ORIGINAL ARTICLE

Optimization of Fixture Number in Large Thin-Walled Parts Assembly Based on IPSO

Changhui Liu[1,2]* , Jing Wang[1], Ying Zheng[1], Ke Jin[1], Jianbo Yu[1] and Jianfeng Liu[3]

Abstract

There are lots of researches on fixture layout optimization for large thin-walled parts. Current researches focus on the positioning problem, i.e., optimizing the positions of a constant number of fixtures. However, how to determine the number of fixtures is ignored. In most cases, the number of fixtures located on large thin-walled parts is determined based on engineering experience, which leads to huge fixture number and extra waste. Therefore, this paper constructs an optimization model to minimize the number of fixtures. The constraints are set in the optimization model to ensure that the part deformation is within the surface profile tolerance. In addition, the assembly gap between two parts is also controlled. To conduct the optimization, this paper develops an improved particle swarm optimization (IPSO) algorithm by integrating the shrinkage factor and adaptive inertia weight. In the algorithm, particles are encoded according to the fixture position. Each dimension of the particle is assigned to a sub-region by constraining the optional position range of each fixture to improve the optimization efficiency. Finally, a case study on ship curved panel assembly is provided to prove that our method can optimize the number of fixtures while meeting the assembly quality requirements. This research proposes a method to optimize the number of fixtures, which can reduce the number of fixtures and achieve deformation control at the same time.

Keywords Assembly quality, Large thin-walled parts, Fixture layout, PSO, FEM

1 Introduction

Large thin-walled parts are commonly-used components in aircraft manufacturing, ship building, and other industrial fields. They are usually assembled to construct outer shells, such as ship hulls and fuselage, which provide necessary space for passengers and cargoes. This kind of parts have large in-plane dimensions including the length and width (3 m−10 m), while the thickness of the part is very small (1 mm−10 mm), which leads to its low out-of-plane stiffness. Due to its low out-of-plane stiffness, deformation under the action of gravity is easy to occur. Its deformation will affect the final assembly quality,

which will reduce the surface quality as well as the service life of products.

To suppress the part deformation in the assembly process, an "N-2-1" locating principle came into being [1]. According to the principle, N (N > 3) fixtures are placed on the main datum plane to reduce the deformation (shown in Figure 1). The research shows that this principle not only restricts the degree of freedom of large thin-walled parts, but also reduces their deformation. However, when arranging fixtures on the basis of "N-2-1" locating principle, how to arrange N fixtures reasonably has become a key problem for engineers. Therefore, lots of researches on the optimal design of fixture layout have been done.

Most researches on fixture layout optimization design are carried out under the situation that the number of fixtures is known. Researchers combine the finite element model with optimization algorithms to optimize fixture layout. Cai et al. [1] divided the plate into meshes

*Correspondence:
Changhui Liu
liuchanghui@tongji.edu.cn
[1] School of Mechanical Engineering, Tongji University, Shanghai 201804, China
[2] Shanghai Jiaotong University, Shanghai 200240, China
[3] Shanghai Waigaoqiao Shipbuilding Co., Ltd., Shanghai 200137, China

Liu *et al. Chinese Journal of Mechanical Engineering*   (2024) 37:1

Page 2 of 15

with the finite element method (FEM). The deformation at the grid nodes was used to express the deformation of the plate. Then they applied the nonlinear programming algorithm to obtain the optimal fixture layout to reduce the overall deformation. Ahmad et al. [2–4] proposed to use the overall strain energy to represent the deformation of the part. They used the FEM to calculate the strain energy, and then used different optimization methods to conduct optimization. Bi et al. [5] used the partial least squares regression method to carry out optimization. Wu et al. [6] combined genetic algorithm (GA) with FEM to get an appropriate layout of the auxiliary support of the blade in order to suppress its deformation. ANSYS and ABAQUS are commonly-used finite element analysis (FEA) software. Liao and Wang [7] combined ANSYS and MATLAB to build the finite element model, and then used the mode purchasing sampling method to search for the appropriate fixture layout. Hajimiri et al. [8] used ABAQUS for FEA and calculated part deformation. They optimized fixture layout and clamping sequence by GA. Xiong et al. [9] and Yang et al. [10] used Python to modify the parameters of the finite element model, so as to calculate the part deformation under different fixture layouts. Different heuristic algorithms were utilized for optimization.

In addition to calculating the deformation directly in the FEA software, some researchers also chose to derive the stiffness matrix and modify it to calculate the deformation. Such behavior can be summarized as building a finite element solver. Du et al. [11] adopted the direct stiffness method (DSM) to obtain the deformation of each node. They modified the stiffness matrix according to modification rules proposed by Wu et al. [12], and then obtained the deformation according to Hooke's law. Liu and Hu [13] adopted the method of influence coefficients (MIC) to deal with the stiffness matrix. Camuz et al. [14] used MIC to obtain the plastic distribution of all nodes on the sheet metal. The method they proposed can efficiently improve the accuracy of deformation prediction.
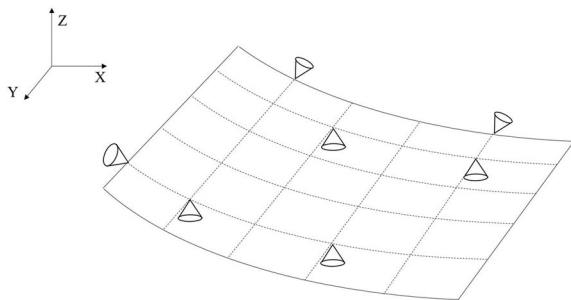
The above studies are carried out when the number of fixtures has been determined based on engineering experience. In most assembly process of large thin-walled parts, the N fixtures are uniformly fixed on the *X-Y* plane (illustrated in Figure 1), while the heights of the "N" locators in *Z* direction are adjusted by workers according to the profile of each part to be assembled. With the large in-plane dimensions, the current dense and uniform fixture layout makes the value of "N" a great number. It brings long fixture setup time, extra assembly workload, and high cost in assembly process. Therefore, to reduce extra cost and improve the assembly efficiency under the requirements of assembly accuracy, the number of fixtures needs to be optimized.

Li and Melkote [15] employed the sequential quadratic programming technique to optimize the number of fixtures. They designed an iterative synthesis algorithm. Simulation results showed that their approach improved the workpiece location accuracy significantly. Wang et al. [16] proposed an approach which combined FEM with nonlinear programming algorithms to get the appropriate position and number of the fixtures. To optimize fixtures' position and number, Liao [17] proposed a method based on GA. The deformation due to the gravity effect was also minimized. The method was applied to an industrial case and the practicability of the approach was demonstrated. Yang et al. [18] constructed the grey model to link the maximum deformations of the parts to the number of fixtures. The number of fixtures was found under the allowable maximum deformation. Khodabandeh et al. [19] came up with a novel idea which combined the FEM and multi-objective ant colony algorithm. The number and position of clamps were both optimized. This method was proved to be effective by the fixture layout optimization for the automotive side reinforcement. Aderiani et al. [20] utilized evolutionary optimization algorithms and compliant variation simulation of the assembly. They intended to optimize several fixture layout parameters simultaneously. Two cases from the automotive industry were studied to prove that the presented method was effective.

Although researchers have done some research on the optimization of fixture number, these studies still have limitations. First, most studies only consider the fixture layout of one part, while the assembly process involves at least two parts. Therefore, it is necessary to consider the fixture layout optimization of two parts at the same time. Second, most of the studies aim to reduce the part overall deformation. However, in large thin-walled part assembly process, in addition to the part deformation itself, there is also the assembly gap between the two parts. As many thin-walled parts are assembled by seam welding, the size of assembly gap has a great impact on the welding quality

Figure 1 "N-2-1" locating principle for large thin-walled part assembly

in the assembly process. Therefore, it is of great significance for improving the assembly quality to control the assembly gap size. Thirdly, the optimization of the number of fixtures was mostly carried out in the form of trial-and-error in the past. Researchers increased or reduced the number of fixtures one by one in each stage, looked for the optimal fixture layout respectively, and finally got the smallest number of fixtures. This optimization method has cumbersome steps and low efficiency. Therefore, an improved particle swarm optimization (IPSO) algorithm is proposed to optimize the number of fixtures for large thin-walled parts in this paper.

The optimization model is constructed in this paper. The DSM is used to calculate the part deformation and lay a foundation for the deformation and assembly gap control. Then, with the IPSO algorithm, the number of fixtures is optimized. Finally, taking ship curved panel assembly as a case, the feasibility of our method is proved. The arrangement of this paper is as follows: Section 2 introduces the construction of the optimization model, and Section 3 introduces the optimization algorithm. The case study of ship curved plane assembly is discussed in Section 4. Finally, the conclusions are summarized in Section 5.

