



Research article

Modified chemical reaction optimization and its application in engineering problems

Shijing Ma¹, Yunhe Wang² and Shouwei Zhang^{1,*}

¹ School of Physical Education, Northeast Normal University, Changchun 130022, China

² School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China

* **Correspondence:** Email: zhangshouweinenu@126.com, zhangsw178@nenu.edu.cn.

Abstract: Chemical Reaction Optimization (CRO) is a simple and efficient evolutionary optimization algorithm by simulating chemical reactions. As far as the current research is concerned, the algorithm has been successfully used for solving a number of real-world optimization tasks. In our paper, a new real encoded chemical reaction optimization algorithm is proposed to boost the efficiency of the optimization operations in standard chemical reactions optimization algorithm. Inspired by the evolutionary operation of the differential evolution algorithm, an improved search operation mechanism is proposed based on the underlying operation. It is modeled to further explore the search space of the algorithm under the best individuals. Afterwards, to control the perturbation frequency of the search strategy, the modification rate is increased to balance between the exploration ability and mining ability of the algorithm. Meanwhile, we also propose a new population initialization method that incorporates several models to produce high-quality initialized populations. To validate the effectiveness of the algorithm, nine unconstrained optimization algorithms are used as benchmark functions. As observed from the experimental results, it is evident that the proposed algorithm is significantly better than the standard chemical reaction algorithm and other evolutionary optimization algorithms. Then, we also apply the proposed model to address the synthesis problem of two antenna array synthesis. The results also reveal that the proposed algorithm is superior to other approaches from different perspectives.

Keywords: chemical reaction optimization; application; optimization

1. Introduction

It is prevalent to solve the optimization problems in our life, which is generally done by selecting a set of variables to reach an optimal value while satisfying a set of constraints. For instance, antenna arrays are a typical global optimization problem that is important for detecting and processing signals from different directions. The goal of antenna array synthesis is to generate a physical layout of the array in

which the radiation pattern is as close as possible to the desired pattern. Unfortunately, most optimization problems are always nonlinear, non-differentiable, and multipolar. Numerous optimization methods have been proposed for solving these optimization problems. Classical optimization methods, including gradient methods and alternating projection methods, are constantly being proposed. Nevertheless, it is known that classical optimization methods require a fixed point that is fairly near to the final value, otherwise they are likely to get trapped in local optimal. Moreover, the quality of the solution strongly depends on the estimation of the initial values. However, the goodness of the final solution scheme of the algorithm mostly depends on the setting of the initial scheme. If the initial solution falls in the worse region of the solution, then the search will only be conducted to find the best solution in the worse solution region. Considering the shortcomings of these techniques, evolutionary algorithms are the best choice for solving large-scale optimization problems since evolutionary algorithms are population-based search algorithms with high convergence rates and have been successfully used for large-scale optimization problems extensively [1, 2]. Therefore, different kinds of evolutionary algorithms, such as particle swarm optimization algorithms (PSO) [3, 4], ant colony optimization (ACO), biogeography-based optimization (BBO) [5–8], animal migration optimization (AMO) [9, 10], cuckoo search algorithm (CS) [11–13], and artificial bee colony (ABC) [14], have been adopted to handle antenna arrays, especially on large-scale optimization problems.

Recently, chemical reaction optimization (CRO) [15] is a population-based heuristic evolutionary algorithm that simulates molecular collisions in chemical reactions. The algorithm simulates the intermolecular interactions of molecules from a high-energy state into a low-energy state to explore the global optimal solution in the search space. To achieve efficient performance and extensive applications of the algorithm, different variations of the CRO algorithm were proposed successively to enhance the performance of the traditional CRO algorithm. In addition, there are many works that focus on the applications of CRO algorithms; for instance, Lam and Li [16] developed a real-coded CRO, called RCCRO for addressing continuous optimization problems. The experimental results indicate that the average performance of RCCRO is superior to other algorithms. In [17], the authors developed a double molecular structure-based chemical reaction optimization (DMSCRO) method and the CRO scheme is employed to outline the directed acyclic graph jobs scheduling in heterogeneous computing systems. Xu *et al.* [18] developed a hybrid chemical reaction optimization framework to solve the DAG-based task scheduling problem. It combines the novel heuristic approaches and a proposed selection strategy. Dam *et al.* [19] proposed a new method that integrates the chemical reaction optimization scheme with the unified tabu search heuristic method for solving the capacitated vehicle routing problem.

Lam and Li [20] solved the peer to peer live streaming with chemical reaction optimization and experimental results demonstrate that CRO beats many commonly-used techniques for containing population transitions in the system applications. In [21], Lam and Li used chemical reaction optimization to resolve the grid scheduling problem. Experimental results indicate that the proposed algorithm surpasses many existing evolutionary methods for the majority of cases tested. Truong *et al.* [22] introduced a chemical reaction optimization algorithm (CROG) coupled with a greedy strategy to tackle the quadratic allocation problem. The algorithm also proposes a design method for the update operator and a parameter tuning method. Unfortunately, it is difficult for these approaches to balance the exploitation and exploration capability of CRO algorithms simultaneously. Therefore, it is necessary to develop new optimization strategies for chemical reaction optimization to further enhance the performance of the algorithms.

To address these issues, we propose a new realistic encoded chemical reaction optimization algorithm that can boost the performance of traditional chemical reaction algorithms. First, inspired by the search mechanism in differential evolution algorithms, we propose a modified chemical reaction optimization (MCRO) model based on the optimal solution in the population. Then, to further improve the performance of the algorithm, the perturbation frequency of individuals is manipulated by increasing the modification rate of individuals and balance the exploitation and exploration ability of the algorithm. Moreover, to further accelerate the convergence of the algorithm, we assemble the stochastic initialization, opposition learning methods, and generalized opposition learning methods to obtain a new population initialization method. Finally, to demonstrate the superiority of the proposed algorithm, the algorithm is compared with several benchmark functions and applied on two antenna arrays. Based on the experimental results, it can be observed that our proposed algorithm is superior to other state-of-the-art algorithms from several perspectives.

2. Chemical reaction optimization

2.1. The basic concept of the CRO

Chemical Reaction Optimization (CRO) is an evolutionary algorithm first published by Lam and Li in [22, 23], which simulates the collision of molecules in a chemical reaction. For each molecule, its molecular structure is similar to the optimization problem and the molecule denotes the solution to the problem under consideration. Potential energy (χ) is the objective function value and kinetic energy (η) can be interpreted as the tolerance of the system, which accepts a solution that is worse than the current one. This approach can assist the solution to be partially replaced locally. Assuming that the numerator is ω , χ can be calculated as follows: $\chi_\omega = f(\omega)$.

