Optimization of the main parts of hydroelectric sets using hybrid genetic algorithm

Zhipin Ye[a], Xiaoli Li[b,*], Chuangyin Dang[a]

[a]*Department of Manufacturing Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong*
[b]*Institute of Precision Engineering, Harbin Institute of Technology, Harbin 150001, China*

Received 14 July 1999

Abstract

This paper presents hybrid genetic algorithms to optimize the structure of the main parts of hydroelectric sets based on the finite element method. Firstly, the optimal model of the main parts of hydroelectric sets is established including an objective function and some constraint conditions. Afterwards, the stochastic direction method (SDM) and a genetic algorithm (GA) method are applied to solve the optimal model. Comparing the computational results of the two methods, some features of the methods are pointed out. In order to overcome the disadvantages of the above two methods, a new method, a hybrid genetic algorithm (HGA), is presented, and is used to optimize the structure of the main parts of hydroelectric sets. The computational results show the new method can obtain a global solution with a high search speed compared to the other two methods. © 2000 Published by Elsevier Science S.A.

Keywords: Genetic algorithm; Stochastic direction method; Finite element method; Optimization design

1. Introduction

The structure of hydroelectric set is commonly designed based on the conventional structure computing method, utilizing the experience of designer, handbooks and standards provided. Taking its high safety requirement into account, a large size is selected for the design of the practical structure, which will increase the structure's weight, and waste materials. Moreover, this need not necessarily improve the structure performance of a hydroelectric set. In recent years, some optimization methods have been proposed for the design of the structure of a hydroelectric set. The common optimization methods mainly rely on the gradient value to search for the optimization solution during computing. For some practical design problems, the optimal function is discontinuous or multi-modal, so the derivative of the function cannot be obtained. These general optimization methods often fail when confronted with the above problems. The stochastic direction method (SDM) has been proposed to overcome the above disadvantages, but SDM easily converges to local solution. The genetic algorithm (GA), which is called a heuristic optimization method, has been presented to overcome the local convergence of the

SDM. The GA can outperform conventional optimization techniques on discontinuous, multi-modal and noisy functions [1], but needs a large amount of computation time.

In this paper, a hybrid genetic algorithm (HGA) is presented by combining the SDM and the GA method for overcoming their disadvantages, and is used to design the upper-frame of the main part of a hydroelectric set.

2. Mathematical model

The structure of the upper-frame is shown in Fig. 1, which includes a center frame and four arms. The upper-frame is an important supporting part of the hydroelectric set which fixes the thrust bearings, the guide bearings, the exciter turbine and brake, etc., and supports the total weight of the rotational components, the deadweight and the axial water-thrust, etc.

2.1. Original finite element model

The upper-frame is simply considered as 12 steel plates of different thickness. Fig. 2 shows the grid of the finite element model of the upper-frame. There are 12 element units, 1256 elements, 624 nodes and 3744 degrees of freedom. The four baseboards of the arms are considered as fixed restraint, i.e., its six degrees of freedom are restrained.

* Corresponding author. Tel.: +852-2788-8449; fax: +852-2788-8423.
E-mail address: me150001@cityu.edu.hk (X. Li).
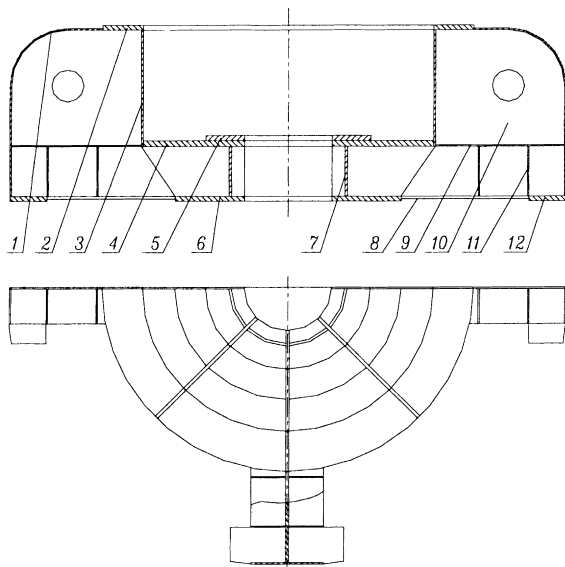
12	baseboard
11	reinforcement
10	reinforcement
9	mid radials
8	lower radials
7	annular reinforcement
6	lower ring
5	antifriction plate
4	annular plate
3	wallboard
2	upper ring
1	upper radials

Fig. 1. The basis configuration of the upper-frame.

