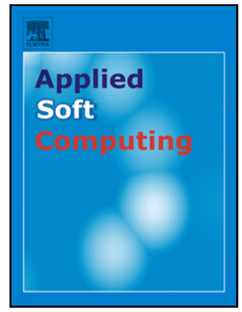
Accepted Manuscript



Title: Dynamic multi-objective evolutionary algorithms for single-objective optimization

Author: Ruwang Jiao Sanyou Zeng Jawdat S. Alkasassbeh Changhe Li

PII: S1568-4946(17)30512-4
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.08.030
Reference: ASOC 4420

To appear in: *Applied Soft Computing*

Received date: 11-1-2017
Revised date: 6-8-2017
Accepted date: 16-8-2017

1. Convert a single-objective optimization problem with many local optima into an equivalent dynamic multi-objective optimization problem.
2. The converted two objectives include the original objective and a niche-count objective, niche-count aims to maintain the population diversity.
3. The niche radius provides a proper balance between the exploitation and the exploration by gradually pushing the niche radius to zero.

Dynamic multi-objective evolutionary algorithms for single-objective optimization

Ruwang Jiao[a], Sanyou Zeng[a,*], Jawdat S. Alkasassbeh[a], Changhe Li[b]

[a] *School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China*
[b] *School of Automation, China University of Geosciences, Wuhan 430074, China*

Abstract

This paper proposes a new method for handling the difficulty of multi-modality for the single-objective optimization problem (SOP). The method converts a SOP to an equivalent dynamic multi-objective optimization problem (DMOP). A new dynamic multi-objective evolutionary algorithm (DMOEA) is implemented to solve the DMOP. The DMOP has two objectives: the original objective and a niche-count objective. The second objective aims to maintain the population diversity for handling the multi-modality difficulty during the search process. Experimental results show that the performance of the proposed algorithm is significantly better than the state-of-the-art competitors on a set of benchmark problems and real world antenna array problems.

Keywords: Evolutionary computation, Multi-objective optimization, Niching, Dynamic optimization, Antenna array.

1. Introduction

Evolutionary algorithms (EAs) have been widely applied to single-objective optimization problems (SOPs). However, given a problem with many local optima, conventional EAs yet have a great tendency to trap in local optima. A crucial issue is the trade-off between the population convergence/exploitation and the population diversity/exploration. Topics on those trade-offs are widely discussed [1]. Techniques for handling the trade-offs, mostly niching techniques as surveyed in [2], are employed to maintain the diversity, such as fitness sharing [3], crowding [4],[5], restricted tournament selection [6], clearing [7], and speciation [8].

Simultaneously, for the last twenty years, a lot of efforts have been made in the area of EAs for multi-objective optimization. Several effective MOEAs have been proposed such as NSGA-II [9], MOEA/D [10], and HypE [11]. To utilize these MOEAs, a newly proposed niching multi-objective technique restates a SOP as a multi-objective optimization problem (MOP) where the diversity measurement function is used as an additional objective([12], [13]). Then an existing MOEA can be applied to solve multi-modal SOP. One issue of this technique is that is not equivalent to the original SOP.

*Corresponding author
Email address: sanyouzeng@gmail.com (Sanyou Zeng)

This paper converts a SOP into an equivalent dynamic multi-objective optimization problem (DMOP) with two objectives: the original objective and an additional niche-count objective. The niche radius gradually decreases to zero as the the environmental changes. Then an existing DMOEA (a dynamic version of an existing MOEA) is applied to solve the DMOP. Minimization of the original objective is beneficial to the population convergence while minimization of the niche-count objective is beneficial to the population diversity. The Pareto domination between the two conflict objectives can maintain a certain population diversity. Through the whole evolutionary process, the gradually-reducing-to-zero niche radius can drive the trade-off between the convergence and the diversity.

The remainder of this paper is organized as follows. Section 2 summarizes some related works. Section 3 introduces our proposed method for converting a SOP into an equivalent DMOP. Section 4 presents a framework of DMOEA for the DMOP, an instantiation of the DMOEA framework denoted as DNSGAII-DE is also introduced. Section 5 presents the simulation results and discussions. Section 6 concludes this paper and briefly discusses some future work.

2. Related work

2.1. Solving SOPs by MOEAs

Using multi-objective methods to enhance the population diversity for SOPs is a promising approach since multi-objective schemes try to simultaneously optimize several objectives. Using diversity as an additional objective might provide a proper balance between exploration and exploitation. In fact, several studies have analyzed the use of MOEAs to promote diversity maintenance in SOPs. A comprehensive survey with detailed discussions of using MOEAs for SOPs can be found in [14]. In this subsection, we briefly review the related work on using multi-objective techniques for SOPs.

De Jong [15] exploited the potential of MOEAs by adding a diversity-related function among the objectives to optimize and applied to genetic programming in order to limit the growth of solutions over time, where only non-dominated individuals are preserved in the population. Abbass and Deb [16] proposed to add an additional objective to promote diversity. Three different artificial objectives were discussed in their work: time-stamp, random and inversion. Toffolo [17] presented a diversity-preserving mechanism called the Genetic Diversity Evaluation Method (GeDEM), which considers a distance-based measure of genetic diversity as a real objective in fitness assignment. This provides a dual selection pressure towards the exploitation of current non-dominated solutions and the exploration of the search space. Jensen [18],[19] proposed the concept of helper-objectives to introduce new objectives for the optimization process, which are used to guide the search towards solutions containing good building blocks and to help the algorithm escape from local optima. Carlos et al.[20] proposed an improved diversity preservation of multi-objective approaches, which provides several advantages in terms of premature convergence avoidance. Segredo et al. [21] focused on the application of parameter control approaches to diversity-based MOEAs. They investigated the methods of Fuzzy Logic and Hyper-heuristic, and results showed the benefits of solving SOPs with diversity-based multi-objective schemes.

Wang et al. [22] transformed a multimodal optimization problem (MMOP) into a MOP with two conflicting objectives. However, a large number of variables may impact the performance of the algorithm. Similarly, Ji [23] constructed a bi-objective optimization problem for a MMOP, the two objectives are constructed conflict by using the distance information and the fitness. To adapt the balance between exploration and exploitation, Segura et al.[24] applied a replacement-based diversity management strategy to transform a SOP into a MOP by considering diversity as an additional objective.

2.2. Solving SOPs with fitness sharing

Compared with the conventional EAs, niching EAs increase the diversity of the population, and decreases the probability of getting trapped in local optima. Niching EAs are able to search multiple global/local peaks in a single run. Among all of the niching techniques, fitness sharing [3],[25] is a well-known niching technique, and a variety of modified fitness sharing schemes have been proposed.

Jelasity [26] suggested a cooling-based mechanism for the niche radius, which adapts the global radius as a function of time during the evolution. Della et al. [27] proposed a characterization of the behavior of an EA with fitness sharing in terms of the mean and the standard deviation of the number of niches. A modified species-based DE (SDE) with a self-adaptive radius was proposed by Qu and Suganthan [28] to overcome the difficulty of selecting the proper radius. Li [29] built a normal model of fitness sharing with a proportionate selection of real-valued functions. The normal model constitutes a platform to investigate the dynamic behavior of the fitness sharing EAs with regard to the niche radius. Shir et al.[30] applied an adaptive individual niche radius to niching with the covariance matrix adaptation evolution strategy (CMA-ES) to solve multi-global optimization problems. They coupled the radius to the step size mechanism, and employed the Mahalanobis distance metric with the covariance matrix mechanism for the distance calculation, for obtaining niches with more complex geometrical shapes.

2.3. Discussions

Compared with the work of this paper, the aforementioned studies convert a SOP into an unequivalent MOP, then an EA adopting the concept of Pareto domination, Pareto ranking or other multi-objective techniques are applied to solve the SOP, rather than the MOP. Furthermore, one crucial issue of the fitness sharing is the usage of the niche radius, specifying this parameter requires a priori knowledge of how far apart optima are. Although some aforementioned works adopt adaptive niche radius strategies, the niche radius does not converge to zero which cannot guarantee the convergence of the population. They are always used to locate multiple optima rather than one global optimum. There are three significant differences between this study and previous works:

1. A SOP is converted into an equivalent DMOP. A DMOEA is implemented to solve the DMOP, thus the SOP is solved.

2. The balance between exploration and exploitation can be adjusted by the niche radius. The aim of exploration is to find promising basins of attraction, and the aim of exploitation is to find the best solution within these promising basins. In this study, the niche radius is slightly reduced from an initial value to zero, which achieves a trade-off between exploration and exploitation.
3. Combined with multi-objective, niching and dynamic techniques, this study aims to find a global optimum rather multiple optima.

3. Converting SOPs to DMOPs

3.1. Single-objective Optimization Problem

Without loss of generality, minimization optimization is assumed in this paper.

Definition 1. *(**SOP**) A general SOP includes a set of n variables and an objective which is a function of the variables. The goal is to*

$$\begin{aligned} \min \quad & y = f(\mathbf{x}) \\ \textit{where} \quad & \mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbf{X} \\ & \mathbf{X} = \{\mathbf{x} | \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \\ & \mathbf{l} = (l_1, l_2, ..., l_n), \mathbf{u} = (u_1, u_2, ..., u_n) \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the solution vector (solution for short) and $\mathbf{X}$ denotes the solution space, $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds of the solution space.

In this paper, the solution space $\mathbf{X} = [\mathbf{l}, \mathbf{u}]$ is normalized into a space $\mathbf{X}^{'} = [\mathbf{0}, \mathbf{1}]$ by

$$\mathbf{x}_i^{'} = \frac{\mathbf{x}_i - \mathbf{l}_i}{\mathbf{u}_i - \mathbf{l}_i} \tag{2}$$

3.2. Conversion of the SOP to the DMOP

To prevent an EA from getting trapped in local optima, an additional mechanism for maintaining the population diversity is usually introduced. One mechanism was proposed by Goldberg [3], namely niche count (sum of sharing function), which is introduced as follows.

The sharing function between two solutions $\mathbf{x}_1 = (x_{11}, x_{12}, \cdots, x_{1n})$ and $\mathbf{x}_2 = (x_{21}, x_{22}, \cdots, x_{2n})$, is defined as follows with a niche, a super-sphere with a radius σ:

$$sh(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} 1 - (\frac{d(\mathbf{x}_1, \mathbf{x}_2)}{\sigma}), & d(\mathbf{x}_1, \mathbf{x}_2) \leq \sigma \\ 0, & \textit{otherwise} \end{cases} \tag{3}$$

where $d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$.

The niche-count is then defined as follows.

Definition 2. *(**Niche-count**) Given $\mathbf{U} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L\}$ (the union of the parent and offspring populations) and σ (the niche radius), the niche-count of $\mathbf{x} \in \mathbf{U}$ is defined as:*

$$nc(\mathbf{x}|\mathbf{U}, \sigma) = \sum_{i=1, \mathbf{x}_i \neq \mathbf{x}}^{L} sh(\mathbf{x}, \mathbf{x}_i). \tag{4}$$

This paper proposes a multi-objective technique to handle the trade-off between the population convergence and the population diversity. By introducing the niche-count $nc(\mathbf{x}|\mathbf{U}, \sigma)$ to the SOP in Eq.(1) as an additional objective, we obtain a bi-objective optimization problem:

$$\begin{array}{ll} \min & \mathbf{y} = (f(\mathbf{x}), nc(\mathbf{x}|\mathbf{U}, \sigma)) \\ \text{where} & \mathbf{x} \in \mathbf{X} = \{\mathbf{x}|\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \end{array} \tag{5}$$

In terms of the population distribution, minimization of the original objective $f(\mathbf{x})$ will push all solutions in the population to the optimum, while minimization of the niche-count $nc(\mathbf{x}|\mathbf{U}, \sigma)$ will disperse any two solutions in the population with a dispersal distance σ between each other. The population will distribute wider with a larger σ, narrower with a smaller σ, and the population will converge with $\sigma = 0$.

