EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Nonuniform antenna array design by parallelizing three-parent crossover genetic algorithm

Xin Zhang[1,2], Qinglian Zhang[1,2] and Xiu Zhang[1,2*]

Abstract

Antenna plays a very important role in wireless communication. Array elements laid out nonuniformly could achieve better frequency reliability and lower sidelobe level than uniform spaced elements. Designing desirable nonuniform antenna array requires the tuning of distances between each element, excitation, amplitude, and so on. Such design problem can be solved by genetic algorithm. Based on a recently genetic algorithm modification, this paper attempts to invent a parallel framework to enhance the efficiency of genetic algorithm. The parallel framework increases exploitation search of genetic algorithm. The effectiveness of the proposed algorithm is firstly demonstrated on toy problems with different kinds of problem complexity. The proposed algorithm is then verified on nonuniform antenna array design. Compared with genetic algorithm without parallelization, the novel algorithm attains better design solution and saves a lot computation time.

Keywords: Nonuniform antenna array, Genetic algorithm, Parallel computing, Antenna design, Numerical optimization

1 Introduction

In radar sensor networks, antenna arrays are quite often used to increase the diversity gain and reduce the probability of missed detection and false alarm [1, 2]. How to deploy the antenna arrays is a science and art. Antenna arrays could be deployed in uniform or nonuniform to achieve the best performance in target detection [3–5].

Since Hertz and Marconi invented the first antenna, it has becoming more and more important in social life, and now, it is indeed an indispensable part of our daily life [3]. Existing in a three-dimensional world that composed of the scope of beam, three-dimensional radian, square angle, and solid angle, antenna can be a transducer or a transverter between the guided wave and free space wave, which can be described by basic electrical paraments such as input impedance, radiation resistance, gain, efficiency, directional diagram, polarization mode, and beam width. It can be seen that antenna is nowadays a crucial factor in wireless communication, though design-efficient and reliable antenna is a difficult problem as such design is a simulation-based time-consuming task [6]. Parallel execution of antenna design is able to greatly save simulation time [7].

Genetic algorithm (GA) was proposed by J. Holland in the 1970s. It is characteristic of imitating biological evolution mechanism in nature. The paradigm of GA is based on Darwin's evolution theory and Mendel's heredity theory. It could automatically acquire and accumulate the acknowledgement of the search space in the exploring process and reasonably control the search process toward optimal solution [8]. GA generally begins with an initial population, according to the fitness function, to evaluate the fitness of all individuals, and then, it chooses excellent individuals, intersects, and mutates to introduce new individuals to population. Finally, GA is able to make its population to evolve until met the given accuracy or reached the largest genetic algebra [9].

Since the 1990s, GA has become popular in solving both combinational problems and continuous optimization problems [10, 11]. Wind farm micro-siting is the decision problem for determining the optimal placement

*Correspondence: zhang210@126.com
[1] Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin, China
[2] College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin, China

© The Author(s). 2017 **Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

of wind turbines in consideration of the wake effect. An enhanced genetic algorithm (EGA), which is customized to the properties of wind farm dimensions, was proposed to solve such problem. The experimental results show that the EGA can obtain the decision both effectively and efficiently as compared to other metaheuristic approaches [12]. A better match to the field data was reached with the optimization parameters set with the genetic algorithm. In order to check to what extent the model replicated reality, model validation was also addressed. Results showed that a genetic algorithm is usefully applicable in the calibration process of the microscopic traffic simulation model [13]. Parallel GA strategy has been applied to optimize reservoir operation [14]. Alansi et al. implement GA in field-programmable gate array for space division multiple access techniques [15].

Besides GA, other evolutionary algorithms have been applied in real-world problems [16–19]. Min et al. considered pulse compression and sidelobe suppression in [20]. Yan et al. studied parallel algorithm in cognitive radar network [21]. Dhaliwal and Pattnaik compared several evolutionary algorithms on the design of fractal antenna [22]. Zhang et al. applied artificial bee colony algorithm to solve antenna design [23].

As a famous evolutionary algorithm, GA shows good performance in dealing with antenna design problem. This paper attempts to propose a parallel framework based on a three-parent crossover GA. This framework plans to enhance exploitation of GA so that solutions could be refined to a better extent than original algorithm in the later evolutionary stage. The goodness of the proposed framework is studied on both toy problems and nonuniform antenna design problem.

In the following, Section 2 introduces the design of nonuniform antenna array and related works. Section 3 describes three-parent crossover GA and the proposed parallel framework. Section 4 reports numerical simulation results and discussions. Conclusion is made in Section 5.