2.2. Elementary reactions

In a standard CRO, the algorithm simulates four different types of chemical reactions, comprising on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. They are mainly employed to control the redistribution of energy between molecules and buffers in the solution. Among them, single operations include wall ineffective collisions and disintegration, and the rest are multiple operations.

When a molecule strikes the wall of a container and then bounces away, the on-wall ineffective collision reaction is invoked. The process will have some change. The molecular structure ω and χ will change to a generated molecular ω' in the neighborhood of ω if the condition is satisfied under Eq 1.

$$\chi_\omega + \eta_\omega \geq \chi_{\omega'} \quad (1)$$

After an invalid collision at the wall, a certain fraction of the η of the transformed molecule is withdrawn to the buffer. Then, the η of the new molecule ω' can be as follows:

$$\eta_{\omega'} = (\chi_\omega + \eta_\omega - \eta_\omega) \times a \quad (2)$$

where $a \in [\eta^{LR}, 1]$, η^{LR} is a statistical parameter that limits the maximum percentage of η that can be lost at each time. As long as there is enough χ at the beginning, it is possible for a molecule with lower χ to

transform into a molecule with higher χ , corresponding to a poorer solution. Following the collision, the molecule has fewer η .

The decomposition collision reaction results from a molecule hitting a wall, which then breaks one molecule into several molecules. Assume that ω produces ω'_1 and ω'_2 , i.e., $\omega \rightarrow \omega'_1 + \omega'_2$. Any mechanism, which can generate ω'_1 and ω'_2 from ω , is allowed. The algorithm considers two cases for the decomposition reaction: first, each molecule needs enough energy to complete the decomposition. The χ of the resulting molecule is denoted by the following:

$$\begin{aligned}\eta_{\omega'_1} &= (\chi_{\omega} + \eta_{\omega} - \chi_{\omega'_1} - \chi_{\omega'_2}) \times q \\ \eta_{\omega'_2} &= (\chi_{\omega} + \eta_{\omega} - \chi_{\omega'_1} - \chi_{\omega'_2}) \times (1 - q)\end{aligned}\quad (3)$$

Based on the above analysis, the following conditions need to be satisfied for molecular decomposition:

$$\chi_{\omega} + \eta_{\omega} \geq \chi_{\omega'_1} + \chi_{\omega'_2} \quad (4)$$

where q is a random number that takes values in the range $[0, 1]$. Meanwhile, the algorithm considers another condition:

$$\begin{aligned}\eta_{\omega'_1} &= (\chi_{\omega} + \eta_{\omega} - \chi_{\omega'_1} - \chi_{\omega'_2} + \text{buffer}) \times k_1 \times k_2 \\ \eta_{\omega'_2} &= (\chi_{\omega} + \eta_{\omega} - \chi_{\omega'_1} - \chi_{\omega'_2} + \text{buffer}) \times k_3 \times k_4\end{aligned}\quad (5)$$

$$\chi_{\omega} + \eta_{\omega} + \text{buffer} \geq \chi_{\omega'_1} + \chi_{\omega'_2} \quad (6)$$

where k_1, k_2, k_3 and k_4 are random numbers which take values in the range of $[0, 1]$. The buffer can be defined as below:

$$\text{buffer} = \chi_{\omega} + \eta_{\omega} - \chi_{\omega'_1} - \chi_{\omega'_2} + \text{buffer} - \eta_{\omega'_1} - \eta_{\omega'_2} \quad (7)$$

Null collisions between molecules happen when two or more molecules collide with each other and then separate. The energy conservation condition can be expressed in the following way:

$$\chi_{\omega} + \eta_{\omega} + \eta_{\omega_1} + \eta_{\omega_2} \geq \chi_{\omega'_1} + \chi_{\omega'_2} \quad (8)$$

Then the η of two transformed molecules can be described as follows:

$$\begin{aligned}\eta_{\omega'_0} &= (\chi_{\omega} + \eta_{\omega} + \eta_{\omega_1} + \eta_{\omega_2} - \chi_{\omega'_1} - \chi_{\omega'_2}) \times q \\ \eta_{\omega'_2} &= (\chi_{\omega} + \eta_{\omega} + \eta_{\omega_1} + \eta_{\omega_2} - \chi_{\omega'_1} - \chi_{\omega'_2}) \times (1 - q)\end{aligned}\quad (9)$$

where q is a random number that takes values in the range $[0, 1]$.

The synthesis is the opposite of decomposition. Synthesis occurs when a molecule collides and bond together. The reaction conditions that allow the synthesis reaction are description below:

$$\chi_{\omega} + \eta_{\omega} + \eta_{\omega_1} + \eta_{\omega_2} \geq \chi_{\omega'} \quad (10)$$

Then the η for the new molecule is considered by the following method.

$$\eta_{\omega'} = \chi_{\omega} + \eta_{\omega} + \eta_{\omega_1} + \eta_{\omega_2} - \chi_{\omega'} \quad (11)$$

To address the continuous optimization problem more efficiently, Lam and Li proposed a real coded version of CRO, namely RCCRO. Subsequently, a similar modified neighborhood search approach was

used. Among them, the Gaussian distribution model was proposed to generate perturbations to search continuous space neighborhood. Their update method can be characterized here below:

$$\omega'(i) = \omega(i) + N(0, \sigma^2) \quad (12)$$

where σ holds the step size and determines the efficiency of the algorithm. Because different problems have different characteristics, various σ should be adopted. In the algorithm, the neighborhood method will be employed for one-wall and intermolecular ineffective collision. The framework of the algorithm is shown in Figure 1. In summary, the CRO algorithm starts with the initialization. Then if it is an inter-molecular ineffective collision, the single molecule reaction is employed. Otherwise we will carry out the bimolecule reaction. In terms of the single molecule reaction, if the composition condition is satisfied, then it will conduct the decomposition, otherwise, it will conduct the on-wall ineffective collision. In terms of the bimolecular reaction, if the synthesis condition is met, the synthesis reaction is conducted, otherwise the inter-molecular ineffective collision reaction is conducted. After these steps, new minimum points are checked. If the stopping criterion is not matched, then the above steps will be repeated. Otherwise, the global minimum point is obtained.