Now we dynamically change the niche radius σ in Eq. (5) by gradually decreasing the value of σ from an initial value to zero. This process actually constructs a sequence of MOPs $\{MOP^{(s)}\}, s = 0, 1, \cdots, S$, i.e., a ***dynamic multi-objective optimization problem (DMOP)*** shown as follows

Definition 3. *(**DMOP**)*

$$\begin{array}{lll} MOP^{(0)} & min & \mathbf{y} = (f(\mathbf{x}), nc(\mathbf{x}|\mathbf{U}, \sigma^{(0)})) \\ MOP^{(1)} & min & \mathbf{y} = (f(\mathbf{x}), nc(\mathbf{x}|\mathbf{U}, \sigma^{(1)})) \\ \ldots\ldots & & \ldots\ldots \\ MOP^{(S)} & min & \mathbf{y} = (f(\mathbf{x}), nc(\mathbf{x}|\mathbf{U}, \sigma^{(S)})) \\ & where & \mathbf{x} \in \mathbf{X} = \{\mathbf{x}|\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \end{array} \tag{6}$$

where $\sigma^{(0)} > \sigma^{(1)} > \cdots > \sigma^{(S)} = 0$. S is a given max number of environmental changes, $\sigma^{(s)}$ denotes the value of σ at environment state s. An environment change is described as a reduction of the niche radius σ from state s to $s+1$.

With a slight change of σ between two successive states, the properties of the problem $\text{MOP}^{(s+1)}$ will usually be similar to that of the previous problem $\text{MOP}^{(s)}$.

A dynamic MOEA (DMOEA) can be employed to solve the DMOP in Eq.(6). In terms of the search of the DMOEA, minimization of the original objective $f(\mathbf{x})$ is of benefit to the exploitation while minimization of the additional objective, the niche-count function $nc(\mathbf{x}|\mathbf{U}, \sigma)$, is of benefit to the exploration. The DMOEA handles the trade-off between the exploration and the exploitation by reducing the niche radius gradually to zero for the diverse population converging to the global optimum.

Note A DMOP definition can be found in [31]. Literature [32] has systemically studied the dynamic single-objective optimizations.

We now explain the relationship between the SOP and the DMOP. At the final state S, the niche radius reduces to zero, i.e., $\sigma^{(S)} = 0$, then $nc(\mathbf{x}, \mathbf{U}|0) = 0$. That is, the final $MOP^{(S)}$ in the DMOP in Eq. (6) has the following simplified form:

$$MOP^{(S)} : min \quad \mathbf{y} = (f(\mathbf{x}), 0) \tag{7}$$

The Pareto-optimal set for $MOP^{(S)}$ is a one-solution set. The one-solution is exactly the optimal solution of the SOP in Eq.(1). In this sense, the $MOP^{(S)}$ is called equivalent to the SOP which is denoted as $MOP^{(S)} \cong SOP$. Given $MOP^{(S)} \cong SOP$, we say ***the DMOP is equivalent to the SOP*** i.e., $DMOP \cong SOP$.

It is easy to prove the following proposition.

Proposition 1. *Denote the Pareto-optimal set of MOP*$^{(s)}$ *as* $\mathbf{P}^{(s)*}$ *at state s in the DMOP in Eq. (6), and opt** *as the optimal solution of the SOP. Suppose the niche radius sequence* $\sigma^{(0)} > \sigma^{(1)} > \cdots$, $\sigma^{(s)} \to 0$ *with* $s \to \infty$. *We then have* $\mathbf{P}^{(s)*} \supset \{opt^*\}$, *and* $\mathbf{P}^{(s)*} \to \{opt^*\}$ *with* $s \to \infty$.

Proposition 1 guarantees an elitism DMOEA converges to the *opt**. It seems that such a conversion unnecessarily makes a problem complex. From another point of view, solving SOPs with many local optima is not easy. There are many niching techniques for EAs but none of them is simple and efficient. Paradoxically converting such a SOP into a DMOP enables a MOEA to handle difficulty of many local optima without any extra effort.

3.3. When to change the environment

An important issue of the use of a DMOEA for solving the SOP is that the DMOEA reaches the final problem MOP$^{(S)}$($\cong SOP$) as fast as possible since the DMOEA in this study is to solve the SOP ($\cong$ MOP$^{(S)}$) rather than the whole DMOPs. We set and change the environment in a round-robin way as follows.

- Set the initial niche radius $\sigma^{(0)}$ and set the initial MOP$^{(0)}$ at the first environment state $s = 0$ (also at the initial generation $t = 0$).
- Change the environment at every generation t by reducing the niche radius slightly from $\sigma^{(s)}$ to $\sigma^{(s+1)}$, i.e., the problem changes from MOP$^{(s)}$ to MOP$^{(s+1)}$.
- Repeat the process till s reaches the final state $s = S$.

Since the environment changes at every generation t, we have state $s = t$ and the given maximum number of the environmental changes S is equal to the maximum number of generations T, i.e., $S = T$.

3.4. How to change the environment

The dynamic environment $\sigma^{(s)}$, $s = 0, 1, \cdots, S$, in the DMOP in Eq. (6) should be pre-defined.

Inspired by the simulated annealing algorithm [33], where the acceptance probability and the annealing program usually use an exponential function, the reduction of the niche radius σ for each environment change as follows:

$$\sigma^{(s)} = Ce^{-(\frac{s}{D})^{cp}} - \varepsilon \tag{8}$$

where ε is a given positive close-to-zero number and cp is a control parameter of the environment change level. C and D are constants to be determined. Note that in most studies, like ε level control function [34], parameter control function [22], they all adopt exponential function. Essentially, they have no significant difference. This paper adopts the exponential function of the Gaussian form, which can guarantee the faster convergence of the population. We think that other functions, e.g., polynomial or logarithmic ones, could also be candidates. We will study the effect of choosing different functions in our future work.

Suppose there are L solutions in the union of the parent and the offspring populations $\mathbf{U}$ spreading over the solution space. Naturally, we set the average space a solution takes to be the initial niche size, i.e., $\frac{\prod_{i=1}^{n}(u_i-l_i)}{L}$. On the basis of the n-dimensional hypersphere volume formula, we obtain the initial niche radius $\sigma^{(0)} = \frac{1}{2}\sqrt[n]{\frac{2n\prod_{i=1}^{n}(u_i-l_i)}{L\pi}}$ at $s = 0$. The final niche radius is $\sigma^{(S)} = 0$ at $s = S$. From the initial and final states of Eq. (8), we obtain a group of two equations

$$\begin{cases} s = 0: & \sigma^{(0)} = C - \varepsilon \\ s = S: & 0 = Ce^{-(\frac{S}{D})^{cp}} - \varepsilon \end{cases} \tag{9}$$

Then C and D are obtained:

$$\begin{cases} C = \sigma^{(0)} + \varepsilon \\ D = \frac{S}{\sqrt[cp]{ln\left(\frac{\sigma^{(0)}+\varepsilon}{\varepsilon}\right)}} \\ where\ \ \sigma^{(0)} = \frac{1}{2}\sqrt[n]{\frac{2n\prod_{i=1}^{n}(u_i-l_i)}{L\pi}} \end{cases} \tag{10}$$

4. DMOEA for solving DMOP

4.1. Framework of DMOEA

The framework of DMOEA for the DMOP is introduced as a dynamic version of a normal MOEA. The details are as follows.

Note that without *Step 1.1* and *Step 2.1* (in **bold**), the DMOEA framework is the same as a general MOEA.

Algorithm 1 Framework of DMOEA

Step 1 Initialization

1.1 **Initialize niche radius $\sigma = \sigma^{(0)}$, MOP$^{(0)}$ and environment state $s = 0$.**

1.2 Initialize parent population, generation counter $t = 0$.

Step 2

2.1 **Reduce $\sigma = \sigma^{(s+1)}$, update MOP$^{(s+1)}$, choose current population as next parent population, $s = s + 1$.**

2.2 $t = t + 1$.

Step 3 Generate offspring population and select the next parent population.

Step 4 IF s approaches to the final state S, THEN goto *Step 5*, ELSE go back to *Step 2*.

Step 5 Output results.

4.2. Instance of DMOEA

The instantiation of the DMOEA (*Algorithm 1*) requires that

- An initialization of parent population in *Step 1.2*;
- A generation of offspring population and a selection of new parent population in *Step 3*.

A traditional MOEA: NSGA-II [9] is selected to instantiate the DMOEA. In this paper, the DE/rand/1/bin operator [35] in *Algorithm 2* is adopted to generate offspring for NSGA-II since this operator is effective for solving continuous optimization problems.

In *Step 2* of *Algorithm 2*, $rndInt(1, n)$ returns an integer number in the range $[1, n]$ while the $rndReal(0, 1)$ returns a real number in the range $(0, 1)$.

The vector $\mathbf{v_i}$ occasionally goes out of the normalized solution space $[\mathbf{0}, \mathbf{1}]$ (the original solution space is normalized in $[\mathbf{0}, \mathbf{1}]$ by Eq. (2) in this paper) due to an inappropriate value of parameter F. In this case, we calculate a reduced F to restrain the vector $\mathbf{v_i}$ to stay in $[\mathbf{0}, \mathbf{1}]$.

The dynamic version of NSGA-II is denoted as **DNSGAII-DE**. The initialization of the parent population in *Step 1.1* of *Algorithm 1*, the generation of offspring population and the selection of the next parent population in *Step 3* are referred directly to NSGA-II [9], except that its generating offspring operator is replaced by *Algorithm 2*.

Algorithm 2 DE/rand/1/bin operator.

Input: Parents $\{\mathbf{p_i}, \mathbf{p_a}, \mathbf{p_b}, \mathbf{p_c}\}$.

Output: Offspring $\mathbf{q_i}$.

Step 1 Perform mutation on $\mathbf{p_a}, \mathbf{p_b}, \mathbf{p_c}$:

$$\mathbf{v_i} = \mathbf{p_a} + F(\mathbf{p_b} - \mathbf{p_c}).$$

Step 2 Crossover on $\mathbf{p_i} = (p_{i1}, p_{i2}, \cdots, p_{in})$ and $\mathbf{v_i} = (v_{i1}, v_{i2}, \cdots, v_{in})$:

$$j_{rnd}= rndInt(1, n),\ r_{rnd}= rndReal(0, 1).$$

$$q_{ij} = \begin{cases} v_{ij} & if\ r_{rnd} < CR\ or\ j = j_{rnd} \\ p_{ij} & if\ r_{rnd} \geq CR\ and\ j \neq j_{rnd} \end{cases}$$

$$j = 1, 2, \cdots, n.$$

Step 3 Output $\mathbf{q_i}$.

