ICAS 2021 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021 Pudong Shangri-La, Shanghai, China

Bionic light-weight grid surface wing structure design for a solar-powered UAV

Ding You[1, 2], Zhou Zhou[1,2], Liu Hongjun [3]

[1]College of Aeronautics，Northwestern Polytechnical University, Xi'an Shaanxi, 710072, P.R.China.
[2]Yangtze River Delta Research Institute, Northwestern Polytechnical University, Taicang Jiangsu, 215400, P.R.China
[3] Science and Technology on UAV Laboratory, Northwestern Polytechnical University, Xi'an 710065, China;

Abstract

A two-step implicit cellular-based evolutionary topology optimization method is implemented for the plane-like structure design. In the first step, a homogenization method based on the ground structure is implemented to get the materials distribution data from the design domain. In the second step, a graph-based interpretation for the bionic path searching is applied to the cellular structure topology generation. A heuristic alternative to this approach couples a genetic algorithm with the split point finding rules in the hierarchical structure system, which encodes design variables and governs the development of the structure when coupled with an interpreter to translate genomic information into structural topologies. The framework allows a short number of design variables to express the number of feasible solutions within the design spaces. A detailed solar-powered UAV wing structure design is demonstrated at last for this method implementation.

Keywords: Bio-inspired; Topology optimization; Ground Structure; Evolutionary algorithms; Mathematical graph;

1. Introduction

Solar-electrically powered fixed-wing Unmanned Aerial Vehicles (UAVs) have extremely structure optimization demands to improve the time of endurance. As the solar cell packaging technology, improved which made these solar cell modules become much more flexible than before. These changes in manufacturing caused a flexible surface structure that may fail before the wing had got the maximum tip displacements due to the local buckling on the compressive side. So it is challenging to minimize the structure weight as well as improve the performance of the structure [1].

Different from the usual optimization method, there are several special features in this paper's layout design. Firstly, the whole solar plate is too flexible, so the stiffener layout should be strong enough to avoid the external load. Secondly, in order to make the whole surface take the external load, the structure should be continuous and have redundancy to avoid uncertain destruction. It would makes this stiffener layout more like a grid surface structure that has well-designed connectivity to maintain the shape of the surface and meanwhile stiffness enough to suffer the load. Thirdly, the uncertain distribution and demanding structural weight requirements make this optimization will not only a topology optimization but also a shape and size optimization and neither one of these optimization methods can well solve it directly. So the balance to combine these methods is the key to solve this grid layout pattern.

Nature structures have various kinds of this surface structure, like insect wings, plant leaves, and diatom shells. These structures fully use the materials and have enough strength to protect themselves from outside damage. To take full advantages of these nature structure, a hierarchical optimization strategy of composite curved-wall wing structure design which based on Lindemayer system (or map L system) and the Ground Structure Method is proposed. A genetic algorithm (GA) optimizes the details of the developmental program with a small number of parameters.

2. Bionic hierarchical optimization strategy

The following sections provide a brief overview of the map L-System, homogenized operations to the topology data from the Ground Structure (GS) method and a Graphic structural topology optimization framework is presented in the end.

2.1 Parameterized L-System approach

Binary Propagating Map 0L-systems with markers shorted by mBPM0Lsystems is a variant of map

L systems and widely used in the structure topology design[2]. The word binary means a two-part splitting operation applied in the next new iteration during propagating. Markers specify points at the edges for cellular division and are analogous to attachment sites for division walls during mitosis. There are three main components to the L-system: an alphabet Σ, axiom ω, and a set of production rules P. General mBPM0L-systems based methods produce the structure topology by the parameterized encoding of the production rules with members in the alphabet. A general rule expression with a usual initiate axiom as $\omega_0=\omega_1\omega_2$ is shown below:

$$P_i : \alpha_i \rightarrow \lambda_1^i \lambda_2^i \ldots \lambda_{N*M}^i, \quad i = 1 \ldots j \tag{1}$$

Where α_i is the member of variable letter characters, for example α = {A, B, C, D…}, these letters in α represent specific command to graphic operation. N is the number of variable characters that used to provide a kernel for the full topology generation string in the next iteration. M is the minimum necessary character number to express a single graph action (translation, rotation, and instantaneous placement). λ_k^i is chosen from the different components in alphabet Σ, like variable letter characters α, constant characters β = {+, _,—}, γ = {[,] }, and etc.. A detailed explanation can be found in [3] and [4]. Once the encoded L-system instructions are generated, the corresponding optimization framework interprets the instructions and performs graphical operations toward the creation of the structure.

