e\right] + \frac{e}{\pi} x_2 - 2ex_1 = 0 \end{cases} \tag{15}$$

where $x_1 \in [0.25, 1]$ and $x_2 \in [1.5, 2\pi]$. It has two roots: (0.299465, 2.836948) and (0.499966, 3.141589).

TABLE S-VIII
ROOTS OF F14.

x_1	x_2
-0.127961	-1.953715
-0.270845	-0.923039
0.086678	2.884255
3.385154	0.073852
3.584428	-1.848127
3.000000	2.000000
-3.779310	-3.283186
-3.073026	-0.081353
-2.805118	3.131313

16) F16:

$$\begin{cases} -\sin(x1)\cos(x2) - 2\cos(x1)\sin(x2) = 0 \\ -\cos(x1)\sin(x2) - 2\sin(x1)\cos(x2) = 0 \end{cases} \tag{16}$$

where $x_i \in [0, 2\pi]$, $i = 1, \ldots, n$, and $n = 2$. It has 13 roots as shown in Table S-IX.

TABLE S-IX
ROOTS OF F16.

x_1	x_2
0.000000	0.000000
3.141593	0.000000
1.570796	1.570796
6.283185	0.000000
0.000000	3.141593
4.712389	1.570796
3.141593	3.141593
1.570796	4.712389
6.283185	3.141593
0.000000	6.283185
4.712389	4.712389
3.141593	6.283185
6.283185	6.283185

TABLE S-X
ROOTS OF F17.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0.1644	-0.9864	-0.9471	-0.3210	-0.9982	-0.0594	0.4110	0.9116
0.1644	-0.9864	-0.9471	-0.3210	-0.9982	0.0594	0.4110	-0.9116
0.1644	-0.9864	-0.9471	-0.3210	0.9982	-0.0594	0.4110	0.9116
0.1644	-0.9864	-0.9471	-0.3210	0.9982	0.0594	0.4110	-0.9116
0.1644	-0.9864	0.7185	-0.6956	-0.9980	-0.0638	-0.5278	0.8494
0.1644	-0.9864	0.7185	-0.6956	-0.9980	0.0638	-0.5278	-0.8494
0.1644	-0.9864	0.7185	-0.6956	0.9980	-0.0638	-0.5278	0.8494
0.1644	-0.9864	0.7185	-0.6956	0.9980	0.0638	-0.5278	-0.8494
0.6716	0.7410	-0.6516	-0.7586	-0.9625	-0.2711	-0.4376	0.8992
0.6716	0.7410	-0.6516	-0.7586	-0.9625	0.2711	-0.4376	-0.8992
0.6716	0.7410	-0.6516	-0.7586	0.9625	-0.2711	-0.4376	0.8992
0.6716	0.7410	-0.6516	-0.7586	0.9625	0.2711	-0.4376	-0.8992
0.6716	0.7410	0.9519	-0.3064	-0.9638	-0.2666	0.4046	0.9145
0.6716	0.7410	0.9519	-0.3064	-0.9638	0.2666	0.4046	-0.9145
0.6716	0.7410	0.9519	-0.3064	0.9638	0.2666	0.4046	-0.9145
0.6716	0.7410	0.9519	-0.3064	0.9638	-0.2666	0.4046	0.9145

17) F17:

$$\left\{ \begin{array}{rcl} x_1^2 + x_2^2 - 1.0 & = & 0 \\ x_3^2 + x_4^2 - 1.0 & = & 0 \\ x_5^2 + x_6^2 - 1.0 & = & 0 \\ x_7^2 + x_8^2 - 1.0 & = & 0 \\ 4.731 \cdot 10^{-3} x_1 x_3 - 0.3578 x_2 x_3 - 0.1238 x_1 + x_7 & & \\ -1.637 \cdot 10^{-3} x_2 - 0.9338 x_4 - 0.3571 & = & 0 \\ 0.2238 x_1 x_3 + 0.7623 x_2 x_3 + 0.2638 x_1 - x_7 & & \\ -0.07745 x_2 - 0.6734 x_4 - 0.6022 & = & 0 \\ x_6 x_8 + 0.3578 x_1 + 4.731 \cdot 10^{-3} x_2 & = & 0 \\ -0.7623 x_1 + 0.2238 x_2 + 0.3461 & = & 0 \end{array} \right. \tag{17}$$

where $x_i \in [-1, 1]$, $i = 1, \ldots, n$, and $n = 8$. It has 16 roots as shown in Table S-X.

18) F18:

$$\begin{cases} 4x_1^3 - 3x_1 - \cos(x_2) = 0 \\ \sin(x_1^2) - |x_2| = 0 \end{cases} \tag{18}$$

where $x_i \in [-2, 2]$, $i = 1, \ldots, n$, and $n = 2$. It has six roots as shown in Table S-XI.

TABLE S-XI
ROOTS OF F18.

x_1	x_2
-0.597167	-0.349098
-0.442758	-0.194781
-0.442758	0.194781
-0.597167	0.349098
0.964499	-0.801774
0.964499	0.801774

19) F19:

$$\begin{cases} x_i + \sum j = 1^n x_j - (n+1) = 0 \quad i = 1, \ldots, n-1 \\ \left[\prod_{j=1}^n x_j \right] - 1 = 0 \end{cases} \tag{19}$$

where $x_i \in [-2, 2]$, $i = 1, \ldots, n$, and $n = 20$. It has two roots: (1, ..., 1) and (0.994922, ..., 0.994922, 1.101551).

20) F20:

$$x_i - \cos\left(2x_i - \sum_{j=1}^n x_j\right) = 0 \quad i = 1, \ldots, n \tag{20}$$

where $x_i \in [-1, 1]$, $i = 1, \ldots, n$, and $n = 3$. It has seven roots as shown in Table S-XII.

TABLE S-XII
ROOTS OF F20.

x_1	x_2	x_3
0.810561	0.810561	-0.625687
0.810561	-0.625687	0.810561
-0.625687	0.810561	0.810561
0.543850	0.995778	0.543850
0.543850	0.543850	0.995778
0.995778	0.543850	0.543850
0.739086	0.739086	0.739086

21) F21:

$$\begin{cases} x_1^2 + x_2^2 - 2 = 0 \\ x_1^2 + x_2^2/4 - 1 = 0 \end{cases} \tag{21}$$

where $x_i \in [-2, 2]$, $i = 1, \ldots, n$, and $n = 2$. It has four roots: (-0.816497, -1.154701), (0.816497, -1.154701), (-0.816497, 1.154701), and (0.816497, 1.154701).

