1 Introduction

Origami designers employ both diagrams and crease patterns (CPs) as tools for documenting and sharing their creations. It would be desirable to build a diagram given a CP, but this process is not straightforward. Each experienced folder has a preferred sequence of folding a specific CP, and frequently they just collapse all the folds together at once. But, by observing diagrams, it is possible to recognize recurrent sequences used to fold specific patterns of folds within a CP.

This paper presents an algorithm based on these observations designed to assist the creation of step-by-step folding sequences given an origami CP. It aims to help both origami designers, by making the diagramming process less time consuming, and folders, by giving an alternative option to fold more complex models without needing to collapse all the folds together.

This paper is organized as follows. Section 2 introduces the development of modern origami and its relation to the birth of computational origami. Section 3 deals with some concepts regarding origami design and flat foldability used by the algorithm. Section 4 describes the algorithm itself. Section 5 discusses the scope of applicability, and Section 6 presents
III. Origami Science, Engineering, and Technology

an example of the application on a CP manually. Section 7 presents some results of a developing software implementation.

2 Computation Origami

Origami is the centuries-old traditional Japanese art of paper folding. Although it is an ancient art, only recently has the number of new models significantly increased, both in quantity and in complexity. Origami has reached other frontiers, such as science, technology and mathematics. The popularization of origami started when Akira Yoshizawa published his first book, *Atarashi Origami Geijutsu (New Origami Art)*, in 1954, introducing a new notation to express how to fold an origami model through diagrams with step-by-step instructions [Yoshizawa 54]. This notation uses arrows and lines to represent the positions of folds and the movements performed when folding the paper (see Figure 1 for some symbols). It became so popular that origami designers from all over the world still use it to document their works [Lang 00].

Humiaki Huzita and Jacques Justin provided, in 1989, the first formal description of mathematical principles of paper folding, the Huzita-Justin Axioms [Huzita 92, Justin 91]. Soon, mathematical modeling made it possible for simple shapes to become multilegged insects, multihorned dragons, and multiwinged creatures. The work of Toshiyuki Meguro, Jun Maekawa, Fumiaki Kawahata, and Robert Lang, among others, has made an enormous contribution to the development of technical origami modeling [Lang 98].

Although the CP is not a new tool, it gained more importance in this context. Most of modern design techniques results in a CP of the desired origami base. Besides being easy to draw, CPs contain very useful information under the structure and design techniques of a base through only one picture. Also, it is difficult to find a step-by-step sequence for the desired

![Figure 1. Common symbols used in origami diagrams.](image-url)
base. CPs can be helpful in situations in which the sequence does not exist, and the folders collapse all the folds simultaneously.

The development of mathematical origami also gave birth to another field of study: computational origami, which deals with development of algorithms and their implementation, based on the mathematical modeling of origami and its design techniques. The software TreeMaker, for example, employs tree theory, which is based on the circle/river packing technique [Lang 96]. This software computes the best scale and arrangements for the circles and builds the CP for the desired base automatically.

Another example of origami software is ORIPA [Mitani 2007]. ORIPA is a CP drawing tool that shows a preview of the folded CP and exports it to a computer-aided design (CAD) file.

3 Basic Definitions on Technical Origami

A crease pattern is the set of creases on a square that defines the structure of an origami. Crease patterns, unlike diagrams, don’t show any arrows, and it is common to see CPs that don’t even differentiate mountain folds from valley folds. That is because the origami base resulting from many CPs are flat, and thus there is no more than one way to fold each crease in most cases. If a CP results in a planar base, it is said to be flat-foldable. Flat-foldability of a CP is related to Maekawa’s theorem and Kawasaki’s theorem [Hull 94], and the CP must satisfy noncrossing conditions [Justin 97].

Another aspect of CPs is that most of the time they represent only the base of the origami. A base is a folded state of the paper that has the same structure as the subject, but without the details that characterize the final model. The detail folds are added later in a process called shaping.

Lang describes mathematical algorithms and techniques to design uniaxial bases with an arbitrary arrangement of flaps known as circle packing and/or tree theory [Lang 98].

In the circle-packing technique, the base is designed in the shape of a specified tree graph with the desired edge lengths and configuration, and then the problem is reduced to packing the circles inside the square of the paper. This technique results in a uniaxial base with a determined configuration and length of flaps. A uniaxial base is a base that has all its flaps lying on a single axis and all its hinges perpendicular to this axis [Lang 96].

