Int J Adv Manuf Technol (2017) 91:987–997
DOI 10.1007/s00170-016-9822-7

ORIGINAL ARTICLE

Assembly sequence planning for reflector panels based on genetic algorithm and ant Colony optimization

Dou Wang[1] **· Xiaodong Shao**[1] **· Simeng Liu**[1]

Received: 25 July 2016 / Accepted: 21 November 2016 / Published online: 30 November 2016

Abstract Assembly sequence planning (ASP) can significantly improve assembly accuracy and reduce assembly costs in modern manufacturing industries. Large reflector antennas are difficult to assemble and urgently need ASP. Based on genetic algorithms (GAs) and ant colony optimization (ACO), an approach for ASP of reflector antennas was developed. An accurate simulation of the assembly of the reflectors was required for the evaluation and optimization of the ASP. The initial population was created by ACO and optimized by GA operators to generate an optimal solution. By releasing the information on the optimal solution to the ant search paths of ACO, convergence to a globally optimal solution was accelerated. A model of the finite element simulation was used to simulate the dynamic assembly process of reflectors according to the algorithm results of the proposed approach (GAACO). The proposed approach was tested and compared to GA, and the results indicate that GAACO can improve the quality of the optimal solution, increase the searching efficiency, and reduce the probability of finding a local optimal solution.

Keywords Reflector panel assembly · Assembly sequence planning · Genetic algorithm · Ant colony optimization

✉ Xiaodong Shao
13991138009@163.com

[1] Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, No. 2 South Taibai Road, Xi'an, Shaanxi 710071, China

1 Introduction

Assembly sequence planning (ASP) is the arrangement of assembly operations in the product manufacturing industry, where an optimal assembly sequence is selected from possible assembly sequences that have been automatically generated using assembly modeling and are based on some quantitative and qualitative criteria [1, 2]. Research has determined that assembly tasks account for 20–70% of the total production work [3], and in the manufacturing industry, the assembly of manufactured products accounts for more than 50% of the total production time and 30–50% of the labor costs, which makes the ASP problem as one of the basic problems in the assembly process [4]. The ASP problem determines many assembly aspects including the number of assembly orientation changes, number of assembly tool changes, and number of assembly operation type changes. The optimization results of the above three aspects can reduce the cost and time required for the assembly process [5]. Because of the importance of the assembly sequence, it has attracted interest from many researchers and engineers in recent decades.

The assembly precedence relation and assembly association figure model were first employed to solve the ASP problem in 1984 [6]. DeFazio and Whitney [7] optimized the method proposed in [6] by improving the form of questions and reducing the number of questions to ask the operators. Mello [8] presented an algorithm method to generate mechanical assembly sequences using a relational model of assemblies and disassemblies. Dong et al. [9] proposed a knowledge-based approach to the ASP problem, which is connection-semantics-based assembly considering both geometric information and non-geometric knowledge. Bai et al. [10] proposed an integration strategy for assembly sequence planning and sequence scheme evaluation and for predicting whether a collision will occur between the assembly tool and assembled components by considering factors such as target components and assembly resources.

Recently, intelligent algorithms have been rapidly developed, including the particle swarm optimization (PSO) algorithm, immune optimization approach (IOA), genetic algorithm (GA), and ant colony optimization (ACO) algorithm. These intelligent algorithms are also widely applied to ASP problems.

The PSO algorithm, which has powerful local and global search abilities, has been widely used in assembly sequence planning and related fields [11, 12]. Lv and Lu [13] proposed the assembly sequence planning approach with a discrete particle swarm algorithm (DPSO). The DPSO algorithm can achieve good results, but is easily affected by the individual optimal fitness value, and easily tends to convergence in the early evolution.

The artificial immune system (AIS) has emerged as a novel branch of computational intelligence [14] during the last decade. Cao and Xiao [15] explored the application of the artificial immune system (AIS) in the problem of assembly planning and proposed a novel approach, called the IOA, to generate the optimal assembly plan based on the bionic principles of AIS.

In addition to powerful random and global research abilities, less time consumption and high robustness, the GA can converge efficiently and is easily implemented. In view of the mentioned advantages, Bonneville et al. [16] first employed a genetic algorithm for ASP in 1995. After that, many contributions have been made in ASP topics using GA.

Kongar and Gupta [17] found an optimal (or near-optimal) disassembly sequence that is crucial to increasing the efficiency of the process based on GA, and a case example was considered to demonstrate the functionality of the algorithm.

Giudice and Fargione [18] proposed an approach to disassembly process planning that supports searching for the disassembly sequence best suited for both aspects, service of the product, and recovery at the end of its useful life, developing two different algorithms.

Seo et al. [19] presented a GA-based approach for an optimal disassembly sequence considering economic and environmental aspects, which is based on a disassembly tree, the disassembly precedence, and the disassembly value matrix to optimize a disassembly sequence.

The ant colony (AC) algorithm was first employed for ASP by Failli et al. [20] in 2000. Based on the assembly by disassembly philosophy, Wang [21] proposed an ant colony algorithm considering feasible and reasonable disassembly operations.

Yu [22] proposed an improved ACO-based ASP method for complex products that combines the advantages of the ant colony system (ACS) and max–min ant system (MMAS) and compared them with the methods of priority rule screening, genetic algorithm, and particle swarm optimization. MMACS is verified to be superior in efficiency and sequence performance.

Cong and Zhuo [23] proposed an approach for solving integrated assembly sequence planning and assembly line balancing with the ant colony algorithm, in which the assembly task priority diagram was built based on the assembly information, and the assembly sequences were evaluated at each work station of the assembly line separately.

Many researchers have also mixed two or more intelligent algorithms, including bacterial chemotaxis combined with the genetic algorithm [24], immune algorithm combined with the particle swarm optimization algorithm [25], and ant colony optimization combined with the genetic algorithm [26, 27] to obtain hybrid algorithms with better searching ability. Due to the large radius, hundreds of parts, and complicated structures, the assembly of a large reflector antenna has a great impact on its development cycle and reflector accuracy [28, 29]. For example, the development cycle—which includes the design, manufacture, assembly, and more—of an antenna with a 35-m radius is 15 weeks, and its assembly requires 8 weeks, where 2/3 of the 8 weeks are used to adjust the reflector panels to meet the surface accuracy requirements. These problems can be improved by ASP during the design stage to shorten the development cycle, reduce labor costs, and improve reflector accuracy.