22) F22:

$$\begin{cases} \exp\left(x_1^2 + x_2^2\right) - 3 = 0 \\ |x_2| + x_1 - \sin(3(|x_2| + x_1)) = 0 \end{cases} \tag{22}$$

where $x_i \in [-2, 2]$, $i = 1, \ldots, n$, and $n = 2$. It has six roots as shown in Table S-XIII.

23) F23:

$$\begin{cases} \beta_{11} + \beta_{12} x_2^2 + \beta_{13} x_3^2 + \beta_{14} x_2 x_3 + \beta_{15} x_2^2 x_3^2 = 0 \\ \beta_{21} + \beta_{22} x_3^2 + \beta_{23} x_1^2 + \beta_{24} x_3 x_1 + \beta_{25} x_3^2 x_1^2 = 0 \\ \beta_{31} + \beta_{32} x_1^2 + \beta_{33} x_2^2 + \beta_{34} x_1 x_2 + \beta_{35} x_1^2 x_2^2 = 0 \end{cases} \tag{23}$$

$$\beta_{ij} = \begin{bmatrix} -13, -1, -1, 24, -1 \\ -13, -1, -1, 24, -1 \\ -13, -1, -1, 24, -1 \end{bmatrix}$$

where $x_i \in [-20, 20]$, $i = 1, \ldots, n$, and $n = 3$. It has 16 roots as shown in Table S-XIV.

TABLE S-XIII
ROOTS OF F22.

x_1	x_2
-0.741152	-0.741152
-0.741152	0.741152
-0.256625	1.016246
-0.256625	-1.016246
-1.016246	-0.256625
-1.016246	0.256625

TABLE S-XIV
ROOTS OF F23.

x_1	x_2	x_3
10.857703600	0.7795480449	0.7795480451
-10.857703600	-0.7795480449	-0.7795480451
0.3320730984	4.6251816010	4.6251816010
-0.3320730984	-4.6251816010	-4.6251816010
0.7795480449	0.7795480449	0.7795480451
-0.7795480449	-0.7795480449	-0.7795480451
0.7795480449	10.857703600	0.7795480451
-0.7795480449	-10.857703600	-0.7795480451
0.7795480440	0.7795480457	10.857703600
-0.7795480440	-0.7795480457	-10.857703600
4.6251816010	4.6251816010	4.6251816010
-4.6251816010	-4.6251816010	-4.6251816010
4.6251816010	0.3320730984	4.6251816010
-4.6251816010	-0.3320730984	-4.6251816010
4.6251816000	4.6251816130	0.3320730984
-4.6251816000	-4.6251816130	-0.3320730984

24) F24:

$$\begin{cases} -3.84x_1^2 + 3.84x_1 - x_2 = 0 \\ -3.84x_2^2 + 3.84x_2 - x_3 = 0 \\ -3.84x_3^2 + 3.84x_3 - x_1 = 0 \end{cases} \tag{24}$$

where $x_i \in [0,1]$, $i = 1, \ldots, n$, and $n = 3$. It has eight roots as shown in Table S-XV.

TABLE S-XV
ROOTS OF F24.

x_1	x_2	x_3
0.000000	0.000000	0.000000
0.488122	0.959435	0.149452
0.540304	0.953754	0.169399
0.959447	0.149373	0.487917
0.149440	0.488092	0.959440
0.953781	0.169343	0.540157
0.169254	0.539937	0.953788
0.739584	0.739584	0.739574

25) F25:

$$\begin{cases} 3u^2 + 2v - 3x_1 + x1x_2 + x_3^2 - 24 = 0 \\ u - 3v^2 - x_1 + 2x_2 - x_1x_3 + 10 = 0 \\ 2u - v + x_1 - x_2^2 + 2x_3 - 5 = 0 \end{cases} \tag{25}$$

where $u = 3x_1 + x_2 - x_3$, $v = x_1^2 - x_2 + x_3$, and $x_i \in [-3,3]$, $i = 1, \ldots, n$, and $n = 3$. It has two roots: (1.140226, -0.448382, 0.135274) and (1, 2, 3).

26) F26:

$$\begin{cases} x_1^3 - 3x_1x_2^2 - 1 = 0 \\ 3x_1^2x_2 - x_2^3 + 1 = 0 \end{cases} \tag{26}$$

where $x_1 \in [-1,-0.1]$ and $x_2 \in [-2,2]$. It has two roots: (-0.793701, -0.793701) and (-0.290515, 1.084215).

27) F27:

$$\begin{cases} 4x_1^3 - 3x_1 - x_2 = 0 \\ x_1^2 - x_2 = 0 \end{cases} \tag{27}$$

where $x_1 \in [-5, 1.5]$ and $x_2 \in [0,5]$. It has three roots: (-0.75, 0.5625), (0, 0), and (1, 1).

28) F28:

$$\begin{cases} x_1^3 - 3x_1x_2^2 + a_1(2x_1^2 + x_1x_2) + b_1x_2^2 + c_1x_1 + a_2x_2 = 0 \\ 3x_1^2x_2 - x_2^3 - a_1(4x_1x_2 - x_2^2) + b_2x_1^2 + c_2 = 0 \end{cases} \tag{28}$$

where $a_1 = 25, b_1 = 1, c_1 = 2, a_2 = 3, b_2 = 4, c_2 = 5$, $x_1 \in [0,2]$, and $x_2 \in [10,30]$. It has two roots: (1.6359718, 13.8476653) and (0.6277425, 22.2444123).

29) F29:

$$\begin{cases} x_1^2 - x_1 - x_2^2 - x_2 + x_3^2 = 0 \\ \sin(x_2 - \exp(x_1)) = 0 \\ x_3 - \log|x_2| = 0 \end{cases} \tag{29}$$

where $x_1 \in [0,2]$, $x_2 \in [-10,10]$, and $x_3 \in [-1,1]$. It has five roots shown in Table S-XVI.

TABLE S-XVI
ROOTS OF F29.

x_1	x_2	x_3
0.825297	-0.859034	-0.151946
1.299490	0.525835	-0.642769
1.533662	-1.648068	0.499604
1.981360	-2.172180	0.775731
1.983283	0.983378	-0.016762

30) F30:

$$\begin{cases} x_1^4 + 4x_2^4 - 6.0 = 0 \\ x_1^2x_2 - 0.6787 = 0 \end{cases} \tag{30}$$

where $x_1 \in [-2,2]$ and $x_2 \in [0,1.1]$. It has four roots: (-1.563533, 0.277628), (-0.789706, 1.088295), (1.563533, 0.277628), and (0.789706, 1.088295).