Otherwise, the overall loads of about 668 000 N fall onto an anti-friction plate (see Fig. 1).

The static and dynamic analyses of the upper-frame are carried out by means of the finite element software DASAP90, and the maximum strain $\delta_{max} = 0.3151$ mm and the maximum stress $\sigma_{max} = 31.55$ MPa are obtained. The static deformation of the upper-frame structure is shown in Fig. 3. The first 10 orders of the natural frequency of the upper-frame for different values of k are computed to be as follows:

$k = 1$,   frequency $= 121.736$ Hz

$k = 2$,   frequency $= 127.289$ Hz

$k = 3$,   frequency $= 130.073$ Hz

$k = 4$,   frequency $= 130.429$ Hz

$k = 5$,   frequency $= 132.658$ Hz

$k = 6$,   frequency $= 139.669$ Hz

$k = 7$,   frequency $= 139.676$ Hz

$k = 8$,   frequency $= 139.676$ Hz

$k = 9$,   frequency $= 139.677$ Hz

$k = 10$,   frequency $= 198.124$ Hz

and its modes are shown in Fig. 4 (enumerating only four typical modes due to limited space).

The allowable stress σ of the framework is about 110–140 MPa, and the allowable strain δ of the framework is about 0.5–1.5 mm. From the above analytical results, it is found that the upper-frame has abundant strength. Thus, the
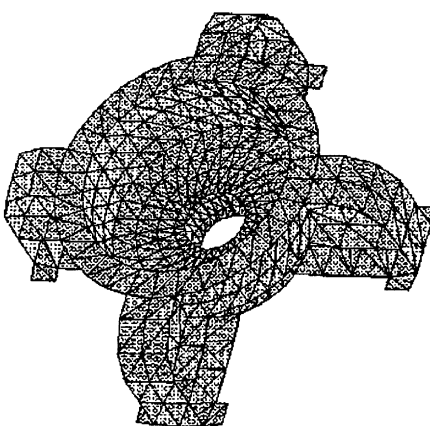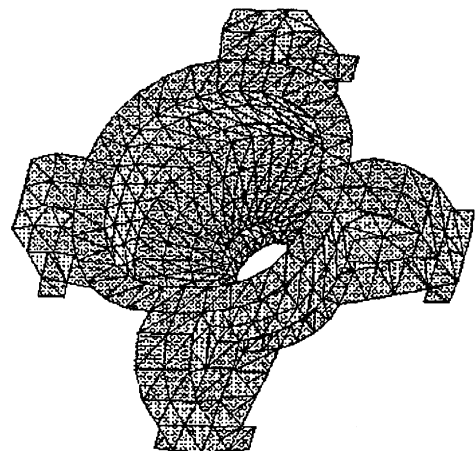

Fig. 2. The grid chart of the upper-frame.

Fig. 3. The static deformation chart of the original scheme.

