

# Linear Array Pattern Synthesis Using an Improved Multiobjective Genetic Algorithm

Chuang HAN, Ling WANG, Zhaolin ZHANG, Jian XIE, Zijian XING

School of Electronics and Information, Northwestern Polytechnical University, 710072 Xi'an, Shannxi, China

hanchuang357@163.com, lingwang@nwpu.edu.cn, z184@sina.com, xiejian@nwpu.edu.cn, zjxing@nwpu.edu.cn

Submitted July 13, 2017 / Accepted August 10, 2017

**Abstract.** *In this paper, we propose an improved nondominated sorting genetic algorithm-II with scope constrained (INSGA-II/SC) with three modifications, which are dynamic nondomination strategy, scope-constrained strategy, and front uniformly distributed strategy. Here, the metric for multiobjective optimization mainly focuses on the computation complexity, convergence, and diversity of the final solutions. For a large search space in the initial process and a fast convergence in the last process, dynamic nondomination factor is considered in the rank operator. We can find a manageable number of Pareto solutions that are in the constrained scope instead of the entire Pareto front (PF) to reduce the computation complexity by scope-constrained strategy. In order to obtain a high performance for good representatives of the entire PF, the solutions closer to the uniformly distributed points on the current front will be chosen. In this paper, the proposed methods and two efficient multiobjective optimization methods are used for the optimization of mathematical problems and array pattern synthesis with lower side lobe level (SLL) and null. Numerical examples show that INSGA-II/SC has a high performance of diversity and convergence for the final solutions when compared with the other techniques published in the literature.*

## Keywords

Multiobjective optimization, convergence, diversity, array pattern synthesis, genetic algorithm

## 1. Introduction

Multiobjective optimization for array pattern synthesis has been an academic issue [1–10]. Thinned arrays have been an object of intense research due to several advantages associated with their lower cost, weight, power, and complexity compared with the fully filled arrays. The objectives of side lobe level (SLL), directive, main beam width, and null depth may conflict with each other.

Many algorithms [1], [2] combine these multiple objectives into a single function, typically in, commonly known as aggregating functions. An evolutionary method based on backtracking search optimization algorithm (BSA) is proposed for linear antenna array pattern synthesis with prescribed nulls and lower SLL [2]. The result obtained by BSA outperforms the results of the seventeen algorithms which include particle swarm optimization (PSO), multiobjective evolutionary algorithm based on decomposition (MOEA/D) and nondominated sorting genetic algorithm-II (NSGA-II). The SLL for 28-element array is  $-21.90$  dB, and three nulls at  $30^\circ$ ,  $32.5^\circ$ , and  $35^\circ$  are lower than  $-60$  dB by BSA [2]. With the same parameter set, a family of position-mutated hierarchical particle swarm optimization algorithms with time-varying acceleration coefficients (PM<sub>4</sub>HPSO-TVAC) [3] outperforms these algorithms for synthesizing unequally spaced 28-element linear array with the best SLL suppression of  $-23.63$  dB and nulls control below  $-60$  dB.

In [4], the authors show that the linear antenna array design can be modeled as a multiobjective optimization problem (MOP). Evolutionary multiobjective optimization methods, such as NSGA-II, differential evolution multiobjective (DEMO), strength-Pareto evolution algorithm (SPEA-2), and multiobjective particle swarm optimization (MOPSO), are provided for the synthesis problem of linear antenna arrays and concentric rings antenna arrays with the objectives of main beam width and SLL [1], [4–7]. NSGA-II [10] is a popular and efficient multiobjective genetic algorithm, which has been used in several engineering design problems [11–14], [30]. For instance, NSGA-II has been extensively applied in the synthesis of antenna arrays [15–18], which is considered to be one of the best evolutionary optimizer for multiobjective problems [7]. multiobjective differential evolution (MODE) algorithm and NSGA-II are employed to design concentric ring arrays with three-dimensional (3D) beam scanning [19]. The authors used NSGA-II for array pattern synthesis with lower SLL and null control, and the values of null depth are just  $-32.7$  and  $-22$  dB for the thinned linear array with 40 elements [20].

There are no improvements for the algorithms when synthesizing array pattern. What can be focused, the simu-

lation results of optimum solutions distributed in the Pareto front (PF) lose the diversity, and not all the solution can converge to the PF close enough. Some hybrid algorithms, such as memetic generalized differential evolution (MGDE3) [21] and the iterative fast Fourier transform (IFFT) technique with a judge factor introduced into the NSGA-II [22], try to improve the convergence and solution diversity.

J. Ouyang presented three modifications based on NSGA-II and applied it for array pattern synthesis. These modifications mainly focused on the distance selection strategy, and from the simulation results, these modifications obtain a better spread of the solutions and are more efficient than the NSGA-II algorithm in some mathematical problems [11]. For conformal phased array [14] and conformal conical surface linear phased array [23], the main beam width and SLL are taken into account simultaneously, and these modifications can get better solutions. From the objective space, the distribution of the final solutions may lose the uniformness for most mathematical problems and array pattern synthesis.

From the literatures proposed, the classical multiobjective optimization algorithms may be directly applied, and the null control for the array pattern synthesis is sparse in the multiobjective optimization problem. In this paper, we propose an improved NSGA-II with scope constrained (INSGA-II/SC) for the optimization of mathematical problems and array pattern synthesis. The metric for multiobjective optimization mainly focuses on the computation complexity, convergence, and diversity of the final solutions. INSGA-II/SC makes three modifications based on NSGA-II.

1) *Dynamic nondomination strategy*: In the initial process, a smaller nondomination factor can make a large search space, and a larger nondomination factor can sort the rank quickly, and the dynamic nondomination factor can get a fast convergence near the true PF.

2) *Scope-constrained strategy*: We can find a manageable number of Pareto solutions that are in the scope constrained instead of the entire PF to reduce the computation complexity.

3) *Front uniformly distributed strategy*: For every front, the best solutions are those closer to the uniformly distributed points on the current front, and the chosen solutions have good diversity compared with the solutions chosen by the crowded distance.

The remaining part of this paper is organized as follows. In Sec. 2, INSGA-II/SC with three modifications is proposed. The multiobjective optimizations of the mathematical problem and array pattern synthesis are presented in Sec. 3. Section 4 gives the metric for evaluating the performance of the multiobjective evolutionary algorithms and simulates numerical examples and describes the comparative performance of the presented technique. Concluding remarks are given in Sec. 5.

## 2. INSGA-II/SC with Three Modifications

### 2.1 Multiobjective Optimization Problem and NSGA-II

The multiobjective optimization problem can be expressed as

$$\begin{aligned} \min F(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))^T \\ \text{subject to } \mathbf{x} &= (x_1, x_2, \dots, x_n) \in \Omega \end{aligned} \quad (1)$$

where  $\mathbf{x}$  is the decision vector, and  $\Omega$  is the feasible region in decision space. Here  $M$  is the number of objectives, and  $n$  is the size of the decision vector.

Very often, since the objectives in (1) contradict each other, no point in  $\Omega$  minimizes all the objectives simultaneously. One has to balance them. The best tradeoffs among the objectives can be defined in terms of Pareto optimality [24].