S-II. Supplemental Results

TABLE S-R-I
EXPERIMENTAL RESULTS OF RADE, RADE-WoR, RADE-WoD, AND RADE-WoA ON 30 TEST PROBLEMS WITH RESPECT TO THE ROOT RATE AND THE SUCCESS RATE.

Prob.	RR				SR			
	RADE	RADE-WoR	RADE-WoD	RADE-WoA	RADE	RADE-WoR	RADE-WoD	RADE-WoA
F01	**1.0000**	**1.0000**	0.5000	0.9950	**1.00**	**1.00**	0.31	0.99
F02	**0.9900**	0.9891	0.9764	0.8882	**0.90**	0.88	0.80	0.22
F03	**0.9960**	0.7965	0.6860	0.8820	**0.95**	0.38	0.02	0.39
F04	**0.9015**	0.9000	0.6469	0.3200	**0.31**	0.28	0.19	0.00
F05	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**	**1.00**
F06	**0.9900**	0.9875	0.2775	0.4925	**0.93**	0.92	0.01	0.00
F07	**1.0000**	**1.0000**	0.7900	**1.0000**	**1.00**	**1.00**	0.58	**1.00**
F08	**0.9971**	0.9871	0.7143	0.8371	**0.98**	0.91	0.00	0.08
F09	**0.9700**	0.9667	0.5800	0.0000	**0.91**	0.90	0.00	0.00
F10	**1.0000**	**1.0000**	0.5500	0.9950	**1.00**	**1.00**	0.10	0.99
F11	**1.0000**	**1.0000**	0.8875	0.2325	**1.00**	**1.00**	0.64	0.00
F12	**0.6310**	0.6230	0.6120	0.3870	0.00	0.00	0.00	0.00
F13	**0.8908**	0.8683	0.8192	0.5717	**0.19**	0.17	0.15	0.00
F14	**0.9867**	0.9778	0.2122	0.8411	**0.89**	**0.89**	0.00	0.11
F15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**	**1.00**
F16	**0.9954**	0.9823	0.9623	0.6823	**0.94**	0.81	0.82	0.00
F17	**0.9444**	0.9281	0.2394	0.7481	**0.43**	0.31	0.00	0.03
F18	**1.0000**	**1.0000**	0.6550	0.9917	**1.00**	**1.00**	0.01	0.95
F19	**0.7950**	0.7850	0.7900	0.0000	**0.69**	0.67	0.68	0.00
F20	**1.0000**	**1.0000**	0.3471	0.8457	**1.00**	**1.00**	0.00	0.17
F21	**1.0000**	**1.0000**	**1.0000**	0.9950	**1.00**	**1.00**	**1.00**	0.98
F22	**1.0000**	0.9983	0.8133	0.9900	**1.00**	**1.00**	0.30	0.94
F23	**0.5619**	0.5513	0.0850	0.1356	0.00	0.00	0.00	0.00
F24	**0.9988**	**0.9988**	0.4150	0.9175	**0.99**	**0.99**	0.00	0.49
F25	**0.8350**	0.8300	0.0350	0.0000	**0.67**	0.66	0.00	0.00
F26	**1.0000**	**1.0000**	0.9850	**1.0000**	**1.00**	**1.00**	0.97	**1.00**
F27	**0.9967**	**0.9967**	0.6933	0.8000	**0.99**	**0.99**	0.19	0.47
F28	**1.0000**	**1.0000**	0.5000	0.5000	**1.00**	**1.00**	0.00	0.00
F29	**0.9940**	0.9880	0.2260	0.4920	**0.97**	0.94	0.00	0.00
F30	**1.0000**	**1.0000**	0.7075	0.9775	**1.00**	**1.00**	0.30	0.91
Avg.	**0.9491**	0.9385	0.6235	0.6839	**0.8247**	0.7900	0.3023	0.3907

TABLE S-R-II
ROOT RATE OF THE SIX COMPARED METHODS ON 30 TEST PROBLEMS.

Prob.	RADE	NCDE	R-JADE	R-CLPSO	I-HS	MONES
F01	**1.0000**	0.9950	0.7250	0.5150	0.0000	0.9450
F02	0.9900	0.8927	0.9118	0.9209	0.9955	**1.0000**
F03	**0.9960**	0.8293	0.7413	0.9700	0.9920	0.9653
F04	0.9015	0.3200	0.9108	0.3908	0.8246	**0.9346**
F05	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F06	0.9900	0.4825	0.7638	0.4263	**0.9988**	0.1250
F07	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9900
F08	**0.9971**	0.7686	0.8057	0.8486	0.7286	0.4386
F09	**0.9700**	0.0000	0.0800	0.0033	0.4467	0.8867
F10	**1.0000**	**1.0000**	0.8950	0.5950	**1.0000**	0.9650
F11	**1.0000**	0.2300	**1.0000**	0.9200	**1.0000**	0.0000
F12	0.6310	0.3870	0.5360	0.6400	**0.9940**	0.4360
F13	**0.8908**	0.5750	0.7092	0.8650	0.7858	0.5433
F14	0.9867	0.8400	0.3433	0.2756	0.8189	**0.9900**
F15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9800
F16	**0.9954**	0.7508	0.9631	0.9392	0.8338	0.4369
F17	**0.9444**	0.7713	0.0088	0.2250	0.7106	0.1663
F18	**1.0000**	0.9967	0.9850	0.9850	0.9750	0.5083
F19	**0.7950**	0.0000	0.7200	0.0950	0.0000	0.3050
F20	**1.0000**	0.8457	0.6400	0.8829	0.8614	0.5443
F21	**1.0000**	0.9950	**1.0000**	**1.0000**	**1.0000**	0.5350
F22	**1.0000**	0.9850	**1.0000**	**1.0000**	**1.0000**	0.5300
F23	**0.5619**	0.1363	0.3769	0.1269	0.3581	0.1463
F24	**0.9988**	0.9138	0.8688	0.8275	0.7488	0.6400
F25	**0.8350**	0.0000	0.6000	0.3550	0.0550	0.3150
F26	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F27	0.9967	0.7500	**1.0000**	**1.0000**	0.9967	**1.0000**
F28	**1.0000**	0.5000	0.9250	0.5100	0.6250	0.8750
F29	**0.9940**	0.4760	0.7440	0.2780	0.7660	0.8220
F30	**1.0000**	0.9575	0.9975	**1.0000**	**1.0000**	**1.0000**
Avg.	**0.9491**	0.6799	0.7750	0.6865	0.7838	0.6675

TABLE S-R-III
SUCCESS RATE OF THE SIX COMPARED METHODS ON 30 TEST PROBLEMS.