When comparing the advantages and disadvantages of GA and ACO, it can be determined that these two intelligent algorithms (IAs) are complementary, which suggests that the combination of GA and ACO could produce a more efficient and superior IA for ASP problems. Based on this idea and the assembly process of a reflector antenna, an ASP method for a reflector antenna was developed by combining the finite element analysis method and GAACO.

There has been some research on the combination of GA and ACO [26, 27]. Kucukkoc and Zhang [26] proposed a novel hybrid-agent-based ant colony optimization–genetic algorithm approach for the solution of a mixed model parallel two-sided assembly line balancing and sequencing problem. Akpinar et al. [27] presented a new hybrid algorithm that executes ant colony optimization in combination with the genetic algorithm (ACO-GA) for a type I mixed-model-assembly line balancing problem with some particular features of real world problems such as parallel workstations, zoning constraints, and sequence-dependent setup times between tasks.

However, these combination algorithms of GA and ACO are unsuitable for the assembly optimization of the reflectors. In [26] and [27], ACO is the main algorithm, and GA is introduced into ACO; in addition, assembly line balancing is also the aim of these two ACO-GA algorithms; third, the assemblies of reflectors are much different from common products, which are more complex and difficult to control.

The basics of our proposed ASP method are as follows: First, we analyze the assembly process of the reflector and propose a finite element simulation method for it. Second, the parameters of the panels are encoded into chromosomes, considering the pre-tightening forces and the assembly sequences of panels. Third, the initial population is generated by the ACO algorithm, which can guarantee the quality and improve the diversity of this initial population to a great extent. Fourth, an optimized solution is obtained and optimized further by GA operations, which includes selection, crossover, and mutation. The pheromones according to the quality of the optimized solution are then released into the ant optimization paths, which are the feedback to guide the searching ants. This operator can obviously accelerate the accumulation of information about the optimal solution. Fifth, when the termination condition is met, the optimal solution is decoded and employed into a finite element simulation for the assembly process of the reflector. The optimal solution is obtained under the condition that the simulation results satisfy the accuracy requirement. Otherwise, the pheromones are updated, the information on each ant is cleared, and the process is repeated until an optimal solution that meets all conditions is obtained. The proposed ASP algorithm considers the finite element method, GA and ACO algorithm, which can improve the quality and searching efficiency to find the best assembly sequence and reduce the probability of finding a local optimal solution.

2 Finite element simulation method of reflector

Because a reflector is divided into several rings and consists of numerous panels, which are limited only by their manufacturing technology, the large reflector antenna is always assembled as follows: first, the back-up structure is assembled, and then the panels are assembled successively to compose the reflector. Each panel is placed in its appropriate position on the backup structure, and then the screws or rivets of the panel are tightened in a fixed sequence to fasten the panel.

To precisely simulate panel assembly and reduce the number of calculations, we propose that the "life–death element technique" be employed for finite element simulation. The "life–death element technique" can control the states of the parts during the assembly simulation by commands named "ekill" and "ealive". This method can overcome the multiple modeling, meshing times, and computationally high expense of the conventional method.

All components of the reflector are created first. All panels are then "killed" via the command "ekill", which means that the panels do not appear in the model. When one panel will be assembled, the panel can be "revived" via the command "ealive" to express the panel appearing in the model. "Killed" in this paper refers to multiplying a very small coefficient [ESTIF] to the stiffness matrix of a panel unit; the panel will disappear in the model, and all its parameters—including mass, damp, and specific heat—are set to zero. "Revived" means using the same method to make the panel appear in the model, and all its parameters are recovered.

This simulation method overcomes the requirements of the conventional simulation method including multiple modeling and meshing times and much algorithmic work, and it is computationally very expensive, which also improves the precision of the simulation of the reflector's assembly.

The simulation process is illustrated by an example of a 9-m reflector antenna shown in Fig. 1. The finite element geometric model of this case is shown in Fig. 2. The reflector is made up of three rings; each ring has 12 large-scale panels (Fig. 2), and each panel has four screws (Fig. 3) to fasten the panel onto the structure. This antenna is also based on the finite element method proposed above; the simulation model is created and the simulation is performed by ANSYS. The test of this simulation method is shown in Section 6.

3 Basic genetic algorithm

3.1 Chromosome code for assembly sequence

According to the features of the assembly process of the panels, each panel can be encoded into genomes of the GA that can be modeled with the following parameters:

$\text{Panel}_i = \{\text{Number}_i, \text{Ring}_i, \text{Gravity}_i, F_i, \text{Seq}_i\}$;

i ith assembly step

Number_i Number of panels

Ring_i Ring number of ith assembly step

Gravity_i Gravity of ith assembly step

F_i Pre-tightening force of screws

Seq_i Tightening sequence of screws

The fixture and fastening tools are not considered in the genomes because these tools will not change during the reflector assembly process. That means all panels will be fastened with the same fastening tools, and all panels will be fixed with the same fixtures.

In this section, a description of the GA formulation in this paper is provided. The example of the 9-m reflector antenna mentioned in Section 2 is used to illustrate this process.

There are several main encoding rules of GAs, including binary encoding rules, floating point number (decimal number) encoding rules, and delta encoding rules. We chose floating point number encoding rules because of their high precision and ease of expansion.

(1) i is the assembly step number, where $i \in \{1, 2, 3, \ldots\ldots, N\}$, which is different from the value of Number_i. N is equal to the number of sheet parts. We use an integer to denote the assembly step number, which can be described as the ith step. In this case, $i \in \{1, 2, 3, \ldots\ldots, 36\}$.

(2) Number_i is the sheet part number, where $\text{Number}_i \in \{1, 2, 3, \ldots\ldots, N\}$. N is equal to the number of sheet parts. In this case, $\text{Number}_i \in \{1, 2, 3, \ldots\ldots, 36\}$.