Mode 2 127.29 Hz

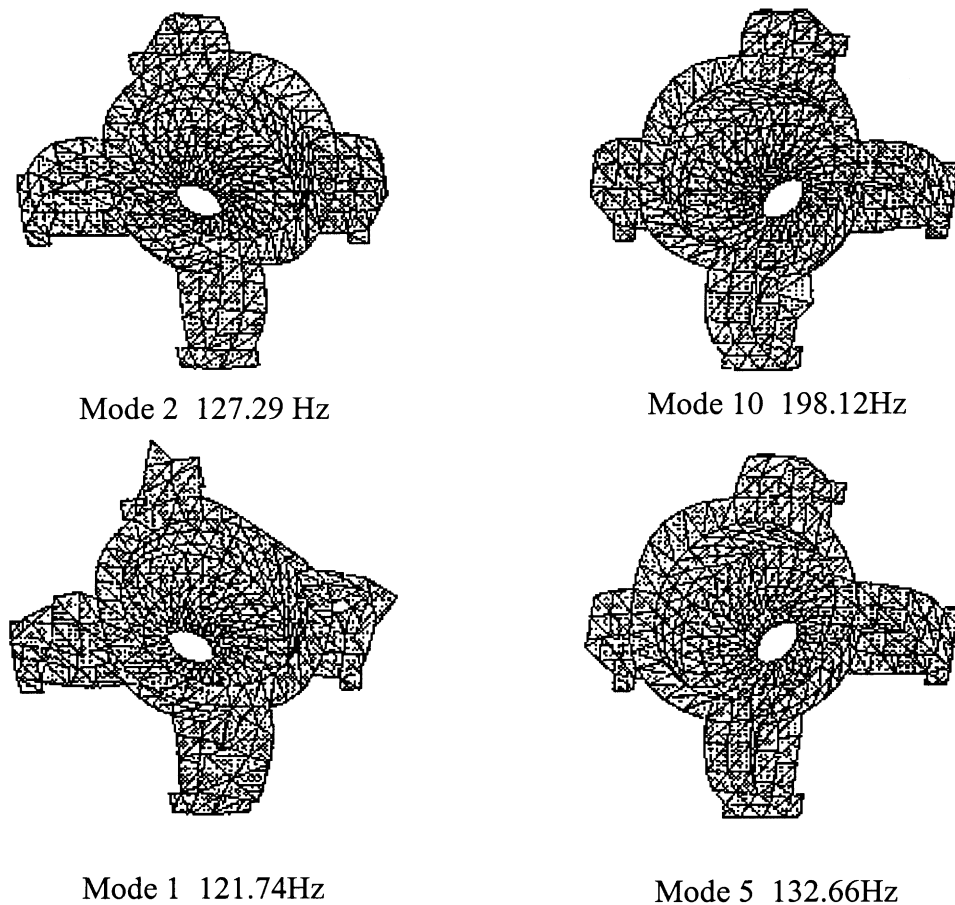Mode 10 198.12Hz

Mode 1 121.74Hz

Mode 5 132.66Hz

Fig. 4. The modes of the original scheme.

upper-frame can be designed as a lightweight structure which satisfies the strength and stiffness conditions. The upper-frame structure will be designed using an optimal algorithm in conjunction with the finite element method.

2.2. Mathematical model

2.2.1. Objective function

The weight of the upper-frame and the thickness of each plate are taken as the objective function and the design variable, respectively. Thus, there are 12 variables, namely, $X = [x_1, x_2, \ldots, x_{12}]^T$. The objective function is as follows:

$$
\begin{aligned}
f(X) = 7850(&1.576024x_1 + 2.6939137x_2 \\
&+ 3.9563333x_3 + 3.243042x_4 + 2.5756x_5 \\
&+ 0.6212x_6 + 1.592x_7 + 1.761604x_8 \\
&+ 0.5881061x_9 + 1.3103583x_{10} + 0.79853x_{11} \\
&+ 1.51372x_{12})
\end{aligned}
\tag{1}
$$

2.2.2. Constraints

For static optimization, the constraints are the strength and stiffness conditions, i.e., both the maximum stress and the maximum strain are less than the allowable values.

Therefore, the optimal mathematical model can be expressed as

$$
\begin{aligned}
&\min \quad f(X) \\
&\text{s.t.} \quad g_1(X) = \frac{\sigma_{\max}}{[\sigma]} - 1 \le 0 \\
&\qquad\quad\, g_2(X) = \frac{\delta_{\max}}{[\delta]} - 1 \le 0
\end{aligned}
$$

3. Static optimization

From the above mathematical model, the constraints are implicit, non-linear, and non-differentiable. SDM and GA have no special demand on the properties of the function, being applied to solve this complex structural engineering problem, the objective being to obtain the maximum strength/weight ratio.