Prob.	RADE	NCDE	R-JADE	R-CLPSO	I-HS	MONES
F01	**1.00**	0.99	0.45	0.21	0.00	0.93
F02	0.90	0.32	0.33	0.39	0.95	**1.00**
F03	**0.95**	0.45	0.00	0.59	0.89	0.58
F04	0.31	0.00	0.47	0.00	0.00	**0.67**
F05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F06	0.93	0.00	0.10	0.00	**0.99**	0.00
F07	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98
F08	**0.98**	0.05	0.03	0.01	0.00	0.00
F09	**0.91**	0.00	0.00	0.00	0.08	0.66
F10	**1.00**	**1.00**	0.80	0.19	**1.00**	0.93
F11	**1.00**	0.00	**1.00**	0.73	**1.00**	0.00
F12	0.00	0.00	0.02	0.00	**0.96**	0.00
F13	**0.19**	0.00	0.00	0.14	0.06	0.00
F14	0.89	0.09	0.00	0.00	0.04	**0.92**
F15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96
F16	**0.94**	0.02	0.60	0.42	0.01	0.00
F17	**0.43**	0.03	0.00	0.00	0.00	0.00
F18	**1.00**	0.98	0.92	0.91	0.86	0.00
F19	**0.69**	0.00	0.66	0.08	0.00	0.07
F20	**1.00**	0.22	0.03	0.31	0.30	0.00
F21	**1.00**	0.98	**1.00**	**1.00**	**1.00**	0.01
F22	**1.00**	0.91	**1.00**	**1.00**	**1.00**	0.00
F23	0.00	0.00	0.00	0.00	0.00	0.00
F24	**0.99**	0.43	0.14	0.17	0.00	0.00
F25	**0.67**	0.00	0.28	0.00	0.00	0.07
F26	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F27	0.99	0.32	**1.00**	**1.00**	0.99	**1.00**
F28	**1.00**	0.00	0.85	0.02	0.25	0.78
F29	**0.97**	0.00	0.16	0.00	0.11	0.13
F30	**1.00**	0.83	0.99	**1.00**	**1.00**	**1.00**
Avg.	**0.8247**	0.3873	0.4943	0.4057	0.5163	0.4230

TABLE S-R-IV
EXPERIMENTAL RESULTS OF RADE WITH DIFFERENT REPULSION TECHNIQUES.

Prob.	RR			SR		
	RADE	RADE-1	RADE-2	RADE	RADE-1	RADE-2
F01	**1.0000**	0.9950	**1.0000**	**1.00**	0.99	**1.00**
F02	**0.9900**	0.9882	0.9873	**0.90**	0.87	0.88
F03	0.9960	**0.9973**	**0.9973**	0.95	**0.96**	**0.96**
F04	**0.9015**	0.8738	0.8792	**0.31**	0.09	0.11
F05	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F06	**0.9900**	0.9875	0.9863	**0.93**	0.91	0.91
F07	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F08	**0.9971**	0.9929	0.9914	**0.98**	0.95	0.94
F09	0.9700	**0.9733**	0.9667	0.91	**0.92**	0.90
F10	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F11	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F12	0.6310	0.6130	**0.6350**	0.00	0.00	0.00
F13	**0.8908**	0.8717	0.8825	0.19	0.17	**0.24**
F14	0.9867	**0.9944**	0.9889	0.89	**0.95**	0.90
F15	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F16	**0.9954**	0.9815	0.9538	**0.94**	0.89	0.53
F17	0.9444	**0.9581**	0.9113	0.43	**0.53**	0.22
F18	**1.0000**	**1.0000**	0.9900	**1.00**	**1.00**	0.94
F19	0.7950	**0.8000**	0.7800	0.69	**0.70**	0.66
F20	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F21	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F22	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F23	0.5619	0.5988	**0.6181**	0.00	0.00	0.00
F24	**0.9988**	**0.9988**	**0.9988**	**0.99**	**0.99**	**0.99**
F25	**0.8350**	0.8300	0.8300	**0.67**	0.66	0.66
F26	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F27	0.9967	**1.0000**	**1.0000**	0.99	**1.00**	**1.00**
F28	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F29	**0.9940**	0.9920	0.9800	**0.97**	0.96	0.90
F30	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
Avg.	**0.9491**	0.9482	0.9459	**0.8247**	0.8180	0.7913

TABLE S-R-V
EXPERIMENTAL RESULTS OF RADE WITH DIFFERENT DIVERSITY PRESERVATION MECHANISMS.

Prob.	RR		SR	
	RADE	RADE-3	RADE	RADE-3
F01	**1.0000**	0.9900	**1.00**	0.98
F02	**0.9900**	0.9445	**0.90**	0.50
F03	**0.9960**	0.9640	**0.95**	0.59
F04	**0.9015**	0.8746	**0.31**	0.07
F05	**1.0000**	**1.0000**	**1.00**	**1.00**
F06	**0.9900**	0.9850	**0.93**	0.89
F07	**1.0000**	**1.0000**	**1.00**	**1.00**
F08	**0.9971**	0.9786	**0.98**	0.85
F09	0.9700	**1.0000**	0.91	**1.00**
F10	**1.0000**	**1.0000**	**1.00**	**1.00**
F11	**1.0000**	**1.0000**	**1.00**	**1.00**
F12	0.6310	**0.8270**	0.00	**0.21**
F13	**0.8908**	0.6925	**0.19**	0.01
F14	**0.9867**	**0.9867**	**0.89**	0.88
F15	**1.0000**	**1.0000**	**1.00**	**1.00**
F16	**0.9954**	0.8462	**0.94**	0.05
F17	**0.9444**	0.8663	**0.43**	0.09
F18	**1.0000**	**1.0000**	**1.00**	**1.00**
F19	0.7950	**0.8750**	0.69	**0.82**
F20	**1.0000**	**1.0000**	**1.00**	**1.00**
F21	**1.0000**	**1.0000**	**1.00**	**1.00**
F22	**1.0000**	0.9950	**1.00**	0.97
F23	**0.5619**	0.5613	0.00	0.00
F24	**0.9988**	0.9813	**0.99**	0.85
F25	0.8350	**0.9250**	0.67	**0.85**
F26	**1.0000**	**1.0000**	**1.00**	**1.00**
F27	0.9967	**1.0000**	0.99	**1.00**
F28	**1.0000**	**1.0000**	**1.00**	**1.00**
F29	**0.9940**	0.9880	**0.97**	0.94
F30	**1.0000**	**1.0000**	**1.00**	**1.00**
Avg.	**0.9491**	0.9427	**0.8247**	0.7517

TABLE S-R-VI
EXPERIMENTAL RESULTS OF RADE WITH DIFFERENT ADAPTIVE PARAMETER CONTROLS.