3. Modified chemical reaction optimization (MCRO)

3.1. Initial population

For evolutionary algorithms, population initialization is a crucial component. The goodness of population initialization directly influences the final convergence speed of the algorithm and the quality of the optimal solution. In fact, most algorithms use random initialization to generate the initial population in the absence of a priori information. Moreover, it is difficult to guarantee the quality of initialized population solutions by individual population initialization methods. Therefore, this paper proposes a new initialization method to generate the initial population by fusing random initialization, opposition-based learning initialization [24] and generalized opposition learning initialization [25]. This method can enhance the performance of CRO by preventing premature convergence to local optimal. Table 4 summarized the framework of this method. Figure 2 depicted the difference between the random population initiation and the opposition-based population initialization discussed above.

3.2. A modified search equation

Differential evolution algorithm (DE) [26] is an efficient stochastic search method which was originally proposed mainly for solving continuous optimization problems. Its simple structure and ease of completion allowed the algorithm to be quickly applied to different optimizations and achieve impressive performance. The algorithm also has the identical evolutionary process as other evolutionary algorithms. In the algorithm, we can see that the algorithm consists of three main processes: mutation, crossover and selection. At the beginning of the algorithm, a mid-autumn is generated using a random initialization, followed by the operations of mutation, crossover and selection to generate a new population. It is worth noting that several different variation operations have been proposed in the DE algorithm, such as:

$$\text{DE/best/1} : V_i = X_{\text{best}} + F \cdot (X_{r1} - X_{r2}) \quad (13)$$

where $i \in \{1, \dots, NP\}$ and r_1 and r_2 are random integer indices selected from $\{1, \dots, NP\}$ that are different from each other. F is a scaling factor or amplification factor. X_{best} , the base vector to be

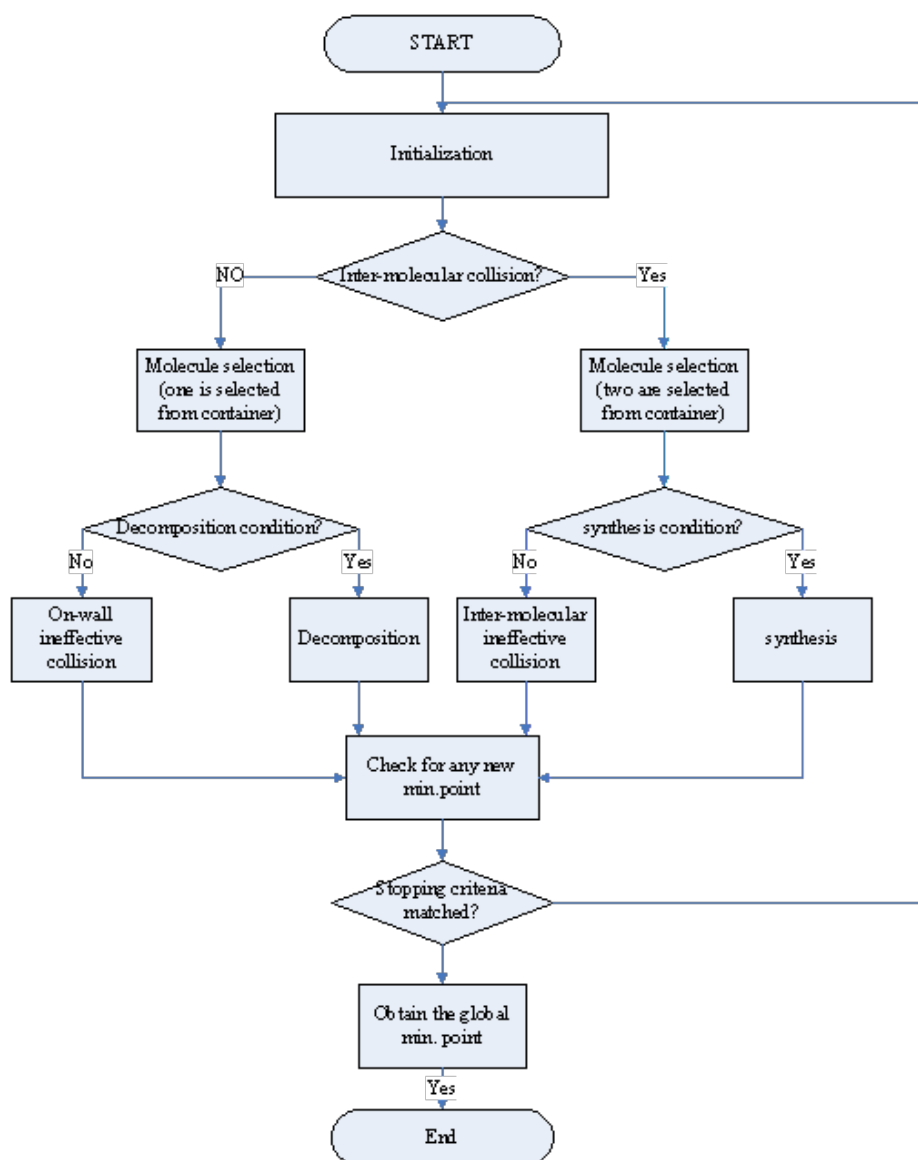


Figure 1. The framework of the CRO algorithm.

perturbed, is the best individual in the current population and indicates the best information in the population.

In fact, the information about the best individuals in the current population is very useful for the whole population, and if the information is used wisely it is possible to accelerate the convergence rate of the population. For the strategy "DE/best/1", the best solution can explore the movement of the current population. Based on DE algorithm and the property of CRO, we propose a novel search mechanism to improve CRO:

$$v_{ij} = x_{bestj} + \varphi_{ij}(x_{ij} - x_{bj}) \quad (14)$$

where φ_{ij} denotes the scaling factor that controls the speed of optimization of the algorithm. It is calculated at the beginning of each generation. If $rand < rand$, then φ_{ij} is equal to $5 \times (rand - rand) \times (rand - rand)$, otherwise, φ_{ij} equals to $randn$. We can note that "rand" denotes a random number generated from