This determined configuration and length of flaps can be expressed by a tree graph in which the edges of the leaf nodes represent the flaps in position and length. On the square of paper, the leaf nodes are represented by circles and the edges that don’t connect any leaf node are represented by rivers. By packing these two elements in the square, one can determine the
portions of paper that each part of the base will consume. Active paths are lines that connect the center of the touching circles or of the circles that have no space between them apart from the space occupied by rivers. The active paths result in folds that lie along the axis (axial folds) and determine the perimeter of polygons within the square. These polygons could be filled in with molecules, which are partial CPs that will bring all the active paths together and will produce the flaps with the desired length. The designed CP is then obtained by filling all the polygons with molecules [Lang 03a].

Packing the circles and rivers within a square often produces unused portions of paper at the corners of the square. The effective CP is the group of all the polygons, excluding these unused areas.

4 Turning a CP into a Folding Sequence

The construction of this algorithm was made by observing some patterns of creases and how we would fold them, as well as following some examples of diagrams and our own experience. We tried then to make some rules modeling our way of folding. As input, it has a circle-packed CP, and as output, it generates partial CPs that lead to intermediary flat states between the square of paper and the folded base, resembling a series of progressive CPs.

This algorithm focuses only on circle/river-packed CPs of uniaxial bases constructed with the universal molecule. This would allow the software implementation to use as input the output of TreeMaker [Lang 98], which already has the major part of information regarding the tree theory [Lang 03a] on which this algorithm relies.

Because the algorithm tries to follow a single strategy, the sequences generated might not be the most intuitive way for folding some specific CP.

As part of the modeling, we created the concept of “maneuvers.” A maneuver is a set of common steps used to fold specific parts of a CP. The next sections describe the maneuvers employed in the proposed algorithm and their structures; Table 1 summarizes them. Through the paper, it is shown that these parts of the CP are defined regarding the leaf nodes location and the molecules that connect these nodes to other leaf nodes. In the flowcharts, the word node refers to these leaf nodes. An internal node refers to a circle the center of which does not lie on the border of the effective CP, and a border node refers to a circle the center of which lies on the border of the effective CP.

Due to space restrictions, the whole algorithm is not presented here; all of the flowcharts can be found at www.origamiracle.cic.umb.br.
<table>
<thead>
<tr>
<th>Maneuver</th>
<th>Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Used in “Choose and fold next internal node” block.</td>
<td>Collapses two symmetrical neighbor triangular molecules. The molecules must have collinear hinges.</td>
</tr>
<tr>
<td></td>
<td><img src="image1" alt="Ex:" /></td>
<td><img src="image2" alt="Ex:" /></td>
</tr>
<tr>
<td>2</td>
<td>Used in “Choose and fold next internal node” block.</td>
<td>Folds small stubs. It is subdivided into three smaller maneuvers according to the stub characteristics.</td>
</tr>
<tr>
<td></td>
<td><img src="image3" alt="Ex:" /></td>
<td><img src="image4" alt="Ex:" /></td>
</tr>
<tr>
<td>3</td>
<td>Used in “Choose and fold next molecule” block.</td>
<td>Reverse fold or crimp fold.</td>
</tr>
<tr>
<td></td>
<td><img src="image5" alt="Ex:" /></td>
<td><img src="image6" alt="Ex:" /></td>
</tr>
<tr>
<td>4</td>
<td>Used in “Choose and fold next molecule” block.</td>
<td>Petal fold.</td>
</tr>
<tr>
<td></td>
<td><img src="image7" alt="Ex:" /></td>
<td><img src="image8" alt="Ex:" /></td>
</tr>
<tr>
<td>5</td>
<td>Used in “Choose and fold next molecule” block.</td>
<td>Maneuver to fold gusset molecules. It must be a corner molecule and have a symmetry line that intercepts the effective CP twice.</td>
</tr>
<tr>
<td></td>
<td><img src="image9" alt="Ex:" /></td>
<td><img src="image10" alt="Ex:" /></td>
</tr>
</tbody>
</table>