Prob.	RR			SR		
	RADE	RADE-4	RADE-5	RADE	RADE-4	RADE-5
F01	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F02	**0.9900**	0.9818	0.9873	**0.90**	0.83	0.87
F03	**0.9960**	0.9940	0.9947	**0.95**	0.91	0.92
F04	0.9015	**0.9054**	0.8931	**0.31**	0.20	0.24
F05	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F06	**0.9900**	0.9863	0.9863	**0.93**	0.91	0.89
F07	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F08	0.9971	**0.9986**	0.9829	0.98	**0.99**	0.88
F09	0.9700	**1.0000**	0.0800	0.91	**1.00**	0.03
F10	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F11	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F12	0.6310	**0.6880**	0.6150	0.00	**0.01**	**0.01**
F13	0.8908	**0.9567**	0.8117	0.19	**0.58**	0.07
F14	0.9867	**0.9889**	**0.9889**	0.89	**0.91**	0.90
F15	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F16	**0.9954**	0.9469	0.9762	**0.94**	0.46	0.73
F17	**0.9444**	0.9250	0.8181	**0.43**	0.35	0.04
F18	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F19	0.7950	**1.0000**	0.2650	0.69	**1.00**	0.18
F20	**1.0000**	0.9986	0.9986	**1.00**	0.99	0.99
F21	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F22	**1.0000**	**1.0000**	0.9967	**1.00**	**1.00**	0.98
F23	0.5619	**0.5813**	0.5300	0.00	0.00	0.00
F24	0.9988	0.9975	**1.0000**	0.99	0.98	**1.00**
F25	0.8350	0.9050	**0.9100**	0.67	0.81	**0.82**
F26	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F27	0.9967	**1.0000**	**1.0000**	0.99	**1.00**	**1.00**
F28	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
F29	**0.9940**	0.9580	0.9800	**0.97**	0.79	0.90
F30	**1.0000**	**1.0000**	**1.0000**	**1.00**	**1.00**	**1.00**
Avg.	0.9491	**0.9604**	0.8938	**0.8247**	0.8240	0.7483

TABLE S-R-VII
EXPERIMENTAL RESULTS OF RADE WITH DIFFERENT DIFFERENT MUTATION OPERATORS WITH RESPECT TO THE ROOT RATE.

Prob.	RADE	RADE-6	RADE-7	RADE-8	RADE-9	SaDE
F01	**1.0000**	0.5050	**1.0000**	0.5150	0.0000	0.9700
F02	**0.9900**	0.4782	0.9791	0.5291	0.9873	0.9882
F03	**0.9960**	0.5440	0.9327	0.5533	0.9647	0.9920
F04	0.9015	0.1585	0.6408	0.1708	0.6685	**0.9408**
F05	**1.0000**	**1.0000**	**1.0000**	0.9900	0.8200	**1.0000**
F06	**0.9900**	0.3475	0.8738	0.3613	0.9363	0.9513
F07	**1.0000**	0.9900	**1.0000**	0.9950	**1.0000**	**1.0000**
F08	**0.9971**	0.7900	0.9443	0.8143	0.9586	0.9829
F09	**0.9700**	0.6500	0.4433	0.6367	0.6567	0.7567
F10	**1.0000**	0.5350	**1.0000**	0.5650	0.9950	0.9950
F11	**1.0000**	0.3225	**1.0000**	0.3250	**1.0000**	**1.0000**
F12	0.6310	0.2160	0.4680	0.2750	0.6730	**0.7280**
F13	0.8908	0.7733	0.7975	0.9083	0.6008	**0.9650**
F14	**0.9867**	0.1222	0.9133	0.1256	0.8767	0.9544
F15	**1.0000**	**1.0000**	**1.0000**	0.9900	**1.0000**	**1.0000**
F16	**0.9954**	0.2046	0.5331	0.2415	0.5415	0.9615
F17	0.9444	0.2644	0.5913	0.2794	0.0000	**0.9456**
F18	**1.0000**	0.8633	0.9983	0.8750	0.9967	**1.0000**
F19	0.7950	**0.8700**	0.0250	0.6550	0.0000	0.8400
F20	**1.0000**	0.7714	0.9586	0.7971	0.8371	0.9957
F21	**1.0000**	0.6550	**1.0000**	0.6775	**1.0000**	**1.0000**
F22	**1.0000**	0.4933	0.9850	0.5400	0.9600	0.9867
F23	**0.5619**	0.0644	0.4838	0.0656	0.3775	0.4781
F24	**0.9988**	0.4300	0.9925	0.4925	0.9213	0.9525
F25	**0.8350**	0.1650	0.5400	0.1750	0.4750	0.8300
F26	**1.0000**	0.5200	**1.0000**	0.5300	**1.0000**	**1.0000**
F27	**0.9967**	0.5667	0.9733	0.5867	0.9033	0.9933
F28	**1.0000**	0.5000	**1.0000**	0.5000	**1.0000**	**1.0000**
F29	**0.9940**	0.3880	0.7940	0.4040	0.5920	0.9420
F30	**1.0000**	0.2950	**1.0000**	0.2925	**1.0000**	**1.0000**
Avg.	**0.9491**	0.5161	0.8289	0.5289	0.7581	0.9383

TABLE S-R-VIII
EXPERIMENTAL RESULTS OF RADE WITH DIFFERENT DIFFERENT MUTATION OPERATORS WITH RESPECT TO THE SUCCESS RATE.

