Xu ML, Gu NB, Xu WW *et al.* Mechanical assembly packing problem using joint constraints. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(6): 1162–1171 Nov. 2017. DOI 10.1007/s11390-017-1791-2

Mechanical Assembly Packing Problem Using Joint Constraints

Ming-Liang Xu[1], *Member*, *CCF*, Ning-Bo Gu[1], Wei-Wei Xu[2,*], *Member*, *ACM*
Ming-Yuan Li[1], *Member*, *ACM*, Jun-Xiao Xue[3], *Member*, *CCF*, and Bing Zhou[1]

[1] *School of Information Engineering, Zhengzhou University, Zhengzhou 450000, China*
[2] *State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University, Hangzhou 310058, China*
[3] *School of Software, Zhengzhou University, Zhengzhou 450000, China*

E-mail: iexumingliang@zzu.edu.cn; gu.ningbo@qq.com; xww@cad.zju.edu.cn; iemyli@gs.zzu.edu.cn
{xuejx, iebzbou}@zzu.edu.cn

Received June 26, 2017; revised October 13, 2017.

Abstract The three-dimensional packing problem is generally on how to pack a set of models into a given bounding box using the smallest packaging volume. It is known as an NP-hard problem. When discussing the packing problem in mechanical field, the space utilization of a mechanism is low due to the constraint of mechanical joints between different mechanical parts. Although such a situation can be improved by breaking the mechanism into components at every joint, it burdens the user when reassembling the mechanism and may also reduce the service life of mechanical parts. In this paper, we propose a novel mechanism packing algorithm that deliberately considers the DOFs (degrees of freedom) of mechanical joints. With this algorithm, we construct the solution space according to each joint. While building the search tree of the splitting scheme, we do not break the joint, but move the joint. Therefore, the algorithm proposed in this paper just requires the minimal number of splits to meet the goal of space utilization. Numerical examples show that the proposed method is convenient and efficient to pack three-dimensional models into a given bounding box with high space utilization.

Keywords NP-hard problem, packing problem, search tree

1 Introduction

The packing problem is on how to pack a group of items into a container with the smallest packaging volume, which is usually based on combinatorial optimization algorithms. It is frequently encountered in the fields of manufacturing and transportation industries.

The research work on the three-dimensional (3D) packing problem can be classified into three categories: bin-packing, knapsack-loading and container-packing. The bin-packing problem targets at how to put boxes of different sizes into as small as possible containers, and all boxes should be parallel to the container[1-2]. The knapsack-loading problem studies how to select the object into the knapsack, so that the value of the objects is maximum[3-4]. And in the container-packing problems, all boxes should be packed into a container with an unlimited size. The goal of the problem is to find a reasonable combined manner of the boxes to make the size of the container to be the smallest. Authors of [5-7] proposed a split-packing algorithm which splits the model into small parts and then wraps the parts into a container with minimum volume.

In this paper, we introduce a novel mechanical assembly packing method using joint constraints. Based on the split packing algorithm, we firstly split the mechanical model into small parts and then pack them into a square container which is as small as possible. The main technical contribution of this paper is that we integrate joint constraints among mechanical components to improve the efficiency of the 3D packing of mechanical assemblies. Our packing algorithm first

Regular Paper
Special Section of CAD/Graphics 2017
The work was supported by the National Key Research and Development Program of China under Grant No. 2017YFC0804401, the National Natural Science Foundation of China under Grant Nos. 61472370, 61672469, 61379079, 61322204, and 61502433, the Natural Science Foundation of Henan Province of China under Grant No. 162300410262, and the Key Research Projects of Henan Higher Education Institutions of China under Grant No. 18A413002.
*Corresponding Author

constructs the solution space of the packing problem at each joint. After that, candidate disassembly schemes are selected using a search algorithm. Finally, we adjust and optimize the mechanical joint parameters involved in the selected schemes for small packaging volumes. Therefore, we can obtain the optimal splitting scheme to meet the target of space utilization by using minimum splitting of the mechanical model.

The rest of the paper is structured as follows. We first review some related work in Section 2, followed by an overview of our packing algorithm in Section 3. Section 4 discusses the construction of search tree. Then, we precisely analyze how to merge parts into groups and pack the groups in Section 5 and Section 6 respectively. In Section 7, numerical examples are given to illustrate the effectiveness of the algorithm. We conclude the paper in Section 8.

2 Related Work

2.1 Mechanical Assembly Designs

In mechanical assembly designs, mechanical parts are connected via joints to achieve the desired function of the designer. There are two main design parameters in such designs: 1) the spatial position and orientation of the parts in the assembly; 2) mechanical connections between parts and the corresponding kinematic chain information. In [8], the connection relations between parts of a machine are represented by semantics, where the semantic web rules language (SWRL) and semantic Internet ontology description language (OWL) are often used to describe the mechanical assembly connection. Mitra *et al.*[9] proposed an automatic method to analyze the type of connection between parts through the shape analysis of mechanical parts. Xu *et al.*[10] presented an interactive system for mechanism modeling from multi-view images. Its key feature is that the generated 3D mechanism models contain not only geometric shapes but also internal motion structures. Coros *et al.*[11] presented an interactive design system. The system used an articulated character. Then the user iteratively creates an animation by sketching motion curves indicating how different parts of the character should move.