2 Nonuniform antenna design and related works

Relatively, though in the early 1960s nonuniform array has begun to be studied, its synthesis remains to be solved [24]. Usually, the synthesis is considered in a given number of arrays and array response, determining the array element position and incentive distribution. Among them, under a given number of elements and a given shape of antenna, the most important subject is how to design array element space and incentive of phase distribution reasonably so as to make lowest the array's peak sidelobe level. On the one hand, array element response is the complex exponential function of its position; hence, the synthesis problem of array element is a nonlinear optimization problem; on the other hand, considering engineering application, the array element space must satisfy certain constraint (e.g., no less than a given value) to reduce the mutual coupling among elements. In addition, the ratio of the largest and the smallest array element incentive (i.e., current taper ratio (CRT)) need to be close to 1, which will make the antenna transmission power as large as possible.

Antenna array is comprised of a set of array elements sequentially arranged based on given discipline. Nonuniform linear antenna array is shown in Fig. 1. We consider thinned array case in which elements are not required to be placed in grid pattern. In thinned array, elements would be arranged anywhere given the distance between adjacent elements was no less than a predefined value for reducing cross-coupling of adjacent elements.

The model of nonuniform linear antenna array can be expressed as:

$$\begin{array}{ll} \min\limits_{d_i,(2\le i\le N-1)} & f\left(d_2, d_3, \ldots, d_{N-1}\right) \\ s.t. & d_i - d_j \ge d_c, 1 \le j < i \le N \\ & d_c > 0 \end{array}, \tag{1}$$

where N is the number of array elements, d_i denotes the position of the ith element, L is the array aperture with $d_1 = 0$ and $d_N = L$. Function f is the peak sidelobe level (PSLL), and the objective is to minimize PSLL.

With the assumption that all elements have equal amplitude and same direction, far directional pattern could be expressed as:

$$E(u) = \sum_{i=1}^{N} e^{jkud_i}, \tag{2}$$

where $k = 2\pi/\lambda, u = \cos\theta - \cos\theta_0$ with θ is the sweeping angle from the direction of array, and θ_0 is the position targeted by the main beam.

The binary particle swarm optimization in conjunction with the time-domain discrete Green's function method was proposed for designing of printed ultra wideband (UWB) antennas [4]. A topology optimization method based on the method of moments for configuration design of planar metallic antenna was proposed in [25]. Bhatia et al. proposed a novel design of modified printed circular monopole antenna with defective ground structure for wideband applications. The basic design of antenna

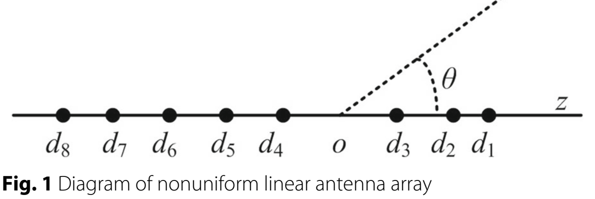

Fig. 1 Diagram of nonuniform linear antenna array

was comprised of a partial rectangular ground plane. The antenna design was further modified by employing a triangular notch in the ground plane to enhance the impedance bandwidth [26].

3 The used algorithms

This section describes three-parent crossover GA and our parallel framework.

3.1 Three-parent crossover genetic algorithm

Three-parent crossover GA (GA-TPC) was proposed by Elsayed and his colleagues [27]. A diversity operator is used in this algorithm to keep good population diversity. The algorithm attains good performance in competition problems on real parameter optimization. By taking this version, it probably outperforms old GA versions in dealing with antenna design problems.

The structure of GA-TPC consists of four steps: initial population, tournament selection, three-parent crossover, and diversity operator. The later three steps constitute main generation of this algorithm.

Initial population. As most evolutionary algorithms, initial population of GA-TPC is randomly created based on uniform distribution in problem's feasible search space as shown below:

$$\mathbf{x}_i = \mathbf{x}^{\min} + \mathbf{r}_i \left(\mathbf{x}^{\max} - \mathbf{x}_{\min}\right), \tag{3}$$

where $\mathbf{x}_i$ is the ith solution ($1 \leq i \leq N_p$ where N_p is the population size), $\mathbf{r}_i$ denotes a vector of real numbers randomly generated between 0 and 1, and $\mathbf{x}^{\max}$ and $\mathbf{x}^{\min}$ are the boundary vectors of search space. For most practical problems, feasible ranges of decision variables are easy to set; in case feasible initial solutions are hard to create, feasibility detection should be added to the algorithm.