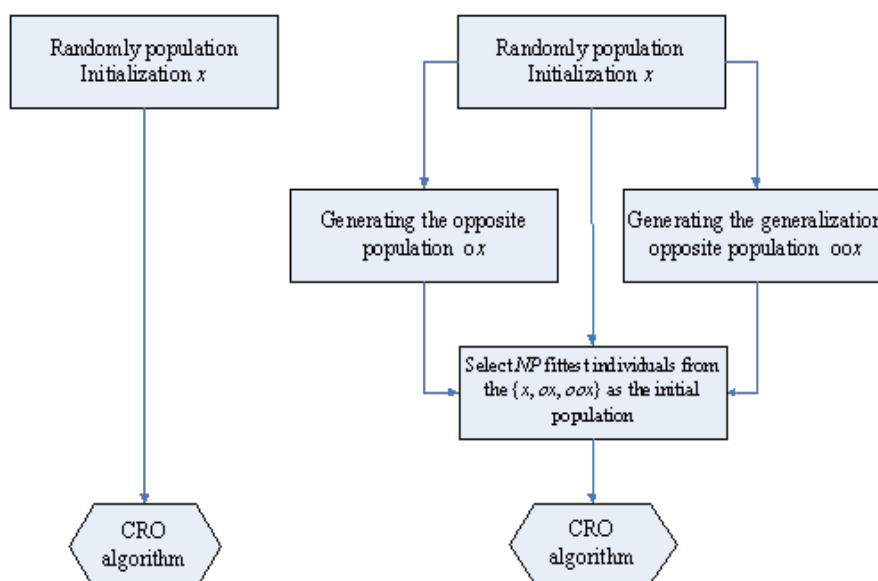


Figure 2. The difference between the random population initialization and above mentioned opposition based population initialization.

the range of $[-1, 1]$, $i \in \{1, 2, \dots, NP\}$, and $j \in \{1, 2, \dots, D\}$ are randomly chosen indexes, "randn" denotes a number generated through a Gaussian distribution with mean "0" and standard deviation "1".

To control the frequency of perturbation, it makes the algorithm can solve the difficult problem better. A control parameter, the modification rate (MR), is proposed, which serves mainly to generate a candidate position v_{ij} from the current memory of x_{ij} as described below:

$$v_{ij} = \begin{cases} x_{\text{best}} + \varphi_{ij}(x_{ij} - x_{bj}), & \text{if } r_{ij} \leq MR \\ x_{ij} & \text{otherwise} \end{cases} \quad (15)$$

where r_{ij} denotes a random number generated from the range of $[-1, 1]$. This new rule is similar with the binomial crossover of the DE algorithm, which can enhance the potential diversity of the population.

3.3. Boundary restrictions

For evolutionary algorithms, the setting of bounds is crucial. During the search process, the individuals in the part of MCRO algorithm can easily move to the boundary of the whole search space, which renders many of the solutions invalid. Therefore, it is important to restrict the individuals around the boundary to the search space once again. In other words, if most of the individuals are on the boundary, it is difficult for the algorithm to find the optimal solution to the problem, which will necessarily make the algorithm fall into local optimal solutions and make the whole population very easy to lose the population diversity. To address this problem, we propose to use the following rule for the setting of off-boundary individuals:

$$x_i = \begin{cases} x_{\min} + \text{mod}((x_{\min} - x_i), (x_i - x_{\min})) \\ x_{\max} - \text{mod}((x_i - x_{\max}), (x_{\max} - x_{\min})) \end{cases} \quad (16)$$

Procedure A new initialization approach

Begin

Randomly initialize each individual in population x .

Set the individual counter $i=1, j=1$

for $i = 1 \rightarrow NP$ **do**

for $j = 1 \rightarrow D$ **do**

$$OX_{i,j} = x_{\min,j} + x_{\max,j} - x_{i,j}$$

end for

end for

Set the individual counter $i = 1, j = 1$

for $i = 1 \rightarrow NP$ **do**

for $j = 1 \rightarrow D$ **do**

$$OOX_{i,j} = x_{\min,j} + x_{\max,j} - x_{i,j}$$

end for

end for

The fitness of each individual was assessed and the most suitable individual for NP from $\{X(NP) \cup OX(NP) \cup OOX(NP)\}$ was selected as the initial population.

End

3.4. Time complexity analysis

In this section, the time complexity analysis of the proposed MCRO algorithm for the benchmark function optimization is analyzed. The time mainly depends on the searching loop. At first, the time complexity of the initialization method is $O(NP \times D)$, where NP is the population size and D is the number of variables. Then for each iteration in the loop, the time is mainly spent in evaluating the fitness of each individual, which also costs $O(NP \times D)$. Therefore, the time complexity of MCRO is $O((I + 1)(NP \times D))$, where I is the number of iterations for the optimization process.

4. Results

4.1. The benchmark functions optimization by MCRO

To validate the effectiveness of the MCRO algorithm proposed in this paper, we first applied the algorithm to solve nine standard global optimization algorithms that are widely used. To ensure the fairness of the algorithm comparison, we did not modify any of the problems, which are tabulated in Table 1. As can be observed from the table, the first five problems are unimodal global optimization problems, where f05 is a noisy quadratic function. The remaining problems are all multimodal global optimization problems. It is worth noting that the local minima of these problems are increasing as the size of the problem increases, which is the reason why the difficulty of solving the problem increases with the size of the problem.

To illustrate the superiority of the MCRO algorithm proposed in our paper, we compared the MCRO algorithm with the standard CRO, the Differential Evolution algorithm (DE), the Adaptive Differential Evolution (jDE) [27] and the Modified Artificial Bee Colony algorithm (MABC) [28]. Across the experiments, the algorithms were computed with a maximum number of functions of 150,000 and

Table 1. Benchmark functions based in our experimental study.

Test Function	D	Range	optimum
$f_{01} = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
$f_{02} = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$f_{03} = \max_i \{x_i , 1 \leq i \leq D\}$	30	[-100,100]	0
$f_{04} = \sum_{i=1}^D ([x_i + 0.5])^2$	30	[-100,100]	0
$f_{05} = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1)$	30	[-1.28,1.28]	0
$f_{06} = \frac{1}{400} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-32,32]	0
$f_{07} = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	30	[-600,600]	0
$f_{08} = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_i + 1) \right] \right. \\ \left. + (yD - 1)^2 + \sum_{i=1}^D u(x_i, 10, 100, 4) \right\}$ $y_i = 1 + \frac{x_i+1}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$f_{09} = 0.1 \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_i + 1) \right] + (yD - 1)^2 \right\} \\ + \sum_{i=1}^D u(x_i, 10, 100, 4)$	30	[-50,50]	0

50 independent experiments were conducted for each problem, and their average values were finally compared as the final results. To compare fairly, the parameter of different algorithm can be set as follows: for the CRO algorithm, the population size NP is 20, KELossRate is 0.1, Molecoll is 0.2, InitialKE is 1000, and the value of decThres and synThres is 150000 and 0, respectively. The value of MR is 0.2. For the DE algorithm, the population size is 100, F is 0.3 and CR is 0.7. The parameter of the jDE is the same with the paper [27]. For modified artificial bee colony algorithm (MABC), the parameters of this algorithm are identical to the paper [28]. The results are listed in Table 2. From Table 2, we can observe that the results of MCRO are better than the other algorithms while Figure 3 shows the 2D plot of f07.