2 Formulation of Fixture Layout Optimization for Large Thin-Walled Parts

According to the "N-2-1" locating principle, the layout scheme of "N" fixtures plays a critical role in reducing the part deformation in the assembly process. However, each additional fixture or adjustment of the fixture will produce additional costs. To reduce the waste and improve profits, it is essential to design an optimization model to minimize the number of fixtures while meet the assembly requirements. This section will describe the optimization problem, decision variables, optimization objectives and constraints, and construct the optimization model of the fixture layout.

2.1 Decision Variables and Optimization Objectives

In "N-2-1" principle, N fixtures placed on the main datum plane are aimed at reducing part deformation. Two fixtures are located on the second datum plane and one fixture is placed on the third plane. Among them, the fixtures on the main datum plane have great influence on the part deformation. Therefore, this paper mainly considers the layout optimization of "N" fixtures on the main datum plane. According to the engineering practice, fixtures cannot locate on sharp curves. Therefore, the fixture design space should subtract these areas. If the fixture positions are represented by the coordinate, it will be difficult to design constraints when the part shape is irregular. To simplify the problem, the fixture positions are expressed by discrete variables $X = [x_1, x_2, x_3, \ldots, x_N]$. X is a vector representing the fixture layout. $x_i (i = 1, 2, 3, \ldots, N)$ means the position of each fixture, which is represented by the index of finite element mesh nodes. N denotes the fixture number.

The specific method is as follows. Using the finite element analysis software, two large thin-walled parts are divided into multiple grids. It is assumed that the fixtures are arranged at the grid nodes. There are n_1 nodes and n_2 nodes for part 1 and part 2, respectively. Considering that the fixture cannot be arranged at the sharp edge, the nodes at the edge of the parts are excluded from the optional range in this paper. NF_1 and NF_2 are used to represent the number of nodes which cannot be selected on two parts respectively. In other words, there are $(n_1 + n_2 - NF_1 - NF_2)$ nodes where fixtures can be located on the two parts. C_s represents the set of optional nodes so that $x_i \in C_s, i = 1, 2, 3, \ldots, N$.

The optimization objective of this paper is as follows to optimize the number of fixtures:

$$\min_{x_i \in C_s} N(X), \tag{1}$$

where $N(X)$ means the number of fixtures when the fixture layout is X. The objective is to minimize the number of fixtures.

2.2 Constraints

Large thin-walled parts have low out-of-plane stiffness. This characteristic makes them easy to deform. Excessive part deformation will also cause excessive assembly gap between two parts, resulting in a decline in assembly quality. Therefore, while optimizing the number of fixtures, constraints are needed to make sure that the part deformation and the assembly gap meet the requirement. The DSM is used to help calculate part deformation and assembly gap.

Firstly, the overall part deformation is considered. The large thin-walled part is divided into multiple grids through finite element analysis, and the nodal deformation is used to characterize the overall deformation. To calculate the deformation, this paper assumes that the deformation at each node is linear elastic. Based on Hooke's law, the overall stiffness equation can be obtained as follows:

$$KU = F, \tag{2}$$

where $F \in R^n$ represents the force vector, $K \in R^{n \times n}$ means the global stiffness matrix, and $U \in R^n$ is the deformation vector. n represents the number of nodal displacements. The deformation at each node of the part can be calculated by expanding Eq. (2) as follows:

$$
\begin{bmatrix}
k_{11} & k_{12} & k_{13} & \cdots & k_{1n} \\
k_{21} & k_{22} & k_{23} & \cdots & k_{2n} \\
k_{31} & k_{32} & k_{33} & \cdots & k_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
k_{n1} & k_{n2} & k_{n3} & \cdots & k_{nn}
\end{bmatrix}
U =
\begin{Bmatrix}
f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n
\end{Bmatrix}. \tag{3}
$$

To improve the calculation efficiency, this paper used the DSM [11, 12] to calculate U. According to the DSM, U is calculated by adjusting the stiffness matrix and force vector. Suppose the αth node displacement u^α is known and $u^\alpha = u^{\alpha\prime}$. The DSM set $f_\alpha' = k_{\alpha\alpha} u^{\alpha\prime}$, $f_i' = f_i - k_{i\alpha} u^{\alpha\prime}$, and $k_{i\alpha}' = k_{\alpha i}' = 0$, where $i = 1, 2, 3, \ldots, n$ and $i \neq \alpha$. In the fixture layout process, when a fixture is set on node α, it means that $u^\alpha = 0$. According to the direct stiffness method, the displacement boundary conditions imposed by the fixture can be easily expressed. For example, if $\alpha=3$, i.e., $u^3=0$, then Eq. (3) can be modified as:

$$
\begin{bmatrix}
k_{11} & k_{12} & k_{13} & \cdots & k_{1n} \\
k_{21} & k_{22} & k_{23} & \cdots & k_{2n} \\
0 & 0 & k_{33} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
k_{n1} & k_{n2} & k_{n3} & \cdots & k_{nn}
\end{bmatrix}
U =
\begin{Bmatrix}
f_1 \\ f_2 \\ 0 \\ \vdots \\ f_n
\end{Bmatrix}. \tag{4}
$$

Through this method, the original Eq. (2) can be rewritten as follows:

$$
K'_1(X)U_1(X) = F'_1(X), K'_2(X)U_2(X) = F'_2(X), \tag{5}
$$

where $F'_1(X)$ and $K'_1(X)$ are the adjusted force vector and stiffness matrix of part 1, respectively. $F'_2(X)$ and $K'_2(X)$ denote the force vector and adjusted stiffness matrix of part 2, respectively. $U_1(X)$ and $U_2(X)$ are the node deformation vectors of part 1 and part 2 respectively.

During assembly, the surface profile tolerance of the large thin-walled part meets the requirement. In this paper, the surface profile is expressed by the node deformation. Therefore, the requirements for surface profile tolerance can be reflected by the following constraints:

$$
\begin{cases}
u_{1,i}(X) = \sqrt{u_{1,ix}^2(X) + u_{1,iy}^2(X) + u_{1,iz}^2(X)} \le \varepsilon, i = 1, 2, \ldots, n_1, \\
u_{2,j}(X) = \sqrt{u_{2,jx}^2(X) + u_{2,jy}^2(X) + u_{2,jz}^2(X)} \le \varepsilon, j = 1, 2, \ldots, n_2,
\end{cases} \tag{6}
$$

where $u_{1,ix}(X)$, $u_{1,iy}(X)$, $u_{1,iz}(X)$, $u_{2,jx}(X)$, $u_{2,jy}(X)$, and $u_{2,jz}(X)$ are linear displacements of the ith node of part 1 and the jth node of part 2 in the X, Y and Z directions, respectively. ε is the surface profile tolerance requirement.

Besides considering the surface profile, the influence of the assembly gap between the two parts on the assembly quality is also considered in this paper. In fact, if the assembly gap between the two plates is too large, it will greatly affect the quality of welding. The calculation formula for assembly gap dimension of two parts is as follows:

$$
\begin{cases}
\psi_k(X) = \sqrt{\left(u_{1,kx}(X) - u_{2,kx}(X)\right)^2 + \left(u_{1,ky}(X) - u_{2,ky}(X)\right)^2 + \left(u_{1,kz}(X) - u_{2,kz}(X)\right)^2}, \\
H(X) = \max(\psi_k(X)), \\
k = 1, 2, \ldots, m_0,
\end{cases} \tag{7}
$$

where $\psi_k(X)$ means the assembly gap at node k under fixture layout X. m_0 is the number of nodes along the assembly gap. $H(X)$ is the maximum assembly gap, which needs to meet the requirement of less than σ. Therefore, the constraint is as follows:

$$
H(X) \le \sigma. \tag{8}
$$

In general, the optimization model is as follows:

$$
\min_{x_i \in C_s} N(X), s.t. K'_1(X)U_1(X) = F'_1(X), K'_2(X)U_2(X) = F'_2(X), u_{1,i}(X) \le \varepsilon, i = 1, 2, \ldots, n_1, u_{2,j}(X) \le \varepsilon, j = 1, 2, \ldots, n_2, H(X) \le \sigma. \tag{9}
$$

It can be found that the optimization model contains three unequal constraints. To simplify the model, the number of unequal constraints should be reduced. Therefore, the penalty function $\delta(X)$ is introduced. $\delta(X)$ is defined as follows:

$$\delta(X) = \begin{cases} 0, X \in C_1 \text{ and } X \in C_2, \\ \delta_0 \times \max\{u_{1,i}(X), u_{2,j}(X)\}, X \notin C_1 \text{ or } X \notin C_2, \end{cases} \quad i = 1, 2, \ldots, n_1, j = 1, 2, \ldots, n_2, \tag{10}$$

where $C_1 = \{X | u_{1,i}(X) \le \varepsilon, i = 1, 2, \ldots, n_1\} C_2 = \{X | u_{2,j}(X), \le \varepsilon, j = 1, 2, \ldots, n_2\}$, and δ_0 is a positive parameter. δ_0 is much greater than σ. At this time, the three unequal constraints of the original problem can be written as follows:

$$H'(X) = H(X) + \delta(X) \le \sigma. \tag{11}$$

It can be proved that Eq. (11) is equivalent to the three unequal constraints in Eq. (9). The proof process is as follows:

(1) When $X \notin C_1$ or $X \notin C_2$, $\delta(X)$ is much greater than σ as δ_0 is much greater than σ. Therefore, $H'(X)$ will be greater than σ.
(2) When $H(X)$ is greater than σ, because $\delta(X)$ is greater than or equal to 0, $H'(X)$ must be greater than σ.
(3) When X satisfies all three unequal constraints, $\delta(X)$ equals to 0 and $H(X)$ is less than or equal to σ. Therefore, $H'(X) \le \sigma$ holds.