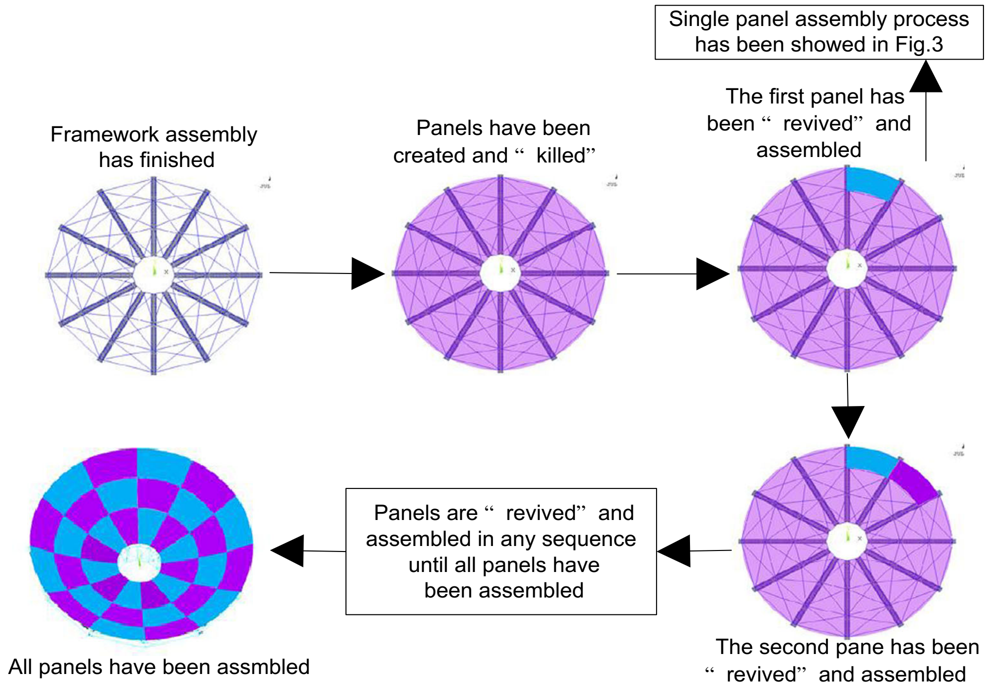


Fig. 1 The finite element simulation model of a reflector antenna

(3) Gravity$_i$ is the gravity number of the ith assembled sheet part, where Gravity$_i$=1 ,2 ,3.....G. In this case, Gravity$_i \in \{1,2,3\}$.

(4) F_i is the pre-tightening force of one screw or rivet of the ith assembled sheet part, where F_i=1, 2, 3, 4.....F. In this case, $F_i \in \{1,2,3,4\}$.

(5) Seq$_i$ is the tightening sequence number of one screw or rivet of the$i$th assembled sheet part, where Seq$_i$=1, 2, 3,......S. For example, in this case, Seq$_i$=1, 2, 3,....19 when the number of screws in one sheet part is four (Fig. 2, Table 1). Plan 1 (1–2–3–4) means tightening the four screws in the sequence of 1, 2, 3, and 4. Plan 13 (1, 2–3, 4) means tightening screws 1 and 2 simultaneously and then tightening 3 and 4 simultaneously. Plan 17 (1, 2, 3–4) means tightening 1, 2 and 3 at the same time and then subsequently tightening 4. Plan 19 (1, 2, 3, 4) means tightening the four screws at the same time.

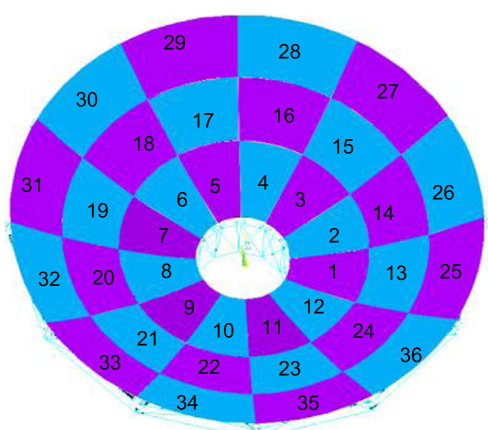


Fig. 2 The number of panels

3.2 Fitness function

A fitness function is used to evaluate the quality of the assembly sequence and can be expressed by considering the properties of certain assembly processes [18–21].

In the actual engineering, the accuracy is not the only factor considered; the cost to develop an antenna—labor cost and time cost—is the other main factor. Thus, the fitness function has considered the accuracy and the cost of the assembly of the reflector to establish a more accurate and comprehensive evaluation criterion. The fitness value of the individual of the algorithm influences the quality of the next generation. The accuracy of each optimal solution considered in the fitness function (obtained assembly sequence of every search) is fed back to the algorithm to reach the optimal solution quickly.

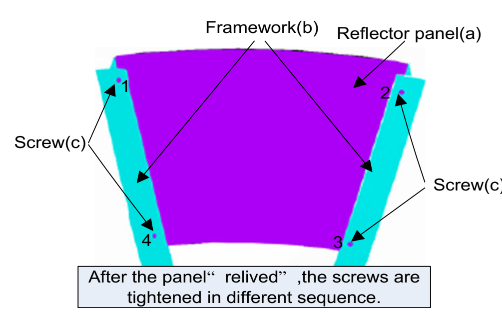


Fig. 3 The assembly process of a single panel

Table 1 Tightening sequences of screws

No.	1	2	3	4	5	6	7	8	9	10
Tightening sequence	1–2–3–4	1–3–4–2	1–2–4–3	1–3–2–4	1–4–3–2	1–4–2–3	3–4–1–2	3–1–2–4	3–1–4–2	3–1–4–2
No.	11	12	13	14	15	16	17	18	19	
Tightening sequence	3–2–1–4	3–2–4–1	1, 3–2, 4	1, 2–3, 4	1, 4–2, 3	3, 4–1, 2	1, 2, 3–4	1, 4, 3–2	1, 2, 3, 4	

This can make the fitness function smarter and more suitable for this case and revise the next generation searching solution of the algorithm. Therefore, in this study, considering the assembly process of this reflector, the fitness function is defined as follows:

$$\mathrm{Fitness}(P_i) = \sum_{i=1}^{N}\left(\frac{1}{g_i(x)}\right) + \sum_{i=1}^{N}\frac{1}{\left(f_{1ij}+f_{2i}\right)} \tag{1}$$

where N is the number of panels in the antenna, and $g_i(x)$, f_{1ij} and f_{2i} represent three major factors of the assembly sequence for this antenna, which are the assembly deformation of panels, the extra cost between the ith panel and $i+1$st panel, and the cost of each panel during assembly, respectively. The cost consists of labor cost and time cost, and the labor cost is judged by the salary (hourly wages multiplied by working time) of the assembly workers, and the time cost is judged by the time to complete the assembly of the reflector.

