

## The Intelligent Pipe-Routing Layout for the Aero-engine Based on Improved Artificial Fish Swarm Algorithm

Yu ZHANG and Xiaolan BAI

School of Mechanical Engineering & Automation  
Northeastern University, Shenyang, 110819, P. R. China  
Tel.: +86-24-83687620  
E-mail: zy4097534@126.com

*Received: 16 September 2013 / Accepted: 15 October 2013 / Published: 23 December 2013*

---

**Abstract:** The pipe-routing layout directly impacts the performance, the reliability, the cost as well as the lifecycle of the aero-engine. With the rapid development of aerospace industries, how to realize the intelligent pipe-routing layout for the aero-engine has become a hot issue to be urgently to be solved. Facing the complexity of the pipe-routing layout and the characteristic of its layout space for the aero-engine, a novel intelligent pipe-routing layout method based on the improved artificial fish swarm algorithm (IAFSA) is put forward in this paper. The method consists of two parts: the build of the local searching space and the searching of the optimized pipe-routing. The build of the local layout space is used to reduce the complexity of the problem solving. The searching of the optimized pipe-routing is used to find out an optimal trajectory in the solution space by the IAFSA which is involved in chaos mutation and gradient setting. At the end, the feasibility and effectiveness of the proposed method is verified by a case study. *Copyright © 2013 IFSA.*

**Keywords:** Aeroengine, Pipe-routing layout, Artificial fish swarm algorithm, Chaos optimization algorithm, Engineering rules.

---

### 1. Introduction

Simply speaking, the pipe-routing layout is to design the appropriate pipe path connecting the starting and goal points in an environment with scattered obstacles, meeting certain engineering rules. It is commonly involved in numerous fields such as aircraft, ship building and robotics. Moreover, it plays an important role in product design (especially for complex products such as aero-engines and ships) because of affecting the reliability, the performance, the maintainability and the life cycle of products.

Because a large number of pipes, diverse design constraints and many complex-shaped accessories considered as obstacles are involved with the pipe-routing layout for the aero-engine, the pipe-routing

layout becomes one of the most difficult works in the aero-engine design. At present, it is mainly done by hand with the aid of the computer in practice. This design mode depends on the experience of the designer to a great extent, which results in many problems such as complex operation, long design cycle and high cost. To solve above issues, the research about the pipe-routing layout for the aero-engine was done. As an earlier research, Chen et al. [1] put forward a human-machine interaction method to realize the pipe-routing layout in aero-engine. The method requires that designers have a strong professional background and knowledge. Subsequently, Li [2] proposed a knowledge-based intelligent pipe-routing layout approach. Based on the approach, the aircraft engine routing system which

mainly consists of knowledge base, intelligent reference machine, data base and computing function set was developed using Visual C++ programming language. However, the knowledge is difficult to be acquired. In 2001, Huang and Chen [3] presented twenty-seven path patterns to support the two-dimensional pipe-routing layout. Because these path patterns are limited, it is hard to comprehensively reflect the actual pipe-routing. Later, Fan et al. [4] adopted the improved Lee algorithm and the least generative method of Stanner tree to implement the three-dimensional pipe-routing layout and developed the aero-engine grid based routing system (AEGRS). However, the computation of this method is exponentially added with the number of grids increasing. At the same year, genetic algorithm was applied to the pipe-routing planning by Fan [5]. In 2009, Liu et al. [6] proposed a new method based on particle swarm optimization algorithm. In the method, a fixed-length encoding mechanism was designed instead of the variable-length one to overcome bad generality. All the above-mentioned research concerns the pipe-routing layout of the single pipe. For the pipe-routing layout of the branch pipe, Bai et al. [7] came up with an automatic pipe-routing layout method based on maze algorithm. When the number of pipeline terminals is over three, the method can't be used. In addition, some scholars [8-14] engaged in the corresponding work for ship and electromechanical products, etc. Because there are numerous unique constraints and highly complex layout space involved in the design of the aero-engine pipe-routing, these methods for other products are hard to be applied to solve the pipe-routing layout for the aero-engine. Based on the above discussion, it is concluded that there is still no a set of mature theories and methods for the aero-engine pipe-routing layout. Therefore, a novel intelligent pipe-routing method for the aero-engine layout based on the improved artificial fish swarm algorithm is proposed in this paper.

The remainder of this paper is organized as follows. The mathematical model of the pipe-routing layout for the aero-engine is established in Section 2. In Section 3, a novel intelligent pipe-routing method for the aero-engine layout based on the improved artificial fish swarm algorithm is proposed and discussed. A case is studied in Section 4. In Section 5, conclusions and future works are given.