Tournament selection. Parent selection is realized by tournament selection method with tournament size $tc = 2$ and with elitism 1, which means the best individual always keeps in the selection pool.

Three-parent crossover is realized by the following four steps. First, pick three individuals from the selection pool obtained in tournament selection step. Note that this selection is done without any duplication. Then, rank the chosen three individuals from the best to the worst based on their fitness. Third, based on normal distribution with zero mean and standard deviation σ ($N(0, \sigma)$), a random number is generated. Fourth, three candidate offsprings are produced below:

$$\mathbf{v}_1 = \mathbf{x}_1 + r_1 \left(\mathbf{x}_2 - \mathbf{x}_3\right) \;, \tag{4}$$

$$\mathbf{v}_2 = \mathbf{x}_2 + r_2 \left(\mathbf{x}_3 - \mathbf{x}_1\right) \;, \tag{5}$$

$$\mathbf{v}_3 = \mathbf{x}_3 + r_3 \left(\mathbf{x}_1 - \mathbf{x}_2\right) \;, \tag{6}$$

where $\mathbf{v}_i$ are candidate offsprings, r_i ($i = 1, 2, 3$) are generated according to ($N(0, \sigma)$).

Diversity operator. Instead of mutation operator, diversity operator is used with a diversity probability to diversify offsprings. Which parent undergoes this operator and the extent of diversity are two issues faced by this operator. For the former issue, an archive pool of size m is chosen from the parent population depending on fitness. For the later issue, three probabilities are respectively set for three children. Diversity operator for three children $\mathbf{v}_i, i = 1, 2, 3$ is given below:

$$u_{i,j} = \begin{cases} x_j^{\text{arc}} & \text{if } r_{i,j} < p_i \\ v_{i,j} & \text{otherwise} \end{cases} , j = 1, 2, \ldots, D \,, \tag{7}$$

where $\mathbf{u}_i, i = 1, 2, 3$ are offsprings after current generation. Survivor selection of GA-TPC is alike to other GA versions. A new population is build by choosing the best N_p individuals from offsprings and archive pool.

3.2 The proposed parallel framework

To run an algorithm in parallel, either multiple central processing units (CPUs) or multiple cores of a central processing unit (CPU) can be used [28]. As nearly all personal computer is a single CPU but multiple-core environment, we decide to design a parallel framework running in multiple cores as shown in Fig. 2. Preliminary experiment shows that the ratio of running time decrease; comparing original GA-TPC with parallel GA-TPC is the greatest when the number of cores is 2. Hence, Fig. 2 shows an example carrying out two cores. That is, two threads are sponsored at the same time to execute two GA-TPC algorithms. These two GA-TPC algorithms have the same procedures and the same algorithmic parameters.

Specifically, GA-TPC is split to two stages as in Fig. 1. One is a coarse exploration search, while the other is a fine exploitation search. In coarse search stage, algorithmic parameters are set making the algorithm performing more search in whole search space. In fine search stage, a good initialization is created as follows:

$$\mathbf{x}_i = \mathbf{x}_{\text{best}} + \mathbf{r}_i \left(\mathbf{x}_p^{\max} - \mathbf{x}_p^{\min}\right) \;, \tag{8}$$

where $\mathbf{x}_{\text{best}}$ is the best so far individual produced in coarse search stage, $\mathbf{r}_i$ ($i \in [1, N_p]$) are random vectors generated according to ($N(0, 1)$), $\mathbf{x}_p^{\max}$ and $\mathbf{x}_p^{\min}$ are respectively the vector of maximal and minimal values of all dimensions in the final population of coarse search stage. The physical meaning of this operation is to create a population of individuals normally distributed around the best so far individual. The standard deviation of normal distribution relies on the range of decision variables obtained in the first stage. Besides good initialization, algorithmic parameters are set causing the algorithm performing more exploitative search in the fine search stage.