Prob.	RADE	RADE-6	RADE-7	RADE-8	RADE-9	SaDE
F01	**1.00**	0.01	**1.00**	0.03	0.00	0.95
F02	0.90	0.00	0.84	0.00	0.89	**0.92**
F03	**0.95**	0.00	0.37	0.00	0.57	0.90
F04	0.31	0.00	0.00	0.00	0.00	**0.42**
F05	**1.00**	**1.00**	**1.00**	0.99	0.82	**1.00**
F06	**0.93**	0.00	0.27	0.01	0.57	0.70
F07	**1.00**	0.98	**1.00**	0.99	**1.00**	**1.00**
F08	**0.98**	0.26	0.63	0.33	0.77	0.88
F09	**0.91**	0.00	0.08	0.00	0.00	0.27
F10	**1.00**	0.07	**1.00**	0.13	0.99	0.99
F11	**1.00**	0.00	**1.00**	0.00	**1.00**	**1.00**
F12	0.00	0.00	0.00	0.00	0.05	**0.21**
F13	0.19	0.03	0.02	0.41	0.00	**0.68**
F14	**0.89**	0.00	0.39	0.00	0.26	0.66
F15	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**
F16	**0.94**	0.00	0.00	0.00	0.00	0.57
F17	**0.43**	0.00	0.00	0.00	0.00	**0.43**
F18	**1.00**	0.47	0.99	0.50	0.98	**1.00**
F19	0.69	**0.74**	0.01	0.34	0.00	0.70
F20	**1.00**	0.11	0.73	0.10	0.25	0.97
F21	**1.00**	0.41	**1.00**	0.45	**1.00**	**1.00**
F22	**1.00**	0.08	0.92	0.05	0.78	0.92
F23	0.00	0.00	0.00	0.00	0.00	0.00
F24	**0.99**	0.00	0.94	0.02	0.40	0.62
F25	**0.67**	0.00	0.08	0.00	0.00	**0.67**
F26	**1.00**	0.04	**1.00**	0.06	**1.00**	**1.00**
F27	**0.99**	0.16	0.92	0.17	0.71	0.98
F28	**1.00**	0.00	**1.00**	0.00	**1.00**	**1.00**
F29	**0.97**	0.00	0.03	0.00	0.00	0.73
F30	**1.00**	0.00	**1.00**	0.00	**1.00**	**1.00**
Avg.	**0.8247**	0.1787	0.5740	0.1853	0.5013	0.7723

TABLE S-R-IX

INFLUENCE OF DIFFERENT NP VALUES IN RADE WITH RESPECT TO THE ROOT RATE.

Prob.	$NP = 50$	$NP = 80$	$NP = 100$	$NP = 120$	$NP = 150$	$NP = 200$
F01	0.9050	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F02	0.9873	0.9855	0.9900	0.9911	0.9930	**0.9935**
F03	0.9920	0.9920	0.9960	0.9987	0.9993	**1.0000**
F04	**0.9300**	0.9200	0.9015	0.8708	0.8446	0.8462
F05	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F06	0.9650	**0.9938**	0.9900	0.9913	0.9875	0.9825
F07	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F08	0.9243	0.9900	0.9971	**1.0000**	0.9971	**1.0000**
F09	0.9933	**1.0000**	0.9700	0.8500	0.6367	0.1900
F10	0.9950	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F11	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F12	0.6290	0.6130	0.6310	0.7190	0.7250	**0.7280**
F13	0.8042	0.8350	0.8908	0.9242	0.9417	**0.9558**
F14	0.9711	0.9856	0.9867	0.9967	**1.0000**	**1.0000**
F15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F16	0.9938	**0.9954**	**0.9954**	0.9831	0.9115	0.8523
F17	0.9731	0.9388	0.9444	0.9463	0.9625	**0.9906**
F18	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F19	**0.9950**	0.9550	0.7950	0.7500	0.1300	0.0000
F20	0.9986	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F21	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F22	0.9800	0.9950	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F23	0.2725	0.5256	0.5619	0.6000	0.6644	**0.7694**
F24	0.9888	**1.0000**	0.9988	0.9988	**1.0000**	**1.0000**
F25	0.6850	0.7450	0.8350	**0.9000**	0.8400	0.6400
F26	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F27	0.9867	**1.0000**	0.9967	**1.0000**	0.9967	**1.0000**
F28	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.7400
F29	0.9660	0.9920	**0.9940**	0.9820	0.9720	0.9480
F30	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Avg.	0.9312	0.9487	0.9491	**0.9501**	0.9201	0.8879

TABLE S-R-X

INFLUENCE OF DIFFERENT NP VALUES IN RADE WITH RESPECT TO THE SUCCESS RATE.

Prob.	$NP = 50$	$NP = 80$	$NP = 100$	$NP = 120$	$NP = 150$	$NP = 200$
F01	0.81	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F02	0.88	0.88	0.90	0.92	0.93	**0.94**
F03	0.89	0.90	0.95	0.98	0.99	**1.00**
F04	**0.50**	0.39	0.31	0.08	0.02	0.01
F05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F06	0.84	**0.96**	0.93	0.93	0.91	0.87
F07	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F08	0.49	0.93	0.98	**1.00**	0.98	**1.00**
F09	0.98	**1.00**	0.91	0.56	0.14	0.00
F10	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F11	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F12	0.00	0.00	0.00	0.09	**0.15**	0.08
F13	0.14	0.12	0.19	0.26	0.35	**0.45**
F14	0.77	0.83	0.89	0.97	**1.00**	**1.00**
F15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F16	0.92	**0.94**	**0.94**	0.78	0.24	0.08
F17	0.69	0.42	0.43	0.41	0.52	**0.87**
F18	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F19	**0.99**	0.91	0.69	0.64	0.09	0.00
F20	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F21	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F22	0.88	0.97	**1.00**	**1.00**	**1.00**	**1.00**
F23	0.00	0.00	0.00	0.00	0.00	0.00
F24	0.91	**1.00**	0.99	0.99	**1.00**	**1.00**
F25	0.37	0.49	0.67	**0.80**	0.68	0.28
F26	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F27	0.96	**1.00**	0.99	**1.00**	0.99	**1.00**
F28	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.48
F29	0.87	0.96	**0.97**	0.91	0.86	0.74
F30	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Avg.	0.7957	0.8233	**0.8247**	0.8107	0.7617	0.7267

TABLE S-R-XI
INFLUENCE OF DIFFERENT H_m VALUES IN RADE WITH RESPECT TO THE ROOT RATE.