## 2. Description of the Pipe-routing Layout Problem for the Aero-engine

### 2.1. The Mathematical Description of the Pipe-routing Layout Space

Fig. 1 shows the simplified pipe-routing layout space in the aero-engine. It is a rotary space with obstacles between the casing and the nacelle of the aero-engine.

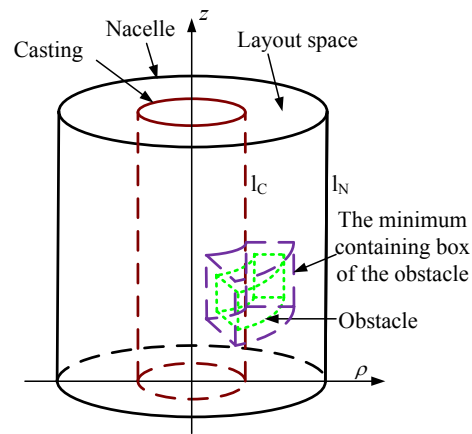


Fig. 1. The simplified pipe-routing layout space in the aero-engine.

Both the surface of the casing and the surface of the nacelle can be considered as the approximate rotatory surface which is generated by rotating a curve around a straight line. Therefore, the pipe-routing layout space in the aero-engine can be easily expressed in the cylindrical coordinate system. Its mathematical expression is described as follows.

$$l_C : \begin{cases} F_C(\rho, z) = 0 & (0 \leq z \leq h_0) \\ \theta = 0 \end{cases}, \quad (1)$$

$$l_N : \begin{cases} F_N(\rho, z) = 0 & (0 \leq z \leq h_0) \\ \theta = 0 \end{cases}, \quad (2)$$

where  $l_C$  means the generatrix of the surface of the casing,  $l_N$  means the generatrix of the surface of the nacelle,  $h_0$  is the axial length of the casing.

For more convenient description,  $l_C$  and  $l_N$  is replaced by  $\rho_C = f_C(z)$  and  $\rho_N = f_N(z)$ , respectively. Thus, the layout space in the aero-engine can be expressed as follows.

$$\begin{cases} f_C(z) < \rho < f_N(z) \\ 0 \leq \theta < 2\pi \\ 0 < z < h_0 \end{cases}, \quad (3)$$

In addition, according to the principle of the minimum containing box, the obstacle in the layout space can be represented as follows.

$$\begin{cases} \rho_{\min} < \rho < \rho_{\max} \\ \theta_{\min} \leq \theta < \theta_{\max} \\ z_{\min} < z < z_{\max} \end{cases}, \quad (4)$$

where  $[\rho_{\min}, \rho_{\max}] \times [\theta_{\min}, \theta_{\max}] \times [z_{\min}, z_{\max}]$  mean the range of the minimum containing box corresponding to the obstacle.

## 2.2. The Description of the Engineering Rules

The engineering rules are various and complex. Some important rules are listed in the following, which are taken into consideration in this paper.

Rule 1: The obstacles should be avoided.

Rule 2: The gap between pipes or between pipes and other accessories should be no less than 3 mm in order to avoid the vibration of the piping system.

Rule 3: To meet the requirement of the assembly process, the tie-in of the pipe can not be bended and the pipe needs to be extended for a distance which is no less than 2.5 times the bending radius of the pipe.

Rule 4: The beeline length of the pipe should be no less than 2.5 times the bending radius of the pipe.

Rule 5: Considering the pipe's manufacturability, the number of elbows should be as few as possible.

Rule 6: The pipe should be across the top of accessories as little as possible in order to maintain the accessories conveniently.

Rule 7: The pipe should be laid as close as possible to the surface of the casing so as to obtain better vibration performance and smaller profile.

Rule 8: For the convenience of the clamping of pipes as well as beautiful appearance, the pipe should be laid along the axial and circumferential direction as possible

Rule 9: Pipes should be laid in the half of aero-engine as possible so as to facilitate the maintenance.

## 2.3. The Mathematical Model of the Pipe-routing Layout for the Aero-engine

The mathematic model of the pipe-routing layout for the aero-engine is involved in two aspects: the objective function and constraint conditions.

### 2.3.1. The objective Function

In the layout space, the pipe-routing can be regarded as be comprised of multiple polygonal lines. Therefore, by means of the variable-length encoding mechanism, the pipe-routing can be expressed in the following.

$$path = \{(x_S, y_S, z_S), \dots, (x_j, y_j, z_j), \dots, (x_D, y_D, z_D)\}, \quad (5)$$

where  $(x_S, y_S, z_S)$  means the coordinate of the starting point,  $(x_j, y_j, z_j)$  means the coordinate of the j point,  $(x_D, y_D, z_D)$  means the coordinate of the destination.