5. Results and discussion

5.1. Benchmark problems

In this subsection, three sets of benchmark problems from the CEC2005, CEC2014, and CEC2017 competitions are chosen to test the performance of DNSGAII-DE. The stopping criteria, population size, and the number of runs are same as in [36], [37] and [38] for CEC2005, CEC2014 and CEC2017, respectively.

The experimental results on CEC2005 obtained by DNSGAII-DE are compared with three classic DE variants (SaDE [39], EPSDE [40], CoDE [41]) and four other methods(ABC [42], HPA [43], NCS-C [44] and IBSA [45]). Different from the dynamic multi-objective and niching technologies used in this paper, SaDE, EPSDE, CoDE, ABC and IBSA are all equipped with parameter/operator adaptation strategies. HPA takes hybrid operators approach. NCS-C uses a method that relies on probability distributions to preserve diversity.

The results on CEC2014 obtained by DNSGAII-DE are compared with four state-of-the-art methods(MOMPSO [46], LX-BBO [47], MERDE [48] and M-PSO-MA [49]). Compared with the multi-objective, niching and dynamic strategies used by DNSGAII-DE, MOMPSO and M-PSO-MA use multiple PSO variants. LX-BBO introduces the Laplace crossover operator. MERDE keeps diversity by combining fitness Euclidean-distance ratio technique with DE.

Furthermore, the results on CEC2017 produced by DNSGAII-DE are compared with two recently developed approaches (D-YYPO[50], TLBO-FL[51]). Differ from traditional optimization variants, D-YYPO and TLBO-FL use two new optimization methods: Yin-Yang pair optimization and teaching learning based optimization.

For CEC2005 functions, F1-F5 are unimodal functions and F6-F25 are multimodal in which F6-F12 are basic functions and F13-F25 are expanded functions. The global optimums of F7 and F25 functions are outside of the initialization range. The CEC2014 test suite expands its CEC2005 counterpart by adding new test functions. The problems can be classified into four groups: F1-F3 are unimodal functions, F4-F16 are simple multimodal functions, F17-F22 are hybrid functions and F23-F30 are composition functions. For the latest CEC2017 benchmark problem functions, F1-F3 are unimodal functions, F4-F10 are simple multimodal functions, F11-F20 are hybrid functions and F21-F30 are composition functions. The detailed description of these benchmark test suites can be found in [36], [37] and [38].

5.2. Algorithm settings

The parameters of DNSGAII-DE are set as follows.

- Parent population size and offspring population size: $N = 100$.
- The settings of the DE operator in Algorithm 2: the scaling factor F is randomly given in the range of [0.0,1.0], crossover ratio $CR = 0.9$.
- The number of runs: 25 for CEC2005 and 51 for CEC2014 and CEC2017.
- Max evaluations: $FEs = D * 10,000$.
- The setting for environmental change in Eq.(8): $\varepsilon = 1e - 8$, $cp = 2$.

The parameters of the other compared algorithms are considered as given in the references respectively.

5.3. Comparison on CEC2005 test suite

In this paper, the performance of an algorithm is measured with function errors, i.e., the difference between the objective function value of the obtained solution and that of the optimal solution. The function errors are recorded and the average function errors on CEC2005 test functions with dimensions ten and thirty are presented in Tables 1 and 2, respectively. The bold font in Tables 1 and 2 indicate the winner. A pair wise multiple-problem Wilcoxon's test at the significance levels α=0.05 and α=0.1 was performed between DNSGAII-DE and its peer algorithms. The Friedman's test was carried out to sort the performance of all compared algorithms.

Tables 3 and 4 present the statistical test results where R+, R- denote the sum of ranks, R+ > R- means that the algorithm of this paper is better than the compared algorithm, and vice versa. $\approx$ denotes there is no significant difference between the two algorithms, + denotes that the algorithm of this paper is significantly superior to the compared algorithm. As shown in Table 3, DNSGAII-DE better than all the compared algorithms where it obtains higher R+ values than R- values in all cases. In terms of the multiple-problem Wilcoxon's test, DNSGAII-DE performs significantly better than SaDE, EPSDE, CoDE, NCS-C and IBSA at both α=0.05 and α=0.1 on problems with D=10; DNSGAII-DE performs significantly better than SaDE, EPSDE, CoDE and IBSA at α=0.1 on problems with D=30. From Table 4, we can observe that DNSGAII-DE ranks the best among all the 8 algorithms in both D=10 and D=30.

Table 1: The average results of 8 algorithms at dimension(D) 10 after reaching D*10000 FEs of CEC2005 test functions are listed in the form of 'mean±standard deviation'. All the results are presented in terms of function errors.

Fun	SaDE	EPSDE	CoDE	ABC	HPA	NCS-C	IBSA	DNSGAII-DE
F1	**0.00E+00**	**0.00E+00**	1.76E-11	**0.00E+00**	**0.00E+00**	3.08E-06	**0.00E+00**	**0.00E+00**
	±0.00E+00	**±0.00E+00**	±7.83E-12	**±0.00E+00**	**±0.00E+00**	±9.46E-07	**±0.00E+00**	**±0.00E+00**
F2	**0.00E+00**	**0.00E+00**	6.89E-02	**0.00E+00**	7.64E-21	5.56E-06	6.62E-02	**0.00E+00**
	±0.00E+00	**±0.00E+00**	±7.04E-04	**±0.00E+00**	±3.76E-20	±3.22E-06	±5.62E-02	**±0.00E+00**
F3	1.66E-13	**2.51E-19**	1.58E+02	6.27E+03	1.11E+05	4.15E+04	1.52E+05	1.41E+02
	±8.02E-13	**±1.24E-18**	±7.83E+01	±2.87E+03	±1.12E+05	±2.83E+04	±9.01E+04	±2.46E+02
F4	**0.00E+00**	6.06E-30	9.88E-03	**0.00E+00**	1.05E+01	8.10E-06	1.03E+00	1.00E+00
	±0.00E+00	±3.03E-29	±2.69E-03	**±0.00E+00**	±3.01E+01	±3.00E-06	±7.54E-01	±2.30E-03
F5	3.01E-07	3.86E-12	2.88E-03	**0.00E+00**	**0.00E+00**	1.94E-01	2.14E+00	5.00E-09
	±2.45E-07	±1.42E-12	±9.92E-04	**±0.00E+00**	**±0.00E+00**	±4.08E-02	±1.54E+00	±1.30E-09
F6	2.83E+01	3.68E+01	2.66E+00	4.69E+00	4.47E-01	3.58E+00	1.62E+00	**2.00E-02**
	±5.85E-01	±1.37E+00	±9.17E-01	±2.24E+00	±7.61E-01	±1.55E+00	±1.74E+00	**±8.18E-02**
F7	1.27E+03	1.44E+03	1.27E+03	3.46E-01	**1.21E-01**	1.27E+03	1.27E+03	1.27E+03
	±5.51E-13	±2.17E-02	±3.48E-10	±6.43E-02	**±9.03E-02**	±8.25E-01	±1.20E-08	±0.00E+00
F8	2.07E+01	2.05E+01	2.05E+01	2.04E+01	2.04E+01	**2.00E+01**	2.03E+01	2.03E+01
	±6.67E-02	±7.57E-02	±6.33E-02	±6.52E-02	±1.30E-01	**±1.37E-02**	±6.98E-02	±6.71E-02
F9	**0.00E+00**	**0.00E+00**	4.10E-06	2.04E+01	**0.00E+00**	1.32E-06	**0.00E+00**	**0.00E+00**
	±0.00E+00	**±0.00E+00**	±3.86E-06	±6.52E-02	**±0.00E+00**	±4.30E+00	**±0.00E+00**	**±0.00E+00**
F10	9.46E+01	6.77E+01	6.37E+01	2.27E+01	1.54E+01	1.56E+01	9.97E+00	**6.96E+00**
	±2.49E+00	±1.59E+00	±3.59E+00	±4.24E+00	±7.10E+00	±6.04E+00	±2.72E+00	**±2.15E+00**
F11	4.48E+00	6.51E+00	6.64E+00	6.13E+00	3.98E+00	7.47E-01	5.48E+00	**1.44E-01**
	±1.73E+00	±8.17E-01	±6.80E-01	±6.65E-01	±1.88E+00	±6.24E-01	±7.04E-01	**±1.51E-04**
F12	8.81E+00	4.88E+02	3.30E+02	3.99E+02	1.65E+01	2.48E+02	7.97E+01	**6.31E+00**
	±6.75E+00	±2.01E+02	±2.17E+02	±2.11E+02	±2.07E+01	±9.60E+02	±6.06E+01	**±5.57E+00**
F13	6.84E-01	6.85E-01	4.49E-01	4.90E-01	**2.92E-02**	7.01E-01	4.19E-01	3.71E-01
	±8.76E-02	±4.72E-02	±1.63E-01	±1.92E-01	**±2.13E-02**	±1.29E-01	±5.75E-02	±4.34E-02
F14	3.81E+00	3.67E+00	3.41E+00	3.51E+00	2.92E+00	3.07E+00	3.28E+00	**2.29E+00**
	±2.68E-01	±1.66E-01	±1.52E-01	±1.55E-01	±4.18E-01	±2.35E-01	±1.48E-01	**±3.69E-01**
F15	5.18E+00	1.24E+02	5.22E-02	1.48E+02	**4.96E-13**	1.86E+02	1.80E+00	1.98E+02
	±1.18E+01	±1.55E+01	±8.95E-02	±2.40E+01	**±2.67E-12**	±2.71E+01	±8.89E+00	±1.84E+02
F16	9.86E+02	1.23E+02	1.44E+02	1.98E+02	1.21E+02	1.28E+02	1.12E+02	**1.02E+02**
	±6.69E+00	±4.28E+00	±9.17E+00	±1.46E+01	±1.93E+01	±1.28E+01	±6.05E+00	**±9.63E+00**
F17	1.77E+02	1.38E+02	1.59E+02	2.16E+02	1.22E+02	1.33E+02	1.31E+02	**1.03E+02**
	±1.08E+01	±8.39E+00	±1.27E+01	±1.61E+01	±3.35E+01	±1.21E+01	±9.60E+00	**±5.10E+00**
F18	7.20E+02	7.93E+02	**3.00E+02**	4.77E+02	5.41E+02	3.15E+02	5.02E+02	4.19E+02
	±1.87E+02	±1.53E+02	**±3.20E-04**	±4.44E+01	±1.44E+02	±2.43E+01	±1.59E+02	±1.19E+02
F19	7.80E+02	7.73E+02	7.00E+02	4.68E+02	5.27E+02	**3.06E+02**	6.95E+02	5.24E+02
	±1.00E+02	±1.82E+02	±1.05E-03	±3.81E+01	±1.50E+02	**±1.68E+01**	±1.63E+02	±2.45E+02
F20	7.40E+02	8.28E+02	7.20E+02	4.77E+02	4.67E+02	**3.15E+02**	6.47E+02	5.00E+02
	±1.66E+02	±1.21E+02	±1.00E+02	±4.60E+01	±1.25E+02	**±2.84E+01**	±2.06E+02	±2.45E+02
F21	5.48E+02	9.26E+02	5.04E+02	6.00E+02	**3.87E+02**	4.30E+02	5.03E+02	5.80E+02
	±1.58E+02	±2.68E+00	±7.24E-11	±9.76E+01	**±9.41E+01**	±1.16E+02	±1.60E+01	±7.72E+01
F22	7.25E+02	6.47E+02	6.96E+02	9.54E+02	5.87E+02	7.62E+02	5.78E+02	**5.00E+02**
	±1.28E+02	±1.50E+00	±1.76E+02	±4.88E+01	±2.34E+02	±9.94E+00	±2.27E+02	**±0.00E+00**
F23	6.31E+02	9.17E+02	5.59E+02	1.00E+03	5.16E+02	5.37E+02	5.51E+02	**4.33E+02**
	±1.22E+02	±2.50E+00	±6.42E-12	±8.19E+01	±6.06E+01	±4.98E+01	±1.13E+02	**±9.42E+01**
F24	2.05E+02	3.75E+02	2.04E+02	2.01E+02	**2.00E+02**	2.12E+02	**2.00E+02**	**2.00E+02**
	±0.00E+00	±2.27E+00	±1.99E-11	±9.01E-02	**±1.23E-12**	±6.00E+01	**±0.00E+00**	**±0.00E+00**
F25	1.82E+03	1.81E+03	1.75E+03	**2.00E+02**	**2.00E+02**	1.75E+03	1.72E+03	1.73E+03
	±5.25E+00	±3.13E+00	±4.37E+00	**±1.95E-03**	**±0.00E+00**	±8.77E+00	±4.30E+00	±1.44E+00

Table 2: The average results of 8 algorithms at dimension(D) 30 after reaching D*10000 FEs of CEC2005 test functions are listed in the form of 'mean±standard deviation'. All the results are presented in terms of function errors.