$g_i(x)$ is related to the tightening sequences and forces of screws and the gravity of panels. It can be computed by the finite element method proposed in Section 2 based on $F=K\delta$, where F is the force, K is the fitness matrix, and δ is the displacement of panels.

$$f_{1ij} = \begin{cases} \left|\mathrm{Ring}_i - \mathrm{Ring}_j\right| + \left|\mathrm{mod}\left(\frac{\mathrm{Number}_i}{12}\right) - \mathrm{mod}\left(\frac{\mathrm{Number}_j}{12}\right)\right|, \left|\mathrm{mod}\left(\frac{\mathrm{Number}_i}{12}\right) - \mathrm{mod}\left(\frac{\mathrm{Number}_j}{12}\right)\right| \le 6 \\ \left|\mathrm{Ring}_i - \mathrm{Ring}_j\right| + \left|12 - \left|\mathrm{mod}\left(\frac{\mathrm{Number}_i}{12}\right) - \mathrm{mod}\left(\frac{\mathrm{Number}_j}{12}\right)\right|\right|, \left|\mathrm{mod}\left(\frac{\mathrm{Number}_i}{12}\right) - \mathrm{mod}\left(\frac{\mathrm{Number}_j}{12}\right)\right| > 6 \end{cases} \tag{2}$$

$f_{1ij}$is computed as Eq. (2). The mass of each panel is always in the tens or hundreds of kilograms, which must be moved by several workers and lifting machinery. The distance between the ith panel and the $i+1$st panel determines the workload and required time for moving the panel and the lifting machinery, which is related to the assembly sequence of panels. The distance between the Nos. 2 and 6 panels is longer than the distance between the Nos. 2 and 3 panels. Eq. (2) is used to estimate the distance between the panels, where "mod" means taking the remainder of the calculation results of $\left(\frac{\mathrm{Number}_i}{12}\right)$ and $\left(\frac{\mathrm{Number}_j}{12}\right)$. $|\mathrm{Ring}_i - \mathrm{Ring}_j|$ is the ring distance between the two panels (radial distance). $\left|\mathrm{mod}\left(\frac{\mathrm{Number}_i}{12}\right)-\mathrm{mod}\left(\frac{\mathrm{Number}_j}{12}\right)\right|$ and $\left|12\text{-}\left|\mathrm{mod}\left(\frac{\mathrm{Number}_i}{12}\right)-\mathrm{mod}\left(\frac{\mathrm{Number}_j}{12}\right)\right|\right|$ are the circumference distances.

f_{2i}, defined as $f_{2i}=F_i+\omega_i\cdot F_i$, estimates the labor costs and required time of the ith panel during the assembly process based on the assumptions that the pre-tightening force and the labor cost for tightening each screw are the same. The value of F_i, illustrated in Section 3.1, is the time cost required to tighten the screws. Pre-tightening forces of screws may be imposed several times, which will cause extra time to tighten the screws. The value of ω_i is illustrated by Eq. (3) and is the time cost of each tightening sequence. $\omega_i\cdot F_i$ is the extra labor cost to tighten screws.

$$\omega_i = \begin{cases} 4, \mathrm{Seq}_i \in [1, 12] \\ 3, \mathrm{Seq}_i \in [13, 16] \\ 2, \mathrm{Seq}_i \in [17, 18] \\ 1, \mathrm{Seq}_i = 19 \end{cases} \tag{3}$$

3.3 Operations of GA

The population selection is based on the roulette wheel method which generates a mating pool from the previous generation to select parent chromosomes [21]. Once the parent chromosomes are selected, the crossover operation that depends on P_c can begin. After the crossover operation, the chromosomes mutate with P_m. The crossover operation will exchange the attributes between the genes of two parents with the same part number, and the crossover genes are selected randomly. If the probability $P_m(P_c)$ set in advance is greater than the random probability $P_{rm}(P_{rc})$ generated by the system in each operation, then the candidate is ready to undergo a mutation (crossover). Otherwise, it maintains its properties and does not undergo change.

4 Basic ant colony optimization

Each panel is an assembly part. The ants are gathered in the first panel (the basic assembly part, which is selected randomly), and then under the guidance of pheromone intensity τ_{ij} (namely, transition probability) and the heuristic information η_{ij}, the ants move to the next node and select the next panel from the candidate list. The assembly information matrices of panels are the genomes of the panels in GA. Thus, the calculation of the main parameters of ACO, including pheromone intensity and heuristic information, are all related to the parameters of the genomes of GA.

The ACO algorithm rules are derived in this section, and there are two basic parameters of ACO to explain. One is the pheromone intensity, τ_{ij}, which is the positive feedback of pheromone accumulation [24]. The more ants are in one path, the greater the probability of the later ants choosing this path. η_{ij} relates to the distance and assembly direction between two cities and is the problem-dependent heuristic information of ACO that can evaluate the pros and cons of pheromone tours. In this study, η_{ij} concerns the changes from the ^{i}th step to the ^{j}th step and can be defined as

$$\eta_{ij} = \frac{1}{f_{1ij}} \tag{4}$$

In the integrated ASP and ACO, the ants should search the tasks with a transition probability. The transition probability through which each ant selects task j after current task i is completed by introducing a new constant, a threshold q_0 of the pheromone, which determines the relative importance of exploitation versus exploration to avoid losing diversity and appearing "stagnant". When ant k of the searching process moves from ^{i} to ^{j}, the algorithm will generate a random number q that is uniformly distributed $(0<q<1)$. Ant k will choose the tour with more pheromones when $q>q_0$. Considering the above, the transition probability P is defined as follows:

$$P = \begin{cases} p_{ij}^k = \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum \tau_{ij}^{\alpha}\eta_{ij}^{\beta}}, q > q_0 \\ \operatorname{argmax}\left\{ \left[\tau_{ij}\right]^{\alpha} \left[\eta_{ij}\right]^{\beta} \right\}, q \le q_0 \end{cases} \tag{5}$$

where the probabilistic state transition rule p_{ij}^k is calculated according to the number of pheromones and heuristic information, and α and β are the parameters that determine the relative importance of pheromone versus distance.