2.2 Ground Structure Method

In this part, a formulation called GRAND proposed by [5-6] is studied to get the structure topology data rapidly. The linear programming problem is simplified as:

$$\begin{aligned} \min_{\mathbf{s}^+,\mathbf{s}^-} \quad & V^* = \frac{V}{\sigma_T} = \mathbf{l}^T\left(\mathbf{s}^+ + \kappa \mathbf{s}^-\right) \\ \text{s.t.} \quad & \mathbf{B}^{\mathbf{T}}\left(\mathbf{s}^+ - \mathbf{s}^-\right) = \mathbf{f} \\ & s_i^+ \,,\; s_i^- \geq 0 \end{aligned} \tag{2}$$

With V the truss' volume, $\mathbf{B}^{\mathbf{T}}$ is the nodal equilibrium matrix of size. $\kappa=\sigma_T/\sigma_C$, $\mathbf{l}^T$ is consist of l_i which is length of the ith truss member. f is the nodal forces. $s_i^+$and $s_i^-$are the slack variables and only one of s_i^+ and s_i^- is non–zero. The member is in tension if $s_i^+ > 0$, $s_i^- = 0$, and in compression if $s_i^- > 0$, $s_i^+ = 0$.

Using matrix notation, (2) can be rewritten as Eq. (3) and can be solved more efficiently by using the interior-point algorithm:

$$\begin{aligned} \min_{\mathbf{s}^+,\mathbf{s}^-} \quad & V^* = \frac{V}{\sigma_T} = \underset{1\times 2N_b}{\left\{\mathbf{l}^T \quad \kappa \mathbf{l}^T\right\}^T} \underset{2N_b\times 1}{\begin{Bmatrix}\mathbf{s}^+ \\ \mathbf{s}^-\end{Bmatrix}} \\ \text{s.t.} \quad & \underset{N_{dof}\times 2N_b}{\left[\mathbf{B}^{\mathbf{T}} \quad -\mathbf{B}^{\mathbf{T}}\right]} \underset{2N_b\times 1}{\begin{Bmatrix}\mathbf{s}^+ \\ \mathbf{s}^-\end{Bmatrix}} = \underset{N_{dof}\times 1}{\mathbf{f}} \\ & s_i^+ \,,\; s_i^- \geq 0 \end{aligned} \tag{3}$$

Here, N_n, N_b and N_{sup} are the number of nodes, truss members and components with supports, respectively, and N_{dof} is the free nodal components with $N_{dof}= n\bullet N_n - N_{sup}$, for n= 2 in a two-dimensional ground structure and n= 3 in three-dimensional.

3. Topology data homogenization

Although GS method can get an optimal topology result at a very fast speed, but the result as Fig.1 (b) often needs to have a post processing. And due to the single load case adaptability, the optimal structure has certain limitations to the multiple loading cases. So a homogenize operation is performed to utilize the GS output topology data and combine these data with the graphic operation to realize a bionic thin-wall structure design both suitable for the multiple loading cases and multi-objective.

As the output data from the GS method, each bar's cross-sectional area and end location are utilized. The main concept to the homogenize operation is to disperse the material of the optimal retained bar members to their affected area. Fig.2 is the brief description of the search operation and homogenize operation. The detailed steps are as follows:

1) Define the influence area Ω_i determined by the bar's two ends location;

2) Find the nodes in the influence area and establish a relationship between the distance and the bar materials as Eq. (4).

$$W_j^i = \frac{S_{ij}}{S_{\text{search}}} \bullet A_j \tag{4}$$

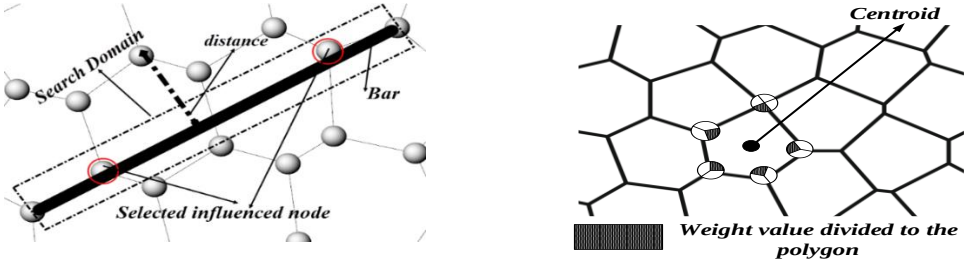
Here, W_j^i means the weight influence value of the jth bar to the ith node, S_{ij} means the distance between the ith node to the jth bar. A_j means the cross-section aero of the jth bar.