Prob. \ H_m	5	10	30	50	100	200	300	400	500	800	1000
F01	**1.0000**	**1.0000**	**1.0000**	0.9950	**1.0000**	**1.0000**	0.9900	0.9950	**1.0000**	**1.0000**	0.9900
F02	0.9745	0.9855	0.9873	0.9827	0.9864	0.9900	**0.9936**	0.9900	0.9882	**0.9936**	0.9891
F03	0.9933	0.9933	0.9947	0.9980	0.9967	0.9960	0.9927	0.9953	0.9980	**0.9987**	0.9953
F04	0.8846	0.8977	0.8769	0.8900	0.8977	0.9015	0.8977	0.9077	0.9146	**0.9269**	0.9223
F05	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F06	0.9913	**0.9950**	0.9875	0.9925	0.9925	0.9900	0.9875	0.9925	0.9925	0.9900	0.9913
F07	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F08	0.9871	0.9914	0.9857	0.9914	0.9914	0.9971	0.9971	**0.9986**	**0.9986**	0.9971	0.9971
F09	0.0000	0.0033	0.4367	0.8333	0.9300	0.9700	0.9867	0.9900	**1.0000**	**1.0000**	0.9933
F10	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F11	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F12	0.6010	0.6280	0.6050	0.6050	0.6150	0.6310	0.6620	**0.6680**	0.6630	0.6510	0.6520
F13	0.7492	0.7700	0.8133	0.8408	0.8658	0.8908	0.8942	0.9175	0.9300	**0.9375**	0.9292
F14	0.9811	0.9844	0.9889	0.9922	0.9889	0.9867	0.9911	0.9889	0.9911	**0.9933**	0.9922
F15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F16	0.9092	0.9300	0.9923	0.9938	0.9931	0.9954	0.9892	**0.9969**	0.9938	0.9900	0.9908
F17	0.8375	0.8488	0.8588	0.8713	0.9075	0.9444	0.9425	0.9475	**0.9556**	0.9531	0.9519
F18	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F19	0.3500	0.0600	0.0550	0.0900	0.3100	0.7950	0.9700	0.9700	0.9900	**1.0000**	0.9950
F20	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F21	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F22	0.9983	0.9983	**1.0000**	0.9983	0.9983	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F23	0.3131	0.3144	0.4631	0.5488	0.5594	0.5619	0.5769	0.5775	0.5819	**0.6013**	0.5988
F24	**1.0000**	0.9988	0.9988	0.9975	0.9988	0.9988	0.9975	0.9988	0.9988	0.9988	**1.0000**
F25	**0.9400**	0.9200	0.9100	0.8600	0.8400	0.8350	0.8350	0.8300	0.8050	0.8550	0.8400
F26	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F27	**1.0000**	0.9967	**1.0000**	**1.0000**	**1.0000**	0.9967	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F28	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F29	0.9400	0.9680	0.9740	0.9800	0.9760	**0.9940**	0.9840	0.9900	**0.9940**	0.9800	0.9820
F30	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Avg.	0.8817	0.8761	0.8976	0.9154	0.9282	0.9491	0.9563	0.9585	0.9598	**0.9622**	0.9603

TABLE S-R-XII
INFLUENCE OF DIFFERENT H_m VALUES IN RADE WITH RESPECT TO THE SUCCESS RATE.

Prob. \ H_m	5	10	30	50	100	200	300	400	500	800	1000
F01	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**	0.98	0.99	**1.00**	**1.00**	0.98
F02	0.73	0.84	0.88	0.81	0.85	0.90	**0.93**	0.89	0.88	**0.93**	0.89
F03	0.92	0.93	0.93	0.97	0.96	0.95	0.91	0.93	0.97	**0.98**	0.95
F04	0.19	0.26	0.14	0.25	0.29	0.31	0.28	0.27	0.30	**0.37**	0.29
F05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F06	0.93	**0.96**	0.91	0.94	0.94	0.93	0.91	0.94	0.94	0.92	0.93
F07	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F08	0.91	0.94	0.90	0.95	0.94	0.98	0.98	**0.99**	**0.99**	0.98	0.98
F09	0.00	0.00	0.20	0.63	0.80	0.91	0.96	0.97	**1.00**	**1.00**	0.98
F10	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F11	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F12	0.01	0.01	0.01	0.00	**0.03**	0.00	0.01	0.00	0.01	0.00	0.00
F13	0.01	0.04	0.08	0.10	0.16	0.19	0.24	0.34	**0.45**	**0.45**	0.42
F14	0.84	0.86	0.91	0.93	0.90	0.89	0.92	0.90	0.92	**0.94**	0.93
F15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F16	0.21	0.34	0.91	0.92	0.93	0.94	0.88	**0.96**	0.92	0.88	0.80
F17	0.06	0.03	0.05	0.16	0.21	0.43	0.43	0.45	**0.56**	0.46	0.47
F18	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F19	0.34	0.05	0.03	0.06	0.22	0.69	0.94	0.94	0.98	**1.00**	0.99
F20	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F21	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F22	0.99	0.99	**1.00**	0.99	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F23	0.00	0.00	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F24	**1.00**	0.99	0.99	0.98	0.99	0.99	0.98	0.99	0.99	0.99	**1.00**
F25	**0.88**	0.84	0.83	0.72	0.68	0.67	0.67	0.66	0.61	0.71	0.68
F26	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F27	**1.00**	0.99	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F28	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F29	0.75	0.86	0.88	0.90	0.88	**0.97**	0.92	0.95	**0.97**	0.90	0.91
F30	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Avg.	0.7257	0.7310	0.7553	0.7767	0.7923	0.8247	0.8313	0.8390	0.8497	**0.8503**	0.8400

TABLE S-R-XIII
INFLUENCE OF DIFFERENT α VALUES IN RADE WITH RESPECT TO THE ROOT RATE.

Prob.	$\alpha = 1$	$\alpha = 2$	$\alpha = 5$	$\alpha = 10$	$\alpha = 20$	$\alpha = 50$	$\alpha = 100$
F01	**1.0000**	0.9950	0.9900	**1.0000**	0.9950	0.9950	0.9950
F02	**0.9955**	0.9927	0.9873	0.9900	0.9855	0.9855	0.9855
F03	0.9973	**1.0000**	0.9953	0.9960	0.9953	0.9987	0.9967
F04	**0.9085**	0.8938	0.9000	0.9015	0.9008	0.9054	0.9008
F05	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F06	0.9850	0.9913	**0.9925**	0.9900	**0.9925**	0.9875	0.9863
F07	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F08	**1.0000**	0.9957	0.9986	0.9971	0.9900	0.9843	0.9914
F09	**0.9800**	0.9767	0.9733	0.9700	0.9600	0.9600	0.9767
F10	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F11	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F12	**0.6490**	0.6460	0.6300	0.6310	0.6340	0.6300	0.6440
F13	0.8883	**0.9375**	0.9275	0.8908	0.8733	0.8650	0.8717
F14	**0.9922**	0.9911	0.9911	0.9867	0.9911	0.9911	0.9889
F15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F16	0.9738	0.9885	0.9946	0.9954	**0.9992**	0.9923	0.9908
F17	**0.9950**	0.9488	0.9288	0.9444	0.9431	0.9450	0.9494
F18	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F19	0.7650	0.7600	0.7650	**0.7950**	0.7750	0.7750	**0.7950**
F20	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F21	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F22	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F23	**0.5819**	0.5656	0.5706	0.5619	0.5731	0.5713	0.5756
F24	**1.0000**	**1.0000**	**1.0000**	0.9988	0.9988	0.9988	0.9988
F25	0.8300	0.8300	0.8300	**0.8350**	0.8250	0.8250	0.8150
F26	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F27	0.9967	**1.0000**	**1.0000**	0.9967	0.9967	**1.0000**	**1.0000**
F28	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F29	0.9920	0.9840	0.9920	**0.9940**	0.9840	0.9900	0.9920
F30	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Avg.	**0.9510**	0.9499	0.9489	0.9491	0.9471	0.9467	0.9484

