Research Article

Heuristic Approach for Packing Identical Rectangular Tiles in an Irregular Marble Slab

Zhe Yuan, Xiaocun Song, and Dong An

School of Mechanical Engineering, Shenyang Jianzhu University, Shenyang 110168, China

Correspondence should be addressed to Dong An; andong@sjzu.edu.cn

Received 7 June 2022; Revised 5 August 2022; Accepted 11 August 2022; Published 2 September 2022

Academic Editor: Laxminarayan Sahoo

Copyright © 2022 Zhe Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Packing identical rectangular tiles in a large marble slab is a necessary task. However, when the region of the marble slab is a concave set, the characteristics of the concave set make it difficult to determine whether the rectangular tiles are in the region. For this purpose, this article proposes a heuristic approach to the problem of packing identical rectangular items with orthogonal constraints in a concave irregular region. The heuristic approach uses a tree-search structure, and the final layout is built by inserting a new item in an irregular polygon. The newly inserted item must satisfy the containment and the nonoverlapping constraints associated with the irregular polygon. Regarding the containment constraint, we propose an improved slope algorithm to obtain the inner-fit polygon (IFP) of irregular polygons and rectangular items and apply IFP to determine the relative position of a rectangular item and an irregular polygon to guarantee containment constraint. Compared with existing methods, this approach can not only be used in convex regions but also in concave regions. Numerical experiments and application examples illustrate the effectiveness of the approach.

1. Introduction

As a fundamental mathematical problem, the rectangular packing problem [1] has many application backgrounds in scientific research, engineering practice, and even daily life. For example, the rectangular packing problem is applied in several commercial and industrial contexts, like stone manufacturing, shoe manufacturing, and furniture making. In the stone industry, slates cut from natural rocks often have irregular contours, which pose challenges for stone processing. At present, the natural stone processing schemes adopt the traditional manual arrangement, which takes a long time and has a low use rate. Therefore, the intelligent packing of irregular natural stone has become an urgent requirement of the stone industry.

The solution to the rectangle packing problem is to use an optimization approach to find a favorable or workable solution in a reasonable amount of time. The goal of packing rectangle items in a container is to find the greatest density of a set of small rectangles arranged in a container. The maximum density depends on the algorithm and is related to the size of the rectangles. For example, if the size of the rectangles becomes smaller, the density will increase. Sometimes, the container’s shape is also a big problem, because when the container has an irregular shape, it is difficult to judge whether the rectangles are in the region.

Cutting rectangular tiles from natural marble plates is a common factory operation. When the shape of a marble slab is rectangular, it is the most common two-dimensional rectangular layout problem, which is packing as many rectangular items as possible into a large fixed rectangle without any two items overlapping. The so-called bottom-left (BL) heuristic [2], which involves placing a rectangle at its lowest possible place and left-justifying. Some algorithms improve the BL algorithm, such as the bottom-left fill (BLF) heuristic [3], deepest bottom-left-fill (DBLF) [4], improved bottom-left (IBL) [5], and bottom-left decreasing (BLD) [6]. However, because of the lack of an iterative optimization process, the heuristic method makes it difficult to get the optimal solution. Some scholars use a combination of meta-heuristic algorithms and layout representation to optimize the layout order. Genetic algorithm [7], simulated annealing
In this paper, a heuristic approach is proposed to the problem of packing identical rectangular items with orthogonal constraints into a concave irregular marble slab. Different from regular-container packing, packing items into irregular shapes is more difficult because of the complex boundary contours and the corresponding geometric calculations. In recent years, the most widely methods for dealing with irregular shapes have been no-fit polygons (NFPs) and inner-fit polygons (IFPs). In the new approach, the final layout is built by adding a new item to the container with the constraint that the new item must be completely inside the container and not overlap with items already packed in the container. The IFP method is applied to solve the concave containment constraint, and the NFP method is applied to decide the position of the newly inserted item.

2. Supporting Concepts

The concepts of NFPs and the analogous IFPs [28] will need to be introduced. The NFP method is applied to find the best position of the newly inserted item, and the IFP method is applied to determine the relative position of rectangular items and the container. When applying NFPs and IFPs to packing problems, a key step is to determine the relative position between the reference point and the polygon.