Considering the shortest pipe-routing as the optimal object, the objective function can be formulated as follows:

$$f(l_1, \dots, l_i, \dots, l_n) = \sum_{i=1}^n l_i, \quad (6)$$

where  $l_i$  means the length of the  $i^{\text{th}}$  pipe,  $l_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$ ,  $0 < i < n + 1$ ,  $n$  means the number of pipes.

### 2.3.2. Constraint Conditions

In this section, constraint conditions corresponding to the engineering rules summarized in Section 2.2 will be taken into account. Corresponding to the engineering rules, constraint conditions can be formulated as follows:

$$n\_elbow \leq P, \quad (7)$$

$$r\_elbow \geq R \quad (8)$$

$$i\_angle \geq \pi / 2 \quad (9)$$

$$e\_length \geq 2.5r\_elbow \quad (10)$$

$$s\_length \geq 2.5r\_elbow \quad (11)$$

$$g\_length \geq \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 \quad (12)$$

$$c\_span < \pi \quad (13)$$

where  $n\_elbow$  means the number of elbows,  $P$  is the constant, which is assigned by the engineer;  $r\_elbow$  means the radius of the elbow,  $R$  is the constant. When  $D \geq 22\text{mm}$ ,  $R = 3D$ ; when  $D \leq 20\text{mm}$ ,  $R = 2D$ ,  $D$  means the outer diameter of the pipe;  $i\_angle$  means the included angle between two adjacent pipes;  $e\_length$  means the extended length;  $s\_length$  means the length of the straight pipe;  $g\_length$  means the gap between pipes or between the pipe and other accessories,  $L_1$  means the smallest gap between pipes or between pipes and general accessories,  $L_2$  means the smallest gap between pipes and special accessories,  $L_3$  means the smallest gap between pipes and rigid fixtures,  $L_1$ ,  $L_2$  and  $L_3$  are constants which are assigned by the engineer,  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  mean the weight. When the pipe is next to other pipes or general accessories,  $\lambda_1 = 1$ ,  $\lambda_2 = 0$ ,  $\lambda_3 = 0$ ; When the pipe is next to special accessories,  $\lambda_1 = 0$ ,  $\lambda_2 = 1$ ,  $\lambda_3 = 0$ ; When the pipe is next to rigid fixtures,  $\lambda_1 = 0$ ,  $\lambda_2 = 0$ ,  $\lambda_3 = 1$ ;  $c\_span$  means the circumferential span of laid pipes.

Based on the above discussion, regarding the coordinate of the endpoint as the variable and the shortest pipe-routing as the optimized object, the mathematic model of the pipe-routing layout is shown as follows.

$$\min f(l_1, \dots, l_i, \dots, l_n) = \min \sum_{i=1}^n l_i$$

$$\text{subject to } n\_elbow \leq P$$

$$r\_elbow \geq R$$

$$i\_angle \geq \pi/2$$

$$e\_length \geq 2.5r\_elbow$$

$$s\_length \geq 2.5r\_elbow$$

$$g\_length \geq \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3$$

$$c\_span < \pi;$$
(14)

$$\begin{cases} \rho_2 \leq \rho \leq \rho_1 \\ \theta_2 \leq \theta \leq \theta_1 \\ z_2 \leq z \leq z_1 \end{cases}, \quad (16)$$

where  $(\rho_1, \theta_1, z_1)$  and  $(\rho_2, \theta_2, z_2)$  respectively means the coordinates of  $P_1$  and  $P_2$  points which are two endpoints in the diagonal of the local searching space. The  $\rho_1, \rho_2, \theta_1, \theta_2, z_1$  and  $z_2$  can be calculated by the following equation.

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \theta_1 \\ \theta_2 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \max\{\rho_s, \rho_D, \rho_{B_1}, \dots, \rho_{B_N}\} \\ \min\{\rho_s, \rho_D, \rho_{B_1}, \dots, \rho_{B_N}\} \\ \max\{\theta_s, \theta_D, \theta_{B_1}, \dots, \theta_{B_N}\} \\ \min\{\theta_s, \theta_D, \theta_{B_1}, \dots, \theta_{B_N}\} \\ \max\{z_s, z_D, z_{B_1}, \dots, z_{B_N}\} \\ \min\{z_s, z_D, z_{B_1}, \dots, z_{B_N}\} \end{bmatrix}, \quad (17)$$

### 3. The Method for the Intelligent Pipe-routing Layout

Facing the complexity of the pipe-routing layout problem and the characteristic of the layout space for the aero-engine, a novel intelligent pipe-routing layout method based on improved artificial fish swarm algorithm is put forward in this section. The method consists of two parts: the build of the local searching space and the searching of the optimized pipe-routing.