Fun	SaDE	EPSDE	CoDE	ABC	HPA	NCS-C	IBSA	DNSGAII-DE
F1	**0.00E+00**	**0.00E+00**	3.41E-08	**0.00E+00**	3.30E-28	**0.00E+00**	**0.00E+00**	**0.00E+00**
	±0.00E+00	**±0.00E+00**	±1.38E-08	**±0.00E+00**	±4.27E-28	**±0.00E+00**	**±0.00E+00**	**±0.00E+00**
F2	2.10E-05	8.73E-24	4.10E-01	**0.00E+00**	1.17E+02	2.10E-10	7.96E+02	3.50E+03
	±3.97E-05	±6.56E-24	±2.99E-01	**±0.00E+00**	±1.36E+02	±1.95E-10	±2.31E+02	±1.33E+03
F3	3.91E+05	1.08E+06	1.43E+05	2.20E+05	4.84E+06	4.55E+05	1.10E+05	**1.21E+02**
	±1.81E+05	±4.49E+06	±1.95E+05	±2.53E+04	±5.00E+06	±2.02E+05	±2.87E+06	**±2.37E+02**
F4	1.74E+01	**2.08E-03**	1.15E+01	2.20E+05	4.68E+02	1.44E+04	6.95E+03	3.17E+03
	±3.99E+01	**±1.01E-02**	±9.17E+00	±2.53E+04	±7.03E+02	±4.50E+03	±1.26E+03	±1.57E+03
F5	2.30E+03	7.27E+02	4.25E+02	6.02E+03	5.19E+03	2.59E+03	3.23E+03	**9.21E+01**
	±3.40E+02	±4.45E+02	±1.05E+02	±7.16E+02	±1.86E+03	±2.32E+03	±2.37E+02	**±5.94E+01**
F6	4.46E+01	2.87E+00	2.52E+01	1.38E+02	**1.52E+00**	2.08E+01	5.90E+01	7.68E+01
	±2.82E+01	±1.38E+01	±1.36E+00	±5.81E+01	**±3.68E+00**	±3.61E+00	±2.77E+01	±2.10E+01
F7	4.80E+03	4.89E+03	4.71E+03	1.05E+02	2.02E-02	**1.69E-02**	4.70E+03	4.70E+03
	±1.29E-12	±1.73E+00	±3.78E-08	±8.20E+01	±1.83E-02	**±1.38E-02**	±9.73E-10	±2.29E-04
F8	2.13E+01	2.10E+01	2.12E+01	2.09E+01	2.11E+01	**2.00E+01**	2.05E+01	2.09E+01
	±6.48E-02	±5.58E-02	±4.08E-02	±5.63E-02	±6.92E-02	**±1.22E-02**	±6.19E-02	±9.25E-02
F9	**0.00E+00**	**0.00E+00**	1.62E+01	6.60E+01	**0.00E+00**	9.36E+01	**0.00E+00**	1.14E-13
	±0.00E+00	**±0.00E+00**	±1.56E+00	±6.74E+00	**±0.00E+00**	±1.38E+01	**±0.00E+00**	±0.00E+00
F10	9.08E+01	9.86E+01	1.59E+02	2.01E+02	1.15E+02	9.03E+01	8.12E+01	**4.42E+01**
	±9.87E+00	±1.00E+01	±1.70E+01	±1.44E+01	±7.49E+01	±1.79E+01	±9.23E+00	**±1.31E+01**
F11	2.18E+01	3.43E+01	3.35E+01	3.56E+01	3.54E+01	1.37E+01	2.78E+01	**3.74E+00**
	±8.04E+00	±3.15E+00	±1.87E+00	±8.84E-01	±7.04E+00	±1.27E+00	±1.69E+00	**±1.18E+00**
F12	2.75E+04	5.72E+04	8.10E+04	9.55E+04	9.86E+03	**1.57E+03**	1.92E+04	4.36E+03
	±1.71E+03	±8.36E+03	±2.53E+04	±1.75E+04	±5.45E+03	**±1.52E+03**	±5.41E+03	±4.70E+03
F13	4.52E+00	2.48E+00	8.11E+00	1.07E+01	**5.40E-01**	1.54E+00	1.80E+00	1.02E+00
	±4.55E-01	±2.27E-01	±5.66E-01	±9.32E-01	**±2.48E-01**	±8.04E-01	±1.11E-01	±1.22E-01
F14	1.49E+01	1.55E+01	1.62E+01	1.36E+01	1.37E+01	**1.24E+01**	1.27E+01	1.28E+01
	±1.68E-01	±2.08E-01	±1.32E-01	±1.33E+01	±2.57E-01	**±3.31E-01**	±2.68E-01	±2.10E-01
F15	4.02E+02	2.28E+02	4.01E+02	2.88E+02	**1.08E+01**	3.15E+02	1.38E+02	3.37E+02
	±5.00E+01	±2.22E+01	±8.45E-08	±3.25E+01	**±2.29E+01**	±5.68E+01	±7.98E+01	±1.36E+02
F16	5.69E+02	1.28E+02	1.84E+02	3.06E+02	2.96E+02	1.21E+02	1.18E+02	**5.08E+01**
	±1.13E+01	±9.05E+01	±1.43E+01	±2.18E+01	±1.04E+02	±1.53E+01	±2.70E+01	**±3.44E+00**
F17	5.63E+02	2.47E+02	2.25E+02	3.01E+02	3.29E+02	1.55E+02	1.68E+02	**7.72E+01**
	±1.12E+01	±8.23E+01	±1.60E+01	±2.00E+01	±1.57E+02	±2.40E+01	±2.63E+01	**±3.39E+01**
F18	8.95E+02	8.98E+02	9.06E+02	**8.12E+02**	9.12E+02	8.79E+02	8.94E+02	8.95E+02
	±5.83E+01	±1.01E+00	±1.97E-01	**±4.07E+01**	±2.92E+00	±8.68E+01	±4.24E+01	±7.70E-01
F19	8.43E+02	8.53E+02	9.06E+02	**8.17E+02**	9.09E+02	8.93E+02	9.05E+02	9.00E+02
	±5.83E+01	±6.78E-01	±1.65E-01	**±4.50E+01**	±2.67E+00	±4.12E+01	±3.18E+01	±3.07E-01
F20	9.38E+02	8.97E+02	9.06E+02	**8.23E+02**	9.13E+02	8.81E+02	8.90E+02	9.00E+02
	±5.66E+01	±1.11E+00	±2.08E-01	**±5.31E+01**	±2.32E+00	±1.23E+02	±4.76E+01	±4.89E-01
F21	5.01E+02	8.59E+02	5.04E+02	6.42E+02	**5.00E+02**	**5.00E+02**	**5.00E+02**	**5.00E+02**
	±5.80E-14	±1.66E+00	±4.73E-09	±1.41E+01	**±3.14E-12**	**±2.32E-13**	**±4.69E-13**	**±3.97E+01**
F22	9.73E+02	9.72E+02	9.71E+02	9.04E+02	9.07E+02	9.06E+02	9.64E+02	**8.76E+02**
	±1.25E+01	±2.62E+00	±1.06E+01	±6.01E+00	±3.19E+01	±1.31E+01	±1.04E+01	**±2.19E+01**
F23	5.35E+02	8.66E+02	5.38E+02	8.20E+02	**5.34E+02**	5.71E+02	**5.34E+02**	**5.34E+02**
	±1.61E-04	±1.31E+00	±3.71E-04	±8.13E+01	**±3.68E-04**	±2.99E+01	**±7.14E-08**	**±5.18E+01**
F24	2.03E+02	2.11E+02	2.05E+02	2.01E+02	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**
	±2.90E-14	±1.39E+00	±1.20E-08	±5.45E-02	**±0.00E+00**	**±2.72E-12**	**±2.90E-14**	**±0.00E+00**
F25	1.67E+03	2.11E+03	1.66E+03	**2.00E+02**	**2.00E+02**	2.22E+02	1.65E+03	1.60E+03
	±5.52E+00	±1.35E+00	±5.65E+00	**±2.23E-03**	**±7.95E-03**	±2.22E+02	±3.80E+00	±5.98E+00

Table 3: Results of the Multiple-Problem Wilcoxon's test for DNSGAII-DE, SaDE, EPSDE, CoDE, ABC, HPA, NCS-C and IBSA (D=10 and D=30) in solving CEC2005 test functions.

Algorithm	D=10				D=30			
	R+	R-	α=0.05	α=0.1	R+	R-	α=0.05	α=0.1
DNSGAII-DE versus SaDE	188.0	43.0	+	+	213.0	63.0	+	+
DNSGAII-DE versus EPSDE	224.0	29.0	+	+	216.5	83.5	+	+
DNSGAII-DE versus CoDE	236.0	64.0	+	+	266.0	59.0	+	+
DNSGAII-DE versus ABC	186.5	89.5	≈	≈	186.0	90.0	≈	≈
DNSGAII-DE versus HPA	142.0	111.0	≈	≈	150.0	103.0	≈	≈
DNSGAII-DE versus NCS-C	194.0	106.0	+	+	139.9	114.0	≈	≈
DNSGAII-DE versus IBSA	168.5	41.5	+	+	153.0	57.0	≈	+

R+, R- represent sum of ranks. That R+ > R- means that the algorithm of this paper is better than the compared algorithm and vice versa.

Table 4: Average ranks of 8 algorithms for CEC2005 (D=10 and D=30) by Friedman's test.