2.1. The No-Fit Polygon. For the two-dimensional packing and cutting problem involving irregular shapes in which no items overlap or protrude from the container is a difficult task for which an essential element is required in any solution approach. An efficient geometric tool to determine the relative position of a pair of pieces, touching, separating, or colliding, must be considered. The no-fit polygon (NFP) method can be applied to packaging problems involving regular or irregular shapes. The NFP and methodologies for implementing this notion in cutting and packing were discussed by Bennell et al. [29].

For a pair of irregular shapes, for example, two polygons A and B, NFP_{AB} is a polygon derived from the polygons A and B. As seen in Figure 1(a), A is a stationary polygon, while B is a tracing polygon that slides around A without rotating. In Figure 1(a), given a reference point on B, when the bottom-left corner of B is selected as the reference point, the locus of the reference point of B forms an NFP_{AB}.

For each pair of pieces, A and B, the properties of NFP_{AB} are subject to the interaction between A and B, summarized as follows: A and B will overlap if the reference point of B is put inside NFP_{AB}; if the reference point of B is placed on the boundary of NFP_{AB}, A and B will touch but not overlap; and if the reference point of B is placed outside NFP_{AB}, A and B will not touch or overlap. Figure 1(b) shows a simple case: B_{1} has its reference point inside NFP_{AB}; B_{2} has its reference point touching NFP_{AB}; and B_{3} has its reference point outside NFP_{AB}. Hence, the boundary and interior of NFP_{AB} represent all possible touching positions between A and B.
2.2. Inner-Fit Polygon. IFP<sub>AB</sub> is a concept related to NFP<sub>AB</sub> for a pair of polygons A and B. It is generated in a similar way to NFP<sub>AB</sub>, that is, the locus of the reference point on B that slides along the internal contour of A without overlapping. An example can be seen in Figure 2(a). As with NFP<sub>AB</sub>, the properties of IFP<sub>AB</sub> illustrate the feasible placement position of B placed inside A. Assuming that the reference point of polygon B is within or on the side of IFP<sub>AB</sub>, B will always be inside polygon A. Otherwise, B will interact with A or lie outside of A, as shown in Figure 2(b).

3. Mathematical Model

This article packs as many identical rectangular items as possible into an irregular region. The rectangular items must be parallel to each other in the cutting process without rotation. Without loss of generality, each rectangle will be parallel to the natural x and y axes of the Cartesian coordinate system. Considering that all formulations of the packing problem consist of two types of constraints: those preventing the collection of items from exceeding the dimensions of the container and those forbidding the items from overlapping with each other.

3.1. Point Inclusion. When using NFP or IFP with a heuristic method, the challenge of determining the relative position of two polygons is translated into a point-inside test, as mentioned above. A mathematical algorithm for the point-inside test is required.

If the detecting point is P<sub>i</sub> (x<sub>i</sub>, y<sub>i</sub>) and the polygon has M sides, then the M vertices are represented by P<sub>1</sub>, . . . , P<sub>M</sub>, whose coordinates are (x<sub>i</sub>, y<sub>i</sub>), and P<sub>M+1</sub> = P<sub>1</sub>. When joining the vertex P<sub>i</sub> and the judged point P, if side P<sub>i</sub>P<sub>i+1</sub> moves to side PP<sub>i+1</sub> in a clockwise orientation then ∠P<sub>i</sub>PP<sub>i+1</sub> is positive, otherwise, it is negative for all i = 1, . . . , M. The absolute value of each angle is less than π. It can be defined as:

\[
f(x, y) = \sum_{i=1}^{M-1} \angle P_iPP_{i+1} + \angle P_MPP_1.
\] (1)

The relative position of a point P and a polygon with M vertices can be obtained using the value of f(x, y) from (1). If f(x, y) = 2π, then P is inside the polygon, see Figure 3(a). If f(x, y) = π, then P is on the side of the polygon, see Figure 3(b). If f(x, y) = 0, then P is outside the polygon, as shown in Figure 3(c).