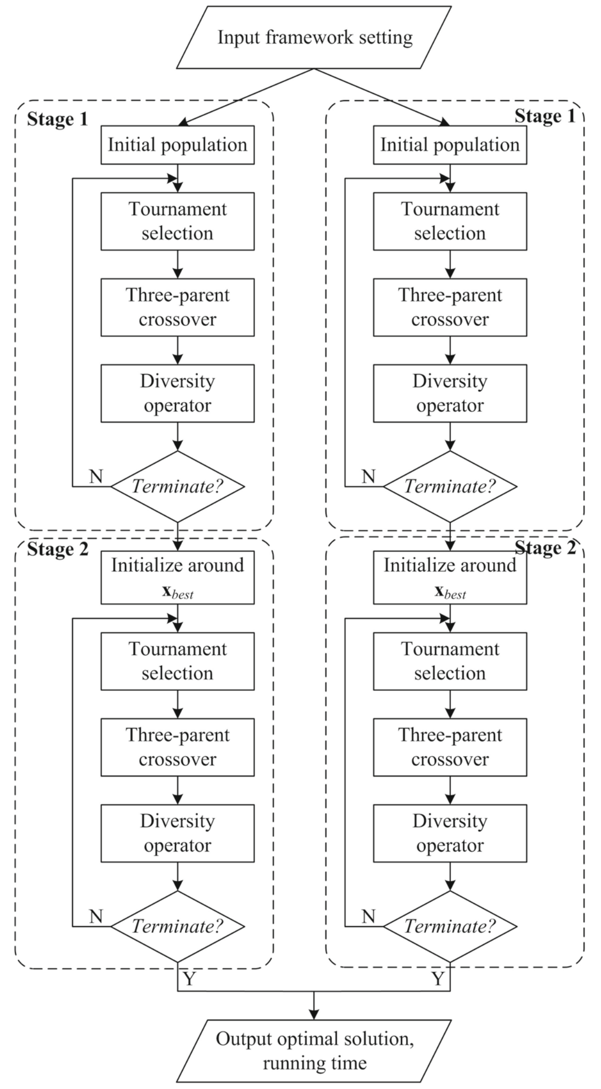

Fig. 2 The proposed parallel framework for GA-TPC algorithm

In terms of algorithmic parameter, large N_p assigns high explorative ability of GA, whereas small N_p causes GA more exploitative ability. Moreover, GA with large N_p converges slower than it with small N_p. The convergence rate of GA is also related with crossover rate, though three-parent crossover relies on normal distribution and does not have crossover rate parameter. As to probabilities in diversity operator, p_i $(i = 1, 2, 3)$ are set to 0.01, 0.1, and 0.1, respectively. They are fixed based on our experience and the author's suggestion.

4 Numerical simulation

In this section, the proposed parallel framework is studied and compared with GA without parallel.

4.1 Experimental setting

Denote parallel GA-TPC as the algorithm instantiating GA-TPC in the proposed parallel framework. A state-of-the-art related parallel method is taken for experiment comparison [14], which improves non-dominated sorting genetic algorithm by incorporating multiple recombination operators. This method is denoted as parallel GA-MR. The simulation configuration is as follows. Algorithmic parameters of GA-TPC, parallel GA-TPC, and parallel GA-MR are given in Table 1. In this table, for GA-TPC, N_p and σ are set the same as in [27]. For parallel GA-TPC, N_p^1 and σ^1 are for parameters used in stage 1, while N_p^2 and σ^2 are for stage 2. All N_p are multiple of 3 as three offsprings are generated by three-parent crossover. σ^2 is greater than σ^1 to keep good diversity in stage 2. A set of 10 toy functions are taken as the first study. Functions as well as their properties are listed in Table 2. This table shows 5 unimodal functions and 5 multimodal functions used in simulation. They are commonly used in preliminary study of the performance of an algorithm [29]. For toy functions, problem dimension D is set to 10. The second study is done on nonuniform antenna array design. As described in Section 2, the parameters of antenna model is $N = 17$, $L = 9.744\lambda, d_c \geq 0.5\lambda$ and $\theta_0 = \pi/2$. As d_1 and d_N are start and end points of antenna, problem dimension D is $N - 2 = 15$.

In the study of toy functions and antenna design, the maximal number of objective function evaluations (MFE) is fixed at 5000D. For each test case, an algorithm is independently run 25 times to reach an average performance of the algorithm. For parallel GA-TPC, computational resources are equally split to stages 1 and 2. That is MFE = 2500D for stage 1 and MFE = 2500D for stage 2. It can be seen from the parallel framework that communication between multiple cores happens when creating initial population at stage 2. Because computational resource is equally split and the same algorithm structure is used, it is expected that multiple cores would finish running nearly the same time so that wasteful waiting of the synchronization of multiple cores is reduced to minimum. Thus, good parallel efficiency could be reached in the proposed parallel framework. Both algorithms and problems are programmed in Matlab. The simulation is performed on a personal computer with four-core 3.40 GHz CPU and 4 GB random access memory.