TABLE S-R-XIV
INFLUENCE OF DIFFERENT α VALUES IN RADE WITH RESPECT TO THE SUCCESS RATE.

Prob.	$\alpha = 1$	$\alpha = 2$	$\alpha = 5$	$\alpha = 10$	$\alpha = 20$	$\alpha = 50$	$\alpha = 100$
F01	**1.00**	0.99	0.98	**1.00**	0.99	0.99	0.99
F02	**0.95**	0.92	0.86	0.90	0.85	0.84	0.84
F03	0.96	**1.00**	0.95	0.95	0.94	0.98	0.96
F04	0.26	0.22	0.21	**0.31**	0.21	0.25	0.26
F05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F06	0.90	**0.95**	**0.95**	0.93	0.94	0.91	0.92
F07	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F08	**1.00**	0.98	0.99	0.98	0.93	0.90	0.94
F09	**0.94**	0.93	0.92	0.91	0.88	0.88	0.93
F10	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F11	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F12	0.00	0.00	0.00	0.00	**0.01**	0.00	0.00
F13	0.19	0.42	**0.44**	0.19	0.19	0.11	0.21
F14	**0.93**	0.92	0.92	0.89	0.92	0.92	0.90
F15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F16	0.70	0.88	0.93	0.94	**0.99**	0.91	0.88
F17	**0.92**	0.43	0.30	0.43	0.38	0.46	0.40
F18	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F19	0.63	0.62	0.63	**0.69**	0.65	0.65	**0.69**
F20	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F21	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F22	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F23	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F24	**1.00**	**1.00**	**1.00**	0.99	0.99	0.99	0.99
F25	0.66	0.66	0.66	**0.67**	0.65	0.65	0.63
F26	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F27	0.99	**1.00**	**1.00**	0.99	0.99	**1.00**	**1.00**
F28	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F29	0.96	0.92	0.96	**0.97**	0.92	0.95	0.96
F30	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Avg.	**0.8330**	0.8280	0.8233	0.8247	0.8143	0.8130	0.8167

TABLE S-R-XV
INFLUENCE OF DIFFERENT ϵ VALUES IN RADE WITH RESPECT TO THE ROOT RATE.

Prob.	ϵ = 1E-08	ϵ = 1E-10	ϵ = 1E-15	ϵ = 1E-20	ϵ = 0
F01	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F02	**0.9900**	**0.9900**	**0.9900**	**0.9900**	**0.9900**
F03	0.9953	0.9960	**0.9973**	**0.9973**	**0.9973**
F04	0.8962	**0.9015**	**0.9015**	**0.9015**	0.8985
F05	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F06	**0.9913**	0.9900	0.9875	0.9875	0.9875
F07	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F08	0.9971	0.9971	**0.9986**	**0.9986**	0.9971
F09	0.9667	0.9700	**0.9733**	**0.9733**	**0.9733**
F10	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F11	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F12	**0.6500**	0.6310	0.6400	0.6400	0.6400
F13	**0.9133**	0.8908	0.8908	0.8908	0.8908
F14	**0.9078**	0.9867	0.9878	0.9867	0.9867
F15	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F16	0.9946	0.9954	**0.9969**	**0.9969**	**0.9969**
F17	**0.9444**	**0.9444**	**0.9444**	**0.9444**	**0.9444**
F18	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F19	**0.7950**	**0.7950**	**0.7950**	**0.7950**	**0.7950**
F20	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F21	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F22	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F23	0.5750	0.5619	0.5738	0.5875	**0.6038**
F24	**0.9988**	**0.9988**	**0.9988**	**0.9988**	**0.9988**
F25	**0.8400**	0.8350	0.8350	0.8300	0.8300
F26	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F27	**1.0000**	0.9967	0.9967	0.9967	0.9967
F28	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
F29	0.9920	0.9940	0.9920	**0.9960**	**0.9960**
F30	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Avg.	0.9482	0.9491	0.9500	0.9504	**0.9508**

TABLE S-R-XVI
INFLUENCE OF DIFFERENT ϵ VALUES IN RADE WITH RESPECT TO THE SUCCESS RATE.

Prob.	ϵ = 1E-08	ϵ = 1E-10	ϵ = 1E-15	ϵ = 1E-20	ϵ = 0
F01	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F02	**0.90**	**0.90**	**0.90**	**0.90**	**0.90**
F03	0.95	0.95	**0.97**	**0.97**	**0.97**
F04	0.23	**0.31**	0.28	0.24	0.23
F05	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F06	**0.94**	0.93	0.91	0.91	0.91
F07	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F08	0.98	0.98	**0.99**	**0.99**	0.98
F09	0.90	0.91	**0.92**	**0.92**	**0.92**
F10	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F11	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F12	0.00	0.00	0.00	0.00	0.00
F13	**0.25**	0.19	0.21	0.20	0.20
F14	**0.91**	0.89	0.90	0.89	0.89
F15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F16	0.93	0.94	**0.96**	**0.96**	**0.96**
F17	0.42	**0.43**	**0.43**	**0.43**	**0.43**
F18	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F19	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**
F20	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F21	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F22	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F23	0.00	0.00	0.00	0.00	0.00
F24	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
F25	**0.68**	0.67	0.67	0.66	0.66
F26	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F27	**1.00**	0.99	0.99	0.99	0.99
F28	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
F29	0.96	0.97	0.96	**0.98**	**0.98**
F30	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Avg.	0.8243	0.8247	**0.8257**	0.8240	0.8233