#### 3.1. The Local Searching Space

To improve the efficiency of the pipe-routing layout, the local searching space including the starting point, the destination and the minimum containing boxes of corresponding obstacles is built, as shown in Fig. 2.

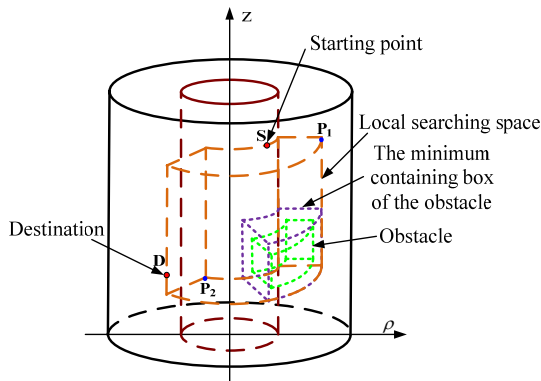


Fig. 2. The local searching space.

The local searching space can be formulated in the following.

$$K = \{S\} \cup \{D\} \cup \{B_1\} \cup \{B_2\} \cup \dots \cup \{B_N\}, \quad (15)$$

where S means the starting point, D means the destination,  $B_N$  means the  $N^{\text{th}}$  minimum containing box corresponding to the  $N^{\text{th}}$  obstacle.

Further, the local searching space can be expressed as follows.

#### 3.2. The Searching of the Optimized Pipe-routing

The searching of the optimized pipe-routing is to find an optimal trajectory in the solution space by an improved artificial fish swarm algorithm.

##### 3.2.1. Artificial Fish Swarm Algorithm (AFSA)

AFSA is a novel swarm intelligence optimization method inspired by fish swarm behaviors, which was proposed in 2002 by Li [15]. The basic idea of the AFSA is to imitate fish swarm behaviors such as praying behavior, swarming behavior, and following behavior with local search of fish individual for reaching the global optimum. Because the algorithm has many advantages such as good robustness and global search ability, it has been applied to many fields [16-17].

In the algorithm, the state of the artificial fish can be denoted as the vector  $X = (x_1, x_2, \dots, x_n)$ , where  $x_i (i = 1, \dots, n)$  means the  $i^{\text{th}}$  variable that needs to be optimized; the food consistence of the artificial fish can be denoted as  $Y = f(X)$ , where  $Y$  means the food consistence (objective function value); the distance between two artificial fishes can be denoted as  $d_{ij} = \|X_i - X_j\|$ ; *Visual* means the vision distance of the artificial fish;  $\delta$  stands for the crowded factor; *Step* is the maximum step length to move on for the artificial fish; *Try\_number* shows the largest trying number of each movement of the artificial fish. In addition, the basic behaviors of the artificial fish can be defined as follows.

1) Preying behavior.

It is a basic biological behavior that tends to the food. Suppose that the  $i^{\text{th}}$  artificial fish's current state

is  $X_i$  and its corresponding food consistence is  $Y_i$ . By Eq. 18, a new state  $X_v$  is randomly chosen in the vision distance and its corresponding food consistence is  $Y_v$ . If  $Y_v < Y_i$  in the minimum problem, it goes forward a step in this direction. Further, a new state  $X_{i|next}$  can be obtained by Eq. 19; otherwise, select a state  $X_v$  randomly again by Eq. 18 and judge whether it satisfies the forward condition. If it can't satisfy after  $Try\_number$  times, it will randomly move a step according to Eq. 20.

$$X_v = X_i + Rand() \times Visual, \quad (18)$$

$$X_{i|next} = X_i + Rand() \times Step \times \frac{X_v - X_i}{\|X_v - X_i\|}, \quad (19)$$

$$X_{i|next} = X_i + Rand() \times Step, \quad (20)$$

where  $Rand()$  produces random numbers between 0 and 1.

2) Swarming behavior.