3.1. Stochastic direction method

SDM is a direct method for solving the constraint optimal problem. Its basic principle is shown in Fig. 5, and its main procedure is given as follows.
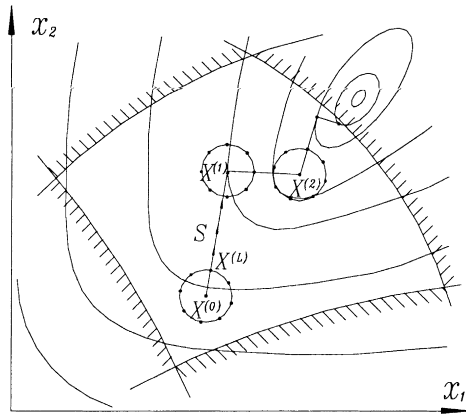
Fig. 5. The basic principle of SDM.

1. Generate an initial point $X^{(0)}$ within the domain.
2. Generate k n-dimensional random unit vector $e^{(j)}$ according to

$$e^{(j)} = \frac{1}{\sqrt{\sum_{i=1}^{n}(r_i^{(j)})^2}} \begin{bmatrix} r_1^{(j)} \\ r_2^{(j)} \\ \vdots \\ r_n^{(j)} \end{bmatrix} \quad (j = 1, 2, \cdots, k)$$

where $r_i^{(j)}$ is a pseudo-random number within $(-1, 1)$.
3. Select a suitable step size α_0, and compute k random points $X^{(j)}$ according to

$$X^{(j)} = X^{(0)} + \alpha_0 e^{(j)} \quad (j = 1, 2, \ldots, k)$$

Obviously, k random points $X^{(j)}$ are distributed on the hypersphere of which the center is $X^{(0)}$ and the radius is α_0; k just represents the k random searching direction.
4. Compute the objective function values of k random points $X^{(j)}$, then the minimum value $f(X^{(L)})$ and the point $X^{(L)}$ are selected from among them, namely,

$$f(X^{(L)}) = \min\{f(X^{(j)})\} \quad (j = 1, 2, \ldots, k)$$

5. If $X^{(L)}$ is feasible and $F(X^{(L)}) < F(X^{(0)})$, the feasible search direction S is determined by

$$S = X^{(L)} - X^{(0)}$$

Moreover, the step $1.3\alpha_0$ is increased along this direction S, then a new point $X^{(1)}$ is obtained as follows:

$$X^{(1)} = X^{(0)} + \alpha(X^{(L)} - X^{(0)})$$

where $\alpha = 1.3\alpha_0$.
6. If $X^{(1)}$ is satisfied with $f(X^{(1)}) < f(X^{(0)})$, continue to increase step 1.3α forward along direction S, or else short step size $0.7\alpha_0$ to search for a new point, until the new point is feasible and its objective function value is less than $f(X^{(0)})$. Afterwards, take the new point for the initial point of the next search. Repeat (2)–(5).
7. Terminate the iterative operation if a solution is found (converged).

3.2. Optimal results of SDM

Because the search direction of SDM is preferably chosen from many directions and the step size can be modified randomly, its speed of convergence is relatively high. To avoid a local solution, the calculation can be made several times with several different initial points. After accurately analyzing these computing results, the best results of all are as follows:

Optim. $X[1] = 0.01552613$ m

Optim. $X[2] = 0.00512837$ m

Optim. $X[3] = 0.00971257$ m

Optim. $X[4] = 0.01460076$ m

Optim. $X[5] = 0.00165203$ m

Optim. $X[6] = 0.00203380$ m

Optim. $X[7] = 0.00574408$ m

Optim. $X[8] = 0.01617512$ m

Optim. $X[9] = 0.05135416$ m

Optim. $X[10] = 0.01723076$ m

Optim. $X[11] = 0.01393475$ m

Optim. $X[12] = 0.01365249$ m

Optim. $FX = 1976.57172450$ kg

Optim. $Str = 6.56817E + 07$ Pa

Optim. $Disp = 0.00049753$ m

where $X[i]$ (i=1–12) is the thickness of the corresponding plate, FX the total weight of the upper-frame, Str the maximum stress, and Disp the maximum strain.

The above computing program is written in language TURBO PASCAL 6.0, and is run on computer COMPAQ 586 CPU166, the total run-time being about 65 min.

3.3. Genetic algorithm

The GA is an adaptive method which searches the solution space of a function through the use of simulated evolution, i.e., the principle of natural selection and the survival-of-the-fittest strategy. GA differs from traditional search techniques. GA manipulates a population of individuals, each representing a possible solution to a given problem [2]. Each individual uses probabilistic genetic-like operators such as selection, crossover and mutation to produce a new population which contains a higher proportion of the more fit individuals. In this way, the GA moves away from unpromising areas of the search space and towards promising areas [3].