Algorithm	Average ranks	
	D=10	D=30
DNSGAII-DE	**2.74(1)**	**3.28(1)**
HPA	3.00(2)	4.60(4)
IBSA	4.00(3)	3.66(3)
ABC	4.76(4)	4.98(5)
NCS-C	4.78(5)	3.30(2)
CoDE	5.34(6)	5.72(8)
SaDE	5.44(7)	5.38(7)
EPSDE	5.94(8)	5.08(6)

5.4. Comparison on CEC2014 test suite

The average function errors and standard deviations on CEC2014 test functions with dimensions ten and thirty are presented in Tables 5 and 6. The statistical test results based on the multiple-problem Wilcoxon's test and the Friedman's test are listed in Tables 7 and 8, respectively. As shown in Table 7, DNSGAII-DE performs better than all compared algorithms. With respect to the multiple-problem Wilcoxon's test at α=0.05 and α=0.1, DNSGAII-DE is statistically superior to the four competitors on the CEC2014 test functions with dimension ten. When D=30, DNSGAII-DE significantly performs better than MOMPSO and LX-BBO at both α=0.05 and α=0.1. In addition, we can observe from Table 8 that DNSGAII-DE ranks the best among all the 4 algorithms in both D=10 and D=30.

5.5. Comparison on CEC2017 test suite

The performance of the proposed DNSGAII-DE algorithm was also examined using CEC2017 benchmark test suites in [38]. This problem set is the latest collection of 30 single objective real-parameter numerical optimization problems, and it is the extension of the CEC2005 and CEC2014. Such an experimentation has been performed only using a few other algorithms, so we make a comparison with two newly developed algorithms D-YYPO and TLBO-FL. The D-YYPO and TLBO-FL results obtained from [50] and [51] directly, and they only retain 2 significant digits.

Table 5: The average results of 5 algorithms at dimension(D) 10 after reaching D*10000 FEs of CEC2014 test functions are listed in the form of 'mean±standard deviation'. All the results are presented in terms of function errors.

Fun	MOMPSO	LX-BBO	MERDE	M-PSO-MA	DNSGAII-DE
F1	1.41E+04±1.77E+04	1.61E+03±1.14E+03	1.58E+00±7.61E+00	4.01E+01±3.17E+01	**6.22E-04±3.05E-03**
F2	8.43E+03±3.80E+03	5.80E+03±2.27E+03	6.31E-05±1.12E-04	3.17E-02±4.63E-02	**0.00E+00±0.00E+00**
F3	1.21E+04±1.18E-04	4.47E+03±5.52E+03	1.35E-03±1.22E-03	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**
F4	6.18E+00±1.07E+01	1.72E+00±4.19E-03	**0.00E+00±0.00E+00**	6.18E+00±1.07E+01	3.34E+00±8.43E+00
F5	2.00E+01±3.81E-02	**1.01E+00±2.81E-01**	1.90E+01±2.71E+00	2.00E+01±5.56E-03	1.61E+01±8.05E+00
F6	3.53E+00±1.77E+00	3.45E+00±1.52E+00	8.93E-01±2.81E-01	1.22E+00±1.32E+00	**1.89E-06±1.16E-06**
F7	1.17E-01±6.19E-02	2.56E-01±1.40E-01	**1.83E-02±1.51E-02**	5.37E-02±2.81E-02	5.17E-02±3.39E-02
F8	1.05E+01±5.36E+00	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00**	3.12E-01±1.08E+00	**0.00E+00±0.00E+00**
F9	1.28E+01±8.22E+00	1.10E+01±4.27E+00	5.58E+00±1.74E+00	5.08E+00±2.02E+00	**3.99E+00±1.88E+00**
F10	3.14E+02±2.03E+02	9.01E+02±5.45E+02	**3.67E-02±3.98E-02**	9.68E+01±9.89E+01	2.92E-01±6.68E-01
F11	4.70E+02±2.54E+02	1.12E+03±5.93E+02	7.55E+01±7.63E+01	**2.52E+00±1.76E+02**	2.43E+01±4.42E+01
F12	1.90E-01±1.23E-01	1.00E-01±4.20E-17	1.17E-01±6.93E-02	**5.29E-02±3.61E-02**	6.71E-02±5.76E-02
F13	**7.31E-02±4.04E-02**	3.12E-01±1.50E-01	1.17E-01±4.43E-02	1.02E-01±4.80E-02	9.34E-02±2.64E-02
F14	2.20E-02±8.27E-03	2.39E-01±2.22E-01	9.37E-02±2.73E-02	2.09E-02±7.87E-03	**1.62E-02±2.82E-02**
F15	6.85E-01±1.93E-01	1.51E+00±7.88E-01	6.72E-01±2.18E-01	6.46E-01±1.58E-01	**6.44E-01±1.99E-01**
F16	2.69E+00±4.30E-01	2.37E+00±4.16E-01	1.53E+00±4.63E-01	1.47E+00±5.68E-01	**7.87E-01±3.45E-01**
F17	1.08E+03±4.01E+02	5.66E+03±6.81E+03	7.93E+00±9.59E+00	2.38E+02±1.37E+02	**5.57E+00±5.55E+00**
F18	7.82E+02±1.20E+03	7.02E+03±7.18E+03	2.72E+00±1.29E+00	4.63E+02±5.90E+02	**9.10E-01±6.92E-01**
F19	2.75E+00±1.43E+00	3.69E+00±7.27E+00	5.10E-01±1.76E-01	9.29E-01±6.98E-01	**7.49E-02±3.50E-02**
F20	3.95E+01±3.89E+01	1.61E+04±2.06E+04	1.70E+00±7.50E-01	2.50E+00±1.71E+00	**1.18E-01±7.06E-02**
F21	3.64E+02±2.63E+02	6.26E+03±7.23E+03	8.54E+00±2.66E+01	6.11E+01±7.63E+01	**1.91E-01±2.12E-01**
F22	5.35E+01±6.99E+01	7.59E+01±7.45E+01	3.24E+00±3.96E+00	1.50E+01±9.03E+00	**2.48E-01±1.80E-01**
F23	3.29E+02±1.38E-12	2.44E+02±5.65E+01	3.29E+02±2.68E-11	3.29E+02±1.38E-12	**2.28E+02±3.76E+01**
F24	1.20E+02±6.74E+00	1.01E+03±8.64E+02	1.15E+02±2.45E+00	**1.14E+02±4.30E+00**	1.15E+02±3.06E+00
F25	1.98E+02±1.60E+01	1.78E+02±1.50E+01	**1.36E+02±8.34E+00**	1.54E+02±4.17E+01	1.63E+02±2.33E+00
F26	1.00E+02±3.34E-02	**3.71E-03±6.58E-03**	1.00E+02±4.18E-02	1.00E+02±3.30E-02	1.05E+02±4.89E-04
F27	3.16E+02±1.71E+02	**1.05E+01±7.68E+00**	2.87E+01±7.60E+01	2.90E+02±1.57E+02	2.06E+02±7.18E+01
F28	4.73E+02±1.07E+02	5.29E+02±1.14E+02	3.66E+02±7.37E+00	4.52E+02±7.24E+01	**3.19E+02±2.41E-01**
F29	3.63E+02±9.23E+01	3.53E+05±7.54E+05	3.17E+02±5.48E+01	2.93E+02±4.57E+01	**2.02E+02±4.59E-01**
F30	7.02E+02±2.63E+02	6.31E+04±6.97E+04	5.34E+02±6.06E+01	6.09E+02±1.78E+02	**2.32E+02±2.24E+01**

Table 6: The average results of 5 algorithms at dimension(D) 30 after reaching D*10000 FEs of CEC2014 test functions are listed in the form of 'mean±standard deviation'. All the results are presented in terms of function errors.

Fun	MOMPSO	LX-BBO	MERDE	M-PSO-MA	DNSGAII-DE
F1	1.98E+06±1.25E+06	1.01E+07±3.81E+06	**5.41E+02±6.40E+02**	3.09E+03±3.19E+03	2.65E+05±2.50E+05
F2	3.79E+02±1.88E+02	5.34E+04±2.14E+04	2.39E-03±3.24E-03	**0.00E+00±0.00E+00**	4.82E+06±3.94E+06
F3	1.20E+04±1.18E+04	1.64E+04±1.71E+04	1.13E-03±7.36E-04	**0.00E+00±0.00E+00**	3.50E+00±3.43E+00
F4	2.35E+02±1.11E+03	9.99E+01±2.85E+01	**6.25E-01±1.46E+00**	5.29E+00±1.77E+01	2.44E+00±3.77E+00
F5	2.06E+01±2.64E-01	**3.06E+00±7.87E-01**	2.00E+01±7.17E-05	2.00E+01±4.45E-02	2.00E+01±4.30E-02
F6	2.76E+01±5.99E+00	1.69E+01±3.12E+00	1.82E+01±1.72E+00	1.05E+01±2.36E+00	**3.47E+00±1.45E+00**
F7	1.28E-02±1.41E-02	1.76E-01±8.56E-02	**0.00E+00±0.00E+00**	1.22E-02±1.25E-02	3.75E-01±1.85E-01
F8	1.35E+02±3.15E+01	5.53E+01±3.78E+02	**0.00E+00±0.00E+00**	1.03E+00±1.23E+00	3.22E+01±9.35E+00
F9	**1.64E+02±2.35E+01**	7.66E+01±1.61E+01	5.50E+01±1.04E+01	7.63E+01±1.36E+01	2.84E+01±6.44E+00
F10	3.86E+03±5.74E+02	1.26E+04±1.16E+02	1.29E+00±1.61E+00	4.54E+02±2.58E+02	**1.90E-01±5.38E-02**
F11	3.81E+03±4.80E+02	1.23E+04±3.42E+02	2.72E+03±4.69E+02	2.59E+03±4.78E+02	**2.44E+03±4.78E+02**
F12	1.35E+00±6.69E-01	**1.11E-02±1.75E-18**	5.64E-01±1.74E-01	8.06E-02±2.86E-02	5.02E-01±6.09E-01
F13	4.34E-01±1.01E-01	6.55E-01±1.56E-01	**2.84E-01±4.41E-02**	3.29E-01±7.47E-02	3.19E-01±05.61E-02
F14	6.97E-01±2.19E-01	6.20E-01±2.96E-01	2.14E-01±2.69E-02	2.16E-01±7.88E-02	**1.86E-01±3.71E-02**
F15	1.03E+01±3.24E+00	1.55E+01±5.50E+00	**4.14E+00±7.77E-01**	9.90E+00±3.27E+00	5.27E+00±1.43E+00
F16	1.18E+01±5.90E-01	1.08E+01±5.84E-01	1.12E+01±4.58E-01	1.03E+01±4.90E-01	**8.11E+00±1.33E+00**
F17	1.13E+05±3.34E+04	1.46E+06±9.34E+05	**1.16E+03±3.73E+02**	1.75E+03±5.32E+02	3.66E+03±2.65E+03
F18	2.06E+04±7.01E+03	2.90E+03±4.27E+03	**2.24E+01±6.47E+00**	2.14E+03±1.83E+03	2.71E+01±3.65E+01
F19	3.15E+01±3.22E+01	5.19E+03±5.67E+03	7.74E+00±7.32E-01	7.35E+00±1.46E+00	**6.23E+00±1.30E+00**
F20	5.45E+02±4.38E+02	2.61E+04±1.56E+04	2.80E+01±1.06E+01	5.56E+01±2.79E+01	**2.20E+01±1.37E+01**
F21	5.16E+04±2.89E+04	1.11E+06±7.95E+05	5.99E+02±2.15E+02	1.89E+03±1.57E+03	**3.04E+02±6.56E+02**
F22	5.39E+02±1.75E+02	1.88E+03±2.04E+02	1.15E+02±7.29E+01	2.80E+02±1.23E+02	**1.78E+01±1.62E+01**
F23	3.39E+02±4.31E+01	4.11E+02±6.43E+01	3.14E+02±3.74E-09	3.15E+02±1.39E-12	**2.91E+02±8.08E+00**
F24	2.26E+02±1.41E+01	1.48E+04±8.37E+03	2.25E+02±5.57E-01	**2.24E+02±5.34E+00**	2.44E+02±1.54E+00
F25	2.06E+02±1.87E+00	5.29E+02±4.37E+01	**2.00E+02±2.83E-02**	2.04E+02±1.56E+00	2.07E+02±5.51E+00
F26	1.05E+02±4.67E+00	2.13E+00±3.46E+00	1.00E+02±7.61E-02	**1.00E+00±6.42E-02**	**1.00E+00±6.99E-02**
F27	1.10E+03±2.41E+02	**1.96E+02±1.04E+02**	3.84E+02±2.32E+01	6.70E+02±1.56E+02	7.01E+02±4.71E+00
F28	1.50E+03±2.03E+02	1.94E+03±5.49E+02	8.05E+02±2.49E+01	1.13E+03±1.67E+02	**3.85E+02±5.05E+00**
F29	3.61E+03±1.34E+03	1.98E+07±3.96E+06	1.19E+03±1.06E+02	1.73E+03±4.62E+02	**2.10E+02±1.55E+01**
F30	2.42E+04±2.46E+04	6.96E+06±1.03E+07	1.07E+03±2.90E+02	2.52E+03±1.24E+03	**4.52E+02±1.54E+02**