3) Data summation and normalization processing for every node in the initial base mesh and define the weight value as w_i^e by Eq. (5):

$$w_i^e = \left(\sum_{j=1}^{m} W_j^i \right) / (w_i^e)_{\max} \tag{5}$$

Here, m means the bars' number that the ith node connected, W_{ji} means the weight influence value of the jth bar to the ith node, (w_i^e)max means the max weight value from the bars to the ith node.

4) Homogenize the weight influence value from base mesh polygon vertex to polygon centroid. This step is to make the irregular distribution value become uniformly distributed. The weight value is evenly divided according to the number of edges to the vertex, then every polygon vertex weight value is transferred to the centroid and divided by the entire polygon area to obtain the final centroid weight value.



(a) Search operation (b) homogenize operation

Fig.1. Schematic allocation diagrams to the search and homogenize operations

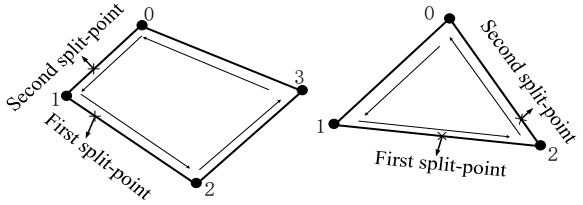
3.1 Parameters definition

Table 2 Design Variable (Genetic code)

Iteration	Variable name	Variable type	Variable range
k=1	First split location compute α_1^1	Real	[0,1]
	Second split location compute α_2^1	Real	[0,1]
k>1	Boundary model judge γ^k	Real	[0,0.66]
	First split location compute α_1^k	Real	[0,1]
	Second split location compute α_2^k	Real	[0,1]

Here, for the first iteration k=1(k is the number of iteration), the boundary is the segments with no weight value assignment. So, the combination of variables which encoded by (α_1^1, α_2^1) is used to define the two split-points' location. And to well interpret the encoded command, a data structure that contains a record of each node, edge, and face (domain) of a planar graph. The number of the boundary are named in a counterclockwise direction as Fig. 4 shows. The floor function $\lfloor x \rfloor$ which will return a greatest integer less than x is used to get an integer value. From equation $\lfloor x \rfloor = \lfloor (N_{sub} + 1) \times \alpha_1^1 \rfloor$, which determines the number of nodes moves from the initial node in a counterclockwise direction. From the value y which calculated by y=x-⌊x⌋, the first split-point location y will be created in between the current node and next node. Here N_{sub} is the number of the boundary. For the second split-point, the same operation is implemented with N_{sub}-1 (remove the segment with first split-point) and α_2^1. In Fig. 4, assume that the node 0 is the initial node, with encoding (0.24, 0.15). The first split-point is calculated by N_{sub}=4 and α_1^1=0.24, by $\lfloor x_1 \rfloor = \lfloor (4+1) \times 0.24 \rfloor = 1$, the first split-point location is on the edge[1,2] and y1= (4+1)× 0.24-$\lfloor (4+1) \times 0.24 \rfloor$=0.2 in between node 1 and node 2. As for second split-point which is calculated by N_{sub}=3 and α_1^1=0.2. $\lfloor x_2 \rfloor = \lfloor (3+1) \times 0.15 \rfloor = 0$. The position is located on edge[0,1], and y2= (3+1)× 0.15-$\lfloor (3+1) \times 0.15 \rfloor$=0.6 in between node 0 and node 1. The same process are worked to the

triangle-like boundary by the encoding (0.4, 0.4). The results are $\lfloor x_1 \rfloor$=1 and y1=0.6, and $\lfloor x_2 \rfloor$=1 and y2=0.2.