The use of GA requires the determination of four fundamental features: a suitable coding (or representation), a fitness function, genetic operators, and initialization

and convergence [4]. These aspects are described in the following.

3.3.1. Coding

A possible solution to the above problem may be represented as a set of parameters, which is the thickness of the upper-frame in the structure design. These parameters (known as genes) are joined together to form a string which is often referred to as a chromosome. The problem is to minimize a function including 12 variables, $f(x_1, x_2, \ldots, x_{12})$, where each variable might be denoted by a 6-bit binary number. The chromosome would, therefore, contain 12 genes, and consist of 72 binary digits.

It is assumed that thickness of each plate is between 0.001 and 0.064 m and each has 64 possible values (0.001 and 0.064 are the variable upper and lower bounds). A 6-bit string 000000 represents 0.001 m thickness, and a 6-bit string 111111 represents 0.064 m thickness: for the other 62 six-bit strings code the rest of the values 0.002–0.063. For example, a certain chromosome string of this upper-frame is shown as follows:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
010100	000111	001011	100101	000101	000001	100000	110011	111100	010101	101010	001100

In this 72-bit string, the first 6-bit codes denote the first thickness x_1 of the plates, whilst the rest may be deduced by analogy, the last 6-bit codes denoting the 12th plate's thickness x_{12}. For example, the first thickness of the plate codes is 010100, which corresponds to 0.021 m, the second plate's thickness codes is 000111, which corresponds to 0.008 m thickness, and the rest of the 10 thickness of plate may be represented with binary digits analogously.

3.3.2. Fitness function

A fitness function must be devised for each problem to be solved. In this upper-frame design problem, the objective function is directly transferred into a fitness function. The upper-frame design is a constrained minimization problem, but it can be transformed into an unconstrained maximization problem by using a penalty function, its expression being as follows:

max $F(X)$

where $F(X)$ is the fitness function, which is computed by

$$F(X) = \frac{1}{1 + (1.1)^{f(X)}} P(X)$$

where $f(X)$ is the objective function, and $P(X)$ a penalty function, defined as

$$P(X) = \frac{1}{(1.1)^{\Phi(X)}}, \quad \Phi(X) = \sum_{u=1}^{2} |g_u(X)|$$

where $g_u(X)$ is a constrained function.

3.3.3. Genetic operators

Genetic operators provide the basic search mechanism of the GA. The operators can create new generations, which are the more fit individuals. These include three basic operators: selection, crossover and mutation.

Selection adopts an expected value method, herein being shown as follows.

1. Compute the expected value:

$$e_i = \frac{nF_i}{\sum_{i=1}^{n} F_i}$$

   where F_i equals the fitness of individual i, and n equals the population size.
2. The integer part of e_i is assigned as the selected times of individual i, the individuals being sorted according to the decimal part of e_i, and selected from large to small, until the population size is n.

Crossover is an important operator in GA, which can improve the search power of GA by leaps and bounds. Herein the uniform crossover is selected because the uniform crossover is more robust and beneficial if the population size is small.

Mutation is applied to a chromosome individually after crossover. In this paper, a new parameter AN is used which is the allowable number of the same individual in a population. If the number SN of the same individual in a population is greater than or equal to AN, the mutation is taken as a useful operation.

3.3.4. Initialization and convergence

GA generates an initial population randomly, then it moves from one generation to the other generation selected and reproduced by parents, until its convergence criterion is met. A specified maximum number M of generations is used to stop the above processes. Since it must load the finite element module programs when computing fitness, and their computing time is considered, the population size n is equal to 30, and the maximum generations number M is equal to 30.

The GA creates an initial generation, $G(0)$, and from each generation $G(i)$, the GA generates a new one, $G(i + 1)$. The main processes of the GA are given as follows.

```
BEGIN /* genetic algorithm */
    Initialize parameters (n=30, M=30, AN=10);
    Generate initial generation G(0);
    Compute fitness of each individual;
    For i=1 to M do
      Begin
        Selection operation from generation G(i − 1);
```

Compute SN; /* the number of same individual in population */
If SN ≥ AN then Mutation operation;
Crossover operation;
Produce next generation $G(i)$;
End;
Print out best solution found;
END.