The optimization model is simplified as follows:

$$\min_{x_i \in C_s} N(X), s.t. K'_1(X) U_1(X) = F'_1(X), K'_2(X) U_2(X) = F'_2(X), H'(X) \le \sigma. \tag{12}$$

3 Method for Fixture Layout Optimization for Large Thin-Walled Parts

After constructing the optimization model, we need to apply the optimization algorithm to calculate the optimal number of fixtures. As a classical heuristic algorithm, particle swarm optimization (PSO) algorithm has simple structure and fast search speed. It is often used to solve fixture layout optimization problems [21–24]. To further improve the search ability for the fixture layout optimization, the IPSO algorithm is proposed. This section introduces the IPSO algorithm and the specific optimization process.

3.1 Improved Particle Swarm Optimization Algorithm

Kennedy and Eberhart [25] first introduced the PSO algorithm. PSO simulates a behavior of birds gathering towards the same target in a multidimensional space. According to the actual requirements, we improve the PSO algorithm.

Section 3.1.1, 3.1.2 and 3.1.3 respectively introduce the IPSO algorithm from three aspects: mapping between fixture layouts and particles, updating particle velocity and position, and selection of optimal fixture layouts.

3.1.1 Mapping Between Fixture Layouts and Particles

There are NP particles in PSO algorithm, and every particle represents a solution of the problem. As for the optimization model in this paper, each particle represents a fixture layout with N^* dimensions, where N^* means the maximum number of fixtures for two parts. The mth particle can be expressed as $X_m = [x_{m,1}, x_{m,2}, x_{m,3}, \ldots, x_{m,N^*}]$, where $x_{m,l}(l = 1, 2, 3, \ldots, N^*)$ indicates the index of the node where the lth fixture are located. For a fixture, it can be placed on any node on the part except the nodes on the edge. However, considering the large number of optional nodes and large solution space, it will become difficult to search feasible solutions. To improve the efficiency, the optional position range of each fixture is constrained. The original set C_s is divided into N^* subsets. These N^* subsets have no intersection with each other, and their combination constitutes C_s. For the lth fixture, its position meets the condition that $x_{m,l} \in C_s^l$, where C_s^l represents the set of optional nodes for the lth fixture.

As the assembly process involves two parts and the fixtures are scattered on them, a particle is also composed of two parts. One part represents the fixture layout on part 1 and the other part represents the fixture layout on part 2. As shown in Figure 2, the first n_1^*-dimensional variable represents the positions of n_1^* fixtures for part 1, while the last n_2^*-dimensional variable represents the positions of n_2^* fixtures for part 2. n_1^* and n_2^* mean the maximum number of fixtures for part 1 and part 2, respectively. The sum of n_1^* and n_2^* equals N^*.

In the general PSO algorithm, the variable has a range. Once the variable exceeds the range, it needs to be pulled back. However, the goal of this paper is to optimize the number of fixtures. Therefore, when the variable is out of range, it is not pulled back to the range, but is regarded as the position of virtual fixtures. The number of fixtures is reduced in this way. For example, suppose after iteration t, $X_m(t) = [x_{m,1}, x_{m,2}, x_{m,3}, \ldots, x_{m,p}, \ldots, x_{m,q}, \ldots, x_{m,N^*}]$, where $x_{m,p}$ is outside the optional range C_s^p, and $x_{m,q}$ is outside the optional range C_s^q. Therefore, $x_{m,p}$ and $x_{m,q}$ are regarded as the position of virtual fixtures. In this case, the number of real fixtures becomes $(N^* - 2)$, which means the fitness value of the mth particle $N(X_m)$ equals to $(N^* - 2)$. In this way, the number of fixtures is optimized.

Liu *et al. Chinese Journal of Mechanical Engineering*   (2024) 37:1

Page 6 of 15

3.1.2 Updating Particle by Integrating Shrinkage Factor and Adaptive Inertia Weight

During the iteration, the velocity $V_m(t)$, the current position $X_m(t)$, the best position a particle reached in previous iterations $X_m^{pbest}(t)$, and the best position of all particle $X_m^{gbest}(t)$, determine the new position of particle m. Every dimension of velocity $V_m(t+1)$ and position $X_m(t+1)$ is updated as follows:

$$
\begin{aligned}
v_{m,d}(t+1) &= \omega v_{m,d}(t) + c_1 r_1\left(x_{m,d}^{pbest}(t) - x_{m,d}(t)\right) + c_2 r_2\left(x_d^{gbest}(t) - x_{m,d}(t)\right) \\
&= \begin{cases} V_{\min}, v_{m,d}(t+1) \leqslant V_{\min}, \\ v_{m,d}(t+1), V_{\min} < v_{m,d}(t+1) < V_{\max}, \\ V_{\max}, V_{\max} \leqslant v_{m,d}(t+1), \end{cases} \\
x_{m,d}(t+1) &= x_{m,d}(t) + v_{m,d}(t+1),
\end{aligned}
\tag{13}
$$

where r_1 and r_2 are the random real numbers in range [0, 1]. c_1 and c_2 are the learning factors. $V_{\max}$ and $V_{\min}$ are the maximum and minimum values of particle velocity. ω is an inertia weight coefficient. The subscript d indicates the dth dimension of the vectors. $v_{m,d}(t+1), v_{m,d}(t), x_{m,d}^{pbest}(t), x_{m,d}(t), x_d^{gbest}(t)$ and $x_{m,d}(t+1)$ represent the dth element of $V_m(t+1), V_m(t), X_m^{pbest}(t), X_m(t), X^{gbest}(t)$ and $X_m(t+1)$, respectively.

Learning factors c_1 and c_2 reflect the information exchange between particles. Setting a larger c_1 will cause too many particles to search in the local range, while a larger c_2 will make the particles trapped in local optimal. To control the velocity of particles and make the algorithm achieve the balance between local search and global search, the shrinkage factor φ is introduced [26]. This adjustment method can ensure the convergence of PSO algorithm [27]. Therefore, the improved velocity and position update equation is as follows:

In PSO algorithm, inertia weight coefficient ω is a very important parameter. It indicates the ability of particles to maintain the motion state of the previous moment. Also, it can help the algorithm keep the balance between local search and global search. The larger ω is, the stronger the global search ability can be. When ω is small, the local search ability become strong. For the optimization algorithm, we hope that it not only has good global search ability to find better solutions, but also can accurately search the local space of better solutions. Therefore, many researchers are interested in the improvement of the inertia weight coefficient ω [28–31]. This paper chooses to introduce adaptive inertia weight [32]. The calculation of ω of the mth particle is as follows:

$$
\omega_m(t) = \begin{cases} \omega_{\min} + (\omega_{\max} - \omega_{\min})\frac{N(X_m(t)) - \min(N(X(t)))}{\overline{N(X(t))} - \min(N(X(t)))}, N(X_m(t)) \leq \overline{N(X(t))}, \\ \omega_{\max}, N(X_m(t)) > \overline{N(X(t))}, \end{cases}
\tag{15}
$$

where $\omega_{\min}$ and $\omega_{\max}$ are the preset minimum and maximum inertia weights respectively, $\overline{N(X(t))}$ is the average fitness of all particles in the tth iteration, $\min(N(X(t)))$ is the minimum fitness in the tth iteration, and $N(X_m(t))$ means the fitness of the mth particle in the tth iteration. The meaning of this equation is as follows: The smaller the fitness value is, the closer it is to the optimal solution, which means a greater need of local search. Conversely, global search is more needed.
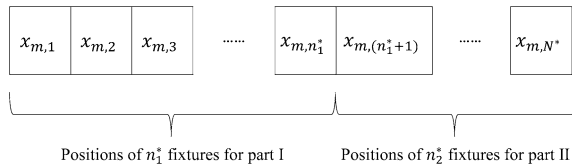
$$
\begin{aligned}
v_{m,d}(t+1) &= \varphi\left(\omega v_{m,d}(t) + c_1 r_1\left(x_{m,d}^{pbest}(t) - x_{m,d}(t)\right) + c_2 r_2\left(x_d^{gbest}(t) - x_{m,d}(t)\right)\right) \\
&= \begin{cases} V_{\min}, v_{m,d}(t+1) \leqslant V_{\min} \\ v_{m,d}(t+1), V_{\min} < v_{m,d}(t+1) < V_{\max}, \\ V_{\max}, V_{\max} \leqslant v_{m,d}(t+1), \end{cases} \\
\varphi &= \frac{2}{\left|2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)}\right|}, \\
x_{m,d}(t+1) &= x_{m,d}(t) + v_{m,d}(t+1).
\end{aligned}
\tag{14}
$$