Table 1. Summary of maneuvers. CP examples are Maneuvers 1, 7, and 8 [Lang 04a]; Maneuvers 2 and 9 [Lang 03b] (“Periodical Cicada,” Opus 377); Maneuvers 3, 4, 6, and 10 [Lang 03b] (“Scorpion Varileg,” Opus 379); and Maneuver 5 [Lang 04b].
<table>
<thead>
<tr>
<th>Maneuver</th>
<th>Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Used in “Choose and fold next internal node” block.</td>
<td>Similar to maneuver 1, but with asymmetrical molecules (with collinear hinges). Requires more prefolding than maneuver 1.</td>
</tr>
<tr>
<td></td>
<td><img src="image1" alt="" /></td>
<td><img src="image2" alt="" /></td>
</tr>
<tr>
<td>7</td>
<td>Used in “Choose and fold next internal node” block.</td>
<td>Used to fold asymmetrical triangular molecules that do not have collinear hinges.</td>
</tr>
<tr>
<td></td>
<td><img src="image3" alt="" /></td>
<td><img src="image4" alt="" /></td>
</tr>
<tr>
<td>8</td>
<td>Used in “Choose and fold next internal node” block.</td>
<td>Normally, as a result of other maneuvers, a node could be sunk. This maneuver is used to unsink these nodes by folding molecules with the same hinges after propagation.</td>
</tr>
<tr>
<td></td>
<td><img src="image5" alt="" /></td>
<td><img src="image6" alt="" /></td>
</tr>
<tr>
<td>9</td>
<td>Used in “Choose and fold next molecule” block.</td>
<td>Rabbit-ear fold.</td>
</tr>
<tr>
<td></td>
<td><img src="image7" alt="" /></td>
<td><img src="image8" alt="" /></td>
</tr>
<tr>
<td>10</td>
<td>Used to fold the first node and after some maneuvers.</td>
<td>Squash folds.</td>
</tr>
<tr>
<td></td>
<td><img src="image9" alt="" /></td>
<td><img src="image10" alt="" /></td>
</tr>
</tbody>
</table>

Table 1. (Continued.)
4.1 Main Algorithm

The process of folding consists of transferring some folds of the input CP to a temporary CP. Then, the model is flattened by a subalgorithm called “Propagate Folds” and is output. This way, the sequences of folds can be seen through the series of CPs created.

The proposed algorithm is shown in the flowchart in Figure 2. Initially, a choice is made that depends on whether there is any internal node in the CP. If there are internal nodes in the CP, a subalgorithm chooses which
internal node will be folded next, and folds it. After folding all internal
nodes, the border molecules should be folded, completing the base. If there
is not an internal node, the folds that come from the center of a border
node should be pleat-folded, and then the remaining molecules should be
folded. The process of folding is based on ten maneuvers, each one with a
different complexity and generating a different number of steps.

A node is said to be folded if all the creases that come from the node’s
location are already transferred to the temporary CP. A molecule is said
to be folded if all its creases are already transferred to the temporary CP.

4.2 Choose First Node

The “Choose First Node” subalgorithm (Figure 3) gives scores to the in-
ternal nodes according to their size and position in order to decide which
node will be folded first.

The “size” of a leaf node is referred here as the length of the edge
connected with it. Thus, the largest node is the one whose edge has the
biggest length. When deciding which of the nodes is going to be folded
first, the largest node is the natural choice because it covers a large area
in the CP and usually has a lot of creases emanating from it. Postponing
the fold of the largest node may lead to particularly difficult unsink folds,
as a fold of another node can cause this node to be sunk.

![Choose First Node flowchart](image-url)
However, symmetry is often a reason that it sometimes is easier to begin with other nodes. A node on the symmetry line is preferable if there are active paths connecting it to the largest nodes symmetrically because the fold of the largest nodes will proceed shortly after the fold of this node and will be replicated symmetrically.

We emphasize that the proposed system models the patterns of a particular folder, so it won’t necessarily be the best way. The point assignment for each situation was made by observing a series of CPs: the first node that was chosen by the folder so that the algorithm’s choice would be similar to that of the folder for the analyzed situations.

A stub is a new leaf node added (together with a new edge) to the conceptual tree of the base used to break down higher-order polygons into several lower-order polygons [Lang 03a]. This usually produces a small circle connected with a bigger one through an active path that does not cross any river. We also observed that the molecules that have this active path as a ring path usually have an opening angle greater than 90 degrees.

The strategy here is to find the equivalent arrowhead molecule. So we consider as stubs only the leaf nodes that have three consecutive paths with an angle greater than 90 degrees from each other. Figure 4 shows an example of what is considered a stub by the algorithm.