3.4. Optimal results of GA

Applying GA to optimize upper-frame, the optimal results are as follows:

Optim. $X[1] = 0.013$ m

Optim. $X[2] = 0.001$ m

Optim. $X[3] = 0.003$ m

Optim. $X[4] = 0.020$ m

Optim. $X[5] = 0.003$ m

Optim. $X[6] = 0.002$ m

Optim. $X[7] = 0.003$ m

Optim. $X[8] = 0.005$ m

Optim. $X[9] = 0.056$ m

Optim. $X[10] = 0.020$ m

Optim. $X[11] = 0.010$ m

Optim. $X[12] = 0.020$ m

Optim. FX $= 1725.94878300$ kg

Optim. Str $= 1.123094E + 08$ Pa

Optim. Disp $= 0.00049967$ m

where $X[i]$ ($i=1$–12) is the thickness of the corresponding plate, FX the total weight of upper-frame, Str the maximum stress, and Disp the maximum strain.

The GA program is written in language TURBO PASCAL 6.0 and run on computer COMPAQ 586 CPU166MHZ, where the total run-time is 4650 min.

3.5. Comparison of the results of SDM and GA

Comparing the run conditions and computing results, SDM can be referred to as a version of *hill-climbing*. It starts from a random point in the search space, and if it can obtain a higher point in the computing processing, it is accepted. Clearly, SDM starts at different starting points, and obtains different optimum solutions. In addition, the SDM possibly locates on the boundary of the search space, but during computing it infrequently load finite element modules, so it takes less time.

GA, by comparison, can provide a more accurate solution but take more run-time. GA starts from many points, a population of individuals, to search for the solution. GA evaluates and operates on several solutions simultaneously, and it can gather information from many current points to directly search. This factor makes GA less susceptible to the problems of local maximum and noise. However, its run-time is proportional to the population size n and the maximum generations number M. For a complicated structure, one static analysis requires about 5 min, as many nodes and elements are divided, so the run-time of GA is very long.

4. Dynamic optimization

In recent years, structure design has translated from the traditional static design to dynamic optimal design, the designed structure not only meeting static demands but also possessing a preferable dynamic property. Herein is developed a dynamic optimal design based on the total minimum weight of the upper-frame, where the designed result can satisfy both static constraints (such as stress and strain) and dynamic constraints (such as avoiding resonance).

The mathematical model, variables, objective function and static constraints are the same as in the above static optimization, but a dynamic constraint is added, i.e., all orders of natural frequency should keep away from the driving base frequency (50 Hz) and its frequency multiplication. It is embodied that the base frequency of the upper-frame should not be less than $50 \times 1.4 = 70$ Hz, so that each order natural frequency ϖ should satisfy the equation

$$50r + 10 \leq \varpi \leq 50s - 10$$
$$(r = 1, 2, 3, \ldots, s = 2, 3, 4, \ldots)$$

Therefore, the mathematical model with static and dynamic constraints is given as follows:

$$\min \quad f(X)$$
$$\text{s.t.} \quad g_1(X) = \frac{\sigma_{\max}}{[\sigma]} - 1 \leq 0$$
$$g_2(X) = \frac{\delta_{\max}}{[\delta]} - 1 \leq 0$$
$$g_3(X) = 1 - \frac{\varpi_{\min}}{70} \leq 0$$
$$50r + 10 \leq \varpi \leq 50s - 10 \quad (r = 1, 2, 3, \ldots, s = 2, 3, 4, \ldots)$$

The objective function $f(X)$ can be transformed into a fitness function according to Section 3.3. During the practical computing, only the first five orders natural frequency is calculated and made outside the range [0–70 Hz], [90–110 Hz] and [140–160 Hz]. Obviously, this dynamic optimization problem is a multi-modal function. For this kind of problem, SDM easily immerses in an improper search area, and it is very difficult to reach global optimum, but one major drawback of GA is low efficiency, and the time of dynamic analysis is twice as much as that of static analysis. Combining the advantages and disadvantages of the above two techniques, the authors present a new HGA which preserves the precision of results, while reducing run-time.