Figure 2 Sample particle in the IPSO algorithm

3.1.3 Selection of Optimal Fixture Layouts

Figure 3 illustrates the specific flow of the IPSO algorithm. As can be seen from the flow chart, selecting and updating X_m^{pbest} and X^{gbest} is a very important step in PSO algorithm. For unconstrained optimization problems, only the fitness value is needed when selecting X_m^{pbest} and X^{gbest}. However, the optimization problem in this paper is a constrained optimization problem. Therefore, when updating X_m^{pbest} and X^{gbest}, the constraint $H'(X_m)$ needs to be carefully considered. $H'(X_m)$ can be obtained based on Eq. (11). In this process, DSM is applied to calculate node deformation vectors, which are necessary for calculating $H'(X_m)$. The specific application of DSM is as shown in Section 2.2. $H'(X_m)$ will be one of the key bases for updating X_m^{pbest} and X^{gbest}.

The criteria of updating X_m^{pbest} and X^{gbest} are as follows:

(1) When there are particles to make H' meet the constraint of $H' \leq \sigma$, select the one with the smallest N;

(2) When there is no particle to make H' meet the constraint of $H' \leq \sigma$, select the one with the smallest H';

(3) When there are multiple particles that can make H' meet the constraint of $H' \leq \sigma$, and their N is equal, select the one with the smallest H'.

The criteria help us avoid selecting a fixture layout that has a small number of fixtures but cannot control part deformation.

As for the mth particle, the best position it reached in previous t iterations is recorded as $X_m^{pbest}(t)$. After the $(t+1)$th iteration, the resulting fixture layout is recorded as $X_m(t+1)$. The process of updating $X_m^{pbest}(t+1)$ is shown in Algorithm 1. The process of updating
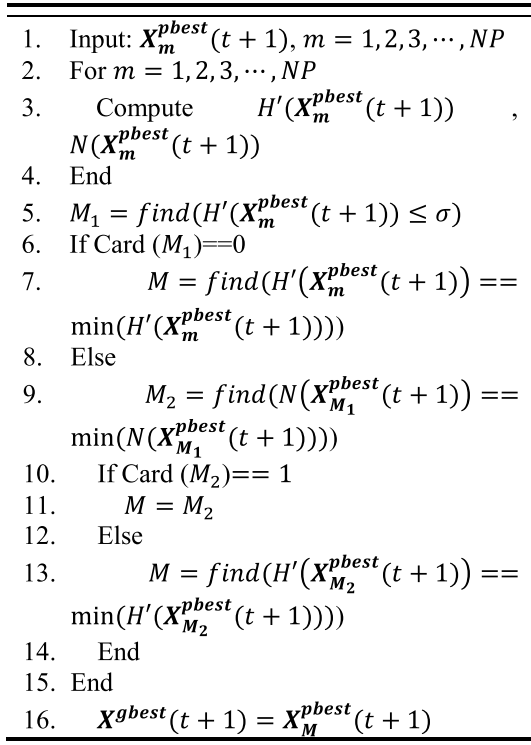
$X_m^{pbest}(t+1)$ is actually a process of selecting a better solution between $X_m^{pbest}(t)$ and $X_m(t+1)$. H' and N are two important criteria for evaluating which solution is better. There are many situations when comparing H' and N, so we listed how to make choices in various situations in Algorithm 1.

After updating $X_m^{pbest}(t+1)$, $X^{gbest}(t+1)$ is waiting to be updated. $X^{gbest}(t+1)$ is recorded as the swarm's best position after $(t+1)$ iterations. It is selected from $\left\{ X_m^{pbest}(t+1), m=1,2,\cdots,NP \right\}$. Because $X_m^{pbest}(t+1)$ is chosen from two solutions, while $X^{gbest}(t+1)$ is chosen from NP solutions, the update procedures are different. The new swarm's best position $X^{gbest}(t+1)$ is updated according to Algorithm 2. H' and N are both considered when updating $X^{gbest}(t+1)$.

Algorithm 1 Procedure of updating $X_m^{pbest}(t+1)$ in the $(t+1)$th iteration

---

1. Input: $X_m^{pbest}(t), X_m(t+1)$
2. Compute $H'(X_m^{pbest}(t)), H'(X_m(t+1)), N(X_m^{pbest}(t)), N(X_m(t+1))$
3. If $H'(X_m(t+1)) \leq H'(X_m^{pbest}(t))$ and $N(X_m(t+1)) \leq N(X_m^{pbest}(t))$
4. $\quad X_m^{pbest}(t+1) = X_m(t+1)$
5. End
6. If $H'(X_m(t+1)) > H'(X_m^{pbest}(t))$ and $N(X_m(t+1)) > N(X_m^{pbest}(t))$
7. $\quad X_m^{pbest}(t+1) = X_m^{pbest}(t)$
8. End
9. If $H'(X_m(t+1)) \leq H'(X_m^{pbest}(t))$ and $N(X_m(t+1)) > N(X_m^{pbest}(t))$
10. $\quad$ If $H'(X_m^{pbest}(t)) \leq \sigma$
11. $\quad\quad X_m^{pbest}(t+1) = X_m^{pbest}(t)$
12. $\quad\quad$ Otherwise, $X_m^{pbest}(t+1) = X_m(t+1)$
13. $\quad$ End
14. End
15. If $H'(X_m(t+1)) > H'(X_m^{pbest}(t))$ and $N(X_m(t+1)) \leq N(X_m^{pbest}(t))$
16. $\quad$ If $H'(X_m(t+1)) \leq \sigma$
17. $\quad\quad X_m^{pbest}(t+1) = X_m(t+1)$
18. $\quad\quad$ Otherwise, $X_m^{pbest}(t+1) = X_m^{pbest}(t)$
19. $\quad$ End
20. End

Algorithm 2 Procedure of updating $X^{gbest}(t+1)$ in the (t+1)th iteration

1. Input: $X_m^{pbest}(t+1), m=1,2,3,\cdots,NP$
2. For $m=1,2,3,\cdots,NP$
3.   Compute $H'(X_m^{pbest}(t+1))$, $N(X_m^{pbest}(t+1))$
4. End
5. $M_1=find(H'(X_m^{pbest}(t+1))\le\sigma)$
6. If Card (M_1)==0
7. $M=find(H'(X_m^{pbest}(t+1))==$ $min(H'(X_m^{pbest}(t+1))))$
8. Else
9. $M_2=find(N(X_{M_1}^{pbest}(t+1))==$ $min(N(X_{M_1}^{pbest}(t+1))))$
10.   If Card (M_2)== 1
11. $M=M_2$
12.   Else
13. $M=find(H'(X_{M_2}^{pbest}(t+1))==$ $min(H'(X_{M_2}^{pbest}(t+1))))$
14.   End
15. End
16. $X^{gbest}(t+1)=X_M^{pbest}(t+1)$