4.3 Maneuver 10

This maneuver is used to complete the fold of a node after the execution of other maneuvers or to fold a node when there is still nothing folded on the partial CP. It uses a series of squash folds to transfer all the folds that

![Diagram](image-url)
come from the node location to the partial CP.

This subalgorithm receives as input a set of neighbor molecules that have a leaf node in common. Then, molecules and groups of molecules are grouped in pairs iteratively until there is just one group. This big molecule corresponds to the initial state of the paper before the execution of the squash folds. Next, the squash folds are made, dividing grouped molecules until all the original molecules are partially folded (have one ridge and two paths transferred to the partial CP).

This grouping process tries to minimize the addition of unnecessary marks to the paper by merging neighbor molecules that have the same opening angle and folding along the symmetry line, aiming at using creases of the original CP as the new ridge of the groups of molecules. Nevertheless, there could be still some undesirable marks, and the diagrammer will be able to choose to take the first and the last output CP and collapse all the folds together.

4.4 Choose and Fold Next Internal Node

To continue folding the internal nodes, the “Choose and Fold Next Internal Node” subalgorithm searches within the CP for folding patterns related to the known maneuvers. The maneuvers to be executed are then chosen, prioritizing the occurrence of simpler maneuvers before the more complex ones.

This subalgorithm tries to fold the remaining nodes from the inside first, and then to the borders; imagine that there is a circle inflating from the first node folded. When the perimeter of the circle touches an unfolded node, it checks whether it can be folded by Maneuver 8, 1, 6, 7, or 2 (in this order) and folds it. If the entire CP (still with uncompleted internal nodes) gets in the circle without applying any maneuver, this CP is not supported by the proposed set of maneuvers. More details can be found in Section 5.

Maneuver 8 is an unsink fold. The reason its execution is preferred over the other maneuvers (which are easier to execute) is that the more folded molecules a sunken node has, the more difficult it is to perform Maneuver 8. We choose to increase the number of unsinks decreasing the individual difficulty of them. On the other hand, we have Maneuver 7 that has also an unsink fold, but is more unusual than Maneuver 8. By being executed after Maneuver 1 and 6, a node previously foldable through Maneuver 7 might be able to be folded with Maneuver 8, which is simpler.

4.5 Maneuvers 1, 2, 6, 7, and 8

Maneuvers 1, 2, 6, 7, and 8 are used to fold internal nodes (Figure 5). Each maneuver folds one node at a time. Its goal is then to transfer the creases
that come from the location of this node to the partial CP. Generally, two paths divide the folded molecules and the not-yet-folded molecules connected to a certain leaf node. It is also noticeable that the two not-yet-folded molecules that have these paths as ring paths are partially folded (they already have a ridge fold transferred to the partial CP).

Maneuvers 1, 2, 6, 7, and 8 transfer two ridge folds and a set of hinge folds of the partially folded molecules (in most cases completing their fold) and partially fold the other molecules.

In the cases of Maneuvers 1, 6, and 7, those two paths are coincident, and there are no folded molecules attached to this node. Maneuver 1
collapses two symmetrical neighbor triangular molecules. Maneuver 6 is similar to Maneuver 1, but with asymmetrical molecules (with collinear hinges). The main difference is that Maneuver 6 requires more prefolding than Maneuver 1. Maneuver 7 folds asymmetrical triangular molecules that don't have collinear hinges.

Maneuver 8 folds sunken nodes for which the partially folded molecules have a coincident path of hinges (Figure 6). Maneuver 2 folds an arrowhead molecule equivalent to the molecules attached to the stub, and then completes it with a spread sink.

4.6 Choose and Fold Next Molecule

Once all internal nodes are folded, there remain only border molecules or molecules with a path for which the two vertices lie on the border of the effective CP. The algorithm tries to fold them by using petal folds (Maneuver 4), reverse folds and crimp folds (Maneuver 3), or rabbit-ear folds (Maneuver 9). Also there is Maneuver 5 to fold some cases of gusset molecules (border).

Since the execution of Maneuver 3, 4, or 5 does not affect the others, they can be applied arbitrarily. So, to minimize movements of flaps, the choice is made by the proximity to the unfolded axial path (more detail in Section 4.8). Maneuver 9, in contrast, must be preceded by Maneuver 3 or 4.