The fish will assemble in groups naturally in the moving process, which is a kind of living habit to guarantee the existence of the colony and avoid dangers. Suppose that the  $i^{\text{th}}$  artificial fish's current state is  $X_i$  and its corresponding food consistence is  $Y_i$ . Taking the  $X_i$  as the center, the number of artificial fishes in the vision distance is  $N_f$  and these artificial fishes can be denoted as the set  $S_i = \{X_j \mid \|X_j - X_i\| \leq Visual, j = 1, 2, \dots, i-1, i+1, \dots, n\}$ .

a) If  $N_f \geq 1$  ( $S_i \neq \emptyset$ ) which means other companions exist in the area, explore the center position. The center position can be denoted as follows.

$$X_{center} = \frac{\sum_{j=1}^{N_f} X_j}{N_f}, \quad (21)$$

If  $Y_{center} < Y_i$  and  $N_f \cdot Y_{center} < \delta \cdot Y$  ( $Y_{center}$  is the food consistence corresponding to  $X_{center}$ ) which means there is more food (higher fitness function value) in the center position and it is not very crowded around the center position, the artificial fish  $X_i$  goes forward a step to the center position according to Eq. 22. Otherwise, the artificial fish executes the preying behavior.

$$X_{i|next} = X_i + Rand() \times Step \times \frac{X_{center} - X_i}{\|X_{center} - X_i\|}, \quad (22)$$

b) If  $N_f = 0$  ( $S_i = \emptyset$ ) which means no companions exist in the area, the artificial fish executes the preying behavior.

3) Following behavior.

In the moving process of the fish swarm, when a fish or several ones find food, the neighboring partners will trail and reach the food quickly. Suppose that the  $i^{\text{th}}$  artificial fish's current state is  $X_i$  and its corresponding food consistence is  $Y_i$ ; in the current area, the artificial fish's state with minimum food consistence is  $X_{min}$  and its corresponding food consistence is  $Y_{min}$ . Taking the  $X_i$  as the center, the number of artificial fishes in the vision distance is  $N_f$ .

a) When  $N_f \geq 1$ : if  $Y_{min} \geq Y_i$ , the artificial fish executes the preying behavior; if  $Y_{min} < Y$  and  $N_f \cdot Y_{min} < \delta \cdot Y_i$  which means the artificial fish  $X_{min}$  has more food (higher fitness function value) and it is not very crowded near the  $X_{min}$ , the artificial fish  $X_i$  goes forward a step to the  $X_{min}$  according to Eq. 23. Otherwise, the artificial fish  $X_i$  executes the preying behavior.

$$X_{i|next} = X_i + Rand() \times Step \times \frac{X_{min} - X_i}{\|X_{min} - X_i\|}, \quad (23)$$

b) When  $N_f = 0$  which means no companions exist in the area, the artificial fish executes the preying behavior.

### 3.2.2. Improved Artificial Fish Swarm Algorithm (IAFSA)

The original AFSA has many advantages such as the intelligence, robustness and global search ability. However, it has also some weaknesses such as low precision and slow convergence speed in the later period of the optimization. Moreover, it is insensitive to initial values. To overcome the above-mentioned deficiencies, some improvements to the algorithm are done.

1) Chaos mutation.

Chaos optimization algorithm (COA) [18] is a novel method of global optimization. Because it has the characteristics of ergodicity, stochasticity and sensitive dependence on initial conditions, COA can carry out easy and rapid searching as well as robust escape from the local optimum. Therefore, it can be used to mutate the state of the artificial fish so as to make up for the deficiencies of the original AFSA. The process of chaos mutation of the artificial fish is described as follows.

Step 1: Suppose that the state of the  $k^{\text{th}}$  generation artificial fish can be denoted as  $X^k = (x_1^k, x_2^k, \dots, x_n^k)$  and corresponding chaos variable is  $Z^k = (z_1^k, z_2^k, \dots, z_n^k)$ . The mapping relationship between them can be denoted as follows.

$$z_i^k = \frac{x_i^k - a_i}{b_i - a_i}, \quad (24)$$

where  $a_i$  and  $b_i$  are the minimum and maximum of the  $i$ th variable of  $X^k$ , respectively.

Step 2: Calculate the new chaos variable  $Z^{k+1}$  by the Logistic Map Equation (25), and the new variables by the linear mapping equation (26).

$$z_i^{k+1} = 4z_i^k(1 - z_i^k), \quad (25)$$

$$x_i^{k+1} = a_i + z_i^{k+1}(b_i - a_i), \quad (26)$$

Step 3: If  $K \leq M$ , then

If  $f(X^{k+1}) \leq f^*$ , then  $X^* = X^{k+1}$ ,  $f^* = f(X^{k+1})$

Set  $K = K + 1$ ,  $Z^k = Z^{k+1}$ , and go to Step 2.