4.2. Array pattern synthesis using MCRO

To further demonstrate the effectiveness of the MCRO algorithm proposed in this paper, the proposed algorithm is applied to two different antenna array problems, including reconfigurable Antenna Array and linear antenna. To enable a fair comparison of the algorithms, the same number of function calculations is necessary. Therefore, in this paper, we use the same number of function evaluations for each algorithm as the final termination condition. As a result, we can evaluate the performance of the algorithms by the results of different algorithms and finally illustrate the efficiency of the proposed algorithm.

4.2.1. Synthesis of reconfigurable antenna-array

In this section, we applied our proposed MCRO to address the reconfigurable dual-beam antenna array. In this problem, when the phase distribution of the array is suitably modified, the amplitude distribution can produce pencil-shaped or sector power patterns. In addition, to reduce the effect of coupling between elements, an additional term is incorporated in the objective function [29]. Therefore,

Table 2. Performance comparisons of DE, jDE, MABC, CRO and MCRO.

F	DE	jDE	MABC	CRO	MCRO
f01	6.1695e-14± 4.7200e-14	2.0141e-28± 4.1612e-28	1.9244e- 39±4.3030e-39	6.7348e- 007±2.1237e- 007	5.5207e-71 ±1.0949e-70
f02	4.2283e-07 ±3.0823e-07	1.6421e-17± 1.0644e-17	6.8088e-18± 1.5132e-17	0.00213.6637e- 004	3.1288e- 43±3.9422e-43
f03	0.0458 ±0.0809	0.8695± 0.7707	13.3649 ± 3.6487	0.0097±0.0035	2.7781e-10 ± 2.2871e-10
f04	0± 0	0±0	0 ± 0	0±0	0± 0
f05	0.0096±0.0021	0.0063±0.0014	0.0311 ± 0.0083	0.0036±0.0013	0.0083± 0.0023
f06	1.3524e-13 ±7.7685e-14	0±0	2.9016e-08± 6.4883e-08	4.7687e- 07±1.5030e-007	0±0
f07	7.8443e-08± 3.1063e-08	7.9936e-15±0	2.8747e-06 ± 6.4281e-06	0.0024±4.2876e- 004	1.6520e-14± 5.9448e-15
f 08	3.5919e-15 ±2.2490e-15	4.6115e-30 ±5.3009e-30	2.3631e-07 ±5.2841e-07	0.1371±0.3257	1.6222e-32 ± 7.0698e-34
f 09	4.7192e-14 ±3.9138e-14	5.5873e- 29±7.6744e-29	1.4035e-09 ±3.1382e-09	7.5513e- 006±5.3363e- 006	7.0479e-31 ± 7.6213e-31

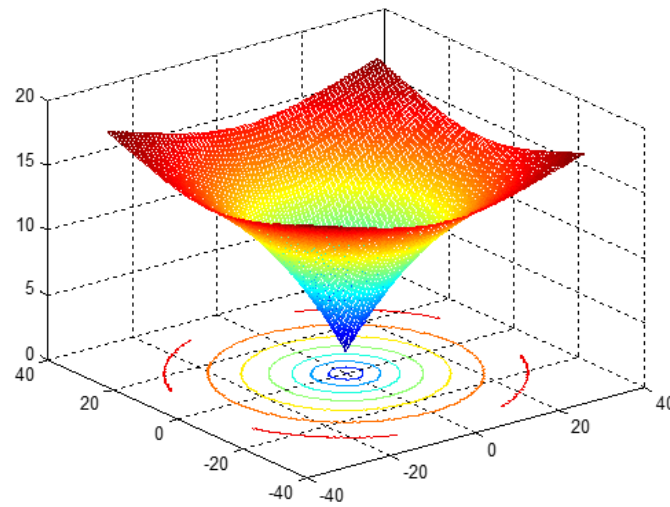


Figure 3. 2D Ackley function.

we can optimize the amplitude-excitation dynamic range (ARD) to minimize the mutual coupling problem [30,31]. Then, the objective function to be optimized in this paper is defined as below.

$$Ec(P) = \sum_{i=1}^3 \left(P_{i,d}^{(p)} - P_i^{(i)} \right)^2 + \sum_{i=1}^4 \left(P_{i,d}^{(s)} - P_i^{(s)} \right)^2 + ADR \quad (17)$$

where ADR denotes the amplitude-dynamic ratio. Based on the above description, it can be noticed that the algorithm can motivate the difference between the amplitudes by optimizing the ADR. Meanwhile, we can reveal that the difference between ADR and amplitude is linear, and minimizing the ADR is accompanied by a smaller difference between amplitudes. Therefore, in this paper, we optimize the reconfigurable antenna-array by optimizing the effect of coupling.

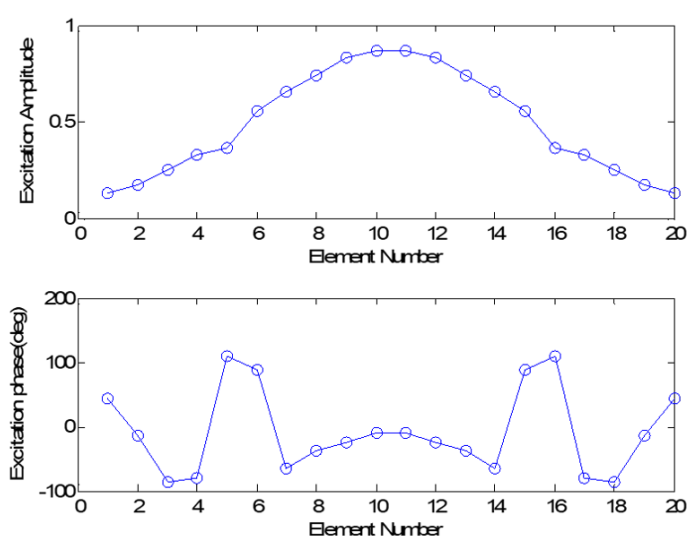
Table 3. The results of Experimental I and Experiment I with ADR.