Table 7: Results of the Multiple-Problem Wilcoxon's test for DNSGAII-DE, MOMPSO, LX-BBO, MERDE and M-PSO-MA (D=10 and D=30) in solving CEC2014 test functions.

Algorithm	D=10				D=30			
	R+	R-	α=0.05	α=0.1	R+	R-	α=0.05	α=0.1
DNSGAII-DE versus MOMPSO	438.5	26.5	+	+	417.0	48.0	+	+
DNSGAII-DE versus LX-BBO	385.0	50.0	+	+	398.0	67.0	+	+
DNSGAII-DE versus MERDE	309.0	97.0	+	+	221.0	185.0	≈	≈
DNSGAII-DE versus M-PSO-MA	368.0	67.0	+	+	270.0	165.0	≈	≈

R+, R- represent sum of ranks. That R+ > R- means that the algorithm of this paper is better than the compared algorithm and vice versa.

Table 8: Average ranks of 5 algorithms for CEC2014 (D=10 and D=30) by Friedman's test.

Algorithm	Average ranks	
	D=10	D=30
DNSGAII-DE	**1.67(1)**	**2.15(1)**
MERDE	2.55(2)	2.18(2)
M-PSO-MA	2.73(3)	2.37(3)
LX-BBO	3.97(4)	4.10(4)
MOMPSO	4.08(5)	4.20(5)

The average function errors and standard deviations obtained on CEC2017 test functions with dimensions ten and thirty are presented in Tables 9 and 10, and the statistical test results based on the multiple-problem Wilcoxon's test and the Friedman's test are presented in Tables 11 and 12. From Table 11, we can clearly observe that DNSGAII-DE have an edge over the two competitors on the basis of higher R+ values than R- values it provided. With respect to the multiple-problem Wilcoxon's test at α=0.05 and α=0.1, significant differences can be observed in all the cases both at D=10 and D=30, which indicates that DNSGAII-DE is statistically better than the two competitors on the CEC2017 functions with dimensions ten and thirty. Besides, from Table 12, we can notice that DNSGAII-DE has the best ranking among 3 algorithms both in D=10 and D=30.

5.6. Comparison on real world antenna array problem

To further evaluate the performance of DNSGAII-DE on real-world optimization problems in the area of communications, a case study is conducted on the antenna array problem. In a typical sparse array or thinned array, elements are allowed to move farther away from adjacent elements to reduce the number of elements, while maintaining a similar radiation pattern as that of a counterpart with uniform spacing. In this work, we attach a sparse subarray to a core subarray with uniform spacing. The core subarray determines the key characteristics of the radiation pattern, while the sparse subarray is used to improve the radiation pattern using fewer elements than other conventional subarray of the same length.

Figure 1 shows a linear subarray of $2N_c$ elements, at a uniform spacing of 0.5λ. It is extended by a sparse subarray of $2N_e$ elements. All the elements are aligned along the z axis. The length of the core subarray is L_c, and the total length of the whole array is L_e.

The fitness function for optimization is defined as

$$\begin{aligned}
\min \quad & y = PSLL(\theta|\mathbf{a},\mathbf{d}) + K|BW(\mathbf{a},\mathbf{d}) - \kappa BW_d)| + K|(0.5\lambda(N_c - 0.5) + \sum_{j=N_c+1}^{N_c+N_e} d_j) - 0.5L_e| \\
\text{where} \quad & 0<a_i \leq 2, \\
& (i = 1, ..., N_c, N_c + 1, ..., N_c + N_e); \\
& 0.5\lambda \leq d_j \leq 2\lambda, \\
& (j = N_c + 1, N_c + 2, ..., N_c + N_e).
\end{aligned} \tag{11}$$

$[\mathbf{a}, \mathbf{d}]$ is the solution vector where $\mathbf{a} = [a_1, ..., a_{N_c}, a_{N_c+1}, ..., a_{N_c+N_e}]$ represents amplitudes of all the elements; $\mathbf{d} = [d_{N_c+1}, ..., d_{N_c+N_e}]$ represents spacings between two adjacent extended elements. λ is the wavelength.

Table 9: The average results of 3 algorithms at dimension(D) 10 after reaching D*10000 FEs of CEC2017 test functions are listed in the form of 'mean±standard deviation'. All the results are presented in terms of function errors.

Fun	D-YYPO	TLBO-FL	DNSGAII-DE
F1	2.9E+03±3.3E+03	2.0E+03±2.5E+03	**0.00E+00±0.00E+00**
F2	**0.0E+00±0.0E+00**	2.0E-02±1.4E-01	6.26E-01±3.07E+00
F3	1.2E-05±4.3E-05	1.1E-04±7.8E-04	**0.00E+00±0.00E+00**
F4	**2.1E+00±8.3E+00**	3.0E+00±1.2E+00	1.60E+01±1.42E+01
F5	1.1E+01±4.2E+00	8.8E+00±5.6E+00	**5.09E+00±2.24E+00**
F6	6.4E-05±6.0E-05	**8.4E-08±4.4E-07**	1.57E-05±7.31E-05
F7	2.2E+01±6.0E+00	2.8E+01±4.0E+00	**1.78E+01±8.78E-01**
F8	1.3E+01±4.5E+00	1.2E+01±4.4E+00	**3.83E+00±1.35E+00**
F9	2.0E-02±9.8E-02	8.9E-03±6.4E-02	**0.00E+00±0.00E+00**
F10	3.7E+02±1.7E+02	9.5E+02±2.1E+02	**1.61E-01±5.94E-02**
F11	9.3E+00±4.8E+00	4.1E+00±1.5E+00	**1.32E+00±1.52E+00**
F12	1.3E+04±1.2E+04	6.6E+04±5.5E+04	**1.27E+01±3.23E+01**
F13	5.1E+03±5.6E+03	2.4E+03±2.2E+03	**4.41E+00±2.92E+00**
F14	2.1E+01±2.2E+01	6.7E+01±1.8E+01	**8.36E-01±9.19E-01**
F15	4.4E+01±1.1E+02	1.3E+02±4.3E+01	**1.13E-02±1.52E-02**
F16	4.4E+01±5.8E+01	8.9E+00±2.2E+01	**3.79E-01±9.44E-01**
F17	1.4E+01±1.4E+01	3.8E+01±7.8E+00	**9.60E-01±6.88E-01**
F18	8.8E+03±6.4E+03	6.2E+03±5.6E+03	**3.90E-01±4.54E-01**
F19	9.3E+01±2.9E+02	6.1E+01±3.2E+01	**6.27E-02±1.31E-01**
F20	8.0E+00±9.1E+00	1.5E+01±9.4E+00	**1.51E-01±1.78E-01**
F21	**1.0E+02±9.0E-01**	1.4E+02±5.2E+01	1.11E+02±2.44E-01
F22	9.7E+01±2.0E+01	**9.3E+01±2.3E+01**	2.05E+02±3.58E+01
F23	3.1E+02±4.5E+01	3.1E+02±3.8E+00	**3.03E+02±8.09E-01**
F24	**1.2E+02±5.0E+01**	3.1E+02±6.9E+01	3.25E+02±3.24E+01
F25	4.2E+02±2.3E+01	4.3E+02±2.2E+01	**2.69E+02±2.75E+01**
F26	3.0E+02±3.1E+01	3.0E+02±4.6E+01	**2.40E+02±9.09E-13**
F27	4.0E+02±3.8E+00	3.9E+02±3.3E+00	**3.68E+02±9.70E-01**
F28	**3.0E+02±5.1E+01**	4.5E+02±1.6E+02	3.46E+02±2.58E+01
F29	2.6E+02±1.9E+01	2.7E+02±1.4E+01	**2.09E+02±2.18E+01**
F30	6.7E+03±6.5E+03	2.8E+05±4.9E+05	**2.30E+02±1.22E+01**

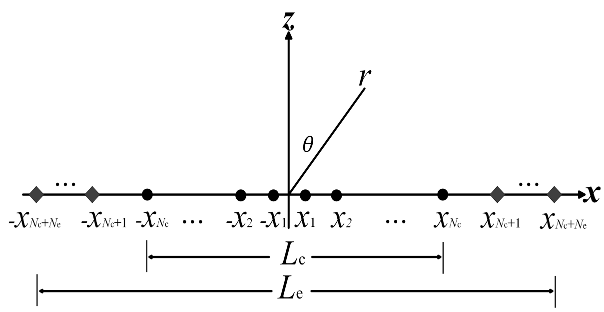


Figure 1: Configuration of a linear subarray with uniform spacing (circle), extended by a linear spare subarray with at unequal spacing (square).

Table 10: The average results of 3 algorithms at dimension(D) 30 after reaching D*10000 FEs of CEC2017 test functions are listed in the form of 'mean±standard deviation'. All the results are presented in terms of function errors.