Considering a minimization problem for each objective and  $\mathbf{p}, \mathbf{q} \in R^M$ ,  $\mathbf{p}$  is said to dominate  $\mathbf{q}$  (written as  $\mathbf{p} > \mathbf{q}$ ) if and only if  $p_i \leq q_i$  for every  $i \in \{1, \dots, M\}$  and  $p_i < q_i$  for at least one  $i$ . We can get a solution  $\mathbf{x}^* \in \Omega$ , the Pareto optimal to (1), if we cannot find a solution  $\mathbf{x} \in \Omega$  such that  $F(\mathbf{x}) > F(\mathbf{x}^*)$ . Then  $F(\mathbf{x}^*)$  is called a Pareto optimal (objective) vector. In other words, any improvement in a Pareto optimal solution in one objective must lead to the deterioration of at least one other objective. The set of all the Pareto optimal solutions is called the Pareto set (PS), and the set of all the Pareto optimal objective vectors is the PF [24]. The multiobjective optimization algorithm is aimed to find the optimum solutions approximating the actual PF.

Genetic algorithm is widely used in multiobjective problems with rooted position because it can produce and choose the solutions in one single iteration. NSGA-II is a popular and efficient multiobjective genetic algorithm that is considered to be one of the best evolutionary optimizers [10]. In this paper, we will present three modifications based on NSGA-II, and the procedure [1] for NSGA-II can be described as follows.

*Step 1)* Set generation number  $r = 0$  and generate an initial population  $P[0]$  with size  $N$ .

*Step 2)* Classify the individuals according to nondominated ranking system.

*Step 3)* Set a counter  $i = 1$ .

*Step 4)* Use Binary Crowded Tournament Selection, and apply crossover with probability  $p_c$  and mutation with probability  $p_m$ .

*Step 5)* Set  $i=i+1$ . If  $i > N$  then go to Step 6; otherwise go to Step 4.

*Step 6)* Replacement. Assign ranks to individuals in the population generated by the union of Parents and Children populations. Copy into the new population individuals from a front with lower index as long as the number of individuals

in the front does not overflow the population size ( $N$ ). In the last front to be copied, sort the individuals according to their crowding distance, eliminating those individuals with smaller crowding distance, until the total number of individuals ( $N$ ) is completed.

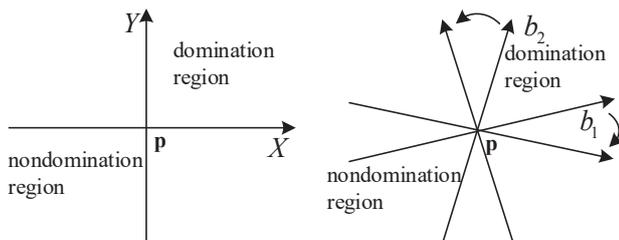
*Step 7)* Set  $r = r + 1$ . If  $r$  reaches the maximum number of generations, then stop; otherwise, go to Step 3.

The main idea in Step 2 is to classify the individuals according to their dominance relation, i.e., the set of nondominated individuals is said to be in front 0. After removing these individuals, the remaining nondominated solutions are in front 1. The procedure continues until all individuals are assigned to a front. Deb [10] explains the procedures involved in each step of this algorithm in detail. The individual representations as well as the crossover and mutation operators are same with those in [10].

## 2.2 Dynamic Nondomination Strategy

The ability of escaping the local optimum and convergence to the PF is the motivation for the improvement of the multiobjective optimization algorithm. For NSGA-II, the fast nondomination sorting approach has high performance and low computation complexity of  $O(MN^2)$  for sorting the front rank for every individual.

The normal nondomination strategy has been stated clearly in published literature [10]. For the bi-objective optimization problem, the domination relationship can be shown in Fig. 1(a). With the origin point  $\mathbf{p}$ , the left bottom is the nondomination region whose solutions dominate the solution  $\mathbf{p}$ , and the top right is the domination region whose solutions are dominated by the solution  $\mathbf{p}$ .



**Fig. 1.** The nondomination relationship for the normal strategy and the dynamic strategy.

In Step 6 of the NSGA-II, the operator of replacement is mainly based on the rank, which is obtained by the operator of the fast nondominated sorting approach. As a result, the nondomination strategy has great influence on the solution evolution. For a large search space, we need keep varied distributed solutions in the initial process. In this case, we need to control the weak nondomination strength, which means that a lot of similar individuals stay in one rank. For a fast convergence in the last process, the nondomination strength should be enhanced to remove the dominated individuals.

In this paper, we define the nondomination factor  $\alpha_M$  to control the nondomination strength. The fast nondominated sorting approach can be shown as follows: for  $\mathbf{p}, \mathbf{q} \in R^M$ , we say that  $\mathbf{p}$  dominates  $\mathbf{q}$  if and only if  $p_i \leq \alpha_M(q_j - p_j) + q_i$  for every  $i, j \in \{1, \dots, M\}$ , where  $i \neq j$  and  $p_i < \alpha_M(q_j - p_j) + q_i$  for at least one  $i, j$ .

Here we use the bi-objective optimization problem as an example to state the meaning of the nondomination factor. Different from the normal nondomination strategy, we set two lines with slope  $b_1$  and  $b_2$  shown in Fig. 1(b), and  $b_1 = 1/b_2 = \alpha_2$ . The nondomination region can be distinguished by linear inequalities that are shown as

$$\begin{cases} p_2 \leq b_1(p_1 - q_1) + q_2 \\ p_2 \geq b_2(p_1 - q_1) + q_2 \end{cases} \quad (2)$$

where the second inequality can also be written as  $p_1 \leq 1/b_2(p_2 - q_2) + q_1$ , where  $b_2 \neq 0$ .

For the case of  $\alpha_2 = 0$ , the nondomination situation is the same with normal nondomination, and the second function can be shown as  $p_1 \leq q_1$ . Combined with the equations, we can describe the nondomination region for the bi-objective optimization problem as

$$\begin{cases} p_1 \leq \alpha_2(q_2 - p_2) + q_1 \\ p_2 \leq \alpha_2(q_1 - p_1) + q_2 \end{cases} \quad (3)$$

Then, we can easily state the domination region as

$$\begin{cases} p_1 > \alpha_2(q_2 - p_2) + q_1 \\ p_2 > \alpha_2(q_1 - p_1) + q_2 \end{cases} \quad (4)$$

To implement the dynamic nondomination strategy to rank the individuals, the value of the nondomination factor must be cautiously chosen for different PFs. The value is always small to prevent the effective individuals to be discarded. In the initial process, a positive nondomination factor can make a large search space, and a negative nondomination factor can sort the rank quickly in the last process. For a fast convergence near the true PF, we present the dynamic nondomination factor, which can be shown as

$$\alpha_M(t) = \alpha_M^p - \frac{t}{T}(\alpha_M^p - \alpha_M^n) \quad (5)$$

where  $\alpha_M^p$  and  $\alpha_M^n$  are the positive nondomination factor and the negative nondomination factor respectively, and  $t$  is the current iteration with the total iteration number  $T$ . This dynamic nondomination factor is easily realized, and we can use different formations according to the optimization problem [25].

The domination count  $n_p$  and the set of solutions  $S_p$  dominated by  $\mathbf{p}$  can be calculated. For the comparison of two individuals, the computation complexity may increase to  $O(M(M-1))$ , so the comparison complexity of one nondomination sorting is  $O(M^2N^2)$ , which is larger than the normal nondomination sorting approach. For most optimization problems, the number of objectives  $M$  is always much smaller compared with the number of individuals  $N$ , and the comparison complexity may have little change.

### 2.3 Front Uniformly Distributed Strategy

To approximate the PF, the final preferred solution should be selected by a decision maker. In many MOPs, many or even infinite Pareto optimal vectors are distributed in the complete PF. Nevertheless, many multiobjective optimization algorithms are to find a manageable number of Pareto optimal vectors that are evenly distributed along the PF and, thus, are good representatives of the entire PF [1–10], [24]. Found in the literature, the goal for distributing individuals is that, along with convergence to the Pareto optimal set. It is also desired that an evolutionary algorithm maintains a good spread of solutions in the obtained set of solutions.