To calculate (1), the points P<sub>i</sub> (x<sub>i</sub>, y<sub>i</sub>) and P<sub>i</sub> (x<sub>i</sub>, y<sub>i</sub>) form the vector v<sub>i</sub> = P<sub>i</sub> − P<sub>i+1</sub>. The dot product formula is as follows:

\[
\mathbf{v_i} \cdot \mathbf{v_{i+1}} = |\mathbf{v_i}| |\mathbf{v_{i+1}}| \cos \theta.
\] (2)

From (2), the value of \( \theta \) can be calculated as:

\[
\theta = \arccos \left( \frac{\left( x_i - x \right) \times \left( x_{i+1} - x \right) + \left( y_i - y \right) \times \left( y_{i+1} - y \right)}{\sqrt{\left( x_i - x \right)^2 + \left( y_i - y \right)^2} \times \sqrt{\left( x_{i+1} - x \right)^2 + \left( y_{i+1} - y \right)^2}} \right)
\] (3)

Because when the P<sub>i</sub>P<sub>i+1</sub> moves to PP<sub>i+1</sub> in a clockwise orientation then ∠P<sub>i</sub>PP<sub>i+1</sub> is positive, otherwise, it is negative for all i = 1, . . . , M. Therefore, the formula for ∠P<sub>i</sub>PP<sub>i+1</sub> is as follows:
Based on Equation (4), Equation (1) can be reformulated as follows:

\[
\angle P_iPP_{i+1} = \arccos \frac{(x_i - x) \times (x_{i+1} - x) + (y_i - y) \times (y_{i+1} - y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2} \times \sqrt{(x_{i+1} - x)^2 + (y_{i+1} - y)^2}} \\
\times \text{sgn}\left[(x_i - x) \times (y_{i+1} - y) - (x_{i+1} - x) \times (y_i - y)\right].
\] (4)

Therefore, based on Equation (4), Equation (1) can be reformulated as follows:

\[
f(x, y) = \sum_{i=1}^{M} \arccos \frac{(x_i - x) \times (x_{i+1} - x) + (y_i - y) \times (y_{i+1} - y)}{\sqrt{(x_i - x)^2 + (y_i - y)^2} \times \sqrt{(x_{i+1} - x)^2 + (y_{i+1} - y)^2}} \\
\times \text{sgn}\left[(x_i - x) \times (y_{i+1} - y) - (x_{i+1} - x) \times (y_i - y)\right].
\] (5)
From (5), the value of $f(x, y)$ can be calculated. Then, the relative position of a point and a polygon can be detected. Additionally, if the detection point is on the vertex of the polygon, (5) cannot generate a value since the denominator is zero.

3.2. Containment Constraint. In contrast to convex regions, for concave regions, determining that all vertices of a rectangular item are in an irregular region is not enough to guarantee that the item is in the region, when the region is a concave region, the IFP is used. As described in Section 2.2, the containment constraint is reduced to a point-inclusion test to determine whether the reference point of the item is inside the IFP formed by the rectangular item and the container.

In this paper, an IFP is applied to determine the relative position of a rectangular item and an irregular polygon; an algorithm for calculating the IFP of a convex polygon and a concave polygon is proposed. An improved slope algorithm is proposed to obtain the IFP of an irregular polygon and a rectangular item. The biggest difference between the improved IFP algorithm and Ghosh’s slope algorithm [30] is the orders of edge vectors and polygon is proposed. An improved slope algorithm is proposed to calculate the IFP of a convex polygon and a concave polygon.

To simplify the constraint to prevent overlap, (6) and (7) are changed to one side, as follows:

$$a^2 - (C_x^i - C_x^j)^2 \leq 0,$$

$$b^2 - (C_y^i - C_y^j)^2 \leq 0.$$  

Then, $t \leq 0$ is substituted with $\max\{0, t\}^2 = 0$ and the two equalities are combined by multiplication. Thus, the nonoverlapping constraint for item $i$ and $j$ can be defined as follows:

$$\max\left(0, a^2 - (C_x^i - C_x^j)^2\right) \times \max\left(0, b^2 - (C_y^i - C_y^j)^2\right) = 0.$$  