4.7 Maneuvers 3, 4, 5, and 9

Maneuvers 3, 4, 5, and 9 are used to fold two kinds of molecules: border molecules and internal molecules that have a path for which the two vertices lie on the border. The second kind of molecules is always connected to a border molecule through the previously referred path. This border molecule should be folded with Maneuver 9 after the application of Maneuvers 3 or 4 to fold the internal molecule.

Maneuver 5 was added later to offer support to border gusset molecules that are symmetrical to the line that connects two opposite border nodes in the polygon. Figure 7 shows the molecules foldable through this maneuver and its axis of symmetry.

4.8 Organize Flaps

The partial CP will always have at least one path of unfolded axial creases that comes from and goes to the edge of the paper. An example is shown in Figure 8. The execution of the maneuvers depends on the relative position
of the unfolded axial path to the molecules that will be folded. There could also be some hinge fold or its prolongation that might cross one of the molecules to be folded.

The goal of the “Organize Flaps” subalgorithm is to dislocate the unfolded axial creases and the hinge paths of folds to enable the execution of the maneuvers.

![Figure 7. Molecules foldable through Maneuver 5 [Lang 04b].](image)

![Figure 8. Path of unfolded axial creases highlighted [Lang 04a].](image)
4.9 Propagate Folds

Figure 9 shows the partial CP after execution of Maneuver 1, with the folded nodes highlighted.

This partial CP is not flat-foldable because there are vertices that don’t satisfy Maekawa’s and Kawasaki’s theorems. The “Propagate Folds” algorithm generates a set of provisory folds to make this partial CP flat-foldable. Figure 10 shows the results of the application of this algorithm on the CP shown in Figure 9.

The already folded portion of the CP is uniaxial and has flat molecules. Every maneuver and “organize flaps” application modifies the layer order-

Figure 9. CP after execution of Maneuver 1.

Figure 10. CP with propagated creases.
ing in some specific way. We haven’t modeled these ways yet, so there is still no guarantee of noncrossing conditions. In the future, we will try to fill this gap to implement three-dimensional simulation.

5 Applicability Scope

The scope of this algorithm is related to the set of documented maneuvers. A maneuver is applicable only to specific situations, so some CPs may not be supported. The major part of the documented maneuvers was designed for folding triangular molecules and later expanded to some quadrangular molecules. So far, a CP is foldable through this algorithm if it is circle packed; has only rabbit-ear, waterbomb, sawhorse, and/or gusset molecules; and obeys the following rules:

- If it has gusset molecules, they must be border molecules and have the symmetry explained in Section 4.7.

- While there are still internal nodes to be folded, the CP must have at least one unfolded internal node foldable through the set of maneuvers. This doesn’t happen when all the pairs of partially folded molecules lack coincident hinge paths toward each other and the other two hinge paths that come from the same ridge intercept closest to the node already folded go through more than one molecule to the edge of the paper.

In Figure 11, the hinge folds of the pair of partially folded molecules m1 and m2 don’t meet. But the other two hinge paths of m1 (which has

![Figure 11](image_url)
the closet ridge interception to the node already folded among the two molecules) go to the edge of the paper through only one molecule. So this model is foldable through Maneuver 7.

It is important to notice that the increase in complexity of the CP results in a need for more specific maneuvers. Adding only a few supported molecules would require a lot more maneuvers than we have already documented.

Also notice that the algorithm can handle CPs with an arbitrary number of internal nodes as long as the CP respects the two rules above. Actually, the number and disposition of rivers is more decisive to the applicability of the algorithm. The algorithm probably will not work for CPs with a greater number of rivers.

The project website has an example of the folding of an infinite grid of bird bases. There we can see that, as the CP doesn’t have any rivers at all, there are only a few situations that occur repeatedly and can be folded without problems.

6 Example

In this section, we describe an example of the manual application of the algorithm into a CP. We are still working on providing an example for every case supported by each flowchart on the website. Consider the CP shown in Figure 12.

![Figure 12. Anteater’s CP [Akitaya 10].](image-url)
The Choose First Node subalgorithm chooses node 3, which scores 7 points for being the largest and on a symmetry line. After the execution of Maneuver 10, the resulting partial CP is as shown in Figure 13.