Else if  $K > M$  is satisfied, then stop.

Where  $K$  means the current iterative number,  $M$  means the number of chaos search,  $f$  means the objective function,  $X^*$  and  $f^*$  are the current optimal variable and function value, respectively.

2) Gradient Visual and Step.

In the original AFSA, there are two important parameters: *Visual* and *Step* which are constants and largely impact on the final result. For the *Visual* parameter, when the value of the *Visual* is larger, the algorithm has better global search ability and faster convergence speed. On the contrary, when the value of the *Visual* is smaller, the algorithm has better local search ability. For the *Step* parameter, when the value of the *Step* is larger, the convergence speed of the algorithm is faster. In contrast, when the value of the *Step* is smaller, the convergence speed of the algorithm is slower, whereas the solving accuracy is higher. Therefore, in order to overcome the above-mentioned drawbacks in the original AFSA, larger values of *Visual* and *Step* are taken in the prior period of the optimization; smaller values of *Visual* and *Step* are taken in the later period of the optimization. Corresponding equations are given as follows.

$$Visual_{t+1} = Visual_t / w, \quad (27)$$

$$Step_{t+1} = Step_t / w, \quad (28)$$

where  $Visual_t$  and  $Step_t$  respectively stand for the current vision distance and step length,  $Visual_{t+1}$  and  $Step_{t+1}$  respectively mean the next vision distance and step length,  $w = 1, 2, \dots, n$ .

Based on the above discussion, the procedure of the improved AFSA is described as follows:

Step 1: Initialize the number of artificial fishes  $N$ , the maximum of the vision distance  $Visual_{max}$ , the maximum of the step length  $Step_{max}$ , the crowed factor  $\delta$ , the largest trying number of each movement of the artificial fish  $Try\_number$ , the iterations  $Q$  and the number of chaos search  $M$ .

Step 2: Generate initial artificial fishes randomly in the search space and perform chaos mutation to them.

Step 3: Create the fitness function using sequential unconstrained minimization technique which is described in the Eq. 29. Based on it, calculate the fitness value of each artificial fish in the initial artificial fish swarm. By comparing these fitness values, record the current best artificial fish in the bulletin board.

$$P(x, M) = F(x) - M \sum_{i=1}^k \ln[-g_i(x)] + \frac{1}{M} \sum_{i=1}^k [h_i(x)]^2 +, \quad (29)$$

$$\frac{1}{M} \sum_{j=1}^l [\max\{0, g_j(x)\}]^2$$

where  $F(x)$  means the original objective function,  $h_i(x)$  means the  $i$ th equality constraint,  $g_j(x)$  means the  $j^{\text{th}}$  non-equality constraint,  $M$  means the penalty factor and  $M_0 < M_1 < M_2 \dots \rightarrow \infty$ ,  $i, j = 1, 2, \dots, n$ .

Step 4: Each artificial fish implements the preying behavior, swarming behavior and following behavior, respectively. Select the best behavior to execute by comparing the function values and perform chaos mutation to the artificial fish after executing the best behavior.

Step 5: Compare the current optimal state of the artificial fish with the one of the artificial fish in the bulletin board. Further, update the bulletin board by the better result.

Step 6: Judge whether the preset iterations have been reached or a satisfactory optimum solution has been obtained. If not satisfied, go to Step 4. Otherwise go to Step 7;

Step 7: Output the optimum solution.

3) Experiment and analysis.

In order to test the effectiveness of the IAFSA, a typical function (see Eq. 30) is utilized and 50 times simulation experiments are done using it for the original AFSA and IAFSA under the same testing software and hardware.

$$\min f(x_1, x_2) = x_1^2 + 2x_2^2 + 3, -5 \leq x_1, x_2 \leq 5, \quad (30)$$

In experiments, the parameters are set as follows: the number of artificial fishes  $N = 100$ , the maximum of the vision distance  $Visual = 4$ , the maximum of the step length  $Step = 1$ , the crowed factor  $\delta = 0.618$ , the largest trying number of each movement of the artificial fish  $Try\_number = 10$  and the number of chaotic search  $M = 100$ . The experiments are performed on the Matlab 7.0 platform under the two different terminal conditions: the convergence precision is 0.00001 and the iterations are 50. The results are listed in the Table 1 and Table 2.