(0.24,0.15) (0.4,0.4)

Fig. 4. Examples for the code interpretation in the initial iteration

After the first iteration γ^k is used to judge the number of split-points that belong to the Initial-Boundary segment besides the first iteration. Fig. 5 gives 6 polygon cases from (a) to (f) when the subdomains' partly boundary is composed of Initial-Boundary segment. In these particular cases the boundary segments are classified in two types: Split-Edge segment and Initial-Boundary segment. The encoding is similar to the first iteration as γ^k (α_1^k, α_2^k), the total number of these variables is 3×k, and this means different encoding variables are used for different iteration. α_1^k, α_2^k are two split location compute value similar as α_1^1, α_2^1 in the first iteration. In this encoding type, different polygon case has its own discrete value set for γ^k. This makes a changeable value range for continuous variable λ to represent corresponding discrete value γ^k.

Define five basic conditions to this code interpretation with two split-point values α_1^k, α_2^k:

① Only one Initial-Boundary segment and only one split-point needed. Define $\alpha_v^k = (\alpha_1^k+\alpha_2^k)/2$, and use this value to calculate the split-point location.

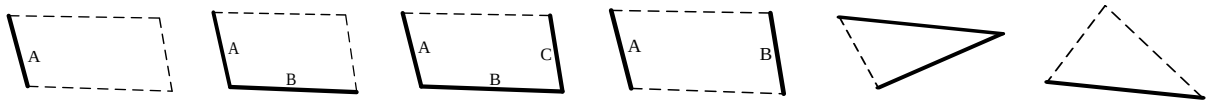
② Two Initial-Boundary segment and two split-points needed. In this situation, using α_1^k, α_2^k to calculate the split-point locations, respectively.

③ Two Initial-Boundary segment and only one split-point needed. A similar operation to the first split-point finding in the first iteration is conducted under the value α_v^k.

④ Three Initial-Boundary segment and two split-point needed. The operations to the split-points finding are based on the code interpretation to the first iteration.

⑤ No split-point is not on the IS or one split-point is on the Split-Edge segment. For the previous situation, the two split-points are got from the previous two maximum weight values from two different split-edges. And for the other situation, the split-pint is the seed with maximum weight value from the corresponding Split-Edge segment.

Different combination for the two split nodes finding is implemented in Fig. 5. For example, in Fig.5 (c), γ^k=1→0≤λ<0.33, and the first split-point is selected by ④ and the second split-point is selected by ⑤. And γ^k=2→0.33≤λ≤0.66, the two split points is selected both by ④.



(a) (b) (c) (d) (e) (f)

Fig.5 Schematic diagram for the combination with two boundary type

3.2 Path search algorithm

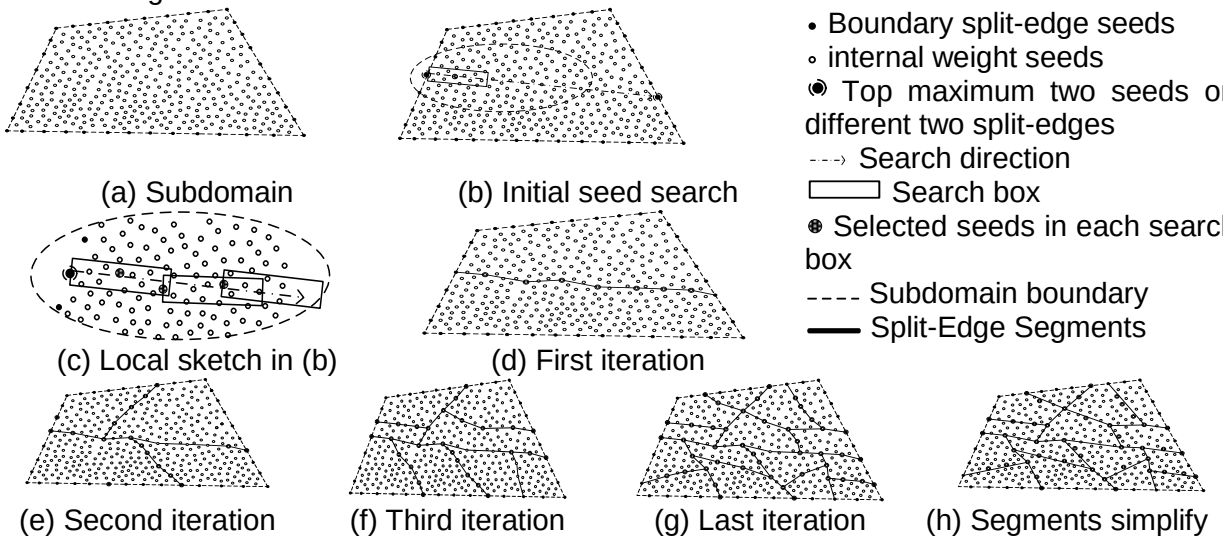
The principle of search algorithm for each component and a propagating process are specified in the following:

a) Two split-points finding: This operation is performed in each iteration after the code interpretation.

b) Max-weighted seed influenced: The searching path was influenced by the max-weighted node along the search domain of the two split-points. This max-weighted seed is regarded as the control-point of the search path curve.

c) Seeds finding along the path between the two split-points: With the determination of the two split-points, the path connected to these two split-points will be regarded as the search direction as Fig. 6(b) shows. A series of search-boxes are established along the search direction to separate the long split-path into several segments and the seeds filtered from the search-boxes criteria will become the candidate for the next iteration's split-point selecting. From Fig. 6(c), the direction for each search box is started at the seed selected from the previous search box (the initial box start at the first split-point) and ended at the second split-point. The judgements based on weight value, angle to the initial path direction and distance make each search box with one seed export. If no suitable seed is in the search

box domain, an operation for the scope extension is applied for the search box until find the suitable seed. Fig. 6(d)-(g) are the front four iterations. Fig. 6(h) is a “line” operation to turn the polyline with no branch to straight line.



(a) Subdomain (b) Initial seed search (c) Local sketch in (b) (d) First iteration

(e) Second iteration (f) Third iteration (g) Last iteration (h) Segments simplify

Fig. 6. Path search algorithm schematic diagram and first four steps in an example cellular-division process

4. Wing structure topology optimization

In this section, our topology optimization method is implemented for a wing structure with curved internal members design. For a wing skin design variables, the half span is 3m and the chord is 0.4m, respectively. A constant cross-section is regarded during each iteration for the topology growth propagated. Fig.7 is a brief introduction for the process of wing structure topology optimization. Fig.7 is the base mesh for the wing structure, and aerodynamic force is considered in this step with 3.5 times overload under the most dangerous situation. The GS result and the final design domain are shown in the GS module. The material properties used for the unidirectional carbon fiber is in Table 4. And in this paper the composite material is entered as engineering constants and shell element in ABAQUS. An optimization software known as SIMULIA ISIGHT is used to integrate an automated data flow processes to multi-programs. As Fig.7 shows, the variables are produced from the “optimizer” and through the encoding and interpretation, a cell-like graph is propagated until satisfy the terminal condition. The structural members of each design iteration are then sampled from the edges of the graph. In this process, a mapping operation from plane graphic to structural geometry is worked under the algorithm established in the Rhino CAD program. A static linear analysis is conducted, and after analysis completed, an ABAQUS postprocessor is implemented to evaluate the fitness of each structure based on multiple objective functions (i.e., weight, stiffness, displacement, force, etc.).