Next, node 2 is foldable through Maneuver 1. The ridge folds of the two partially folded molecules attached to node 3 are transferred to the partial CP as mountain folds. Then their coincident hinge path is transferred as a valley fold. Then, the axial folds of such molecules are transferred to the partial CP (one as a valley and the other unfolded). An added provisory mountain fold bisects those paths to flatten the model. Results are shown in Figure 14.
Inside the algorithm of Maneuver 1, Maneuver 10 is executed to partially fold the other molecules attached to node 3 (Figure 15, left). And then Maneuver 4 folds the two symmetrical partially folded molecules (Figure 15, right).

Now, the “Choose and Fold Next Internal Node” subalgorithm chooses node 7 or 8 arbitrarily. These nodes are connected to border nodes (5 and 6) through triangular molecules. A stub is said to be connected to a node if there is no river between that node and the stub. Maneuver 2.2 folds molecules attached to the stub to make an arrowhead molecule, as shown in Figure 16.

Then, Maneuver 2 completes with a squash fold, by transferring the remaining ridge folds of the molecules (the one that comes from the stub is a valley fold) and the hinge path they have in common as a mountain fold. Results are shown in Figure 17.

The “Choose and Fold Next Internal Node” subalgorithm chooses node 15 or 16. They are folded with Maneuver 2 also. Results are shown in Figure 18.

Having all internal nodes folded, the algorithm does Maneuver 3 (reverse fold) to complete the remaining molecules. Results are shown in Figure 19.
Figure 17. Maneuver 2.2 spread sink.

Figure 18. Maneuver 2.2 (nodes 15 and 16).

Figure 19. Completed CP.
Figure 20. Bird base CP created with TreeMaker and temporary CPs: (a) CP created with TreeMaker; (b) first temporary CP, created after Maneuver 10; (c) second temporary CP, created after Maneuver 4; (d) last temporary CP, created after Maneuver 4.

7 Software Implementation

The software implementation is meant to receive a CP file created with TreeMaker as input, create several temporary CPs, and save them as ORIPA files. This way, we can assume that the TreeMaker files are circle/river packed and that the temporary CPs can be viewed and folded in ORIPA. TreeMaker can also triangulate the CP so that all molecules would be triangular or quadrangular, one of the conditions required by the algorithm, and ORIPA can also create .dxf CAD files.

So far, we have started Maneuvers 10 and 4. With these maneuvers, we’re able to fold the first steps of some origamis. One of them is the bird base. The original CP created with TreeMaker and the temporary CPs created by the software are shown in Figure 20.

The bird base has only one internal node. It is folded using Maneuver 10 (Figure 20(b)). Then there are four symmetrical molecules that can be folded using Maneuver 4 (Figure 20(c) and (d)). ORIPA can fold these
Development of an Intuitive Algorithm for Diagramming and 3D Animated...

Figure 21. Temporary CPs folded by ORIPA and SVG images created based on ORIPA folding.

temporary CPs and, by using the points of the folded CPs, we can create vectorized images in scalable vector graphic (SVG) format. The folded CPs and the vectorized images created are shown in Figure 21.

The result is a vectorized image, so it can be edited easily on any vector graphics editor. The diagrammer is allowed to change the ordering of facets to change line strokes (to create dashed lines, for example), and to change the drawing size. An example of a diagram created this way is shown in Figure 22. Steps 2 through 5 were done with the SVG images. Step 1 is just a square, and each step from 6 to 9 copies the previous step with some small edits.

8 Conclusion

This work describes an algorithm to fold a certain type of CP constructed by a circle/river packing technique. Although it is very strict about the type of CP, the results obtained show that, when implemented, this algorithm would be able to reduce substantially the work spent on diagramming.
Figure 22. Paper crane diagram.

Combined with the TreeMaker software, it would offer a rapid solution for an origami design, helping the designer to test fold and accelerating the task of diagramming.

As future work, we will go on with the implementation of the proposed algorithm in software. Also, we will add new maneuvers and their rules of application to expand the group of covered molecules. Because the algorithm is basically an expert system, it reflects the experience of only one folder. To suppress this limitation, a system could be built to help an origami designer to create a diagram for a CP, not only proposing maneuvers at each step of the folding process, but also allowing the user to add his or her own maneuvers when the proposed ones are not acceptable. We can expect that the addition of new maneuvers, as well as the rules for their application by many designers and folders, will expand the algorithm so that it would be able to fold any flat origami base.

Bibliography


Development of an Intuitive Algorithm for Diagramming and 3D Animated... 373