Fun	D-YYPO	TLBO-FL	DNSGAII-DE
F1	3.7E+03±5.0E+03	**3.5E+03±3.6E+03**	6.67E+05±6.47E+05
F2	3.2E+09±2.3E+10	8.5E+16±5.8E+17	**1.36E+04±3.42E+03**
F3	**5.3E+02±4.3E-05**	3.0E+03±1.1E+03	6.97E+02±4.44E+02
F4	9.1E+01±2.5E+01	9.0E+01±2.4E+01	**1.28E+01±2.66E+01**
F5	9.1E+01±2.4E+01	4.0E+01±2.1E+01	**2.98E+01±7.87E+00**
F6	8.6E-01±7.2E-01	4.9E-01±4.2E-01	**6.94E-04±3.45E-04**
F7	1.4E+02±3.1E+01	1.4E+02±4.7E+01	**7.27E+01±1.11E+01**
F8	9.6E+01±2.5E+01	3.7E+01±1.8E+01	**3.10E+01±6.36E+00**
F9	6.5E+02±7.7E+02	3.4E+01±2.7E+01	**3.10E-01±2.74E-01**
F10	2.8E+03±6.0E+02	6.7E+03±2.8E+02	**1.99E-01±5.09E-02**
F11	1.2E+02±4.1E+01	8.2E+01±4.1E+01	**1.70E+01±6.84E+00**
F12	1.5E+06±1.2E+06	5.7E+04±9.0E+04	**1.06E+04±2.60E+04**
F13	9.6E+03±1.3E+04	2.0E+04±1.8E+04	**2.77E+02±2.47E+02**
F14	2.1E+03±2.3E+03	7.1E+03±5.8E+03	**1.37E+01±6.08E+00**
F15	1.1E+04±9.4E+03	2.2E+04±2.3E+04	**1.92E+01±1.15E+01**
F16	6.7E+02±2.3E+02	4.9E+02±3.5E+02	**2.62E+02±1.51E+02**
F17	2.5E+02±1.5E+02	1.4E+02±6.6E+01	**2.71E+01±7.41E+01**
F18	1.2E+05±1.0E+05	3.7E+05±1.7E+05	**1.14E+02±1.26E+02**
F19	1.4E+04±1.6E+04	1.1E+04±1.1E+04	**8.13E+00±3.54E+00**
F20	2.5E+02±1.5E+02	2.2E+02±1.2E+02	**5.76E+01±7.75E+01**
F21	3.0E+02±2.3E+01	**2.3E+02±1.2E+01**	2.33E+02±8.80E+00
F22	**1.0E+02±9.9E-01**	**1.0E+02±1.9E+00**	1.05E+03±4.65E+02
F23	4.5E+02±3.2E+01	**4.0E+02±1.6E+01**	4.62E+02±1.57E+01
F24	5.6E+02±5.1E+01	**4.7E+02±1.6E+01**	5.24E+02±1.30E+01
F25	3.9E+02±1.4E+00	4.0E+02±1.8E+01	**3.84E+02±1.83E+01**
F26	2.2E+03±7.3E+02	1.4E+03±4.7E+02	**4.66E+02±0.00E+00**
F27	5.4E+02±1.6E+01	5.3E+02±2.1E+01	**4.00E+02±7.73E+00**
F28	**3.9E+02±4.3E+01**	4.3E+02±2.7E+01	4.25E+02±9.82E+00
F29	7.5E+02±2.0E+02	6.2E+02±9.1E+01	**2.79E+02±1.36E+02**
F30	3.3E+04±3.1E+04	2.6E+04±2.8E+04	**3.13E+03±3.03E+02**

Table 11: Results of the Multiple-Problem Wilcoxon's test for DNSGAII-DE, D-YYPO and TLBO-FL (D=10 and D=30) in solving CEC2017 test functions.

Algorithm	D=10				D=30			
	R+	R-	α=0.05	α=0.1	R+	R-	α=0.05	α=0.1
DNSGAII-DE versus D-YYPO	373.0	92.0	+	+	344.0	62.0	+	+
DNSGAII-DE versus TLBO-FL	414.0	51.0	+	+	398.0	67.0	+	+

R+, R- represent sum of ranks. That R+ > R- means that the algorithm of this paper is better than the compared algorithm and vice versa.

Table 12: Average ranks of 3 algorithms for CEC2017 (D=10 and D=30) by Friedman's test.

Algorithm	Average ranks	
	D=10	D=30
DNSGAII-DE	**1.37(1)**	**1.40(1)**
D-YYPO	2.23(2)	2.45(3)
TLBO-FL	2.40(3)	2.15(2)

The first term on the right-hand side of (11) is used to minimize the peak sidelobe level (PSLL) and it can be calculated as:

$$PSLL(\theta|\mathbf{a},\mathbf{d}) = \max_{\forall\theta\in S} |\frac{A_F(\mathbf{a},\mathbf{x},\theta)}{A_F(\mathbf{a},\mathbf{x},\theta_0)}| \tag{12}$$

where S is the space spanned by angle θ excluding the mainlobe with the center at θ_0, which is set to $0°$. $A_F(\mathbf{a},\mathbf{x},\theta)$ is the array factor of the linear antenna array at angle θ and can be calculated as

$$A_F(\mathbf{a},\mathbf{x},\theta) = \sum_{n=1}^{N_c} a_n cos(\frac{2\pi x_n cos\theta}{\lambda}) + \sum_{m=N_c+1}^{N_c+N_e} a_m cos(\frac{2\pi x_m cos\theta}{\lambda}) \tag{13}$$

where $x_n = 0.5\lambda(n-0.5\lambda)$ is the position of the nth core element, a_n is the amplitude of the nth core element, with $1 \leq n \leq N_x$; $x_m = 0.5\lambda(N_c - 0.5\lambda) + \sum_{j=N_c+1}^{m} d_j$ is the position of the mth extended element, and a_m is the amplitude of the mth extended element, with $N_c + 1 \leq m \leq N_c + N_e$.

The second term on the right-hand side of (11) is used to achieve the desired beamwidth κBW_d (in degrees), where BW_d is the ideal beamwidth, which is measured from null to null. The factor κ can be viewed as a relaxation parameter to achieve the ideal beamwidth BW_d of a uniformly spaced array. The last term on the right-hand side of (11) is used to fix the position of the outmost extended element which is equal to $0.5L_e$. K is an empirical penalty coefficient (K=1000 in this paper).

Three well-known algorithms: PSO [52],[53], GA [54] and EHS [55] are compared in solving large sparse arrays with $N_c = 50$ and $N_e = 50$ under three different beamwidth requirements: κ=1.0, κ=1.2 and κ=1.5. The maximum number of evaluations set to 600,000. A trial is claimed successful if the beamwidth and the position of the outmost extended element satisfy the requirements. Note that in this paper, we adopt a classic method to calculate the ideal beamwith BW_d as [55], which uses twice amount of the extended elements and the spacing between each two adjacent extended elements is 0.5λ. The amplitudes of all the elements are set to 1. In this way, we can obtain $BW_d = 0.8$ and $L_e = 149.5\lambda$.

Table 13 lists success rates, best fitness, average fitness and standard deviation results under three different beamwidth. We can observe that GA fails in all three cases. The PSO algorithm fails to fulfill the narrowest beamwidth constraint (κ=1), and achieve 50% and 80% of success rates under the constraints of κ=1.2 and κ=1.5, respectively. The EHS and DNSGAII-DE reach 100% success ratio in all three cases. However, the best fitness and the average fitness obtained by DNSGAII-DE are exceeded 2.5 dB, which is lower than the best results in [55]. Fig.2 shows the field pattern obtained by DNSGAII-DE at κ=1.

Table 13: Performance of DNSGAII-DE, EHS, PSO and GA on large antenna array problem.

Algorithms	κ	Success Rate(%)	Best	Mean	Std
GA	1.0	0	failed	failed	failed
PSO	1.0	0	failed	failed	failed
EHS	1.0	100	-18.05	-17.79	0.13
DNSGAII-DE	1.0	**100**	**-21.36**	**-20.39**	**0.63**
GA	1.2	0	failed	failed	failed
PSO	1.2	50	-20.54	-19.40	0.51
EHS	1.2	100	-19.70	-19.51	0.20
DNSGAII-DE	1.2	**100**	**-23.89**	**-22.41**	**1.49**
GA	1.5	0	failed	failed	failed
PSO	1.5	80	-21.05	-11.70	5.58
EHS	1.5	100	-20.30	-20.01	0.10
DNSGAII-DE	1.5	**100**	**-23.92**	**-22.51**	**1.21**

Best and Mean represent the best and mean fitness value(dB) respectively, Std represents the standard deviation of the fitness value(dB), failed means the algorithm can not satisfy the beamwidth requirements.

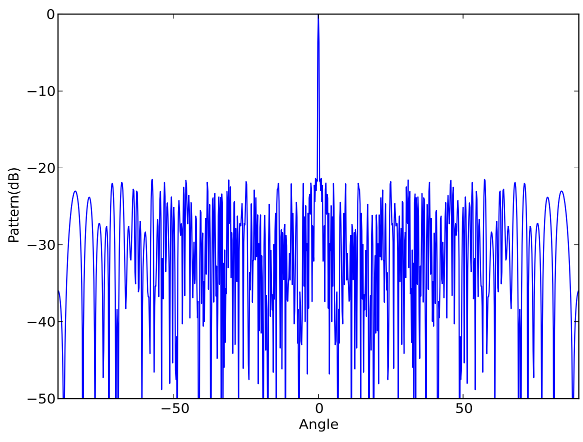


Figure 2: The antenna array radiation pattern designed by DNSGAII-DE at κ=1.0.

Table 14: Characteristics of DE, FitnessSharing-DE and DNSGAII-DE.

Algorithm	niche-count	niche radius
DE	no	no
FitnessSharing-DE	yes	constant radius =0.01
DNSGAII-DE	yes	dynamic radius $\rightarrow$ 0

Table 15: Ranking of DNSGAII-DE, Fitness Sharing-DE and DE by the Friedman's test on 30 test functions with 30-D from IEEE CEC2017.

Algorithm	Average ranks
DNSGAII-DE	**1.50**
Fitness sharing-DE	2.17
DE	2.33

5.7. *Effectiveness of the niche-count*

In order to ascertain the effectiveness of the niche-count, an additional experiment has been carried out. We compare three methods: DNSGAII-DE, DE and fitnessSharing-DE [3]. FitnessSharing-DE is the same as DE except that the objective function $f(\mathbf{x})$ replaced by $f(\mathbf{x})/nc(\mathbf{x}, \sigma = 0.01)$. The characteristics of the three approaches are shown in Table 14. $Algorithm2$ was used as the offspring generator for all the three algorithms.

From the Friedman's test results in Table 15, we can see that DNSGAII-DE achieves the best ranking in solving CEC2017 test functions with 30 dimensions, followed by fitnessSharing-DE. In fact, without niche-count, DE can achieve a higher precision, but it is easy to get trapped in local optima. With niche-count, fitness sharing can maintain the population diversity with higher capacity in global search, but the convergence precision is not high due to the constant niche radius. In the proposed DNSGAII-DE, it provides a proper balance between convergence and diversity by gradually pushing the niche radius to zero with best performance among the three methods.

5.8. *Parameter sensitivity analysis*

In this subsection, we perform a sensitivity analysis for parameter cp in Eq. (8). As aforementioned, this parameter has the capability of controlling the environmental change level. The decrease rate of the exponential function changes as cp changes. If the value of this parameter is too big, it may has a negative effect on the convergence of the population. However, a too small value may lead to premature of the algorithm.

Figure 3 presents the graph of the decrease of the exponential functions over the environment state changes with $cp = 1, 2, 3, 4$, where we set $\sigma_0 = 1.0$ and $S = 1000$. From Figure 3, the radius decreases quickly at the early stage and very slow at the late stage, which indicates that the DNSGAII-DE initially focuses on exploration for the diversity of the population, while in the late stage it will focus on exploitation of high quality solutions.