In MOPs, however, domination does not define a complete ordering among the solutions in the objective space, and a preferred selection operator is needed to produce a number of Pareto optimal solutions as diverse as possible for representing the whole PF.

In the well-known algorithm of NSGA-II [10], the authors use the crowded-comparison operator for maintaining diversity among population members instead of a user-defined parameter. This approach has a better computational complexity, and the performance is better than NSGA.

MOEA/D [24] explicitly decomposes the MOP into  $N$  scalar optimization subproblems, in which the objective is an aggregation of all the functions but not limited to the weighted sum approach and the Tchebycheff approach [26]. At each generation, the best solutions found so far for each subproblem are evolved simultaneously to compose the new population. The neighborhood relations among these subproblems are defined based on the distances between their aggregation coefficient vectors. The optimal solutions to two neighboring subproblems should be very similar. Each subproblem (i.e., scalar aggregation function) is optimized in MOEA/D using information only from its neighboring subproblems.

For the uniformness of final solutions, NSGA-II has no directly driven ability to select the uniformly distributed individuals. MOEA/D is better than NSGA-II for most ZDT test instances and three-objective test, but it has poor performance on ZDT3, which is attributed to the fact that the objectives in ZDT3 are disparately scaled. Although the result of ZDT4 has good distribution, some individuals deviate from the PF. The reason may be that the fixed aggregation coefficient vectors and mere neighborhood relations lead to lose the diversity of the individuals.

In this paper, front uniformly distributed strategy is presented to choose the individuals with high performance for good representatives of the entire PF. There are three steps to assign the expected error order for every individual. The first step is to find the expected positions that may be uniformly distributed or other distribution mode in current front. The next step is to calculate the expected error between the expected positions and every individual. We can choose the preferred individuals according to the lower expected error in the last step. The detail procedure can be shown below.

To find the expected positions, we should calculate the entire length of the current front, which is the total distance of every two nearest individuals after sorting the population according to each objective function value in ascending order of magnitude. The distance of two individuals is the Euclidean distance. The quantity  $\zeta$  serves as an estimate of the expected positions whose value is the total distance divided by the expected number  $K$ . All solutions in the current front are in a nondominated set  $P$  with the size  $L$ . The process is shown as

*Step 1)* Set  $\zeta(i) = 0$  for each  $i$ , and assign the size of  $P$  with  $L$ , the number of objectives with  $M$ . Initialize  $m = 1$  and  $i = 2$ .

*Step 2)* If  $m > M$ , go to Step 4; else go to Step 3, and sort  $P$  according to the  $m$ th objective value.

*Step 3)* If  $i > L$ , set  $m = m + 1$  and go to Step 2, else stay in this step with  $i = i + 1$  and calculate the Euclidean distance for every two individuals

$$\zeta(i) = \sqrt{\zeta^2(i) + ((f_m^i - f_m^{i-1}) / (f_m^{\max} - f_m^{\min}))^2}. \quad (6)$$

*Step 4)* Assign the number of the expected points as

$$K = \begin{cases} L & \text{if } L \leq N, \\ N & \text{if } L > N, \end{cases} \quad (7)$$

and calculate the estimate of expected positions as

$$\hat{\zeta} = \frac{\sum_{i=2}^L \zeta(i)}{K - 1}. \quad (8)$$

Here,  $f_m^i$  refers to the  $m$ th objective function value of the  $i$ th individual in set  $P$ , and the parameters  $f_m^{\max}$  and  $f_m^{\min}$  are the maximum and minimum values of the  $m$ th objective function. Each objective function is normalized before calculating the distance. Thereafter, for each objective function, the solutions with the smallest function value are assigned to zero. All other intermediate solutions are assigned a distance value equal to the square root of the normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The total distance is calculated as the sum of every individual distance value, and the expected number  $K$  is constrained as a maximum value of the population size  $N$ .

For the  $i$ th individual in set  $P$ , its expected error to the  $j$ th expected position can be calculated as

$$\hat{\Delta}_{ij} = \left| \hat{\zeta} \times (j - 1) - \sum_{i=1}^i \zeta(i) \right|. \quad (9)$$

This expected error  $\hat{\Delta}_{ij}$  can constitute the expected error matrix  $\hat{R}$ . In this matrix, we can find  $k$  individuals with a smaller expected error. The error-comparison operator to obtain the expected error order can be described below.

For every  $t$  that is set from 1 to  $K$ , we should find the minimum element  $\hat{\sigma}_t$  in  $\hat{R}$ , and the individual corresponding to the row is assigned with the expected error order of  $t$ ,

while the values in  $\hat{R}$  corresponding to the row and column are assigned with infinite values.

For this error-comparison operator, one expected position can map only one individual in  $P$  with the smallest expected error of distance. Other individuals without order will be set to infinite values. After all population members in set  $P$  are assigned an expected error order, we can compare the value with other solutions. The individuals with a smaller expected error order will be copied to the new population, which are, in some sense, closer to the expected positions. It is obvious that  $\Delta_{11}$  and  $\Delta_{lk}$  are zero, which state that the first and last individuals must be chosen for the new population.

The complexity of this procedure is governed by the sorting algorithm, which is the same with the computation of crowded density in NSGA-II. Therefore, the front uniformly distributed strategy has  $O(MN \log N)$  computational complexity.

### 2.4 Scope-Constrained Strategy

For most multiobjective optimization method, limited individuals evolve to approximate and represent the PF, but not all of them are useful for the application. In some cases, a part of the PF is enough with the constrained scope of certain objectives.

In this paper, under scope-constrained  $F_i^l$  and upper scope-constrained  $F_i^u$  for the  $i$ th objective are presented to get a part of PF. Then, the procedure for replacement is described as follows.

The new population individuals will be copied from the front with lower index as long as the number of individuals in the front does not overflow the population size  $N$ . In the last front, the individuals with lower expected error will be copied. For the last front, scope-constrained strategy should be adopted when the front uniformly distributed strategy is used to compute the expected error. When the number of the new population with the chosen number between  $F_i^l$  and  $F_i^u$  exceeds the total number  $N$  of individuals, only the individuals in the scope constrained are used to calculate the expected error distributed in the front. Thereafter, the individuals with smaller expected error in this scope constrained would be copied to the new population. On the contrary, the number of individuals in the scope constrained cannot meet the totality of the new population, and all the individuals in the last front will be used to calculate the expected error. When choosing the new population, the individuals in the scope constrained are all selected, then the individuals with smaller expected error out of the scope constrained will be copied to the new population until the total number  $N$  of individuals is completed.

For this strategy, the advantage is that the smaller size of individuals can achieve the goal of optimization. In the following examples, we use half the size of the population for the optimization with the scope of certain objectives constrained, and the result also has high performance.

## 3. Multiobjective Optimization Problems

### 3.1 Mathematical Optimization Problems

We use five widely used bi-objective ZDT test instances in comparing the proposed method with MOEA/D [24] and NSGA-II [10]. All these test instances are minimization of the objectives.

- ZDT1

$$f_1(x) = x_1,$$

$$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$$

where  $g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n - 1)$  and  $x = (x_1, \dots, x_n)^T \in [0, 1]^n$ . Its PF is convex, and we choose  $n = 30$  for our experiments.

- ZDT2

$$f_1(x) = x_1,$$

$$f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$$

where  $g(x)$  and the range and dimensionality of  $x$  are the same as that in ZDT1. Its PF is nonconvex.

- ZDT3

$$f_1(x) = x_1,$$

$$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)} - (f_1(x)/g(x)) \sin(10\pi x_1)]$$

where  $g(x)$  and the range and dimensionality of  $x$  are the same as that in ZDT1. Its PF is disconnected. The two objectives are disparately scaled in the PF, and  $f_1$  is from 0 to 0.852, while  $f_2$  is from  $-0.773$  to 1.