It can be seen from Table 1 that when the terminal condition is the fixed convergence precision, the IAFSA is superior to the AFSA whatever it is from the point of the least, the most and average iterations

of the convergence or from the point of the rate of the convergence. On the average, the iterations are reduced about 2-6 times. In addition, it can be seen from Table 2 that when the terminal condition is the fixed iterations, the worst, the best and the average optimal value from the IAFSA are better than those

from the AFSA. On the average, the optimal values are reduced about  $10^2 \sim 10^8$  times. Therefore, it can be concluded that the IAFSA has better performances such as faster convergence and higher precision than the AFSA.

**Table 1.** The comparison of the simulation results under the first terminal condition

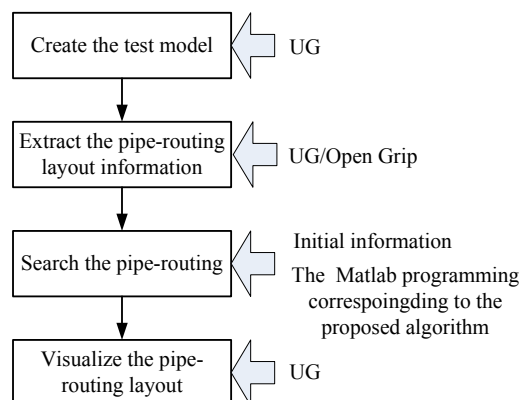
Function	Algorithm	The least iterations	The most iterations	The average iterations	The rate of the convergence
$f(x_1, x_2)$	AFSA	30	115	70.5000	50/50
	IAFSA	8	25	12.2542	50/50

**Table 2.** The comparison of the simulation results under the second terminal condition

Function	Algorithm	The best optimal value	The worst optimal value	The average optimal value
$f(x_1, x_2)$	AFSA	6.376794649359625E-008	0.63348152864264	0.03619186278757
	IAFSA	3.031760774717635E-010	3.024027765502062E-008	0.677657111177661E-008

#### 4. Case Study

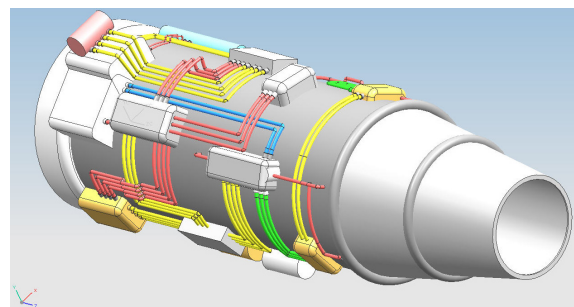
By using UG and Matlab software, the prototype system of the intelligent pipe-routing layout for the aero-engine is developed. Fig. 3 shows the workflow. Firstly, a test model is created by UG software. Secondly, the pipe-routing layout information is extracted by using the secondary development tool of UG: UG/Open GRIP. Then, the pipe-routing is searched by the Matlab programming corresponding to the proposed algorithm. Finally, the computation results are imported into UG to realize the visualized pipe-routing layout.



**Fig. 3.** The workflow of the prototype system.

A simplified aero-engine model which includes 15 accessories is created. The range of its pipe-

routing layout space is  $\theta \in [0, 2\pi]$ ,  $r \in [1000, 1560]$  and  $z \in [0, 2310]$ . In the pipe-routing layout space, there are 46 pipes to be laid. Moreover, the outer diameter of the pipe is 12 mm; the minimum clearance between pipes and obstacles is 3 mm; the bending radius is 24 mm and the length of the straight line is no less than 60 mm. In addition, the parameters in the proposed algorithm are set as follows: the number of artificial fishes  $N=60$ , the maximum of the vision distance  $Visual=8$ , the maximum of the step length  $Step=7$ , the crowed factor  $\delta=0.36$ , the largest trying number of each movement of the artificial fish  $Try\_number=3$ , the number of chaotic search  $M=200$ , and the iterations  $Q=300$ . Fig. 4 shows the pipe-routing layout applying the proposed algorithm. The entire computation time for the pipe-routing layout costs 62 seconds on the personal computer (Intel CPU 2.93 GHz, 1 GB memory).



**Fig. 4.** The aero-engine model with the pipe-routing layout.

## 5. Conclusions and Future Works

Because the intelligent pipe-routing layout can realize the automatic and optimal pipe-routing layout, it is an important approach to improve the quality and efficiency of the pipe-routing layout. This paper gave the mathematical model of the pipe-routing layout for the aero-engine and presented a novel intelligent pipe-routing layout method. In the method, the 3D rotational space of aero-engine is divided into local layout spaces so that the complexity of the problem solving is reduced. Moreover, an improved AFSA is constructed by adding chaos mutation and gradient setting so that precocious phenomenon is suppressed, the convergence rate and the accuracy are improved. Further, it is applied to the searching of the optimal pipe-routing. It is concluded that the presented approach can quickly find the optimal pipe-routing in 3D rotational space with obstacles, meeting certain engineering constraints. In the future, the intelligent pipe-routing layout of the branch pipe will be studied.