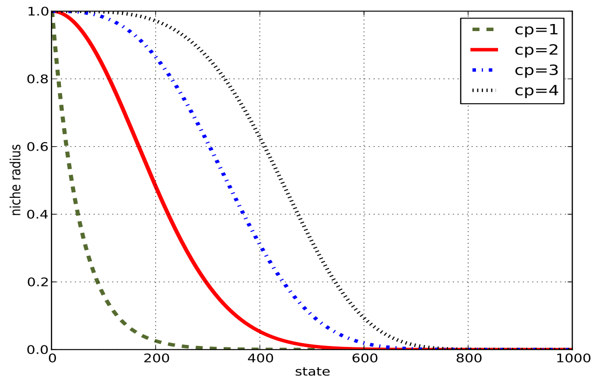


Figure 3: Exponential function value over the environmental change.

Table 16: Results of the Multiple-Problem Wilcoxon's test for DNSGAII-DE with *cp*=1, 2, 3 and 4 on 30 test functions with 30-D from IEEE CEC2017.

Algorithm	R+	R-	α=0.05	α=0.1
DNSGAII-DE-2 versus DNSGAII-DE-1	332.0	133.0	+	+
DNSGAII-DE-2 versus DNSGAII-DE-3	294.0	171.0	$\approx$	$\approx$
DNSGAII-DE-2 versus DNSGAII-DE-4	308.0	157.0	$\approx$	$\approx$

In order to ascertain the effect of the parameter *cp* on the performance of DNSGAII-DE, we test four different values: *cp*=1, 2, 3 and 4 respectively. Considering the statistical test results in Table 16, we can see that DNSGAII-DE achieves the best results with *cp* =2 from the Friedman's test in Table 17, where it has the best ranking among four cases.

Based on the above discussion, a value with $cp = 2$ is recommended in this paper.

5.9. Computational time complexity

The computational time complexity is governed by the following two processes when combining DMOEA with the NSGA-II variant.

- Computational cost for NSGA-II variant is $O(2N^2)$ where N is the population size.
- Computational cost for niche-count is $O(nN^2)$ where n is the dimension of the solution vector.

Therefore, the overall computational complexity of DNSGAII-DE is $O(2N^2 + nN^2)$.

6. Conclusion and future work

To address the issue of the trade-off between the convergence and diversity, this paper proposes a method which converts a SOP into an equivalent DMOP with two objectives. Then a well-established DMOEA is applied to the DMOP, thus the SOP is solved. The Pareto domination between the two conflict objectives

Table 17: Ranking of DNSGAII-DE with cp=1, 2, 3 and 4 by the Friedman's test on 30 test functions with 30-D from IEEE CEC2017.

Algorithm	Average ranks
cp=2	**2.00**
cp=3	2.43
cp=4	2.55
cp=1	3.02

maintains a certain population diversity. The dynamic environment of the gradually-reducing-to-zero niche radius drives the trade-off between the convergence and the diversity through the whole evolutionary process.

The benchmark experimental results show that the performance of DNSGAII-DE is statistically better than that of the other state-of-the-art algorithms. In solving real-world optimization problems in the area of communication, DNSGAII-DE performs much better than three well-established methods.

Future work:

1. By introducing the niche-count objective, a multi-objective optimization problem (MOP) can be converted into an equivalent dynamic multi-objective optimization problem (DMOP). Then this methodology can be employed to solve a MOP with many local optima.
2. Exploration of other candidates similar to the niche-count function and exploration of other candidates of the dynamic environment.
3. Research on dynamic version of other typical types of MOEAs such as decomposition-based MOEA/D [10], hypervolume-based HypE [11], and so on to solve SOPs with many local optima.

Acknowledgment

The authors are very grateful to the anonymous reviewers for their constructive comments to this paper. This work is supported by the National Science Foundation of China under Grant 61673355, 61271140 and 61203306.

Reference

[1] M. Crepinsek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Comput. Surv. 45 (3) (2013) 35.

[2] X. Li, M. Epitropakis, K. Deb, A. Engelbrecht, Seeking multiple solutions: an updated survey on niching methods and their applications, IEEE Trans. Evol. Comput. 21 (4) (2016) 518–538.

[3] D. E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[4] S. W. Mahfoud, Simple analytical models of genetic algorithms for multimodal function optimization, in: ICGA, Citeseer, 1993, pp. 643–654.

[5] O. J. Mengshoel, D. E. Goldberg, Probabilistic crowding: Deterministic crowding with probabilistic replacement, in: Proceedings of the Genetic and Evolutionary Computation, 1999, pp. 409–416.

[6] G. R. Harik, Finding multimodal solutions using restricted tournament selection, in: ICGA, 1995, pp. 24–31.

[7] A. Pétrowski, A clearing procedure as a niching method for genetic algorithms, in: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE, 1996, pp. 798–803.

[8] J.-P. Li, M. E. Balazs, G. T. Parks, P. J. Clarkson, A species conserving genetic algorithm for multimodal function optimization, Evol. Comput. 10 (3) (2002) 207–234.

[9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[10] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.

[11] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.

[12] S. Bandaru, K. Deb, A parameterless-niching-assisted bi-objective approach to multimodal optimization, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 95–102.

[13] L. Siwik, R. Dreżewski, Evolutionary multi-modal optimization with the use of multi-objective techniques, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2014, pp. 428–439.

[14] C. Segura, C. A. C. Coello, G. Miranda, C. León, Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization, Ann. Oper. Res. 240 (1) (2016) 217–250.

[15] E. De Jong, R. Watson, J. Pollack, Reducing bloat and promoting diversity using multi-objective methods, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Citeseer, 2001, pp. 11–18.

[16] H. A. Abbass, K. Deb, Searching under multi-evolutionary pressures, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2003, pp. 391–404.

[17] A. Toffolo, E. Benini, Genetic diversity as an objective in multi-objective evolutionary algorithms, Evol. Comput. 11 (2) (2003) 151–167.

[18] M. T. Jensen, Guiding single-objective optimization using multi-objective methods, in: Lecture Notes in Computer Science, Springer, 2003, pp. 268–279.

[19] M. T. Jensen, Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation, J. Math. Model. Algor. 3 (4) (2005) 323–347.

[20] C. Segura, C. A. C. Coello, E. Segredo, G. Miranda, C. León, Improving the diversity preservation of multi-objective approaches used for single-objective optimization, in: 2013 IEEE congress on evolutionary computation, IEEE, 2013, pp. 3198–3205.

[21] E. Segredo, C. Segura, C. Leon, E. Hart, A fuzzy logic controller applied to a diversity-based multi-objective evolutionary algorithm for single-objective optimisation, Soft Comput. 19 (10) (2015) 2927–2945.

[22] Y. Wang, H.-X. Li, G. G. Yen, W. Song, Mommop: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems, IEEE Trans. Cybern. 45 (4) (2015) 830–843.

[23] J.-Y. Ji, W.-J. Yu, W.-N. Chen, Z.-H. Zhan, J. Zhang, Solving multimodal optimization problems through a multiobjective optimization approach, in: 2017 Seventh International Conference on Information Science and Technology, IEEE, 2017, pp. 458–463.

[24] C. Segura, Improving diversity in evolutionary algorithms: New best solutions for frequency assignment, IEEE Trans. Evol. Comput. 21 (4) (2017) 539–553.

[25] D. E. Golberg, Genetic algorithms in search, optimization, and machine learning, Addion wesley 1989 (1989) 102.

[26] M. Jelasity, Uego, an abstract niching technique for global optimization, in: International Conference on Parallel Problem Solving from Nature, Springer, 1998, pp. 378–387.

[27] A. Della Cioppa, C. De Stefano, A. Marcelli, On the role of population size and niche radius in fitness sharing, IEEE Trans. Evol. Comput. 8 (6) (2004) 580–592.

[28] B. Qu, P. Suganthan, Modified species-based differential evolution with self-adaptive radius for multimodal optimization, in: International Conference on Computational Problem-Solving, IEEE, 2010, pp. 326–331.

[29] M. Li, D. Lin, J. Kou, Dynamics of fitness sharing evolutionary algorithms for coevolution of multiple species, Appl. Soft Comput. 10 (3) (2010) 832–848.

[30] O. M. Shir, M. Emmerich, T. Bäck, Adaptive niche radii and niche shapes approaches for niching with the cma-es, Evol. Comput. 18 (1) (2010) 97–126.

[31] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximations, and applications, IEEE Trans. Evol. Comput. 8 (5) (2004) 425–442.

[32] T. T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, Swarm Evol. Comput. 6 (2012) 1–24.

[33] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al., Optimization by simulated annealing, science 220 (4598) (1983) 671–680.

[34] T. Takahama, S. Sakai, Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation, in: 2010 IEEE Congress on Evolutionary Computation, IEEE, 2010, pp. 1–9.

[35] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.

[36] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization, Natur. Comput. (2005) 341–357.

[37] J. J. Liang, B. Y. Qu, P. N. Suganthan, Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization, Tech. Rep. (2013) 1–32.

[38] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization, Tech. Rep. (2016) 1–34.

[39] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.

[40] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, M. F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2) (2011) 1679–1696.

[41] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–66.

[42] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142.

[43] M. S. KıRan, M. GüNdüZ, A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems, Appl. Soft Comput. 13 (4) (2013) 2188–2203.

[44] K. Tang, P. Yang, X. Yao, Negatively correlated search, IEEE J. Sel. Areas Commun. 34 (3) (2016) 542–550.

[45] S. Nama, A. K. Saha, S. Ghosh, Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c- backfill, Appl. Soft Comput. 52 (2017) 885–897.

[46] G. Singh, K. Deep, Effectiveness of new multiple-pso based membrane optimization algorithms on cec 2014 benchmarks and iris classification, Natur. Comput. (2016) 1–24.

[47] V. Garg, K. Deep, Performance of laplacian biogeography-based optimization algorithm on cec 2014 continuous optimization benchmarks and camera calibration problem, Swarm Evol. Comput. 27 (2016) 132–144.

[48] B. Qu, J. Liang, J. Xiao, Z. Shang, Memetic differential evolution based on fitness euclidean-distance ratio, in: 2014 IEEE Congress on Evolutionary Computation, IEEE, 2014, pp. 2266–2273.

[49] G. Singh, K. Deep, A. K. Nagar, Cell-like p-systems based on rules of particle swarm optimization, Appl. Math Comput. 246 (2014) 546–560.

[50] D. Maharana, R. Kommadath, P. Kotecha, Dynamic yin-yang pair optimization and its performance on single objective real parameter problems of cec 2017, in: 2017 IEEE Congress on Evolutionary Computation, IEEE, 2017, pp. 2390–2396.

[51] R. Kommadath, P. Kotecha, Teaching learning based optimization with focused learning and its performance on cec2017 functions, in: 2017 IEEE Congress on Evolutionary Computation, IEEE, 2017, pp. 2397–2403.

[52] J. Robinson, Y. Rahmat-Samii, Particle swarm optimization in electromagnetics, IEEE Trans. Antennas Propag. 52 (2) (2004) 397–407.

[53] S. Xu, Y. Rahmat-Samii, Boundary conditions in particle swarm optimization revisited, IEEE Trans. Antennas Propag. 55 (3) (2007) 760–765.

[54] J. M. Johnson, V. Rahmat-Samii, Genetic algorithms in engineering electromagnetics, IEEE Antennas Propag. Mag. 39 (4) (1997) 7–21.

[55] S.-H. Yang, J.-F. Kiang, Optimization of sparse linear arrays using harmony search algorithms, IEEE Trans. Antennas Propag. 63 (11) (2015) 4732–4738.