Table 1 Configuration of GA-TPC, parallel GA-TPC, and parallel GA-MR algorithms

Algorithm	Parameters
GA-TPC	$N_p = 90, \sigma = 0.5$
Parallel GA-TPC	$N_p^1 = 60, \sigma^1 = 0.5, N_p^2 = 30, \sigma^2 = 1.0$
Parallel GA-MR	$N_p = 320, N_g = 8, P_g = 80, R_m = 0.5, P_c = 1000$

Table 2 Toy functions: 5 unimodal functions and 5 multimodal functions

Number	Function	Property
f_1	Spherical function	Unimodal, separable, scalable
f_2	Axis parallel hyper-ellipsoid, weighted sphere	Unimodal, separable, scalable
f_3	Perm function	Unimodal, nonseparable, scalable
f_4	High-conditioned elliptic function	Unimodal, separable, scalable
f_5	Schwefel's problem 1.2	Unimodal, nonseparable, scalable
f_6	Rosenbrock's function	Multimodal, nonseparable, scalable
f_7	Rastrigin's function	Multimodal, separable, scalable
f_8	Griewank function	Multimodal, nonseparable, scalable
f_9	Ackley's function	Multimodal, nonseparable, scalable
f_{10}	Zakharov function	Multimodal, nonseparable, scalable

4.2 Results on toy functions

Table 3 shows the statistics of optimal objective function values attained by GA-TPC and parallel GA-TPC. This table also contains hypothesis test of the median function values over 25 runs. It can be seen from the table that both algorithms find global optimum of f_3 and f_9. Parallel GA-TPC reaches global optimum on f_1,f_2, and f_4, while GA-TPC could not attain satisfiable accuracy on the three functions. For f_5 and f_7, parallel GA-TPC could not outperform GA-TPC, though the difference is not significant at level $\alpha = 0.05$. For f_3 and f_9, both algorithms reach global optimum. For the other 6 functions, Parallel GA-TPC significantly outperforms GA-TPC at level $\alpha = 0.05$. Thus, it is able to conclude that the proposed parallel framework improves the effectiveness of GA-TPC.

Figure 3 shows the running time comparison of GA-TPC and Parallel GA-TPC algorithms on toy functions. In this figure, "x" axis denotes the 10 functions, "y" axis is the time ratio of GA-TPC over Parallel GA-TPC. Time ratio is shown in dotted line. The solid line is ratio 1 designating algorithms costing same computational time. It can be seen from Fig. 3 that time ratio curve from f_1 to f_{10} is above 1.8. This means that computational time is saved about 80% when using the proposed parallel framework. Note that the ratio of f_1 is above 2, this is because this function is executed first and Matlab needs some time to warm up. For other functions, the function call in Matlab is much faster than f_1.

Table 4 shows the statistics of the optimal results attained by parallel GA-MR and parallel GA-TPC. This table also contains statistical Mann-Whitney U test (U test) of the median best function values over 25 runs. It can be seen from the table that both algorithms find global optimum of f_1, f_3, and f_9. Parallel GA-TPC reaches global optimum on f_2 and f_4, while parallel GA-MR could not attain satisfiable accuracy on both functions. For f_7, parallel GA-TPC could not outperform parallel GA-MR, though the difference is not significant at level $\alpha = 0.05$. For f_8 and f_{10}, parallel GA-TPC significantly outperforms parallel GA-MR at level $\alpha = 0.05$. Thus, it is able to conclude that the proposed parallel framework compares favorable with the state-of-the-art parallel modifications of GA.

Table 3 Optimal function values found by GA-TPC and parallel GA-TPC over 25 independent runs

Algorithm	GA-TPC		Parallel GA-TPC		U test
$f(\cdot)$	med	std	med	std	p-value
f_1	6.56E−04	2.64E+00	0.00	0.00	4.58E−08
f_2	6.07E−08	6.38E−03	0.00	0.00	1.10E−06
f_3	0.00	0.00	0.00	0.00	N/A[a]
f_4	6.17E−08	5.19E−04	0.00	0.00	1.90E−05
f_5	1.16E+12	9.85E+13	1.49E+12	1.68E+13	0.4377
f_6	1.04E+01	1.16E+02	6.43E+00	1.88E+01	0.0052
f_7	3.98E+00	1.68E+00	4.97E+00	3.22E+00	0.0708
f_8	1.28E−01	1.30E−01	9.07E−02	5.13E−02	0.0478
f_9	0.00	0.00	0.00	0.00	N/A[a]
f_{10}	4.39E−02	7.88E−01	0.00E+00	2.31E−01	6.34E−08