## Acknowledgements

This research is supported by National Natural Science Foundation of China (No. 51205054), Research Fund for Talents of Northeastern University (No. 26333002) and the Fundamental Research Funds for the Central Universities (No. N120403009).

## References

- [1]. Z. Y. Chen, J. Fan, W. Q. Wei, M. Ma, Computer aided routing system for 3-D aircraft engine pipe, *Journal of Propulsion Technology*, Vol. 20, Issue 6, 1999, pp. 58-61.
- [2]. G. Y. Li, Intelligent routing system research of aircraft engine, *Beihang University*, Beijing, 2000.
- [3]. Y. Huang, Z. Y. Chen, Research on the path pattern of the pipe-routing layout for aero-engine, *Aero-Engine*, Issue 4, 2001, pp. 11-14.
- [4]. J. Fan, M. Ma, X. G. Yang, Research on automatic laying out for external pipeline of aeroengine, *Journal of Machine Design*, Vol. 20, Issue 7, 2003, pp. 21-23.
- [5]. J. Fan, Research on MAS based distributed cooperative aeroengine outside pipe system design, *Beihang University*, Beijing, 2003.
- [6]. Q. Liu, C. E. Wang, T. Ren, X. L. Bai, PSO-based pipe routing approach for aircraft engine, *Journal of Northeastern University (Natural Science)*, Vol. 30, Issue 7, 2009, pp. 940-943.
- [7]. X. L. Bai, C. E. Wang, Y. Zhang, Q. Liu, On the automatic route layout for connection of three pipeline terminals, *Journal of Northeastern University (Natural Science)*, Vol. 30, Issue. 2, 2009, pp. 283-286.
- [8]. J. H. Park, R. L. Storch, Pipe-routing algorithm development: case of a ship engine room design, *Expert Systems with Applications*, Vol. 23, Issue 3, 2002, pp. 299-309.
- [9]. T. Ito, A genetic algorithm approach to piping route path planning, *Journal of Intelligent Manufacturing*, Vol. 10, Issue, 1, 1999, pp. 103-114.
- [10]. X. N. Fan, Y. Lin, Z. S. Ji, The ant colony optimization for ship pipe route design in 3D space, in *Proceedings of the 6<sup>th</sup> World Congress on Intelligent Control and Automation*, Dalian, China, 21-23 June 2006, pp. 3103-3108.
- [11]. Y. L. Fu, H. B. Feng, J. X. Sun, R. Li, Y. L. Ma, Automatic pipe-routing particle swarm optimization algorithm in electromechanical products, *Chinese Journal of Mechanical Engineering*, Vol. 43, Issue 11, 2007, pp. 194-199.
- [12]. C. Van der Velden, C. Bil, X. G. Yu, A. Smith, An intelligent system for automatic layout routing in aerospace design, *Innovations in Systems and Software Engineering*, Vol. 3, Issue, 2, 2007, pp. 117-128.
- [13]. S. Sandurkar, W. Chen, GAPRUS-genetic algorithms based pipe routing using tessellated objects, *Computers in Industry*, Vol. 38, Issue 3, 1999, pp. 209-223.
- [14]. K. Fujita, S. Akagi, T. Nakatogawa, Hybrid approach to plant layout design using constraint directed search and an optimization technique, *ASME Journal of Mechanical Design*, Vol. 116, Issue 4, 1994, pp. 1026-1033.
- [15]. X. L. Li, Z. J. Shao, J. X. Qian, An optimizing method based on autonomous animats: fish-swarm algorithm, *System Engineering Theory and Practice*, Vol. 22, Issue 11, 2002, pp. 32-38.
- [16]. Y. Q. Zhou, Z. C. Xie, Improved artificial fish-school swarm algorithm for solving TSP, *Systems Engineering and Electronics*, Vol. 31, Issue 6, 2009, pp. 1458-1461.
- [17]. X. J. Shan, M. Y. Jiang, J. P. Li, The routing optimization based on improved artificial fish swarm algorithm, in *Proceedings of the 6<sup>th</sup> World Congress on Intelligent Control and Automation*, Dalian, China, 21-23 June 2006, pp. 3658-3662.
- [18]. B. Li, W. S. Jiang, Optimization complex functions by chaos search, *Cybernetics and Systems*, Vol. 29, Issue, 4, 1998, pp. 409-419.