It is obvious that if the center points of any pair of rectangular items satisfy Eq. (10), they will not overlap with each other. The formulation that item $m + 1$ does not overlap with the $m$ item already packed in the container on the basis of Equation (10) is given as follows:

$$\sum_{i=1}^{m} \max\left(0, a^2 - (C_x^i - C_{x^{m+1}})^2\right)^2 \times \max\left(0, b^2 - (C_y^i - C_{y^{m+1}})^2\right)^2 = 0.$$  

4. Solution Method

In this section, a heuristic algorithm is proposed to solve the problem of packing as many identical rectangles as possible without rotation in an irregular region. The heuristic algorithm has two issues to consider. One is to decide the position of the newly inserted item, and the other is to decide the position of the previously packed item. In general, the final layout is obtained by adding a new item to a partial solution of already placed items until no item can be inserted into the container. When the position of the first packed item is decided, the final layout can be decided as well. Therefore, different positions of the first packed item will produce different packing layouts.

4.1. Gridding of NFP and IFP. In the packing problem, the position in the coordinate system of the corresponding item is determined by the center point $C$ of the rectangular item. The values for the horizontal and vertical directions of the rectangular item are $a$ and $b$, respectively. Item $i$ is the $i$th packed item in the container, whose center point is $(C_x^i, C_y^i)$, as shown in Figure 7(a). NFP $(i)$ is defined as the NFP of item $i$, whose reference point is the center point of item $i$. Obviously, NFP $(i)$ is a rectangle whose dimensions are $2 \times a$ and $2 \times b$ with the same center point as item $i$ (Figure 7(b)). Due to all items being identical rectangles, the corresponding NFPS are identical in dimension with different center positions. NFP $(i)$ provides all possible touching positions of each item touching item $i$. On this basis, all possible positions of the next packed item, item $i + 1$, can be decided.
A grid is constructed on NFP \((i)\). The grid has appropriate resolutions \(\Delta x\) in the \(x\) dimension and \(\Delta y\) in the \(y\) dimension. It is assumed that \(\Delta x\) divides \(a\) and \(\Delta y\) divides \(b\), that is, \(a = k_1 \Delta x\) and \(b = k_2 \Delta y\), where \(k_1\) and \(k_2\) are both positive integers; see Figure 8(a). All nodes of the grid form a set \(T_i\). Nodes on the edge of NFP \((i)\) represent all admitted positions where each item touches item \(i\). These nodes can be defined as NFP_1\((i)\) (see Figure 8(b)).

A grid is constructed on IFP, formed by a rectangular item and the container. The grid has the same resolutions as a grid over NFP; the bottom-left node of the grid and the bottom-left point of the minimum enclosed rectangle of IFP
Figure 6: Cases of the nonoverlapping constraint: (a) item (i) and item (j) satisfying Equation (6); (b) item (i) and item (j) satisfying Equation (7).

Figure 7: The center point of item i and its NFP rectangle: (a) description of item (i); (b) the corresponding NFP (i) of item i.

Figure 8: The nodes on the edge of NFP (i) obtained by gridding: (a) gridding of NFP (i); (b) set of nodes on the edge of NFP (i).

Figure 9: The nodes inside the IFP rectangle obtained by gridding: (a) gridding of IFP; (b) set of nodes inside IFP.
**Figure 10:** Process of selecting the position of item 3.

**Figure 11:** Nodes representing all possible positions of the first packed item.

**Figure 12:** Illustration of the heuristic.
Heuristic approach for packing

\( A = \text{length of the item} \)
\( \Delta x = \text{gridding accuracy of x axis} \)
\( B = \text{width of the item} \)
\( \Delta y = \text{gridding accuracy of y axis} \)

\[ \text{Step 1} \]
Construct resolution grid
Set \( k_1 = 10, k_2 = 10, t = 1 \)
Set \( T_2 \) by nodes inside IFP

\[ \text{Step 2} \]
Generate root nodes and child nodes
\( T_2 \rightarrow \text{root nodes} \rightarrow \text{child nodes} \)

\[ \text{Step 3} \]
Choose the branch
Choose Node = Whose depth is largest

\[ \text{Step 4} \]
Form new set \( T_2 \)
Filtering nodes which are not searched from \( T_2 \)

\[ \text{Step 5} \]
Loop judgement
\[ T_2 \text{ is empty?} \]
\( \text{Yes, output the result} \)
\( \text{No, } t = t + 1, \text{ go to step 2} \)