[a] N/A: hypothesis test is not available as algorithms have the same median values
Median and standard deviation are respectively abbreviated as med and std

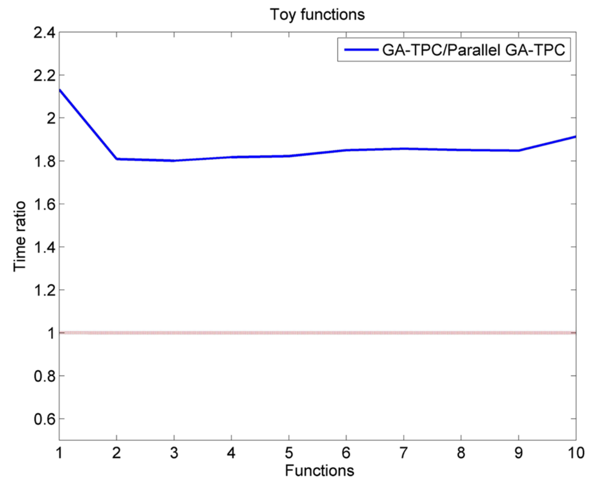

Fig. 3 Running time comparison of GA-TPC and Parallel GA-TPC

Table 4 Results found by parallel GA-MR and parallel GA-TPC over 25 independent runs

Algorithm	Parallel GA-MR		Parallel GA-TPC		*U* test
$f(\cdot)$	med	std	med	std	*p* value
f_1	0.00	0.00	0.00	0.00	N/A[a]
f_2	4.92E−05	1.43E−07	0.00	0.00	8.66E−05
f_3	0.00	0.00	0.00	0.00	N/A[a]
f_4	5.60E−08	8.93E−05	0.00	0.00	5.25E−05
f_5	9.90E+14	9.54E+13	1.49E+12	1.68E+13	0.3077
f_6	7.29E+00	1.96E+01	6.43E+00	1.88E+01	0.2072
f_7	4.05E+00	2.86E+00	4.97E+00	3.22E+00	0.0644
f_8	2.08E−01	1.36E−01	9.07E−02	5.13E−02	0.0130
f_9	0.00	0.00	0.00	0.00	N/A[a]
f_{10}	2.02E−06	5.47E−01	0.00E+00	2.31E−01	0.0379

[a] N/A: hypothesis test is not available as algorithms have the same median values
Median and standard deviation are respectively abbreviated as med and std

Table 5 Results found by GA-TPC and parallel GA-TPC over 25 independent runs for antenna design

Algorithm	GA-TPC			Parallel GA-TPC		
Metric	min	med	max	min	med	max
f	−20.88	−20.57	−19.43	−21.36	−21.02	−20.78
d_1	0	0	0	0	0	0
d_2	0.72	0.67	1.65	0.99	0.77	0.78
d_3	1.56	1.52	2.31	1.74	1.55	1.56
d_4	2.21	2.22	2.92	2.59	2.26	2.24
d_5	2.73	2.81	3.47	3.21	2.75	2.75
d_6	3.31	3.39	4.03	3.71	3.28	3.29
d_7	3.77	3.82	4.51	4.30	3.76	3.81
d_8	4.23	4.34	5.03	4.73	4.30	4.37
d_9	4.73	4.89	5.44	5.22	4.81	4.69
d_{10}	5.25	5.31	5.97	5.61	5.24	5.24
d_{11}	5.62	5.77	6.45	6.11	5.65	5.75
d_{12}	6.25	6.30	6.98	6.57	6.25	6.28
d_{13}	6.75	6.92	7.47	7.09	6.74	6.75
d_{14}	7.33	7.34	7.97	7.49	7.31	7.31
d_{15}	8.07	8.10	8.52	8.16	8.09	8.14
d_{16}	8.89	8.87	9.03	9.02	8.84	8.88
d_{17}	9.74	9.74	9.74	9.74	9.74	9.74

Minimum, median, and maximum are respectively abbreviated as min, med, and max

4.3 Results on antenna design

Both GA-TPC and parallel GA-TPC are applied to deal with nonuniform linear antenna array design. The results are shown in Table 5. This table gives the min, med, and max objective function values found by algorithms as well as the associated solutions. Clearly, parallel GA-TPC attains better results than GA-TPC on the three metrics. Hence, the proposed parallel framework is effective on antenna design. Moreover, it is observed from Table 5 that for the min and med solutions, distance $d_2 - d_1$ of parallel GA-TPC is longer than that of GA-TPC; on the other hand, $d_2 - d_1$ of the max solution of parallel GA-TPC is shorter than that of GA-TPC. This indicates that antenna array design problem has a multimodal landscape and parallel GA-TPC is able to reach a better solution, while GA-TPC searches around margin of promising solution but could not converge to the promising solution.