To design a new hybrid method, GA is integrated with SDM effectively, combining the merits of the two original algorithms. The initial populations of HGA can be obtained by SDM. Namely, first n feasible starting points are randomly chosen, from which n optimum results are obtained via SDM, then these n results are taken as initial population of GA.

Applying the new HGA to dynamically optimize the upper-frame, the optimal results follow:

Optim. $X[1] = 0.019$ m

Optim. $X[2] = 0.009$ m

Optim. $X[3] = 0.008$ m

Optim. $X[4] = 0.021$ m

Optim. $X[5] = 0.007$ m

Optim. $X[6] = 0.010$ m

Optim. $X[7] = 0.010$ m

Optim. $X[8] = 0.020$ m

Optim. $X[9] = 0.057$ m

Optim. $X[10] = 0.019$ m

Optim. $X[11] = 0.012$ m

Optim. $X[12] = 0.020$ m

Optim. FX $= 2564.56168620$ kg

Optim. Str $= 5.905293E + 07$ Pa

Optim. Disp $= 0.00037959$ m

Optim. Fre1 $= 85.81517104$ Hz

Optim. Fre2 $= 87.04540734$ Hz

Optim. Fre3 $= 87.04755860$ Hz

Optim. Fre4 $= 88.19955590$ Hz

Optim. Fre5 $= 110.12744290$ Hz

where $X[i]$ (i=1–12) is the thickness of the corresponding plate, FX the total weight of the upper-frame, Str the maximum stress, Disp the maximum strain, and Fre1–Fre5 represent the first five orders natural frequency. The HGA program is written in language TURBO PASCAL 6.0 and run on computer COMPAQ 586 CPU166MHZ, and the total run-time is 4770 min.

5. Comparison: original scheme and optimal scheme

To meet the international standard specified for the plate, the final optimal scheme is obtained as follows:

$X[1] = 0.020$ m, $X[2] = 0.009$ m

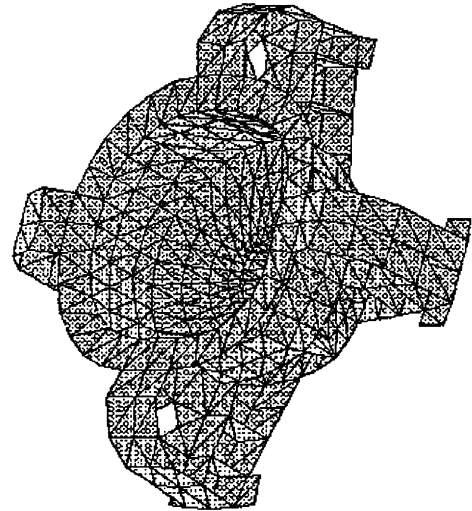$X[3] = 0.008$ m, $X[4] = 0.020$ m

$X[5] = 0.007$ m, $X[6] = 0.010$ m

Fig. 6. The static deformation chart of the final optimal scheme.

$X[7] = 0.010$ m, $X[8] = 0.020$ m

$X[9] = 0.055$ m, $X[10] = 0.018$ m

$X[11] = 0.012$ m, $X[12] = 0.020$ m

The static and dynamic analysis of the optimal scheme is also carried out by means of finite element software DASAP90 so as to enable comparison with the original scheme. The static deformation structure of the optimal upper-frame is shown in Fig. 6, and its modes are shown in Fig. 7 (enumerating only four typical modes). The comparison of results of the different schemes is shown in Table 1.

The maximum stress and maximum strain of the optimal upper-frame scheme are both greater than those of original scheme, but its strength and stiffness can meet the requirements and it has a substantial weight reduction. After dynamic optimization, the vibration performance also attains the design goal. The relatively optimal scheme possesses good structure performance, which indicates that the optimal structure design given in this paper is successful. Although there is some difference between the run-time of GAs and HGAs, the efficiency of the HGA is much greater than that of GA, since dynamic finite element analysis often spends double the time of static analysis. From Table 1, it can be established that for the same population size, the general GA needs to iterate 30 generations to converge, while the HGA needs only 16 generations.