3.2 Process of Fixture Layout Optimization
By combining the FEM with the IPSO algorithm, this paper optimizes the number of fixtures for large
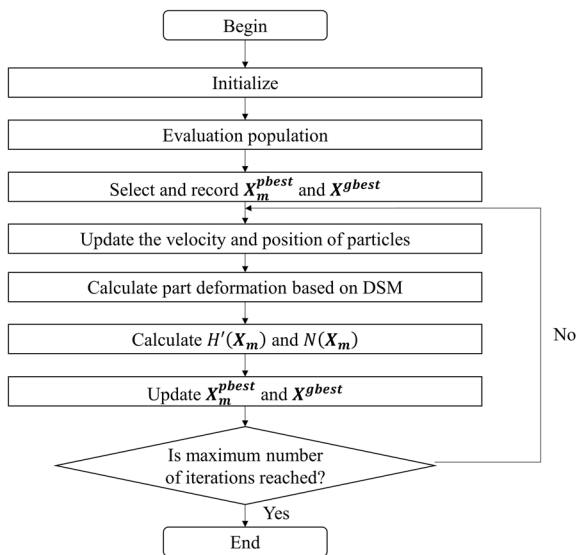
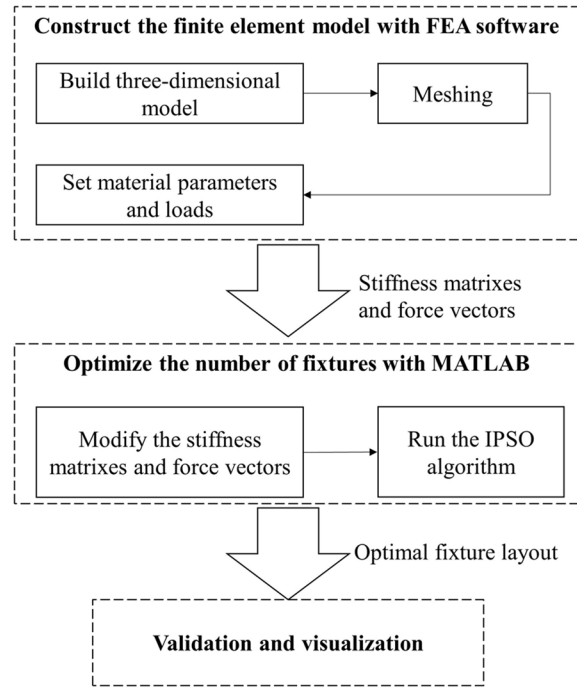**Figure 3** Flow chart of the IPSO algorithm

Figure 4 Fixture layout optimization process

thin-walled parts. The specific optimization steps are shown in Figure 4.

Step 1. Build a three-dimensional model according to the shape and size of the part. After modeling, export the data interaction file of the model.

Step 2. Meshed the model by the finite element meshing tool, and derive the data interaction file of the model. The number of finite element mesh nodes of two parts and the coordinates of each node are extracted from the data interaction file.

Step 3. Read the data interaction file using the FEA software. Set the material parameters. According to the actual situation of the assembly and splicing of the two thin plates, reasonably simplify the model, set that there is no interaction between the two parts and load gravity. After the parameter setting is completed, the stiffness matrixes and force vectors of the two parts are derived and recorded as K_1, K_2, F_1 and F_2 respectively.

Step 4. Use MATLAB to read the derived stiffness matrixes and force vectors. Set the positions of the three fixtures placed on the 2nd and 3rd datum planes according to the "N-2-1" positioning principle, and modify K_1, K_2, F_1 and F_2.

Step 5. Apply the IPSO algorithm to iteratively find the number and position of fixtures that meet all

constraints and minimize the objective function. Use DSM to calculate H' in each iteration. Calculate N. Select the optimal solution according to H' and N.

Step 6. Validate and visualize the optimization results. According to the optimal fixture layout, we found out that the FEA software is used to design constraints at the corresponding grid nodes of the finite element model. Carry out finite element simulation, obtain the deformation nephogram of the parts, and complete the result verification and visualization.

4 Case Study

A case of ship curved plane assembly is studied to demonstrate the effectiveness of our method. The curved plane used in ship assembly has the characteristics of large size and low out-of-plane stiffness, which is easy to deform under the effect of force. Therefore, it is essential to reasonably design the fixture layout to reduce the deformation. Section 4.1 introduces the background and significance of the problem and constructs the finite element model. The optimization results are shown in 4.2. The optimization result of this method are compared with that obtained by iterative trial-and-error procedure based on simulated annealing (SA) algorithm [11] in Section 4.3, which shows the advantages of this method.

4.1 Problem Description of Ship Curved Panel Assembly

Large curved panels are widely used in ship building. Hundreds of large curved panels with different shape and size are assembled and welded together to be ship hull. With the characteristics of large in-plane dimensions, small thickness and low out-of-plane stiffness, large curved panels are very easy to deform under the action of gravity. The deformation of panels will introduce assembly gap in the assembly process. Incompliant assembly gap will affect the welding quality and efficiency [33]. Therefore, it is necessary to control the assembly gap dimension in the ship assembly process.

It is a common approach to restrain the part deformation by using reasonably arranged fixtures. At present, the uniformly distributed jig frame is usually used to support the large cured panels and restrain the deformation in the assembly process. In fact, this experience-based fixture layout not only makes it difficult to achieve satisfactory results in restraining deformation, but also causes unnecessary waste due to the excessive number of fixtures. To reduce the assembly cost and effectively restrain the deformation, the fixture number and the corresponding layout need to be optimized.

The research objects of this case are two ship parts located at the bow. Part 1 and part 2 are two curved parts.
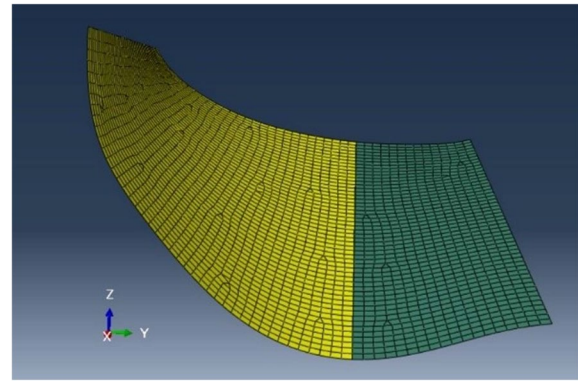

Figure 5 Finite element models of two parts to be assembled

The welding method used in the part assembly process is laser welding or arc welding. During the welding process, two curved parts are welded by long continuous welding seam [11]. The lengths of four sides for part 1 are 5200 mm, 4800 mm, 2800 mm and 4200 mm. The lengths of four sides for part 2 are 5200 mm, 5100 mm, 1500 mm and 1900 mm. Both parts are 6 mm thick. The density, Poisson's ratio, and Young's modulus of the two parts are 7.85×10^{-3} g/mm^3, 0.3, and 210000 N/mm^2 respectively.

Using HyperMesh 13.0 for mesh generation. The size of the mesh is 100 mm × 100 mm. Part 1 is divided into 1751 elements with 1828 nodes; Part 2 is divided into 826 elements with 892 nodes. Figure 5 shows the finite element model. This case only considers the influence of the gravity of the parts themselves on the deformation during the assembly process. The gravity is evenly distributed on the two parts in Z direction, and the gravitational acceleration is set as 9.81 m/s^2.

After establishing the finite element model, the stiffness matrixes K_1 and K_2, and force vectors F_1 and F_2 are derived. Table 1 shows the input parameters which are set according to the obtained model and the known information in the assembly process. The parameter settings of the IPSO algorithm are shown in Table 2. The number of particles, the number of iterations, and the particle velocity variation range are determined through trial and error. To balance the search time and search scope, these parameters are set to the values in Table 2. The variable dimension is determined according to the maximum number of fixtures on two parts. The values of learning factors and the variation range of inertia weight are set according to Ref. [26].

### 4.2 Results of Optimal Design

After setting the initial parameters, the appropriate fixture layout is found after several iterations. This section introduces the optimized fixture layout and the changes of deformation and assembly gap before and after

Table 1 Setting of input parameters

Parameter symbol	Parameter description	Parameter value
n_1	Total number of nodes on part 1	1828
n_2	Total number of nodes on part 2	892
NF_1	Total number of nodes on part 1 where fixtures cannot be placed	170
NF_2	Total number of nodes on part 2 where fixtures cannot be placed	135
m_0	Number of nodes along the interface	53
n_1^*	Maximum number of fixtures for part 1	28
n_2^*	Maximum number of fixtures for part 2	14
N^*	Maximum number of fixtures on two parts	42
ε	Surface profile tolerance for two parts (mm)	3
σ	Maximum assembly gap (mm)	1
δ_0	Penalty factor	300

Table 2 Setting of algorithm parameters

Parameter symbol	Parameter description	Parameter value
NP	Number of particles	40
G	Number of iterations	25
D	Variable dimension, $D = N^*$	42
c_1	Individual learning factor of particles	2.05
c_2	Social learning factor of particles	2.05
$[\omega_{\min}, \omega_{\max}]$	Inertia weight variation range	[0.4, 0.9]
$[V_{\min}, V_{\max}]$	Particle velocity variation range	[−95, 95]
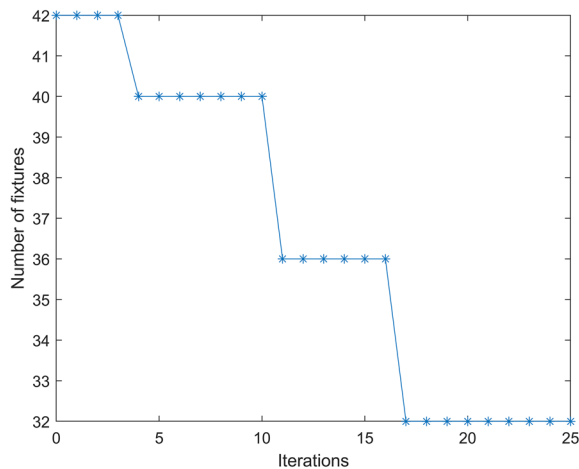
Figure 6 Iterative process of the IPSO algorithm

optimization. The comparison demonstrates the effectiveness of our method.