By conducting U test on 25 trials on this problem, parallel GA-TPC significantly outperforms GA-TPC with significance level $\alpha = 0.05$. In terms of running time, the design simulation of GA-TPC costs 5295.2 (s) over 25 trials, while the design simulation of parallel GA-TPC takes 2891.5 (s). The time ratio on this problem is 1.83, which is alike to the ratio on toy functions. Thus, it is reasonable to conclude that the proposed parallel framework can effectively and efficiently improve the performance of GA-TPC algorithm.

5 Conclusions

Recently, many researchers or users apply genetic algorithm (GA) to design antennas. As antenna design is a difficult and a simulation-based time-consuming task, GA is a useful tool to deal with such problem. This paper focuses on improving the efficiency of GA based on the idea of parallel computing. Different existing approaches, our research is based on a state-of-the-art version of GA, denoted as GA-TPC. Moreover, in the proposed parallel framework, algorithm evolution is equally split to two stages. In stage 1, algorithm search is guided toward exploration through the use of large population size and relatively large standard deviation (std) in normal distribution. In stage 2, algorithm search is guided toward exploitation through using small population size and large std. Moreover, initial individuals of stage 2 are created around the best individual returned in stage 1. The distribution of individuals in stage 2 also relies on the variance of individuals in the last population of stage 1.

The proposed framework is instantiated on GA-TPC, denoted as parallel GA-TPC. Tested on both toy problems and antenna design problem, parallel GA-TPC finds better solutions than GA-TPC and a state-of-the-art parallel modification of GA (i.e., parallel GA-MR). Moreover, parallel GA-TPC reaches smaller std of function values of final solutions than GA-TPC and parallel GA-MR. This means that the proposed framework is effective and stable. Furthermore, computational time of parallel GA-TPC is saved about 80% compared with the time of GA-TPC under the same simulation environment.

The proposed framework shows good performance in antenna design, though it has an overhead that the extension of the framework to more cores is not easy. The difficulty of multiple cores extension attributes to algorithmic parameter setting. Dividing algorithm evolution to two stages is intuitively feasible as GA has exploration and exploitation. Using more cores means to split evolution to more stages which has to be meaningful and effective. This direction will be studied in the future.

Acknowledgements
This research was supported in part by the National Science Foundation of China (Project No. 61603275, 61601329) and the Doctoral Fund Project of Tianjin Normal University (Project No. 043-135202XB1602).

Authors' contributions
XZ proposes the parallel framework based on GA-TPC and writes most of this paper except that XZ writes the numerical simulation section. QZ is in charge of the computer simulation of the algorithm and the collection of the results. XZ organizes the whole paper as well as the proofreading. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 3 April 2017 Accepted: 29 May 2017
Published online: 12 June 2017