- ZDT4

$$f_1(x) = x_1,$$

$$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$$

where  $g(x) = 1 + 10(n - 1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)]$  and  $x = (x_1, \dots, x_n)^T \in [0, 1]^n \times [-5, 5]^{n-1}$ . It has many local PFs, and we choose  $n = 10$  for our experiments.

- ZDT6

$$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1),$$

$$f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$$

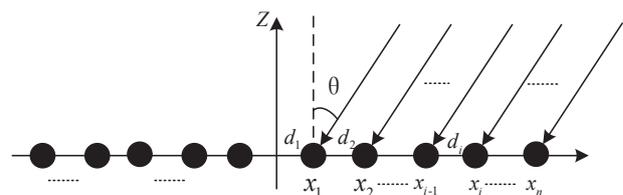


Fig. 2. Geometry of the 2n-elements linear array placed along the x-axis.

where  $g(x) = 1 + 9[(\sum_{i=2}^n x_i)/(n-1)]^{0.25}$  and  $x = (x_1, \dots, x_n)^T \in [0, 1]^n$ . Its PF is nonconvex. The distribution of the Pareto solutions in the Pareto front is very nonuniform. For a set of uniformly distributed points in the Pareto set in the decision space, their images are crowded in a corner of the Pareto front in the objective space. We choose  $n = 10$  for our experiments.

### 3.2 Array Pattern Optimization Problems

Consider a linear array of  $2n$  isotropic elements with a spacing of  $d$  as shown in Fig. 2. The position of each sensor along the  $x$ -axis is symmetric about the origin, which can be written as  $x_i$ . We just consider the array factor, which can be expressed as

$$AF(\theta) = \sum_{i=1}^n w_i \cos(2\pi x_i \sin \theta / \lambda) \quad (10)$$

where  $x_i = \sum_{j=1}^i d_j$ , and  $\theta$  (measured from  $z$ -axis) is the pitch angle of radiation for the transmit array and the incidence of the plane wave for the receive array.  $\lambda$  is the wavelength, and  $w_i$  is the weight coefficient of the  $i$ th sensor. Here, the decision vector is the sensor positions, and in this paper, we set  $d$  with a search range of  $[0.25, 1]$  normalized by  $\lambda$ .

As the objective of optimization is to minimize the SLL and null control of the array pattern by adjusting the geometry of the array subject to given design specifications and constraints, the fitness function can be defined as

$$f_1 = 20 \log_{10} \left( \max_{\theta_{SL} \in [-90^\circ, \theta_b] \cap (\theta_e, 90^\circ]} \left| \frac{AF(\theta_{SL})}{\max AF(\theta)} \right| \right), \quad (11)$$

$$f_2 = 20 \log_{10} \left( \sum_{i=1}^J \left| \frac{AF(\theta_i)}{\max AF(\theta)} \right| \right) \quad (12)$$

where  $\theta_{SL}$  is the spanned angle within the side lobe band except the range of the main lobe.  $[\theta_b, \theta_e]$  is the range of the main lobe, which will be fixed to a given range according to design specifications.  $\theta_i$  is the direction of  $i$ th null, and  $J$  is the number of nulls. The first objective is then evaluated excluding the main beam, and the SLL is measured in decibel. The  $f_1$  works for optimizing SLL, and  $f_2$  works for optimizing nulls. It is well known that array pattern optimization should be organized along a specific trade-off rule between the SLL and the null depth.

## 4. Numerical Examples and Results Analysis

### 4.1 Metrics for Evaluating the Performance of Algorithms

All the mathematical optimization problems have been independently run for 30 times for each instance. For NSGA-II, we have set the crossover and mutation probabilities equal to 0.9 and 0.1, respectively. All algorithms are compiled

using the same compiler (MATLAB 2012a) in a PC with Intel Core i7-7500U, at 2.70 GHz with 8 Gb RAM running Windows 10.

To compare the performances of different algorithms, several metrics should be used according to the nature of MOPs. Both generational distance and spread ( $\Delta$ ) [10], [27] have been widely used for evaluating the performance of the multiobjective evolutionary algorithms. In [24], the distance from representatives in the PF ( $D$ -metric) is adopted to measure both the diversity and convergence of the final population when the number of uniformly distributed points is large enough to represent the PF very well. For the mathematical optimization problems, the expected positions  $\zeta$  calculated from the actual PF are used to evaluate the performance of the final population. However, for the practical problem in engineering, the true Pareto front cannot be obtained [28], [29]. In the case when we cannot get the actual PF, we use the expected positions  $\hat{\zeta}$  calculated from the optimum front to evaluate the performance of the final population.

In this paper, we present two metrics, expectation value and variance value, to evaluate the performance.

#### 1) Expectation value

To have good diversity and convergence to the PF, we hope that all the individuals in the optimized population set  $P^*$  can approximate the expected positions with one-to-one correspondence. The performance index of the expectation value is calculated by the expected error for every individual. The expectation value can be defined as

$$E(P^*, \hat{\zeta}) = \frac{\sum_{t=1}^k \hat{\sigma}_t}{|P^*|}. \quad (13)$$

Here, the number  $k$  of expected positions is set as the size of  $P^*$ . The minimum expected error  $\hat{\sigma}_t$  for every individual can be obtained through the error-comparison operator. If we know the actual PF, we can calculate the expectation value by  $\zeta$  and  $\sigma_t$ . The smaller expectation value illustrates that most individuals can approximate the PF very well.

#### 2) Variance value

Ensuring that every individual approximates the corresponding position without a large error, we define the performance index of variance value as

$$D(P^*, \hat{\zeta}) = \frac{\sum_{t=1}^k [\hat{\sigma}_t - E(P^*, \hat{\zeta})]^2}{|P^*|}. \quad (14)$$

The smaller variance value illustrates that no one in the population deviates the PF.

### 4.2 Experimental Results of Mathematical Optimization Problems

In this experimental study, we compare the performance of MOEA/D, NSGA-II with the proposed INSGA-II (com-

bined with dynamic nondomination strategy and front uniformly distributed strategy), and INSGA-II/SC (combined with the proposed three strategies).

The parameters for all algorithms follow the same set with NSGA-II and MOEA/D in [10], [24]. In mathematical optimization problems, the population size in NSGA-II, MOEA/D, and INSGA-II is set to be 100 for all test instances, while this is 50 in INSGA-II/SC. All the algorithms have 2000 generations.

In MOEA/D, the setting of weight vectors is used by the Tchebycheff approach, and  $T$  is set to be 20 in this section.

Initial populations are generated by uniform random sampling from the feasible search space in both algorithms. MOEA/D is initialized as the lowest value found in the initial population. The simulated binary crossover (SBX) and polynomial mutation are used in these algorithms. The setting of the control parameters in these two operators is the same with [10], [24]. Following the practice in [19], the distribution indices in both SBX and the polynomial mutation are set to be 20. The crossover rate is 1.00, while the mutation rate is  $1/n$ , where  $n$  is the number of decision variables. We use the same distribution indices [8] for crossover and mutation operators as  $\eta_c = 20$  and  $\eta_m = 20$ , respectively. All the algorithms have been run 30 times independently for each test instance. The positive nondomination factor  $\alpha_m^p$  is set to 0.01, and the negative nondomination factor  $\alpha_m^n$  is set to -0.01 for INSGA-II and INSGA-II/SC.