Element Number	Experiment I		Experiment I with ADR	
	Amplitude	Phase[deg.]	Amplitude	Phase[deg.]
1/20	0.129861	-10.6276	0.149815	-0.80199
2/19	0.174584	-25.4484	0.149813	-38.653
3/18	0.255019	-36.7229	0.154374	-29.1362
4/17	0.331274	-64.5232	0.256851	-71.1288
5/16	0.367296	87.84077	0.307724	-72.5951
6/15	0.556305	108.9342	0.412374	-97.438
7/14	0.652548	-80.0141	0.483299	97.30298
8/13	0.737711	-87.0023	0.564111	53.28282
9/12	0.831561	-13.7764	0.616299	-106.146
10/11	0.868279	43.79851	0.645976	88.67032
ADR	6.68		4.31	
Fitness value	0.16		0.04	

Table 4. Performance comparison of MCRO with G3-GA, DE, jDE, MABC and MCRO.

	Exp-I without ADR		Exp-I with ADR	
	fitness		ADR	fitness
G3-GA	0.16		4.4137	0.1028
DE	0.16		4.3470	0.04
jDE	0.36		4.40	0.10
MABC	0.16		4.31	0.06
MCRO	0.16		4.31	0.04

In order to efficiently design the above reconfigurable antenna array problem, we use our algorithm MCRO algorithm to tackle the problem. We compare the MCRO algorithm with some other algorithms including G3-GA, DE, jDE, MABC and MCRO. In the MCRO algorithm, the population size is 20 and the number of function computations is 20000, and the value of MR is 0.7. The rest of the parameters are the same as those of the previous global optimization algorithm. The experimental results are tabulated in Tables 3 and 4, mainly to determine the amplitude and phase excitation patterns for the dual-beam optimization. From these tables, we can see that the MCRO algorithm is superior to other algorithms, especially for reconfigurable antenna array designs with coupling effects. For the problems without coupling effects, we can find that most of these algorithms can find the optimal solutions. For problems with coupling effects, we can find that MCRO gives the best ADR and fitness values. These results show that our proposed algorithm is superior to other evolutionary algorithms. Figures 4 and 5 show the optimized excitation pattern and the dual-beam pattern, respectively. Figure 5 shows the design parameters that satisfy both pencil and sector beams. Figures 6 and 7 show the same case as Figures 4 and 5.

**Figure 4.** Amplitude and phase excitation without coupling effect.

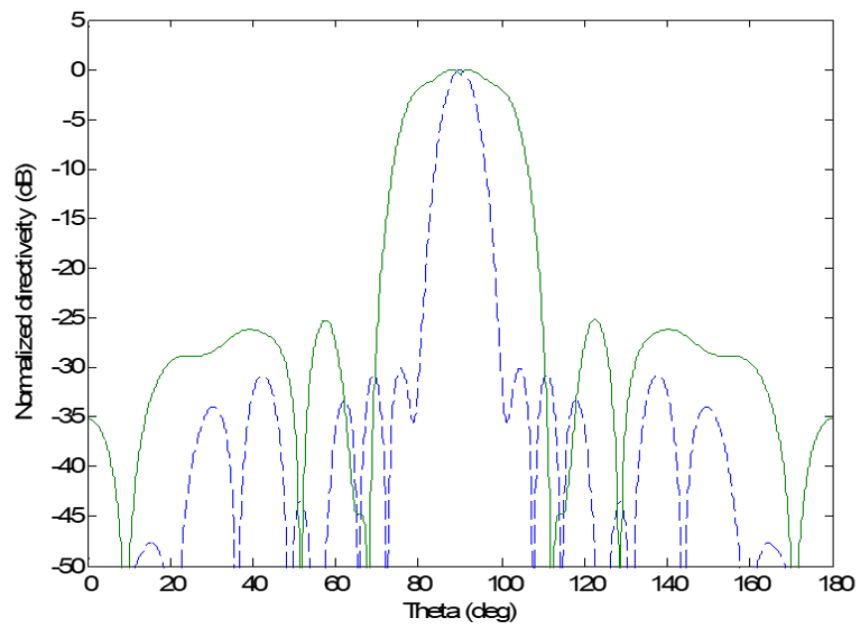


Figure 5. Dual-beam array patterns without coupling effect.

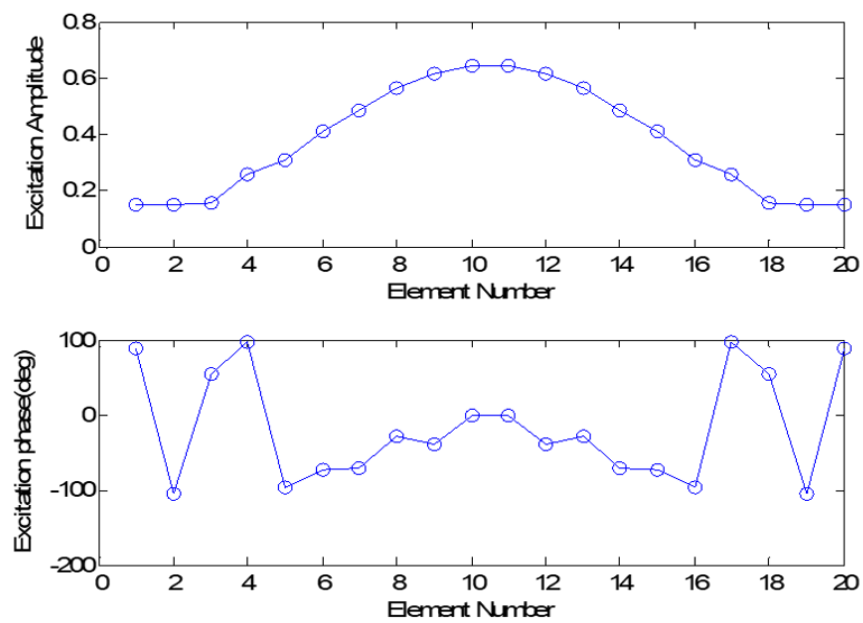


Figure 6. Dual-beam array patterns with coupling effect.