6. Conclusions

This paper successfully presents a new optimal algorithm in conjunction with a finite element module to design a lightweight upper-frame. The final optimal scheme gains a weight reduction of 28.64%, which provides a reliable scientific basis for improving the carrying capacity of a product and new product development.
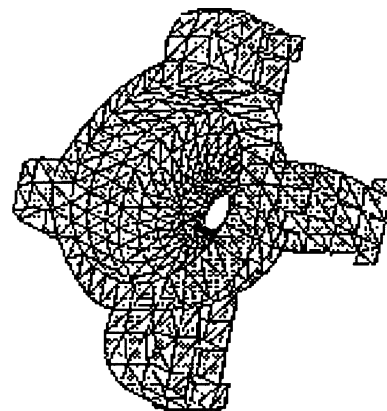
Fig. 7. The modes of the static deformation structure of the optimal upper-frame.
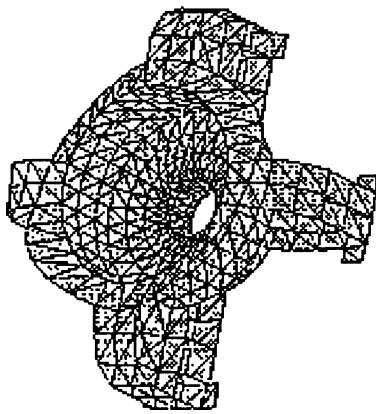
Mode 3 87.66Hz
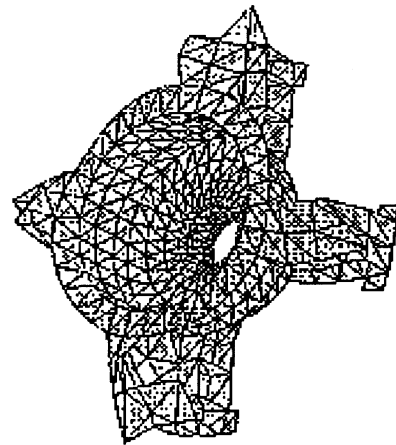
Mode 5 110.30Hz
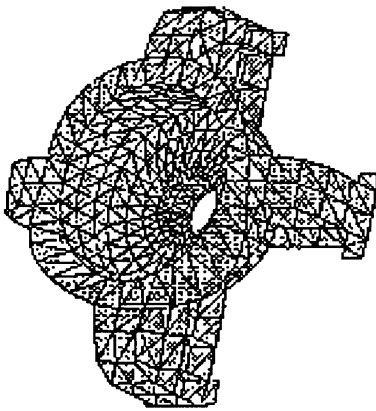
Mode 1 87.00Hz

Mode 4 88.90Hz

Table 1

Factors	Original scheme	Optimal scheme			
		Static optimization of SDM	Static optimization of GA	Dynamic optimization of HGA	Final revision scheme
Run-time (min)		65	4650	4770	
Population size, n			30	30	
Maximum generations number, M			30	16	
Maximum strain (mm)	0.3151	0.4975	0.4997	0.3796	0.38645
Maximum stress (MPa)	31.55	65.68	112.31	59.05	42.75
First-order natural frequency (Hz)	121.74			85.82	87.00
Second-order natural frequency (Hz)	127.29			87.04	87.01
Third-order natural frequency (Hz)	130.07			87.05	87.66
Fourth-order natural frequency (Hz)	130.43			88.20	88.90
Total weight of upper-frame (kg)	3558	1977	1726	2565	2539
Weight reduction (kg)					1019
Proportion reduction (%)					28.64

References

[1] J.S. Chung, S.M. Hwang, Application of a genetic algorithm to the optimal design of the die shape in extrusion, J. Mater. Process. Technol. 72 (1) (1997) 69–77.

[2] F.Y. Cheng, D. Li, Multiobjective optimization design with Pareto genetic algorithm, J. Struct. Eng. 123 (9) (1997) 1252–1261.

[3] C. McCulley, C.L. Bloebaum, Genetic tool for optimal design sequencing in complex engineering systems, Struct. Opt. 12 (2–3) (1996) 186–201.

[4] C.-Y. Lin, P. Hajela, Design optimization with advanced genetic search strategies, Adv. Eng. Software 21 (3) (1994) 179–189.