Table 1 presents the average CPU time used by each algorithm for each test instance. It is clear that MOEA/D can run about twice as fast as NSGA-II with the same number of function evaluations, which is coincident with the results in [24]. INSGA-II need more time than NSGA-II because front uniformly distributed strategy spends more time in choosing the preferred individuals. The complexity of INSGA-II/SC is consistent with the population size, which is about half the time of INSGA-II and lesser time than NSGA-II.

To compare the performances of these algorithms, the performance indices of expectation value and variance value are used for each instance. The number of uniformly distributed points in the PF is much larger than the size of the population in the  $D$ -metric [24]. A better value of the  $D$ -metric cannot guarantee the final solutions uniformly distributed in the PF. In this paper, for the two performance indices, the number of uniformly distributed points is the same with the size of population, and every individual can be constrained.

Tables 2 and 3 present the average expectation value and variance value by different algorithms for each instance. It is clear that NSGA-II is better than MOEA/D with the Tchebycheff approach, which is opposite with the results in [24] because we change the number of uniformly distributed points

in the PF. These tables reveal that the final solutions obtained by INSGA-II and INSGA-II/SC have great improved performance with several orders of magnitude. The smaller expectation and variance values illustrate that the final solution obtained by the proposed method can approximate the PF very well.

Figure 3 shows in the objective space the distributions of the final solutions obtained in the run with the lowest expectation value for each test instance using NSGA-II, MOEA/D, and INSGA-II. It is evident that as to the uniformness of the final solutions, INSGA-II is better than others for all instances. For NSGA-II, the convergence has poor performance with several individuals deviating from the PF. The dynamic nondomination strategy overcomes this disadvantage, by which all the individuals can be distributed in the PF in INSGA-II. In terms of the solution quality of uniformity, INSGA-II obviously has high performance, which is attributed to the front uniformly distributed strategy.

Figure 4 shows the distributions of the final solutions obtained by INSGA-II/SC with  $F_2^l$  of 0 and  $F_2^u$  of 1 for ZDT3, and  $F_2^l$  of 0.5 and  $F_2^u$  of 1 for other instances. This method can achieve the same effect with INSGA-II in the scope constrained by half the size of the population. The convergence and diversity can match the analysis hereinbefore.

Instance	NSGA-II	MOEA/D	INSGA-II	INSGA-II/SC
ZDT1	82.50	45.30	97.34	51.40
ZDT2	82.42	45.27	96.48	49.20
ZDT3	86.92	48.03	108.15	53.99
ZDT4	78.96	41.63	93.95	47.41
ZDT6	73.94	40.49	89.09	44.59

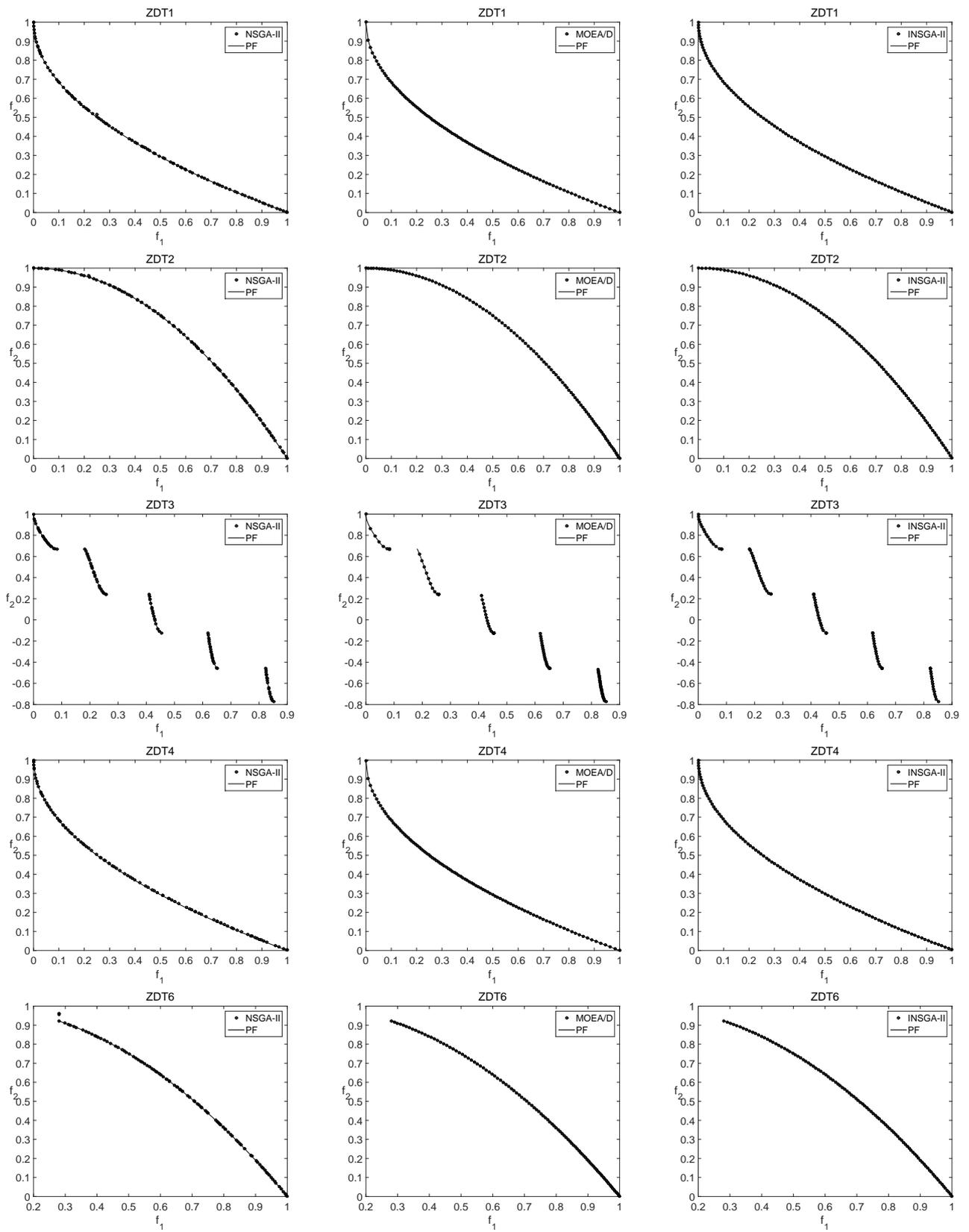
Tab. 1. Average CPU time (in seconds) used by MOEA/D with the Tchebycheff approach, NSGA-II, INSGA-II and INSGA-II/SC.

Instance	NSGA-II	MOEA/D	INSGA-II	INSGA-II/SC
ZDT1	0.0304	0.0695	2.2418e-4	3.3478e-4
ZDT2	0.0300	0.0343	1.9252e-4	6.5512e-4
ZDT3	0.0974	0.2944	0.0094	0.0068
ZDT4	0.0247	0.0691	3.1623e-4	4.1927e-4
ZDT6	0.0284	0.0402	4.8340e-4	4.4752e-4

Tab. 2. Average expectation values of the solutions found by MOEA/D, NSGA-II, INSGA-II and INSGA-II/SC.

Instance	NSGA-II	MOEA/D	INSGA-II	INSGA-II/SC
ZDT1	0.0171	0.0316	1.1409e-7	2.4260e-7
ZDT2	0.0166	0.0161	8.8773e-8	8.3267e-7
ZDT3	0.1049	0.4252	3.9017e-4	3.5462e-4
ZDT4	0.0117	0.0314	1.6993e-7	3.6739e-7
ZDT6	0.0147	0.0252	3.3582e-7	2.6759e-7

Tab. 3. Average variance values of the solutions found by MOEA/D, NSGA-II, INSGA-II and INSGA-II/SC.