After 25 iterations, the optimized number and position of fixtures are obtained. The iterative process is shown in Figure 6. Table 3 shows the number of fixtures on the two parts and nodes where the fixtures are located before and after optimization. Before optimization, there are 28 fixtures on part 1 and 14 fixtures on part 2. The fixtures are evenly distributed on the two parts in the *X-Y* plane (shown in Figure 7). After optimization, the number of fixtures on part 1 is $m_1 = 21$, and the number of fixtures on part 2 is $m_2 = 11$. The total number of fixtures is reduced from 42 to 32. Figure 8 shows the *X-Y* plane projection of the fixture layout.

Although the number of fixtures is reduced, the optimized fixture layout still has a good effect of controlling deformation through the optimization of fixture position. Table 4 shows the deformation of parts with different fixture layouts. Comparing the dimension of the gap between the two parts, it can be found that the mean gap is 1.32 mm and the maximum gap is 1.86 mm when the fixture is evenly distributed. The optimized fixture layout controls the mean gap dimension to 0.37 mm and the maximum gap to 0.69 mm. They are reduced by 72.0% and 62.9% respectively compared with the former. It can be found that when the number of fixtures is 42, uniformly distributed fixtures cannot effectively control the part deformation. The maximum node deformation of the part is 4.53 mm. With the optimized fixture layout, the maximum deformation of parts is 2.20 mm, 51.4% reduced than the former. In terms of average deformation, before optimization, the mean deformation is 0.78 mm, while after optimization, the mean deformation is 0.36 mm. The average deformation is reduced by 53.8%.

To intuitively show the part deformation under different fixture layouts, the deformation diagram of the part is drawn using ABAQUS 6.14. Figure 9 shows the deformation of parts under uniform layout and optimized layout. It can be found from the figure that the most of nodal deformation of the two parts is well-controlled when the fixtures are evenly distributed. However, the deformation

Table 3 Comparison of the number and position of fixtures

	Before optimization	After optimization
Number of fixtures on part 1	28	21
Position of fixtures on part 1	[540, 496, 713, 203, 1383, 1538, 1418, 810, 575, 287, 196, 1718, 1132, 1591, 885, 845, 370, 187, 1820, 1451, 986, 457, 927, 630, 177, 949, 1410, 1502]	[250, 324, 366, 431, 513, 554, 604, 686, 750, 795, 836, 914, 968, 1014, 1154, 1202, 1324, 1450, 1482, 1713, 1812]
Number of fixtures on part 2	14	11
Position of fixtures on part 2	[831, 786, 666, 140, 524, 197, 328, 782, 859, 583, 148, 436, 292, 237]	[156, 226, 341, 354, 444, 471, 563, 591, 657, 754, 854]
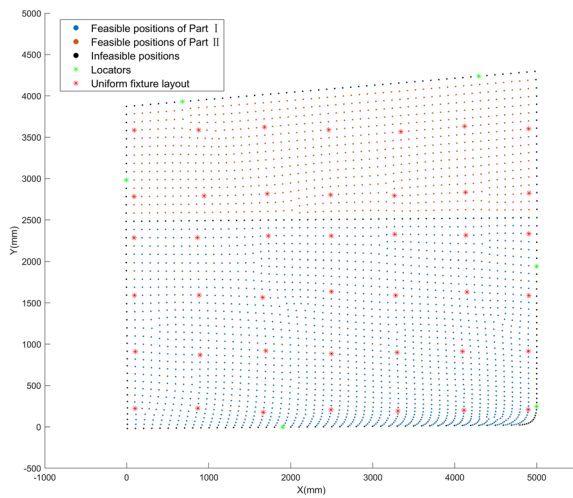
Figure 7 *X-Y* plane projection of fixture layout before optimization
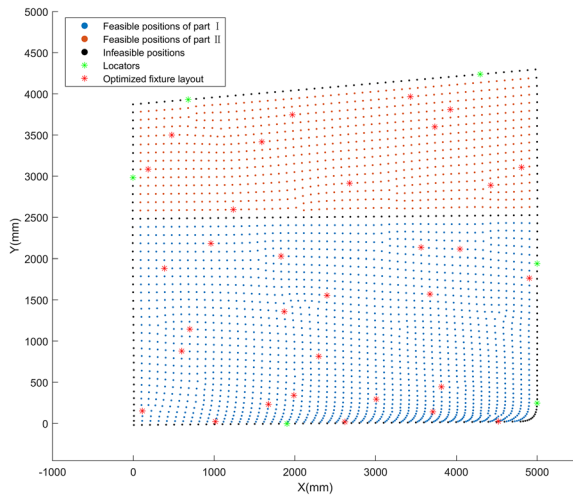
Figure 8 *X-Y* plane projection of fixture layout after optimization

of some nodes at the edge of part 2 is serious. This is because when the fixtures are evenly distributed, the fixtures at the edge are arranged sparsely due to the size and shape of part 2. After optimization, the deformation of

the two parts is well controlled because the deformation constraints are set before optimization. The maximum deformation of part 2 is reduced from 4.53 mm to 2.20 mm. Figure 9 also shows the assembly gap between two parts before and after optimization. It can be found from Figure 9 that when fixtures are uniformly distributed, the size of the assembly gap between the two parts increases along the *X*-direction and the gap is obvious. With the same deformation scale factor in ABAQUS 6.14, the assembly gap between these two parts is significantly reduced after optimization.

Figure 10 shows two different distribution histograms of node deformation before and after optimization. Figure 10(a) shows the deformation when a uniform layout is adopted. It can be found that the displacement of most nodes is less than 1 mm, and the displacement is mostly concentrated in 0.5−1 mm. However, the displacement of a few nodes has not been well controlled, exceeding 3 mm. Figure 10(b) shows the deformation after the fixture layout is optimized. It can be found that the nodal displacement is concentrated between 0−0.5 mm, and only a small part of the nodal displacement exceeds 1 mm. Figure 11 shows the gap dimensions at each node along the assembly edge. Compared with that before optimization, the gap dimensions are significantly reduced. Figures 10 and 11 illustrate that although our objective is to reduce the number of fixtures, the deformation and assembly gap dimensions of the two parts are reduced due to the setting of constraints.

4.3 Comparison and Discussion

This section proves the reliability and advantages of the proposed method from two aspects. Firstly, we validate the calculation accuracy of DSM. Secondly, the performance of IPSO algorithm for the number of fixtures optimization are compared with the traditional trial-and-error procedure.

To prove the accuracy of the calculation results of DSM, the deformation obtained by the DSM is compared with that obtained by ABAQUS 6.14. Taking the optimized fixture layout as an example, the absolute differences of the deformation results obtained by the two

Table 4 Comparison of part deformation and assembly gap under different fixture layout