In this case, all ants can deposit pheromones, and the pheromones will guide the ants to select a tour and volatilize at the same time [23–25]. The global updating rule is used for updating the pheromone level of all paths after the ants finish the tour, after which the pheromones on all paths will be evaporated, and the extra pheromones will be added only to the paths of the current global optimal solution. The global updating rule can be represented by

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} \tag{6}$$

$$\Delta\tau_{ij} = \begin{cases} L_{gb}^{-1}, \text{path}_{ij}\in \text{gbt} \\ 0, \text{otherwise} \end{cases} \tag{7}$$

where $0<\rho<1$ is the global pheromone release coefficient that can evaluate the decay of pheromones, gbt is the global best tour, $\Delta\tau_{ij}$ is the addition of the extra pheromone of the ant moving from i to j, and L_{gb} is the length of the global best tour of the trial. In this case, L_{gb} is the fitness of the global optimal solution found at the current evolution stage.

The local updating is used for updating the pheromone level of the path only when the ants visit it, and it can be represented by

$$\tau_{ij}(t) = (1-\zeta)\tau_{ij} + \zeta\cdot\tau_0 \tag{8}$$

where $0<\zeta<1$ is the local pheromone release coefficient, which determines the pheromone volatility on the path from task $^{\mathrm{i}}$ to task $^{\mathrm{j}}$, and τ_0 is the initial pheromone level on the tour.

The candidate list allowed$_i^k$ is based on the assembly relations between panels, which are all contained in the genomes of panels. In the dynamic update of the candidate list, k will be removed from candidate list allowed$_i^k = \{1, 2, 3, \cdots\cdots, N\}$ after the assembly of the No.k panel.

5 The algorithm framework

Combining the GA rules and ACO rules, Fig. 4 shows the framework of GAACO for assembly sequence planning. The following is the detailed procedure:

Step 1: Set the initial parameters of GA and ACO, including the initial population, mutation rate, crossover rate, and stoppage criteria, and set the initial generation n = 1 and cycle counter nc = 1.

Step 2: Set the initial population; each ant selects its initial feasible node and starts its tour.

Step 3: Generate the initial population by the sequences of m ants obtained from the completed tour. Then go to step 6.

Step 4: Choose the next feasible node according to the ACO rules described in Section 3. Add the component of the feasible node to the tabu list of the ant. Locally update the pheromone according to Eq. (8), and return to step 3.

Step 5: Calculate the fitness of each chromosome, which will serve as the criterion for the evaluation of the chromosome (Section 3.3), and sort the chromosomes by fitness value.

Step 6: If the termination condition is met, execute step 11.

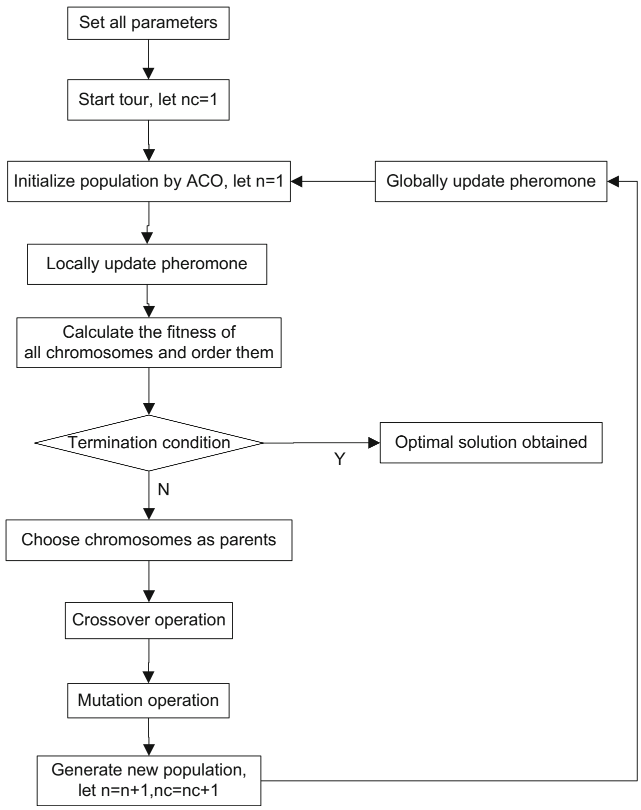


Fig. 4 Flowchart of GAACO

Step 7: Choose chromosomes from parents using the roulette wheel method (Section 3.3) and according to the fitness value calculated in step 5.

Step 8: Crossover operations are performed, and the unfeasible chromosomes are removed.

Step 9: Mutate chromosomes according to Section 3.3. After the operations of crossover and mutation, the attributions of parent chromosomes are changed.

Step 10: A new population is generated, and the information of the new population will feed back to the tour of ACO. Globally update the pheromone using Eq. (6); empty the sequence, candidate list, and tabu list of each ant; set $n = n + 1$ and nc = nc + 1; and return to step 2.

Step 11: Output the reversed best sequence.

6 Case study

6.1 Case study

To validate the proposed method and algorithm, the finite element simulation model of this 9-m reflector antenna (described in Section 2) was created with ANSYS according to the method illustrated in Section 2, as shown in Fig. 5. The simulation results can be judged by the RMS (root mean square) value of the panel deformation, which is to judge the accuracy of the reflector. The calculation of the RMS is shown in reference [27]. In this section, Max-Z represents the max displacement of the Z-axis; δ_z represents the RMS value of the Z-axis displacement; and Max-δ_z represents the maximum value of δ_z of the assembled panels.