**Fig. 3.** The distributions of the final solutions obtained in the run with the lowest expectation value for each test instance using NSGA-II, MOEA/D, and INSGA-II.

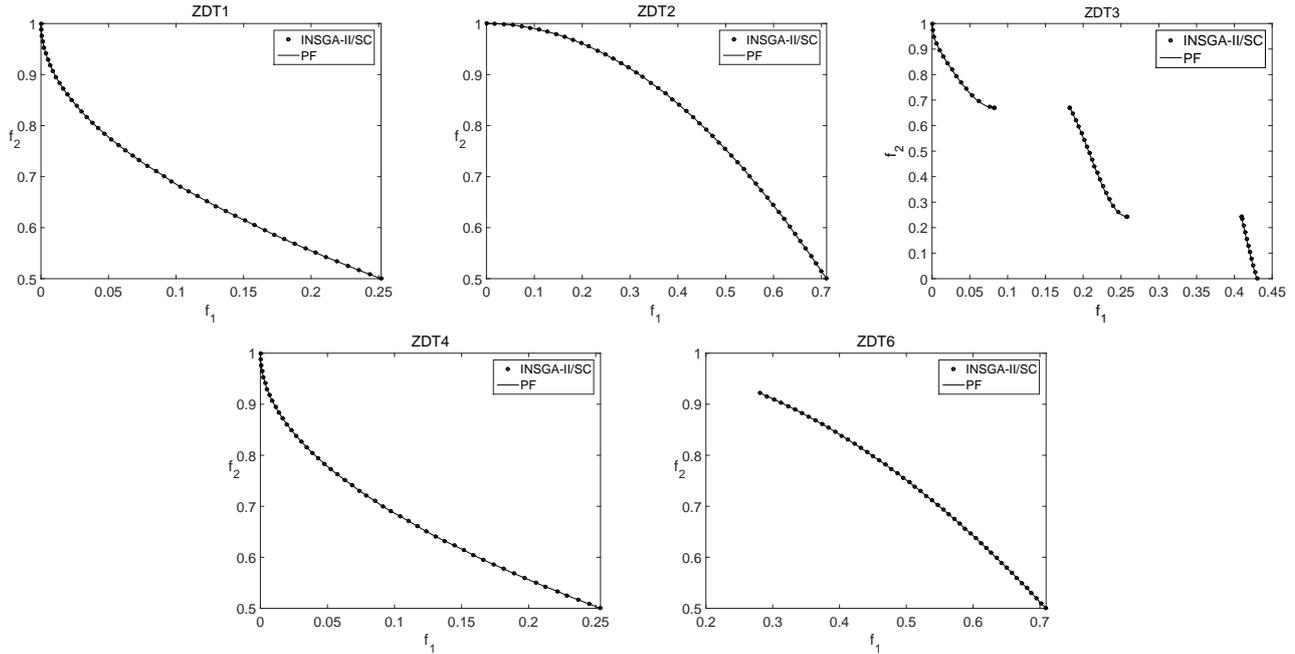


Fig. 4. The distributions of the final solutions obtained in the run with the lowest expectation value for each test instance using INSGA-II/SC.

Method	NSGA-II	MOEA/D	NSGA-II/SC	INSGA-II/SC
Expectation	32.1001	77.2667	9.2885	0.0563
Variance	1.6542e3	2.8320e3	249.8846	0.0137
SLL (dB) (Null=-60dB)	-23.98	-20.32	-24.05	-24.12

Tab. 4. Average expectation and variance values of the solutions found by MOEA/D with the Tchebycheff approach, NSGA-II, NSGA-II/SC and INSGA-II/SC.

### 4.3 Experimental Results of Array Pattern Synthesis

In this experimental study, we compare the performance of MOEA/D, NSGA-II with the proposed NSGA-II/SC (combined with scope-constrained strategy), and INSGA-II/SC. We synthesize a symmetric 28-element array with nulls in six specified directions at  $\pm 30^\circ$ ,  $\pm 32.5^\circ$ , and  $\pm 35^\circ$  [1], [3]. The desired first null position is set to  $8.5^\circ$  with  $\pm 12\%$  tolerance [3]. Two objectives of SLL and null depth are adopted to array pattern.

In the optimization problems of array pattern synthesis, the population size in each algorithm is set to be 400, and the algorithms have 2000 generations. The parameters about crossover and mutation and the size of neighborhood  $T$  are the same with that in mathematical optimization problems. All the algorithms have been run 30 times independently.

For NSGA-II/SC and INSGA-II/SC, the scope constrained is set to null depth with a lower scope of  $-100$  dB and an upper scope of  $-50$  dB. The positive nondomination factor  $\alpha_m^p$  is set to 0.01, and the negative nondomination factor  $\alpha_m^n$  is set to  $-0.01$  for INSGA-II/SC.

Table 4 shows the average expectation and variance values for each algorithm in 30 trials and lists the SLL with null depth of  $-60$  dB in the best case. It is clear that the proposed NSGA-II/SC and INSGA-II/SC have lower expected

and variance values, which means good convergence and diversity of the final solutions. At the same time, the SLL obtained by the two proposed methods can be much lower than others. Attributing to the dynamic nondomination strategy and scope-constrained strategy, INSGA-II/SC can have faster convergence in the setting part of PF with the same size of the population. INSGA-II/SC has the lowest expectation and variance values through the front uniformly distributed strategy, which means good performance on uniform diversity of the final solutions.

Figure 5 shows the distributions of the final solutions obtained in the run with the lowest expectation value for array pattern synthesis using MOEA/D. Compared with other figures, the solutions obtained by MOEA/D gather more seriously in a small scope. MOEA/D explicitly decomposes the MOP into  $N$  scalar optimization subproblems by different aggregation coefficient vectors. The weighted sum approach and the Tchebycheff approach were introduced in [10], [24], and other decomposition approaches could also be used in MOEA/D. Under the condition of a wide range of one-function value and a narrow range of one-function value, there are many difficulties to choose the appropriate aggregation coefficient vectors. The fixed vectors may lead individuals to the small scope where the objective rapidly changes without fault tolerance. In INSGA-II/SC, the error-comparison operator for  $\hat{R}$  can avoid this problem.

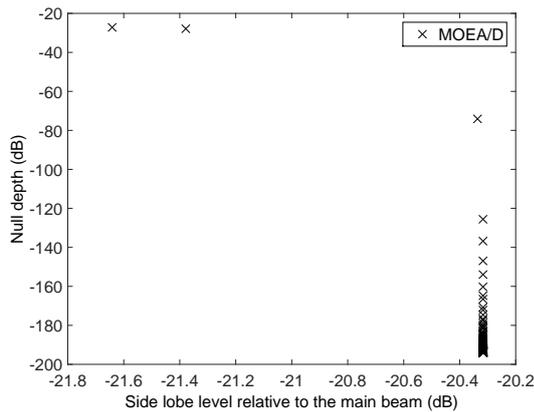


Fig. 5. The distributions of the final solutions obtained in the run with the lowest expectation value for array pattern synthesis using MOEA/D.

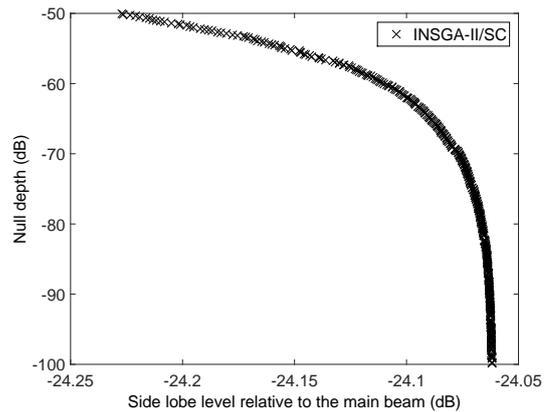


Fig. 8. The distributions of the final solutions obtained in the run with the lowest expectation value for array pattern synthesis using INSGA-II/SC.