---

**Figure 13:** Flow diagram of the algorithm.

**Table 1:** Computational results.

<table>
<thead>
<tr>
<th>No.</th>
<th>Irregular region</th>
<th>Rectangle item Dimensions</th>
<th>Area</th>
<th>Number of items</th>
<th>CPU time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{(0,0), (40,0), (20,20)}</td>
<td>400</td>
<td>3 × 2</td>
<td>6</td>
<td>56</td>
</tr>
<tr>
<td>2</td>
<td>{(0,0), (20,0), (18,20), (2,20)}</td>
<td>360</td>
<td>4 × 4</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>{(0,0), (15,0), (18,12), (3,12)}</td>
<td>180</td>
<td>2 × 3</td>
<td>6</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>{(0,0), (30,0), (15,15), (15,30)}</td>
<td>337.5</td>
<td>3 × 3</td>
<td>9</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>{(0,0), (20,0), (24,24), (10,16), (0,22)}</td>
<td>422</td>
<td>4 × 3</td>
<td>12</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>{(0,0), (14,0), (30,−10), (26,20), (6,16)}</td>
<td>508</td>
<td>3 × 3</td>
<td>9</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>{(0,0), (21,0), (30,20), (6,30), (−4,20)}</td>
<td>720</td>
<td>4 × 3</td>
<td>12</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>{(0,0), (10,0), (15,12), (10,24), (0,24), (−5,12)}</td>
<td>360</td>
<td>3 × 3</td>
<td>9</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>{(0,0), (15,0), (7,5,18), (15,36), (0,36), (−7,5,18)}</td>
<td>540</td>
<td>4 × 3</td>
<td>12</td>
<td>36</td>
</tr>
<tr>
<td>10</td>
<td>{(0,0), (24,0), (21,6,12), (24,24), (0,24), (4.8,12)}</td>
<td>489.6</td>
<td>4 × 2</td>
<td>8</td>
<td>54</td>
</tr>
<tr>
<td>11</td>
<td>{(0,0), (10,0), (15,−6), (22,10), (14,24), (0,24), (−5,12)}</td>
<td>526</td>
<td>3 × 3</td>
<td>9</td>
<td>46</td>
</tr>
<tr>
<td>12</td>
<td>{(0,0), (6,−6), (4,−20), (20,−12), (36,−20), (30,−6), (40,0), (24,0), (20,16), (10,0)}</td>
<td>568</td>
<td>2 × 2</td>
<td>4</td>
<td>105</td>
</tr>
</tbody>
</table>
coincide with each other (Figure 9(a)). Using Eq. (5), some dots of the grid that lie outside the IFP are deleted; the remaining dots form a set $T_2$, where the reference point of the rectangular item can be positioned such that, the item is totally in the container (Figure 9(b)).

4.2. Position of Next Packed Item. Assuming a partial solution with $m$ already packed rectangular items in the container, there may be many positions for the next item inside the container not overlapping with the previously packed items. The reference point of item $m + 1$ must belong
to $T_2$ and the coordinates of the center points must satisfy (8). On this basis, the positions where item $m + 1$ touches item $m$ are selected to decrease computational time and obtain better partial solutions. As described in section 4.1, grids on NFPs and IFPs have the same resolution, so if the reference point of an item is on the node $T_2$, the contour of the corresponding NFP has $4(k_1 + k_2)$ nodes belonging to $T_2$. Therefore, positions where item $m + 1$ touch item $m$ are those nodes belonging to NFP $(i)$. Afterward, one of the nodes for the position of item $m + 1$ is selected on the basis of the bottom-left approach.

Provided that two items (item 1 and item 2) have been packed in the container, the position of item 3 needs to be decided; see Figure 10. The nodes marked with the symbol $\times$ represent those satisfying the containment constraint, nonoverlapping constraint, and touching item 2. The next step is to select a node from these that satisfies the bottom-left approach as the position of item 3. The bottom-left approach is to select a node whose $x$ coordinate is the minimum. If more than one node can be selected, the one whose $y$ coordinate is the minimum is the best choice.