References

1. Q Liang, X Cheng, S Samn, NEW: network-enabled electronic warfare for target recognition. IEEE Trans. Aerosp. Electron. Syst. **46**(2), 558–568 (2010)
2. Q Liang, X Cheng, S Huang, D Chen, Opportunistic sensing in wireless sensor networks: theory and applications. IEEE Trans. Comput. **63**(8), 2002–2010 (2014)
3. MH Bataineh, JI Ababneh, Synthesis of aperiodic linear phased antenna arrays using particle swarm optimization. Electromagnetics. **26**(26), 531–541 (2006)
4. S Mirhadi, N Komjani, M Soleimani, Ultra wideband antenna design using discrete green's functions in conjunction with binary particle swarm optimisation. IET Microwaves Antennas Propag. **10**(2), 184–192 (2016)
5. Q Liang, Situation understanding based on heterogeneous sensor networks and human-inspired favor weak fuzzy logic system. IEEE Syst. J. **5**(2), 156–163 (2011)
6. V Kyovtorov, I Georgiev, S Margenov, D Stoychev, F Oliveri, D Tarchi, New antenna design approach—3d polymer printing and metallization. experimental test at 14-18 GHz. AEU - Int. J. Electron. Commun. **73**, 119–128 (2017)
7. C Han, T Feng, G He, T Guo, Parallel variable distribution algorithm for constrained optimization with nonmonotone technique. J Appl. Math. **2013**(1), 401–420 (2013)
8. SY Yuen, CK Chow, A genetic algorithm that adaptively mutates and never revisits. IEEE Trans. Evol. Comput. **13**(2), 454–472 (2009)
9. CB Kalayci, O Polat, SM Gupta, A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. Ann. Oper. Res. **242**(2), 321–354 (2016)
10. MA Salido, J Escamilla, A Giret, F Barber, A genetic algorithm for energy-efficiency in job-shop scheduling. Int. J. Adv. Manuf. Technol. **85**(5), 1303–1314 (2016)
11. J Yu, M Li, Y Wang, G He, A decomposition method for large-scale box constrained optimization. Appl. Math. Comput. **231**(12), 9–15 (2014)
12. P-Y Yin, T-H Wu, P-Y Hsu, Risk management of wind farm micro-siting using an enhanced genetic algorithm with simulation optimization. Renew. Energy. **107**, 508–521 (2017)
13. S Chiappone, O Giuffre, A Grana, R Mauro, A Sferlazza, Traffic simulation models calibration using speed–density relationship: an automated procedure based on genetic algorithm. Expert Syst. Appl. **44**, 147–155 (2016)
14. D Chen, Q Chen, AS Leon, R Li, A genetic algorithm parallel strategy for optimizing the operation of reservoir with multiple eco-environmental objectives. Water Resour. Manag. **30**(7), 2127–2142 (2016)
15. M Alansi, I Elshafiey, A Al-Sanie, A Mabrouk, FPGA implementation of multi-user detection genetic algorithm tool for SDMA-OFDM systems. Wirel. Pers. Commun. **86**(3), 1241–1263 (2016)
16. S Pang, T Li, F Dai, M Yu, Particle swarm optimization algorithm for multisalesman problem with time and capacity constraints. Appl. Math. Inf. Sci. **7**(6), 2439–2444 (2013)
17. Z Wu, X Xia, Optimal switching renewable energy system for demand side management. Sol. Energy. **114**, 278–288 (2015)
18. W Li, X Shi, Y Hei, An improved particle swarm optimization algorithm for pattern synthesis of phased arrays. Prog. Electromagn. Res. **82**, 319–332 (2008)
19. B Wang, Z Wu, X Xia, A multistate-based control system approach toward optimal maintenance planning. IEEE Trans. Control. Syst. Technol. **25**(1), 374–381 (2017)
20. HK Min, MS Song, I Song, JH Lim, A frequency-sharing weather radar network system using pulse compression and sidelobe suppression. EURASIP J. Wirel. Commun. Netw. **2016**, 100 (2016)
21. B Yan, S Wang, Y Bao, Research on an improved parallel algorithm based on user's requirement in cognitive radar network. EURASIP J. Wirel. Commun. Netw. **2015**, 47 (2015)
22. BS Dhaliwal, SS Pattnaik, Performance comparison of bio-inspired optimization algorithms for sierpinski gasket fractal antenna design. Neural. Comput. Applic. **27**(3), 585–592 (2016)
23. X Zhang, X Zhang, A novel artificial bee colony algorithm for radar polyphase code and antenna array designs. EURASIP J. Wirel. Commun. Netw. **2016**(1), 40 (2016)
24. Q Liang, X Cheng, KUPS: Knowledge-based ubiquitous and persistent sensor networks for threat assessment. IEEE Trans. Aerosp. Electron. Syst. **44**(3), 1060–1069 (2008)
25. S Liu, Q Wang, R Gao, MoM-based topology optimization method for planar metallic antenna design. Acta Mech. Sinica. **32**(6), 1058–1064 (2016)
26. SS Bhatia, JS Sivia, A novel design of circular monopole antenna for wireless applications. Wirel. Pers. Commun. **91**(3), 1153–1161 (2016)
27. SM Elsayed, RA Sarker, DL Essam, in *IEEE Congress on Evolutionary Computation*. A genetic algorithm for solving the CEC2013 competition problems on real-parameter optimization (IEEE, Cancun, 2013), pp. 356–360
28. CY Han, FY Zheng, TD Guo, GP He, Parallel algorithms for large-scale linearly constrained minimization problem. Acta Math. Applicatae Sinica Eng. Ser. **30**(3), 707–720 (2014)
29. Z Wu, TWS Chow, Neighborhood field for cooperative optimization. Soft Comput. **17**(17), 819–834 (2013)

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com