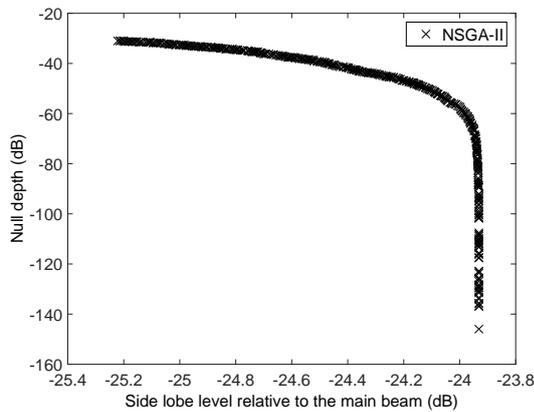


Fig. 6. The distributions of the final solutions obtained in the run with the lowest expectation value for array pattern synthesis using NSGA-II.

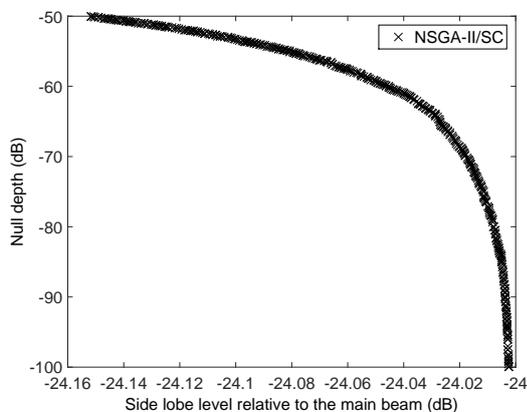


Fig. 7. The distributions of the final solutions obtained in the run with the lowest expectation value for array pattern synthesis using NSGA-II/SC.

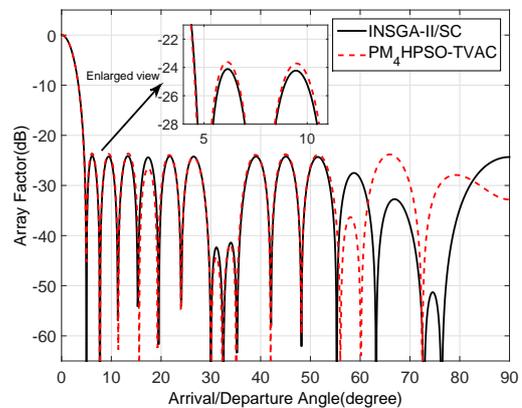


Fig. 9. The optimum array patterns obtained by INSGA-II/SC and PM<sub>4</sub>HPSO-TVAC.

Figures 6 and 7 show the distributions of the final solutions obtained in the run with the lowest expectation value for array pattern synthesis using NSGA-II and NSGA-II/SC. NSGA-II may have good performance on the convergence and diversity, but there are many useless solutions for the oversized null depth or oversized SLL. Hereafter, we find the excellent characteristic of scope-constrained strategy in NSGA-II/SC. It also promotes the convergence and diversity of final solutions further.

The final solutions shown in Fig. 8 by INSGA-II/SC have good spread in the PF, and the convergence is best than others. The best SLL of PM<sub>4</sub>HPSO-TVAC in [3] is -23.63 dB, whereas this value found by INSGA-II/SC is -24.12 dB when the null depth is below -60 dB. Figure 9 shows the array patterns obtained by INSGA-II/SC and PM<sub>4</sub>HPSO-TVAC. INSGA-II/SC outperforms other algorithms for synthesizing unequally spaced linear array with SLL suppression and null control. Table 5 shows the geometry of the best array found by INSGA-II/SC with a null depth of -60 dB for 28-element non-uniform linear array. In this paper, only linear arrays are dealt with. However, there are other antenna array structures that could provide a wider coverage than the linear geometry such as circular and planar geometries.

Element	Position (Normalized to $\lambda$ )	Element	Position (Normalized to $\lambda$ )
1	0.170	8	0.518
2	0.434	9	0.508
3	0.311	10	0.534
4	0.490	11	0.503
5	0.320	12	0.827
6	0.498	13	0.769
7	0.326	14	0.801

**Tab. 5.** Geometry of the best array found by INSGA-II/SC with null depth of  $-60$  dB for 28-element non-uniform linear array.

## 5. Conclusion

The objective of the proposed INSGA-II/SC is to get a set of solutions with good convergence and diversity. Contributing to the dynamic nondomination strategy, scope-constrained strategy, and front uniformly distributed strategy, the final solutions can converge nearly to the PF and be uniformly distributed on the PF through a fewer size of the population. INSGA-II/SC needs less computation time because it can achieve the same effect with a smaller size of the population. The results of five ZDT instances reveal that the final solutions obtained by INSGA-II and INSGA-II/SC have greatly improved performance with several orders of magnitude in the expectation value and variance value. For the array pattern synthesis of the 28-element linear array, the final solutions obtained by INSGA-II/SC have good convergence and diversity. The SLL obtained by the two proposed methods can be much lower than NSGA-II and MOEA/D with the Tchebycheff approach. The best SLL found by INSGA-II/SC is  $-24.12$  dB with the null depth below  $-60$  dB, which is better than the SLL of PM<sub>4</sub>HPSO-TVAC of  $-23.63$  dB.

## Acknowledgments

This study is supported in part by the National Natural Science Foundation of China (No. 61301093, 61601372, and 61601373).