Fixture layout	Mean gap (mm)	Max gap (mm)	Mean deformation (mm)	Max deformation (mm)
Uniform layout of 42 fixtures	1.32	1.86	0.78	4.53
Optimized layout of 32 fixtures	0.37	0.69	0.36	2.20
Reduction percentage (%)	72.0	62.9	53.8	51.4
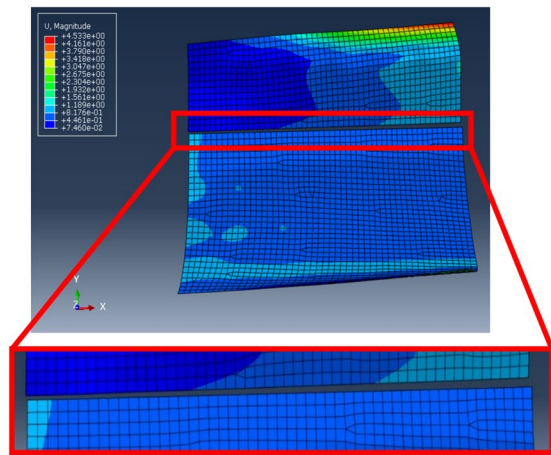
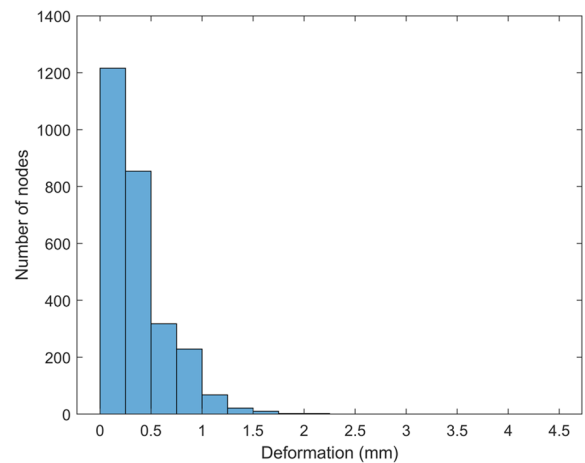(a) Deformation and assembly gap with uniform fixture layout
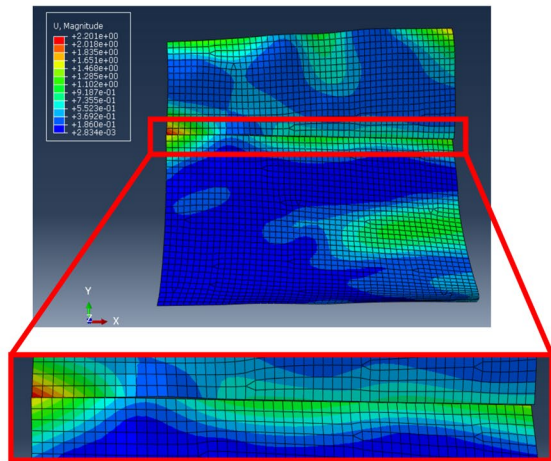
(b) Deformation and assembly gap with optimized fixture layout

Figure 9 Deformation and assembly gap with different fixture layouts
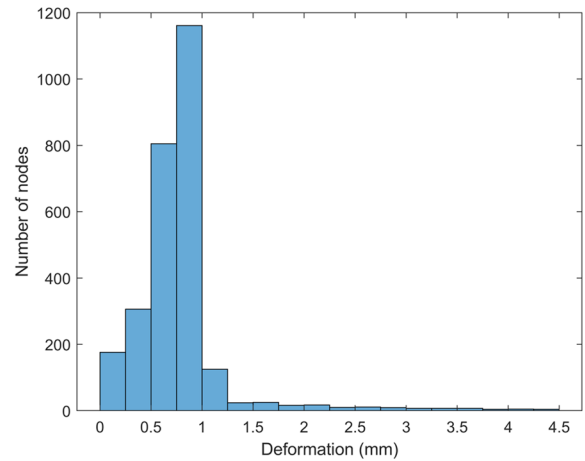
(a) Nodal deformation distribution before optimization

(b) Nodal deformation distribution after optimization

Figure 10 Nodal deformation distribution

methods are shown in Table 5. It can be found that the order of magnitude of the absolute differences is 10^{-4}, so the calculation result of DSM is credible. This means that the method of using DSM to calculate deformation does not need to call the FEA software frequently, and the calculation process is simpler.

Based on general PSO algorithm, the IPSO algorithm integrates the shrinkage factor and adaptive inertia weight to improve the search ability. To verify the superiority of the IPSO algorithm, we compare general PSO and IPSO through experiments. We run two algorithms 10 times respectively. The parameters of the IPSO algorithm are shown in Table 2, while PSO algorithm does not have shrinkage factor, and the inertia weight is fixed at 0.9. The results are plotted as a boxplot, as shown in Figure 12. It can be seen from the figure that the
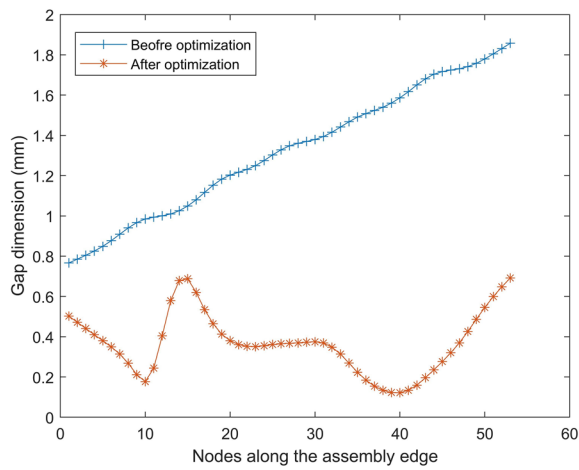
Figure 11 Comparison of gap dimensions of nodes along the assembly edge before and after optimization

Table 5 Deformation differences between DSM and FEA with the optimized fixture layout

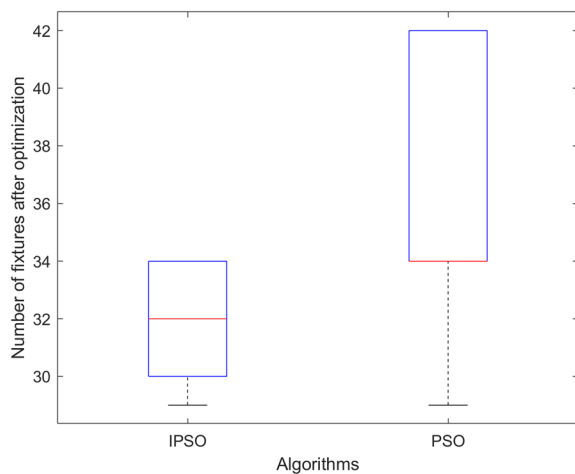| Absolute difference	Part 1	Part 2
Maximum	5.0015×10^{-4}	4.9962×10^{-4}
Mean	1.7220×10^{-4}	2.4202×10^{-4}

Figure 12 Comparison of IPSO and PSO

overall optimization result obtained by IPSO is better. The median of the numbers of fixtures obtained by IPSO algorithm is 32, while that obtained by PSO algorithm is 34. In addition, the results obtained by PSO algorithm are more scattered, which indicates that the search ability of PSO is unstable. The comparison results show that IPSO algorithm has better and more stable search ability.

It is also essential to compare the method with the traditional iterative trial-and-error process to evaluate the performance of our method. Du et al. [11] used the combination of DSM and SA algorithm to optimize the fixture layout. Their purpose was to minimize the assembly gap between the two parts. The number of fixtures was determined before optimization. Based on the method proposed by Du et al., this paper carries out iterative trial-and-error to reduce the number of fixtures. Then, the trial-and-error procedure is compared with the IPSO algorithm. The iterative trial-and-error procedure on the basis of the method developed by Du et al. is as follows: Firstly, optimize the fixture layout with a constant number of fixtures until the assembly gap and deformation meet the requirements. Then, remove a fixture at random and continue to optimize the fixture layout until the requirements are met. Iterate until the number of fixtures is reduced to the target.

According to the optimization results shown in the previous section, the initial number of fixtures is 42 and the target number is 32. The relevant parameters of SA algorithm are consistent with those in the research conducted by Du et al. [11]. The data related to the finite element models of the parts are modified according to the case in this paper.

Through the above method, the number of fixtures is reduced. Table 6 shows the comparison between the result found by the iterative trial-and-error process and the proposed method. Run these two programs on a laptop with 11th Gen Intel (R) Core (TM) i5-11400H @ 2.70 GHz processor, 16 GB of RAM, it takes longer time for the iterative trial-and-error procedure to reduce the number of fixtures by the same degree. Besides, the ability of the fixture layout found by iterative trial-and-error procedure in controlling deformation and assembly gap is not as good as that found by the proposed method. With the fixture layout obtained by iterative trial-and-error

Table 6 Comparison of two different methods

	Running time (s)	Mean gap (mm)	Max gap (mm)	Mean deformation (mm)	Max deformation (mm)
Proposed method	6309.39453	0.37	0.69	0.36	2.20
Iterative trial-and-error based on SA	8346.97434	0.71	0.99	0.39	2.12

Liu *et al. Chinese Journal of Mechanical Engineering*   (2024) 37:1

Page 14 of 15

method, the maximum deformation of the two parts is smaller, but the maximum gap, average gap and average deformation are not as small as that with the fixture layout obtained by the IPSO. The results in Table 6 show that the IPSO algorithm has higher efficiency and higher solution accuracy.

5 Conclusions

(1) An optimization model with the objective of minimizing the number of fixtures is proposed. We designed constraints to ensure that the deformation of parts and the assembly gap meet the requirements while reducing the number of fixtures. The stiffness matrixes and force vectors are derived by FEA software. The DSM is used to calculate the deformation. Using DSM can avoid frequently calling FEA software and make the process of calculating deformation easier.