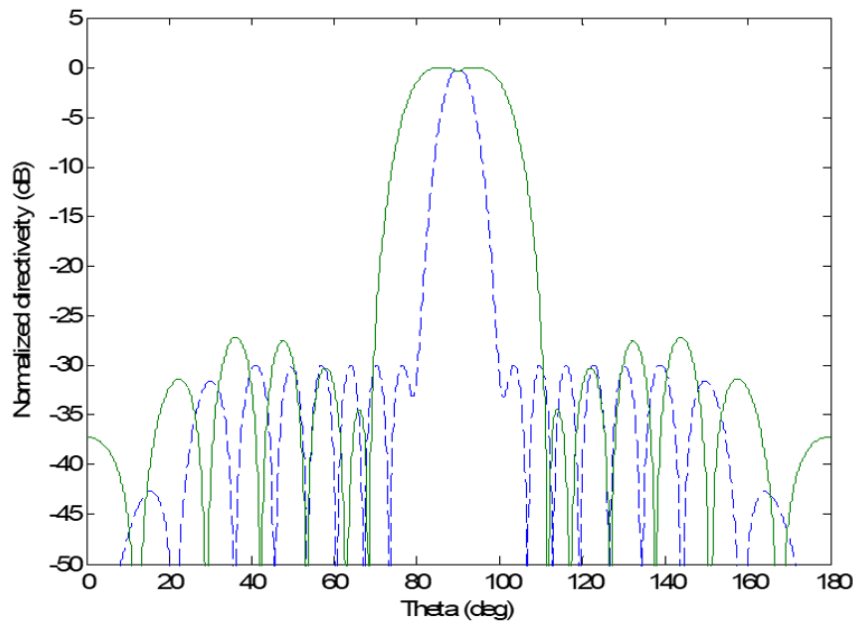


Figure 7. Dual-beam array patterns with coupling effect.

4.3. Synthesis of linear antenna

In this section, we applied our proposed MCRO to address the linear antenna. To design of a linear antenna array [32], let us assume that we have $2N$ isotropic radiators placed symmetrically along x-axis. The objective function of side lobe suppression is:

$$f_1 = \sum_i \frac{1}{\Delta\phi_i} \int_{\phi_{ii}}^{\phi_{ii}} |AF(\phi)|^2 d\phi \quad (18)$$

And for null control:

$$f_2 = \sum_k |AF(\phi_k)|^2 d\phi \quad (19)$$

where $\Delta\phi_i$ is the bandwidth to be restrained as $\phi_{ui} - \phi_{li}$, ϕ_k is the null direction. To optimize both objectives simultaneously, we sum Eqs 18 and 19 and complete their sum to make the objective function that our algorithm needs to optimize.

In order to elaborate the efficiency of the proposed MCRO algorithm, we came to test the reduction of the level of the side lobes in the linear array. Concurrently, we ran each algorithm independently for 25 times in each problem. The termination condition for each run of each algorithm was 5×10^4 number of function evaluations. All other parameters of the algorithms were consistent with the previous experiments. Meanwhile, the same number of function evaluations is applied to the other algorithms to construct a fair experiment.

In this section, we design an array of 26 elements that has minimum SLL in bands $[0^\circ, 82^\circ]$ and $[98^\circ, 180^\circ]$ and null direction in 20° . Tables 5 and 6 summarize the results of the eight different evolutionary algorithms. From these tables, it can be viewed that our proposed MCRO algorithm is to be superior to other algorithms. Also, Figure 8 summarizes the array patterns of MCRO algorithm with

other algorithms. From Figure 8, it can be seen that MCRO can suppress the generation of side lobes to the maximum extent. Moreover, it also obtains the minimum gain in the 20^{circ} null.

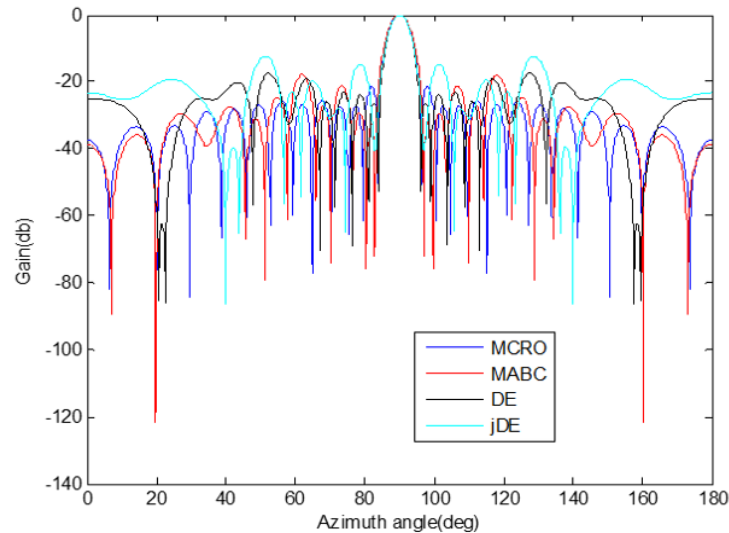


Figure 8. 13-element array for minimum SLL $[0^\circ, 82^\circ]$ and $[98^\circ, 180^\circ]$ and NULL 20° .

Table 5. Geometry of the 26 element linear array, normalized numbers with respect to $\lambda/2$ and null at 20° .

MCRO	0.301	0.939	1.98	2.012	3.367	3.541	5.284	7.049	4.590	8.021	6.093	10.881	9.411
MABC	0.330	0.979	1.821	2.039	3.267	2.897	4.151	6.574	7.749	4.880	5.799	9.544	10.909
DE	0.472	1.049	2.090	2.058	2.897	3.799	4.825	6.318	5.373	7.475	8.61	10.231	11.558
jDE	0.338	1.252	1.527	1.527	2.743	3.077	4.341	6.834	8.475	5.132	10.25	7.954	11.614

Table 6. Mean final objective function value, standard deviation, best, worst, median, and the Rank.