## References

- [1] KHODIER, M. M., CHRISTIDOULOU, C. G. Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, 2005, vol. 53, no. 8, p. 2674–2679. DOI: 10.1109/TAP.2005.851762
- [2] GUNAY, K., DURMUS, A. Pattern nulling of linear antenna arrays using backtracking search optimization algorithm. *International Journal of Antennas and Propagation*, 2015, p. 1–10. DOI: 10.1155/2015/713080
- [3] BHATTACHARYA, R., BHATTACHARYYA, T. K., GARG, R. Position mutated hierarchical particle swarm optimization and its application in synthesis of unequally spaced antenna arrays. *IEEE Transactions on Antennas and Propagation*, 2012, vol. 60, no. 7, p. 3174–3181. DOI: 10.1109/TAP.2012.2196917
- [4] PANDURO, M. A., COVARRUBIAS, D. H., BRIZUELA, C. A., et al. A multi-objective approach in the linear antenna array design. *AEU-International Journal of Electronics and Communications*, 2005, vol. 59, no. 4, p. 205–212. DOI: 10.1016/j.aeue.2004.11.017
- [5] CIVICIOGLU, P. Circular antenna array design by using evolutionary search algorithms. *Progress in Electromagnetics Research B*, 2013, no. 54, p. 265–284. ISSN: 19376472
- [6] PANDURO, M. A., BRIZUELA, C. A., COVARRUBIAS, D. H. Design of electronically steerable linear arrays with evolutionary algorithms. *Applied Soft Computing*, 2008, vol. 8, no. 1, p. 46–54. DOI: 10.1016/j.asoc.2006.10.011
- [7] PANDURO, M. A., BRIZUELA, C. A. Evolutionary multi-objective design of non-uniform circular phased arrays. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 2008, vol. 27, no. 2, p. 551–566. DOI: 10.1108/03321640810847823
- [8] KADLEC, P., RAIDA, Z. Self-organizing migrating algorithm for optimization with general number of objectives. In *Proceedings of the 22nd International Conference, Radioelektronika 2012*, Brno, 2012, p. 111–115. ISBN: 9788021444683
- [9] JAYASINGHE, J., ANGUERA, J., UDUWAWALA, D., et al. A multipurpose genetically engineered microstrip patch antennas: bandwidth, gain, and polarization. *Microwave and Optical Technology Letters*, 2017, vol. 59, no. 4, p. 941–949. DOI: 10.1002/mop.30439
- [10] DEB, K., PRATAP, A., AGARWAL, S., et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, vol. 6, no. 2, p. 182–197. DOI: 10.1109/4235.996017
- [11] OUYANG, J., YANG, F., YANG, S. W., et al. The improved NSGA-II approach. *Journal of Electromagnetic Waves and Applications*, 2008, vol. 22, no. 11, p. 163–172. DOI: 10.1163/156939308784160703
- [12] JIANG, L., CUI, J., SHI, L., et al. Pareto optimal design of multilayer microwave absorbers for wide-angle incidence using genetic algorithms. *IET Microwaves Antennas and Propagation*, 2009, vol. 3, no. 4, p. 572–579. DOI: 10.1049/iet-map.2008.0059
- [13] JAYASINGHE, J., ANGUERA, J., UDUWAWALA, D. Genetic algorithm optimization of a high-directivity microstrip patch antenna having a rectangular profile. *Radioengineering*, 2013, vol. 22, no. 3, p. 700–707. ISSN: 1210-2512
- [14] OUYANG, J., YUAN, Q. R., YANG, F., et al. Synthesis of conformal phased array with improved NSGA-II algorithm. *IEEE Transactions on Antennas and Propagation*, 2009, vol. 57, no. 12, p. 4006–4009. DOI: 10.1109/TAP.2009.2026714
- [15] BIANCHI, D., GENOVESI, S., MONORCHIO, A. A multi-objective genetic algorithm applied to array synthesis at multiple frequencies. In *Proceeding of IEEE Antennas and Propagation International Symposium /USNC/URSI National Radio Science Meeting*, N Charleston, SC, 2009, p. 1172–1175. ISBN: 978-1-4244-3646-0
- [16] LIU, L., LU, J., YANG, S. Many-objective optimization of antenna arrays using an improved multiple-single-objective pareto sampling algorithm. *IEEE Antennas and Wireless Propagation Letters*, 2012, vol. 11, p. 399–402. DOI: 10.1109/LAWP.2012.2193653
- [17] BIANCHI, D., FONTANA, N., GENOVESI, S., et al. Multi-objective optimization of wideband spiral arrays. In *Proceeding of the 2010 IEEE Antennas and Propagation Society International Symposium*, Canada, 2010, p. 1–4. ISBN: 978-1-4244-4968-2
- [18] ADRA, S. F., FLEMING, P. J. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2011, vol. 15, no. 2, p. 183–195. DOI: 10.1109/TEVC.2010.2058117

- [19] ZHANG, L., JIAO, Y. C., CHEN, B., et al. Multiobjective optimization design of concentric ring arrays with 3D beam scanning using differential evolution algorithm. *International Journal of Numerical Modelling-Electronic Networks Devices and Fields*, 2013, vol. 26, no. 6, p. 602–619. DOI: 10.1002/jnm.1864
- [20] GUNES, F., TOKAN, F. Pareto optimal synthesis of the linear array geometry for minimum sidelobe level and null control during beam scanning. *International Journal of RF and Microwaves Computer-Aided Engineering*, 2010, vol. 20, no. 5, p. 557–566. DOI: 10.1002/mmce.20461
- [21] GOUDOS, S. K., GOTSIS, K. A., SIAKAVARA, K., et al. A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution. *IEEE Transactions on Antennas and Propagation*, 2013, vol. 61, no. 6, p. 3042–3052. DOI: 10.1109/TAP.2013.2254437
- [22] CHENG, Y. F., SHAO, W., ZHANG, S. J., et al. An improved multi-objective genetic algorithm for large planar array thinning. *IEEE Transactions on Magnetics*, 2016, vol. 52, no. 3. DOI: 10.1109/TMAG.2015.2481883
- [23] OUYANG, J., LUO, X., YANG, J., et al. Analysis and synthesis of conformal conical surface linear phased array with volume surface integral equation plus AEP (Active Element Pattern) and INSGA-II. *IET Microwaves Antennas and Propagation*, 2012, vol. 6, no. 11, p. 1277–1285. DOI: 10.1049/iet-map.2012.0054
- [24] ZHANG, Q. F., LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, vol. 11, no. 6, p. 712–731. DOI: 10.1109/TEVC.2007.892759
- [25] HAN, C., WANG, L. Array pattern synthesis using particle swarm optimization with dynamic inertia weight. *International Journal of Antennas and Propagation*, 2016, 7 p. DOI: 10.1155/2016/1829458
- [26] YUAN, Y., XU, H., WANG, B., et al. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 2016, vol. 20, no. 2, p. 180–198. DOI: 10.1109/TEVC.2015.2443001
- [27] DEB, K., JAIN, H. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 2014, vol. 18, no. 4, p. 577–601. DOI: 10.1109/TEVC.2013.2281535
- [28] DEB, K. *Multi-objective Optimization using Evolutionary Algorithms*. Chichester: John Wiley and Sons, 2001. ISBN: 978-0-471-87339-6
- [29] COELLO, C. A. C., LAMONT, G. B., VELDHIJZEN, D. A. V. *Evolutionary Algorithms for Solving Multi-objective Problems*. New York: Springer, 2007. ISBN: 978-0-387-36797-2
- [30] CAI, Q., GONG, M. G., RUAN, S. S., et al. Network structural balance based on evolutionary multiobjective optimization: a two-step approach. *IEEE Transactions on Evolutionary Computation*, 2015, vol. 19, no. 6, p. 903–916. DOI: 10.1109/TEVC.2015.2424081

## About the Authors ...

**Chuang HAN** was born in Hebei, China, in 1989. He received the B.Sc. and M.Sc. degrees in Electronic Engineering from the Northwestern Polytechnical University in 2012 and 2015, respectively. Currently, he is working on a Ph.D. degree in Northwestern Polytechnical University. His recent research interests include array signal process, antenna analysis and synthesis, satellite communication systems, and satellite navigation systems.

**Ling WANG** was born in Guizhou, China, in 1978. He received the B.Sc., M.Sc., and Ph.D. degrees from Xidian University, Xi'an, China, in 1999, 2002 and 2004, respectively. From 2004 to 2007, he worked at Siemens Academy of wireless communication unit and Nokia Siemens Networks research center. Since 2007, he has been with School of Electronic and Information in Northwestern Polytechnical University. His current research interests include B3G/4G mobile communications; vehicle tracking, telemetry, and command (TT&C); anti-jamming for communications and navigation systems; cognitive radio.

**Zhaolin ZHANG** received the B.Sc., M.Sc., and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2000, 2005 and 2012, respectively. Since 2000, he has been a teacher with School of Electronic and Information in Northwestern Polytechnical University, where he was promoted to be an associate professor. His current research interests include anti-jamming for communications and navigation systems; array signal process, and multimedia communication.

**Jian XIE** was born in Jiangsu, China, in 1986. He received the M.Sc. and Ph.D. degree from the School of Electronic Engineering, Xidian University in 2012 and 2015, respectively. He is now an associate professor with Northwestern Polytechnical University. His research interests include antenna array processing and radar signal processing.

**Zijian XING** received the B.Eng. degree, M.Eng. degree and Ph.D degree from Northwestern Polytechnical University in 2008, 2011 and 2014 respectively. He worked as assistant professor in Northwestern Polytechnical University from 2014. His research interests include antenna array, circular polarization antenna and antennas for RFID application.