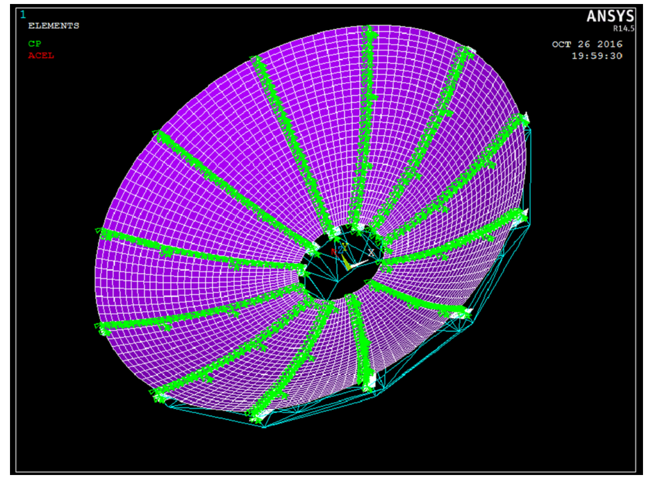


Fig. 5 The finite element simulation model (meshing) of the antenna reflector

Based on the guidelines of studies in [18, 26], the parameters were selected as follows: termination generation N=400, P_c=0.9, P_m=0.05, α=1, β=1, ζ=0.1, ρ=0.2, and q_0=0.2. With the above parameters and simulation model, the ASP algorithm model of the reflector was completed, and the flowchart of the reflector assembly based on ASP is shown in Fig. 6. The GAACO algorithm, programmed by the MATLAB software program, and the finite element simulation model, created by ANSYS (APDL), are joined by programming (C programming language). The calculation results of GAACO will be parsed and employed into the finite element simulation automatically by programming, and then the simulation is started, and the simulation results (deformation) are fed back to the algorithm to calculate the fitness function (including the deformation and the cost) by programming.

It is noted that the unfeasible chromosomes (identified in this case) were removed and not classified in the new population. The unfeasible chromosomes were identified if the following were not met: first, the new genes should be in line with part features in engineering; and second, at least one of the new genes should be different from the genes of the original chromosome. The unfeasible chromosomes are identified, detected, and removed automatically by the algorithm programming with MATLAB after each searching and optimization.

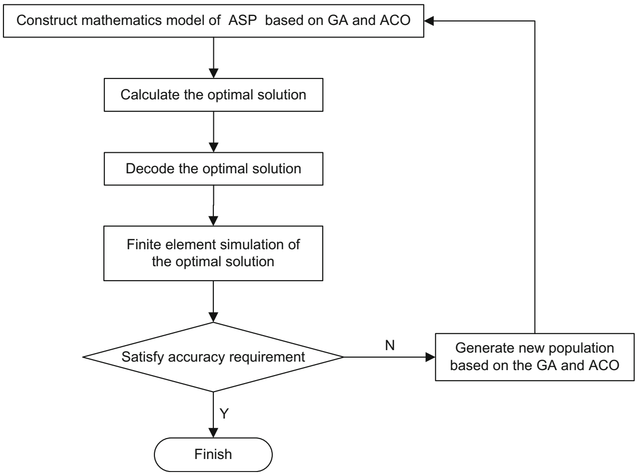


Fig. 6 Flowchart of the reflector assembly based on ASP

6.2 Discussion

Figure 7 shows the comparison of the simulation results and the measurement results of the assembly of the reflector in actual engineering. The Max-Z values of the simulation results and measurement results are 5.362 and 5.563 mm, respectively, the Max-$\delta_z$values are 0.604 and 0.62 mm, respectively, and the final (36th assembly step) δ_z values are 0.482 and 0.5 mm, respectively. The error of δ_z (3.6%) is less than 5% (value of engineering experience), which indicates that the simulation results conform to the measurement results. This demonstrates that the reflector antenna model and simulation method are correct and effective.

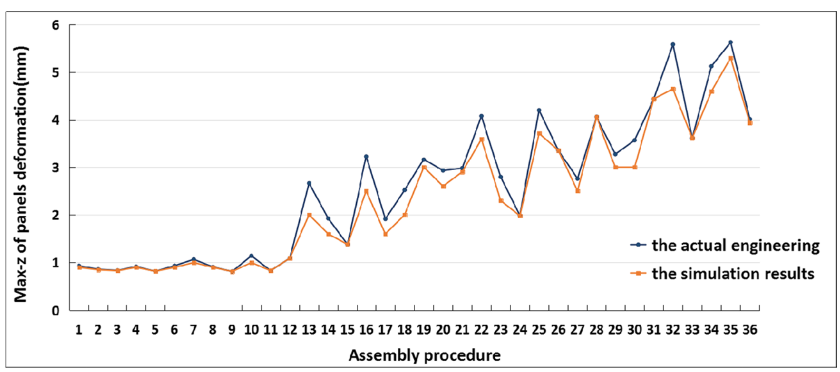
The algorithm that contains only GA was used as the basis of the comparison, and all parameters were the same as in GAACO. The algorithm results are discussed in this section.

Based on the assembly optimization process in Section 5, the convergence curves of GA and GAACO are shown in Fig. 8 (calculation results of MATLAB). The multi-objective fitness value is on the *Y*-axis, and the iteration number is on the *X*-axis. The best assembly sequence (BAS)—that is, the global optimal solution obtained by GA ($n = 372$)—is 9–4–3–10–1–6–14–2–7–11–8–20–12–5–23–16–22–15–18–21–13–33–17–19-34–28–36–30–26–24–32–31–27–29–25–35. The BAS obtained by GAACO ($n = 298$) is 2–6–10–5–3–1–9–4–14–11–7–12–8–20–16–36–30–26–22–13–23–25–15–18–19–17–34–28–33–24–32–21–31–27–29–35. The above two BAS are employed in the finite element simulation model (explained in Section 2) to calculate the deformation results of the reflector, which are shown in Fig. 9, and to compare the measurement results to those of the actual assembly of this reflector without considering the ASP problem.

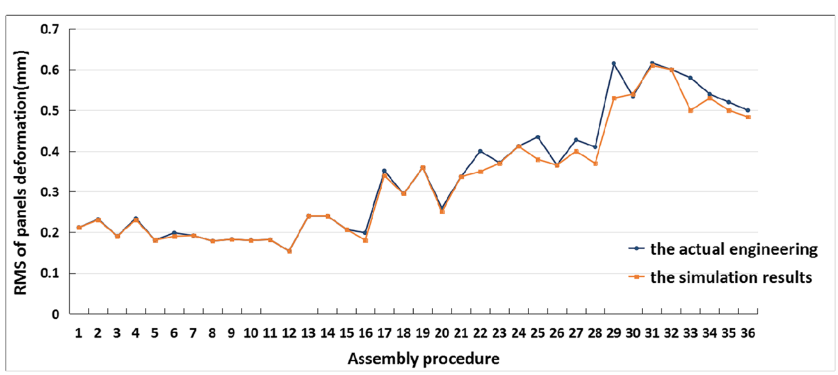
The following conclusions can be obtained through the calculation results and experimental data:

The ASP algorithms based on GA and GAACO for the reflector converge to $n = 372$ and $n = 298$, respectively, which