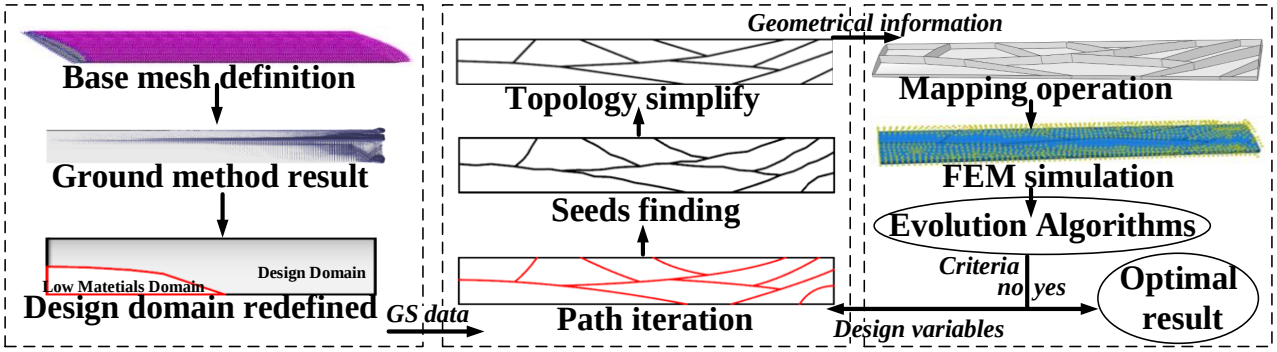


Fig. 7 A flow chart of curved stiffener layout design

To compare the optimization result from the proposed algorithm, a traditional wing structure with three spars at the 15%, 45%, and 60% chords is established. Additionally, there were 9 ribs spaced at 300 mm intervals from the wing root. Finally, the wing skin in front of the 15% spar and behind the 60% spar was removed, leaving only the wing-box, which takes all the flight loading. Additionally, besides the front spars and the rear spars have constant thickness and ply orientations for both wing designs, the other spars in our method have different thicknesses in the different cell-level. The Pareto frontier associated with the multi-objective optimization described in Table 5 is shown in Fig. 8 and these

topology features associated with a series of feasible solutions, which have been shown mathematically to be optimal in the sense of maximizing stiffness per unit mass. The von mises stress distribution and the tip deflection is shown in Fig. 9.A comparison between the curved cell-division composite wing and the traditional type wing, the main variables that were compared were the mass, normalized buckling load, twist stiffness, maximum stress and tip deflection is in Table-7.

Table 4 Composite material engineering constants.

Material Properties	**Carbon fiber**
Longitudinal stiffness, E_1 [N/mm^2]	134,000
Transverse stiffness, $E_2 = E_3$ [N/mm^2]	7900
Shear stiffness, $G_{12} = G_{13}$ [N/mm^2]	4620
In-plane shear stiffness, G_{23} [N/mm^2]	3200
Poisson's ratio, $v_{12} = v_{13}$	0.33
Poisson's ratio, v_{23}	0.41
Density, ρ [kg/m^3]	1,590

Table 5 Specification of the multi-objective problem for curved spars optimization

Design problem statement	
Minimize	Normalized mass
maximize	Normalized stiffness & Normalized Buckling Load
Max iteration number *k*	4
Min split area	100000mm^2
Max total length	7m
By varying	(2+3*4)=14 variables
twist angle φ	$[-3°, 3°]$
Maximum stress limit	900 MPa
Maximum tip displacement limit	300 mm
Tsai-Wu failure criteria	<1
NSGA-II parameters	
40 members for 30 generations	
Cross-over probability =0.9, η =20	
Mutation probability= 0.01	