Algorithm	MCRO	DE	jDE	MABC
Mean	0.01194	0.02763	0.02356	0.0177
Best	0.00921	0.01984	0.01736	0.0131
Median	0.01219	0.02915	0.02319	0.0167
Worst	0.01488	0.03396	0.02982	0.0224
std	0.00209	0.00368	0.00443	0.0039
Rank	1	4	3	2

5. Conclusions

In this paper, a new evolutionary search strategy is employed to enhance the performance of the algorithm. Moreover, we present a new population initialization by fusing stochastic initialization,

opposition learning method, and generalized opposition learning method. It is worth noting that our proposed algorithm is simple in structure and easy to implement. To demonstrate the superiority of our proposed MCRO algorithm, we used nine benchmark functions as test functions. The results show that the proposed algorithm significantly outperforms the other algorithms. In addition, the algorithm was tested on the synthesis problem of two antenna arrays. We analyzed and evaluated the experimental results and compared them with other evolutionary optimization algorithms. In terms of the algorithm's performance, our algorithm is significantly better than the other algorithms. In the future, we would like to apply our proposed algorithm MCRO to solve other antenna arrays, such as circular antenna arrays. More quantifiable experiments on those arrays are expected. Besides, we will focus on exploring other strategies such as multiobjective mechanism to enhance the efficiency of the algorithm. Meanwhile, other evaluation criteria will be computed to estimate the performance of those extended algorithms.

Acknowledgments

The work described in this paper was substantially supported by the National Natural Science Foundation of China under Grant No. 62076109.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. H. Zhu, Y. Wang, Z. Ma, X. Li, A comparative study of swarm intelligence algorithms for ucav path-planning problems, *Mathematics*, **9** (2021), 171.
2. Y. Wang, Z. Ma, K. Wong, X. Li, Nature-inspired multiobjective patient stratification from cancer gene expression data, *Inf. Sci.*, **526** (2020), 245–262.
3. S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, J. C. Bansal, A survey on parallel particle swarm optimization algorithms, *Arab. J. Sci. Eng.*, **44** (2019), 2899–2923.
4. Y. Wang, X. Li, K.-C. Wong, Y. Chang, S. Yang, Evolutionary multiobjective clustering algorithms with ensemble for patient stratification, *IEEE Trans. Cybern.*, 2021.
5. X. Li, J. Wang, J. Zhou, M. Yin, A perturb biogeography based optimization with mutation for global numerical optimization, *Appl. Math. Comput.*, **218** (2011), 598–609.
6. X. Li, M. Yin, Multiobjective binary biogeography based optimization for feature selection using gene expression data, *IEEE Trans. NanoBiosci.*, **12** (2013), 343–353.
7. B. Liu, M. Tian, C. Zhang, X. Li, Discrete biogeography based optimization for feature selection in molecular signatures, *Mol. Inform.*, **34** (2015), 197–215.
8. X. Li, M. Yin, Hybrid differential evolution with biogeography-based optimization for design of a reconfigurable antenna array with discrete phase shifters, *Int. J. Antenn. Propag.*, **2011** (2011).
9. X. Li, J. Zhang, M. Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, *Neural. Comput. Appl.*, **24** (2014), 1867–1877.

10. Y. Cao, X. Li, J. Wang, Opposition-based animal migration optimization, *Math. Probl. Eng.*, **2013** (2013).
11. X. Li, J. Wang, M. Yin, Enhancing the performance of cuckoo search algorithm using orthogonal learning method, *Neural. Comput. Appl.*, **24** (2014), 1233–1247.
12. X. Li, M. Yin, Modified cuckoo search algorithm with self adaptive parameter method, *Inf. Sci.*, **298** (2015), 80–97.
13. X. Li, M. Yin, A particle swarm inspired cuckoo search algorithm for real parameter optimization, *Soft Comput.*, **20** (2016), 1389–1413.
14. X. Li, S. Ma, Multiobjective discrete artificial bee colony algorithm for multiobjective permutation flow shop scheduling problem with sequence dependent setup times, *IEEE Trans. Eng. Manag.*, **64** (2017), 149–165.
15. A. Lam, V. Li, Chemical-reaction-inspired metaheuristic for optimization, *IEEE Trans. Evol. Comput.*, **14** (2010), 381–399.
16. A. Lam, V. Li, J. Yu, Real-coded chemical reaction optimization, *IEEE Trans. Evol. Comput.*, **16** (2012), 339–353.
17. Y. Xu, K. Li, L. He, T. K. Truong, A dag scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization, *J. Parallel Distrib. Comput.*, **73** (2013), 1306–1322.
18. Y. Xu, K. Li, L. He, L. Zhang, K. Li, A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems, *IEEE T. Parall. Distr.*, **26** (2014), 3208–3222.
19. T.-L. Dam, K. Li, P. Fournier-Viger, Chemical reaction optimization with unified tabu search for the vehicle routing problem, *Soft Comput.*, **21** (2017), 6421–6433.
20. A. Lam, J. Xu, V. Li, Chemical reaction optimization for population transition in peer-to-peer live streaming, In *IEEE Congress Evol. Comput.*, 2010.
21. J. Xu, A. Lam, V. Li, Chemical reaction optimization for the grid scheduling problem, *IEEE Int. Conf. Commun.*, 2010, 1–5.
22. T. K. Truong, K. Li, Y. Xu, Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem, *Appl. Soft Comput.*, **13** (2013), 1774–1780.
23. K. Güney, A. Akdag, Null steering of linear antenna arrays using a modified tabu search algorithm—abstract, *J. Electromagn. Waves Appl.*, **15** (2001), 915–916.
24. H. R. Tizhoosh, Opposition-based reinforcement learning, *J. Adv. Comput. Intell. Intell. Inform.*, **10** (2006), 578–585.
25. W. Hui, A. Zw, C. Sr, L. D. Yong, E. Mv, Enhancing particle swarm optimization using generalized opposition-based learning, *Inf. Sci.*, **181** (2011), 4699–4714.
26. R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.*, **11** (1997), 341–359.
27. J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, *IEEE Trans. Evol. Comput.*, **10** (2006), 646–657.

28. X. Li, X. Zhao, J. N. Wang, M. Yin, Improved artificial bee colony for design of a reconfigurable antenna array with discrete phase shifters, *Prog. Electromagn. Res.*, **25** (2012), 193–208.
29. S. Baskar, A. Alphones, P. N. Suganthan, Genetic algorithm based design of a reconfigurable antenna array with discrete phase shifter, *Microw. Opt. Technol. Lett.*, **45** (2005), 461–465.
30. J. A. Rodriguez, F. Ares, E. Moreno, Linear array pattern synthesis optimizing array element excitations using the simulated annealing technique, *Microw. Opt. Technol. Lett.*, **23** (1999), 224–226.
31. R. C. Hansen, *Phased array antennas*, volume 213. John Wiley & Sons, 2009.
32. X. Li, M. Yin, Optimal synthesis of linear antenna array with composite differential evolution algorithm - sciencedirect, *Sci. Iran.*, **19** (2012), 1780–1787.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)