Fig. 7 Comparison of the actual process and simulation results. **a** The comparison of Max-Z of panel deformation. **b** The comparison of RMS of panel deformation



a) The comparison of Max-Z of panel deformation



b) The comparison of RMS of panel deformation

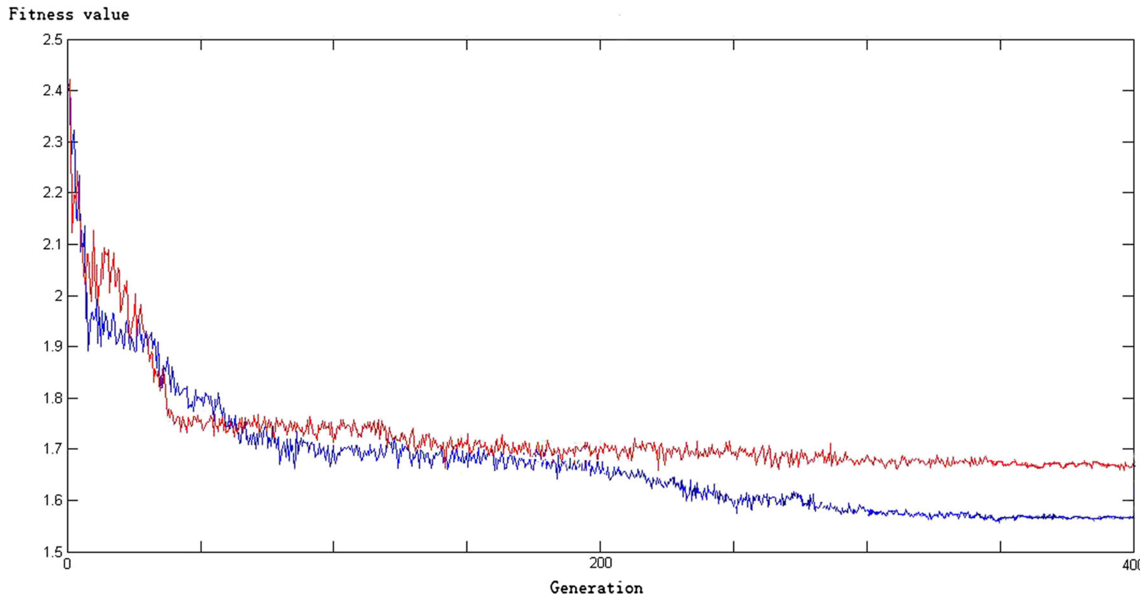
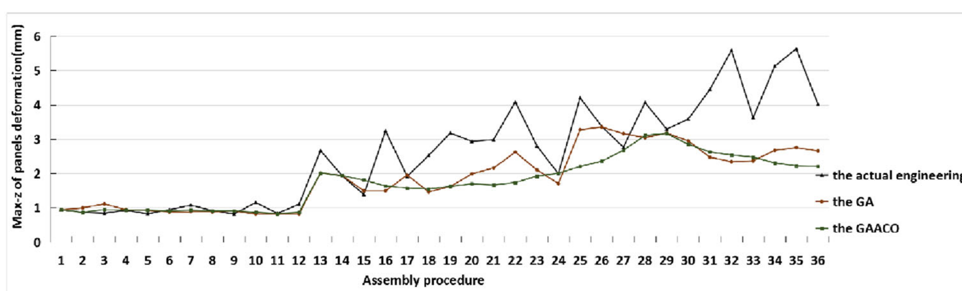


Fig. 8 The calculation results of GAACO

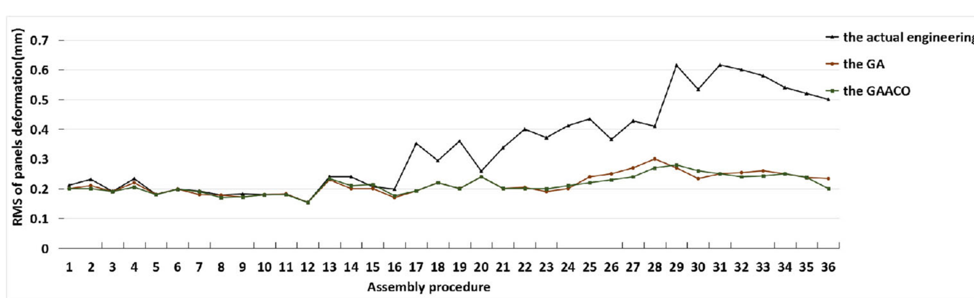
intuitively suggest that the two ASP algorithms are both correct. It also proves that the GAACO converges more quickly than GA. This is because the initial population in the GAACO is of higher quality than that in the GA. The initial populations are obtained from the ants that completed the tour in the ACO and randomly by system in the GA. In addition, the samples of GAACO are more diverse than GA in the later searching stage.

1) The reflector accuracies of the BAS based on GA and GAACO are 0.225 and 0.2 mm, respectively (Fig. 9) and are much smaller than the measurement result (0.447 mm), which demonstrates that the ability of ASP to improve the accuracy of the reflector is highly efficient, and the better accuracy of GAACO than GA may be because the evaluations of GAACO are made more precise and efficient by introducing the simulation results.
2) The best fitness values (Fig. 8) of GA and GAACO are 1.7104 (n = 392) and 1.5689 (n = 332), respectively, which also indicate that GAACO has a faster convergence rate than GA and a superior optimal solution. The fitness function of GAACO considering the simulation results is

Fig. 9 Comparison of the actual process with the GA and GAACO results. **a** The comparison of Max-Z of panel deformation. **b** The comparison of RMS of panel deformation



a) The comparison of Max-Z of panel deformation



b) The comparison of RMS of panel deformation smarter than the fitness functions of other algorithms and more precise and efficient to guide the research.

3) According to Fig. 8, the convergence curve of GAACO decreases more rapidly than GA, which is evidence that the searching ability of GAACO is more efficient than that of GA. The faster convergence rate of GAACO benefits from the information of the optimized solution feedback to the tour of ACO, which accelerates the progress towards an optimal solution.
4) According to the accumulation results (Fig. 8), the computation speed of the GAACO algorithm is relatively faster than GA, and the algorithms of GA and GAACO have great stability. The combination of GA and ACO improves the convergence to an optimal sample.