(2) The IPSO algorithm is used for optimization. Each dimension of a particle represents the position of a fixture. To improve the solution efficiency, we restrict the feasible position of each fixture to ensure that the feasible positions of a fixture are in a small area. The number of feasible solutions is greatly reduced, and the phenomenon that multiple fixtures concentrated in a unified area is avoided. In addition, this paper introduces adaptive inertia weight and shrinkage factor to weigh local search and global search.

(3) A case of ship curved plane assembly is prepared. This case demonstrates that our approach can significantly reduce the number of fixtures while controlling the deformation and assembly gap very well. Comparing the deformation calculated by DSM with that calculated by FEA, it is proved that DSM can accurately calculate the deformation. In addition, the IPSO algorithm is compared with the general PSO algorithm, and the result shows that the IPSO algorithm has stronger search ability. Finally, comparing our method with the Iterative trial-and-error method, it is proved that our method has higher computational efficiency.

Authors' Contributions
CL was in charge of the whole trial, he also provided ideas and participated in manuscript writing; JW wrote the manuscript; YZ, KJ, JY, and JL assisted in modifying the structure and content of this paper. All authors read and approved the final manuscript.

Author's Information
Changhui Liu, born in 1986, is currently an assistant professor at *School of Mechanical Engineering, Tongji University*, *China*. He received his doctorate from *Shanghai Jiaotong University, China,* in 2016. His research interests include quality control and intelligent assembly.
Jing Wang, born in 1999, is currently a master candidate at *School of Mechanical Engineering, Tongji University, China*.
Ying Zheng, born in 2002, is currently an undergraduate student at *School of Mechanical Engineering, Tongji University, China*.
Ke Jin, born in 2001, is currently an undergraduate student at *School of Mechanical Engineering, Tongji University, China*.
Jianbo Yu, born in 1978, is currently a professor at *School of Mechanical Engineering, Tongji University, China*. He received his doctorate from *Shanghai Jiaotong University, China*, in 2009.
Jianfeng Liu, born in 1964, is currently a chief technical officer at *Shanghai Waigaoqiao Shipbuilding Co., Ltd., China*. He received his doctorate from *Shanghai Jiaotong University, China*, in 2000.

Data Availability
The data that supports the study is available from the corresponding author upon reasonable request.

Declarations

Competing Interests
The authors declare no competing financial interests.

References
[1] W Cai, S J Hu, J X Yuan. Deformable sheet metal fixturing: principles, algorithms, and simulations. *Journal of Manufacturing Science and Engineering,* 1996, 118(3): 318-324.
[2] Z Ahmad, M Zoppi, R Molfino. Fixture layout optimization using element strain energy and genetic algorithm. *World Academy of Science, Engineering and Technology,* 2013, 7(10): 1924 - 1930.
[3] Z Ahmad, M Zoppi, R Molfino. Preliminary study on fixture layout optimization using element strain energy. *World Academy of Science, Engineering and Technology,* 2013, 7(4): 557 - 563.
[4] Z Ahmad, T Sultan, M Asad, et al. Fixture layout optimization for multi point respot welding of sheet metals. *Journal of Mechanical Science and Technology,* 2018, 32(4): 1749-1760.
[5] Y Bi, W Yan, Y Ke. Multi load-transmitting device based support layout optimization for large fuselage panels in digital assembly. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* 2014, 229(10): 1792-1804.
[6] B Wu, Z Zheng, J Wang, et al. Layout optimization of auxiliary support for deflection errors suppression in end milling of flexible blade. *International Journal of Advanced Manufacturing Technology,* 2021, 115(5-6): 1889-1905.
[7] X Liao, G G Wang. Simultaneous optimization of fixture and joint positions for non-rigid sheet metal assembly. *International Journal of Advanced Manufacturing Technology,* 2007, 36(3-4): 386-394.
[8] H Hajimiri, V Abedini, M Shakeri, et al. Simultaneous fixturing layout and sequence optimization based on genetic algorithm and finite element method. *International Journal of Advanced Manufacturing Technology,* 2018, 97(9-12): 3191-3204.

Liu *et al. Chinese Journal of Mechanical Engineering*   (2024) 37:1

Page 15 of 15

[9] L Xiong, R Molfino, M Zoppi. Fixture layout optimization for flexible aerospace parts based on self-reconfigurable swarm intelligent fixture system. *International Journal of Advanced Manufacturing Technology,* 2012, 66(9-12): 1305-1313.

[10] B Yang, Z Wang, Y Yang, et al. Optimization of fixture locating layout for sheet metal part by cuckoo search algorithm combined with finite element analysis. *Advances in Mechanical Engineering,* 2017, 9(6): 1-10.

[11] J Du, C Liu, J Liu, et al. Optimal design of fixture layout for compliant part with application in ship curved panel assembly. *Journal of Manufacturing Science and Engineering,* 2021, 143(6): 061007.

[12] B Wu, Z Xu, Z Li. A note on imposing displacement boundary conditions in finite element analysis. *Communications in Numerical Methods in Engineering,* 2007, 24(9): 777-784.

[13] S C Liu, S J Hu. Variation simulation for deformable sheet metal assemblies using finite element methods. *Journal of Manufacturing Science and Engineering,* 1997, 119(3): 368-374.

[14] S Camuz, S Lorin, K Wärmefjord, et al. Nonlinear material model in part variation simulations of sheet metals. *Journal of Computing and Information Science in Engineering,* 2019, 19(2): 021012.

[15] B Li, S N Melkote. Optimal fixture design accounting for the effect of workpiece dynamics. *International Journal of Advanced Manufacturing Technology,* 2001, 18(10): 701-707.

[16] Z Q Wang, Y Yang, Y G Kang, et al. A location optimization method for aircraft weakly-rigid structures. *International Journal for Simulation and Multidisciplinary Design Optimization,* 2014, 5: 1-4.

[17] Y G Liao. A genetic algorithm-based fixture locating positions and clamping schemes optimization. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 2016, 217(8): 1075-1083.

[18] B Yang, Z Wang, Y Yang, et al. Determination of the number of fixture locating points for sheet metal by grey model. *2016 the 3rd International Conference on Mechatronics and Mechanical Engineering,* Shanghai, China, October 21-23, 2016.

[19] M Khodabandeh, M G Saryazdi, A Ohadi. Multi-objective optimization of auto-body fixture layout based on an ant colony algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* 2019, 234(6): 1137-1145.

[20] A Rezaei Aderiani, K Wärmefjord, R Söderberg, et al. Optimal design of fixture layouts for compliant sheet metal assemblies. *International Journal of Advanced Manufacturing Technology,* 2020, 110(7-8): 2181-2201.

[21] Y Xing, W Chen, X Li, et al. Multi-station fixture location layout optimization design for sheet metal parts. *Journal of Computational and Theoretical Nanoscience,* 2015, 12(9): 2903-2908.

[22] X H Zhou, W Liu, Q Niu, et al. Locator layout optimization for checking fixture design of thin-walled parts. *Key Engineering Materials,* 2013, 572: 593-596.

[23] K A Sundararaman, K P Padmanaban, M Sabareeswaran. Optimization of machining fixture layout using integrated response surface methodology and evolutionary techniques. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* 2015, 230(13): 2245-2259.

[24] K A Sundararaman, K P Padmanaban, M Sabareeswaran, et al. An integrated finite element method, response surface methodology, and evolutionary techniques for modeling and optimization of machining fixture layout for 3D hollow workpiece geometry. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* 2016, 231(23): 4344-4359.

[25] J Kennedy, R C Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks,* Perth, Australia, November 27-December 1, 1995: 1942-1948.

[26] R C Eberhart, Y Shi. Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of the 2000 Congress on Evolutionary Computation,* La Jolla, USA, July 16-19, 2000: 84-88.

[27] M Clerc, J Kennedy. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation,* 2002, 6(1): 58-73.

[28] P Chauhan, K Deep, M Pant. Novel inertia weight strategies for particle swarm optimization. *Memetic Computing,* 2013, 5(3): 229-251.

[29] M Li, H Chen, X Shi, et al. A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application. *Applied Soft Computing,* 2019, 84: 105677.

[30] A A Nagra, F Han, Q H Ling. An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search. *Engineering Optimization,* 2019, 51(7): 1115-1132.

[31] J Zhang, J Sheng, J Lu, et al. UCPSO: A uniform initialized particle swarm optimization algorithm with cosine inertia weight. *Computational Intelligence and Neuroscience,* 2021: 881933.

[32] C B Li, X F Jia, Y Zhou, et al. A microgrids energy management model based on multi-agent system using adaptive weight and chaotic search particle swarm optimization considering demand response. *Journal of Cleaner Production,* 2020, 262: 121247.

[33] W Liang, D Deng. Influences of heat input, welding sequence and external restraint on twisting distortion in an asymmetrical curved stiffened panel. *Advances in Engineering Software,* 2018, 115: 439-451.