4.3. Positions of the First Packed Item. As described above, different positions of the first packed item represent different layouts. To find the best layout, all positions where the first packed item could be packed need to be searched. Based on the IFP gridding, nodes inside the IFP represent all possible positions. The nodes marked with the symbol $\times$ may represent all possible positions of the first packed item in Figure 11. These nodes belong to $T_2$. The rectangle has dimensions $2a \times 2b$, and the center point is the bottom-left node of the grid over the IFP.

4.4. The Heuristic Approach. The heuristic approach uses a tree-search structure. In the search tree, nodes represent the positions of rectangle items. Different nodes on the same level correspond to different possible positions of the same rectangular item, and the level of the search tree represents the number of rectangles packed in the container. Thus, the depth of the tree is the maximum number of rectangles the container can contain. Child nodes represent the positions of the next packed item compared with its parent nodes. Each parent node generates $4(k_1 + k_2)$ child nodes, representing positions where the item represented by the child nodes touches the item represented by the parent node, as described in Section 4.2.

The process of building the search tree is as follows; see Figure 12. Generating the root node by selecting the bottom-left node of the IFP grid, as described in Section 4.1, is the first step of constructing the search tree. Figure 13.

Nodes on level 1, the child nodes of the root node, are those points marked with an $\times$ as described in Section 4.3. The number of nodes on level 1 is set to $N_1$. Suppose $k = 4(k_1 + k_2)$, then each node on level 1 generates $k$ child nodes. Only one child node that belongs to $T_1$ and $T_2$ satisfies the bottom-left approach, as described in Section 4.2. Nodes on the subsequent levels are generated in the same fashion. Nodes of different branches represent different layouts of packing identical rectangular items, and the branch whose depth is the largest will be the final layout.

There may be some nodes in the IFP that have not been searched. In order to solve the drawbacks, the process of building the search tree will be looped. The layout of the first search is saved as a partial solution. The remaining nodes represent positions, where the newly inserted item will not overlap with items already packed in the container. The branching tree is re-established for the unselected nodes until the set of remaining nodes in the IFP is empty. All partial solutions form the final layout. Figure 13 gives a graphical representation of the approach.

5. Numerical Experiments and Application

In this section, the algorithm is applied to a computational simulation and stone-plate packing experiment.
5.1. Packing Rectangles in Polygons. The algorithm involved in the experiment was simulated by MatlabR2018b and calculated on an Intel Corei7 CPU with 16 GB memory and 2.6 GHz. To prove the effectiveness of the heuristic algorithm, 12 instances with different polygons were considered, including convex polygons and concave polygons. Table 1 shows the description of each problem, including vertices and areas of Ω, dimensions, and area of the rectangles to be packed, the number of packed items, and the operation time. All problems are randomly generated. Figure 14 shows the solutions.

5.2. Packing Rectangles in a Stone Plate. After the numerical simulation, the heuristic algorithm is applied to packing rectangles in a stone plate. The image of the stone plate, shown in Figure 15, was captured with a CCD camera. The boundary of the stone plate is an irregular curve, and it can be fitted onto a polygon. The polygon is shown in Figure 16.

The size of the small stone piece is 25 mm × 30 mm. The layout result is shown in Figure 17:

6. Conclusion

In this article, we presented a heuristic approach using tree search structures for packing identical rectangles in irregular regions. The final layout is built by inserting a new item into an irregular polygon. In the search tree, nodes represent the positions of rectangle items. Different nodes on the same level correspond to different possible positions of the same rectangular item, and the level of the search tree represents the number of rectangles packed in the container. The model considers containment and nonoverlapping constraints. An IFP is used to overcome the containment constraint. Compared to the existing packing algorithms, the most obvious advantage is packing rectangles in an irregular shape, including convex and concave shapes. In contrast, most algorithms work only with restricted shapes such as rectangles and circles. Numerical simulations and practical experiments proved that the approach can effectively pack identical rectangular tiles into an irregular marble slab.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key Laboratory of Vibration and Control of Aero-Propulsion System, Ministry of Education, Northeastern University (no: VCAME202106); the Fundamental Scientific Research Project of Liaoning Provincial Department of Education (no: LIJKZ0570); the National Science Foundation of China (no: 52075348).

References