7 Conclusion

The difficulties faced in accurately assembling and evaluating large reflector antennas have significant influences on their electrical properties. This paper presents an assembly optimization approach based on the GA and ACO algorithms to predict and control the assembly of reflectors.

Using the GA as a base algorithm and introducing ACO into the GA, the proposed GAACO algorithm compensates for the low diversity of samples in later GA searches and offsets the “stagnation” and “blind random searching” of the single ACO algorithm, thus accelerating the convergence rate of the algorithm. Simultaneously, the finite element simulation results of the optimal solutions are introduced into the fitness function and termination criteria to evaluate the performance of the proposed approach precisely, which make the fitness function more efficient, smarter, and more suitable for the assembly optimization of the reflector. In addition, the finite element simulation method of this optimization approach is achieved using the “life–death element technique”, which requires less computation time and is computationally cheaper than the conventional method. A case study of a 9-m reflector antenna is performed, and the algorithm results based on the proposed approach were compared against the measurement results and the results of GA, which indicate that the proposed approach can obtain the optimized results faster and better, accurately predict the reflector’s assembly results, significantly reduce the assembly errors, shorten the assembly cycle, and reduce the labor consumption.

Further work may examine the performance of this approach with other constraints in assembly sequence planning, including human factors, and machine and workstation assignment, and develop an algorithm with faster convergence and possibly specialized software.

Acknowledgments The authors wish to thank the anonymous reviewers for their helpful comments on this paper. The Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No.2014jz016).

References

1. Lai HY, Huang CT (2004) A systematic approach for automatic assembly sequence plan generation. Int J Adv Manuf Technol 24(9–0):752–763
2. Xing Y, Chen G, Lai X, Jin S, Zhou J (2007) Assembly sequence planning of automobile body components based on liaison graph. Assem Autom 27(2):157–164
3. Su Q (2007) Computer aided geometric feasible assembly sequence planning and optimizing. Int J Adv Manuf Technol 33(1–2):48–58
4. Wang LH, Keshavarzmanesh S, Feng HY, Buchal RO (2009) Assembly process planning and its future in collaborative manufacturing: a review. Int J Adv Manuf Technol (1–2):132–144
5. Li MY, Wu B, Hu YM, Jin C, Shi TL (2013) A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation. Int J Adv Manuf Technol 68:617–630
6. Bourjault A (1984) Contribution à une approche méthodologique de l’assemblage automatisé: élaboration automatique des séquences opératoires, Thèse d’Etat. Université de Basançon Franche-Comté, France
7. DeFazio TL, Whitney DE (1987) Simplified generation of all mechanical assembly sequences. IEEE Trans Robot Autom 3(6):640–658
8. DeMello LSH, Sanderson AC (1991) A correct and complete algorithm for the generation of mechanical assembly sequences. IEEE Trans Robot Autom 7(2):228–240
9. Dong TY, Tong RF, Zhang L, Dong JX (2007) A knowledge-based approach to assembly sequence planning. Int J Adv Manuf Technol 32(11–12):1232–1244
10. Bai YW, Chen ZN, Bin HZ, Hun J (2005) An effective integration approach toward assembly sequence planning and evaluation. Int J Adv Manuf Technol 27(1–2):96–105
11. Liao C, Tseng C, Luarn P (2007) A discrete version of particle swarm optimization for flowshop scheduling problems. Comput Oper Res 34(10):3099–3111
12. Tseng YJ, Yu FY, Huang FY (2011) A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method. Int J Adv Manuf Technol 57(9–12):1183–1197
13. Lv HG, Lu C (2010) An assembly sequence planning approach with a discrete particle swarm optimization algorithm. Int J Adv Manuf Technol 50(5–8):761–770
14. DeCastro LN, Timmis JI (2002) Artificial immune system: a new computational intelligence approach. Springer, London
15. Cao PB, Xiao RB (2007) Assembly planning using a novel immune approach. Int J Adv Manuf Technol 31(7):770–782
16. Bonneville F, Perrard C, Henrioud JM (1995) A genetic algorithm to generate and evaluate assembly plans. Proceedings of the IEEE Symposium on Emerging Technology and Factory Automation, 231–239
17. Kongar E, Gupta SM (2006) Disassembly sequencing using genetic algorithm. Int J Adv Manuf Technol 30:497–506
18. Giudice F, Fargione G (2007) Disassembly planning of mechanical systems for service and recovery: a genetic algorithms based approach. J Intell Manuf 18:313–329

19. Seo KK, Park JH, Jang DS (2001) Optimal disassembly sequence using genetic algorithms considering economic and environmental aspects. Int J Adv Manuf Technol 18:371–380
20. Failli F, Dini G (2000) Ant colony systems in assembly planning: a new approach to sequence detection and optimization. Proceedings of the 2nd CIRP International Seminar on Intelligent Computationin Manufacturing Engineering, 227–232
21. Wang JF, Liu JH, Zhong YF (2005) A novel ant colony algorithm for assembly sequence planning. Int J Adv Manuf Technol (2005) 25:1137–1143
22. Yu JP, Wang CE (2013) A max–min ant colony system for assembly sequence planning. Int J Adv Manuf Technol 67:2819–2835
23. Cong L, Zhuo Z (2016) Integrated assembly sequence planning and assembly line balancing with ant colony optimization approach. Int J Adv Manuf Technol 83:243–256
24. Zhou W, Zheng JR, Yan JJ, Wang JF (2011) A hybrid assembly sequence planning approach combining bacterial chemotaxis with genetic algorithm. Int J Adv Manuf Technol 52(5–8):715–724
25. Zhang HY, Liu HJ, Li LY (2014) Research on a kind of assembly sequence planning based on immune algorithm and particle swarm optimization algorithm. Int J Adv Manuf Technol 71(5–8):795–808
26. Kucukkoc I, Zhang DZ (2016) Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines. Int J Adv Manuf Technol 82(1–4):265–285
27. Akpınara S, Bayhanb M, Baykasoglub A (2013) Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks. Appl Soft Comput 13(1):574–589
28. Wang W, Wang CS, Li P, Song LW (2008) Panel adjustment error of large reflector antennas considering electromechanical coupling. In: Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, (pp. 775–779).
29. Wang W, Duan BY, Li P (2010) Optimal surface adjustment by the error-transformation matrix for a segmented-reflector antenna. IEEE Antennas and Propagation Magazine 52(3):80–87