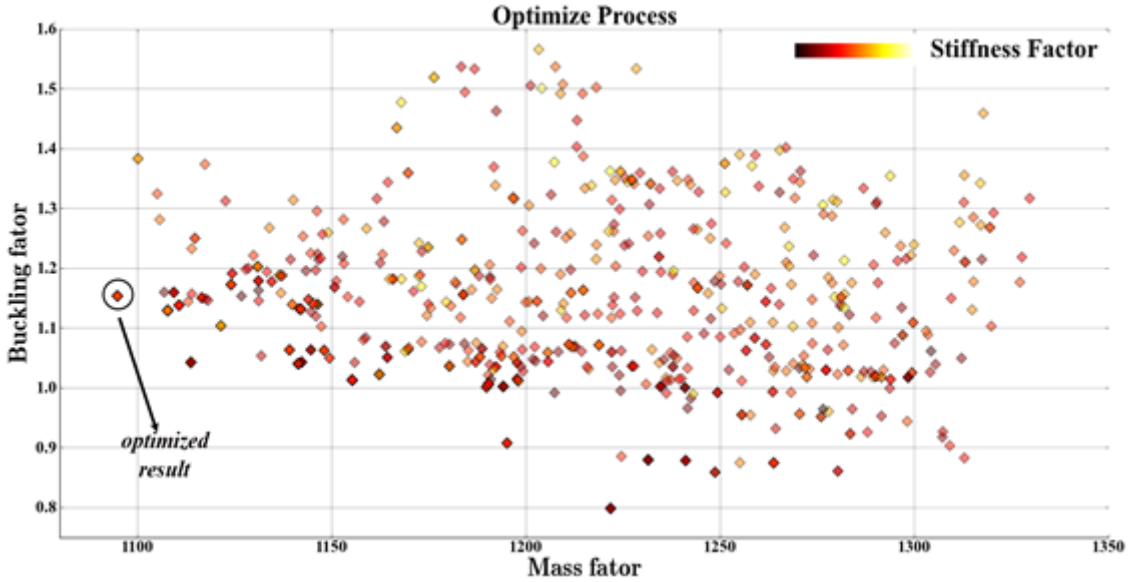


Fig.8 Multi-objective optimization result for the curved spars wing structure

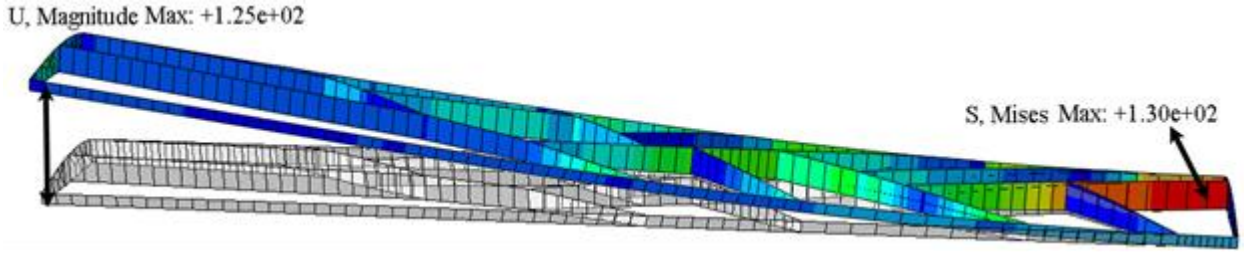


Fig 9 Von Mises stress distribution for carbon fiber composite wing. There is a high stress area near the wing root with high level spar thickness, and a tip deflection is also proposed to reflect a bending stiffness

Table 6 Comparison between the traditional type and optimized curved cell-division type

Property	Traditional type	Optimized curved cell-division type
Mass	1	0.74
Normalized Buckling Load	1	1.13
Twist angle/ °	1.34	2.12
Maximum stress/ MPa	138	130
Tip Deflection/mm	210	125

5. Summary and Conclusions

This work has detailed the development of a new graph-based interpreter for a plane-like structure topology generation and a design problems with wing structure design is considered.at last to demonstrate the potential of this approach to the topology optimization of structures and mechanisms. The curved spares take a great improvements for the stiffness increase with less mass cost. Here, our algorithm was shown to be effective in exploring the design space, discovering preliminary topologies that satisfy objectives and constraints, and in particular finding underlying geometries common to designs generated using other topology optimization methods.

Acknowledgment: This research was supported by Key R&D Projects in Shanxi Province (2021GY-339, 2021ZDLGY09-08 and 2019JM-044)

References

[1] Alsahlani A, Rahulan T, Abdulhassan N. Composite structural analysis of a high altitude, solar powered unmanned aerial vehicle[J]. International Journal of Mechanical Engineering and Robotics Research, 2017, 6(1): 71-76.

[2] Kobayashi M, Pedro H T C, Reich G, et al. On a cellular division model for topology optimization[C]//Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. 2009.

[3] Hartl D J, Reich G W, Beran P S. Additive topological optimization of muscular-skeletal structures via genetic L-system programming[C]//24th AIAA/AHS Adaptive Structures Conference. 2016: 1569.

[4] Bielefeldt B R, Akleman E, Reich G W, et al. Expanding the Design Space Via Graph-Based Interpretation of L-System Encodings for Topology Optimization of Multifunctional Structures Expanding the Design Space Via Graph-Based[C]//Proceedings of the 29th International Conference on Adaptive Structures and Technologies, Seoul, Korea. 2018, 30.

[5] Zegard, T.; Paulino, G.H. GRAND—Ground structure based topology optimization for arbitrary 2D domains using MATLAB. Struct. Multidiscip. Optim. 2014, 50, 861–882.

[6] Zegard, T.; Paulino, G.H. GRAND3—Ground structure based topology optimization for arbitrary 3D domains using MATLAB. Struct. Multidiscip. Optim. 2015, 52, 1161–1184.

Contact Author Email Address

zhouzhou@nwpu.edu.cn

Copyright Statement