2.2 Model Packing

The existing model packing algorithms are mostly developed in the context of manufacturing industry. In [12], several kinds of furniture are selected from the specified range to maximize the filling rate of the target container. The packaging efficiency of the algorithm of [12] is as high as 91.3%. However, the algorithm in [12] cannot be directly applied to the mechanism packing problem, since it requires that all parts be fully packed. In [13], researchers gave a group of objects that need to be packed. In the literature, objects are packed into a container with the smallest volume. Gomes and Oliveira[14] adapted a simulated annealing algorithm to solve the 3D packing problem and achieved good results. Crainic *et al.*[15] proposed an extreme point-based heuristics algorithm to solve the packing problem. However, in this algorithm, it is assumed that the object is packed as a rectangular box. In the mechanism packing problem, due to that the geometric shape of the mechanical parts is irregular, there is a lot of wasted space if axis-aligned bounding boxes of the parts are used.

3 Overview of the Packing Algorithm

In our algorithm, the input is the assembly information of the mechanical model and target usage efficiency of the space, where the assembly information contains the geometric information of the mechanical model and the joint constraints. The output is a packing scheme which meets the target of space utilization with the minimum number of splitting operations for the mechanical model. The basic steps are as follows.

1) Building the solution space of the search tree that contains all the splitting schemes, where the splitting indicates the designated splitting information at the joints of the input mechanism model. Each node in the search tree corresponds to a splitting scheme.

2) Selecting a splitting scheme and splitting the mechanical model into small parts according to the rule of the selected scheme.

3) Minimizing the volume of each group of the mechanical parts through optimizing joint parameters.

4) Packing all splitting part groups into a container, and calculating whether the space utilization of the container meets the specified target.

Fig.1 shows the packing procedure of a 3D crank slider model. First, the mechanical model is split up into multiple groups of parts according to the splitting scheme in the search tree. Second, we minimize the volume of each group with respect to joint constraints. Finally, we pack the optimized groups into a pre-specified container.

We first define several concepts in our packing algorithm.

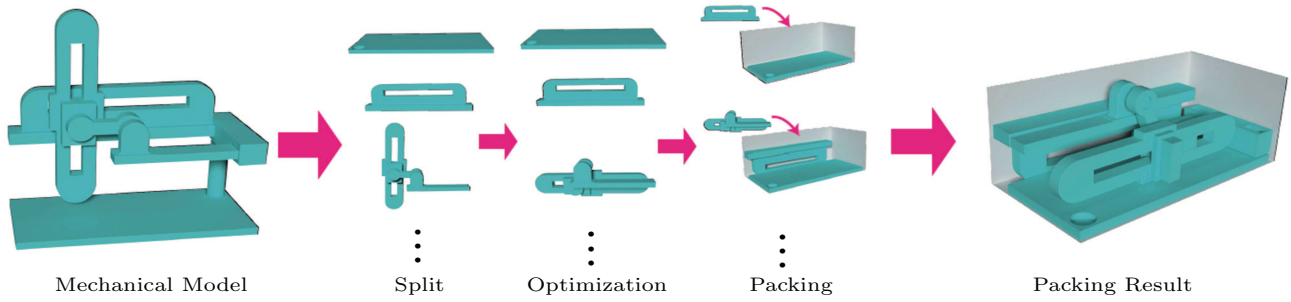


Fig.1. Packing procedure of a crank slider model.

Space Utilization. It indicates that the ratio of the volume of all parts of a model to the volume of a container. It is expressed by "E". Moreover, the higher the space utilization, the less free space in the container.

Part. It represents a separate part of the model, expressed by "p".

Joint. It defines the connection between two parts, indicated by "c". In this paper, the joint type is divided into the fixed joint, rotating joint and translational joint. We also store the rotating or translational joints.

Joint Set. It defines the set of all joints in a model, indicated by "C". The set of joints in a model can be expressed as $C = \{c_1, c_2, \cdots, c_m\}$. In this paper, in order to show whether the joint is split between the parts, we define the failure state and the active state of a joint. In the failure state, it means that part A and part B are two separate parts. When the joint is in the active state, the parts A and B have the space constraints generated by the joint. We define the set F to represent the set of the active state joints.

Group. It defines one or more parts interconnected via joints, indicated by "g".

Group Set. It defines a set of splitting parts groups, and can be indicated by "G". $G = \{g_1, g_2, \cdots, g_n\}$. In parts group set G, $\forall g_i, g_j \in G, g_i \cap g_j = \emptyset$.

The algorithm begins with obtaining the joint set C according to the assembly semantic information of the input model. Then, a search tree is constructed by the bottom-up method according to the joint set C. After that, we traverse the set using the depth-first algorithm to access the nodes in the search tree. We define the default state as the failure state if the root node of the search tree corresponding to the group set is disconnected. And then, if the packing space utilization satisfies the target space utilization, the packing scheme is recorded and the other branches are traversed to check if there is a better result compared with the rerecorded scheme.

In Fig.2, we present a crank slider model which contains assembly information. First, we obtain the set of joints of the crank slider according to the assembly information, where $C = \{c_1, c_2, \cdots, c_6\}$. $F = \emptyset$ is the set of the active state joints of the root node. Each part constitutes a part group. Finally, we can find the packing scheme after visiting the whole search tree. When traversing the search tree, there is at most one joint change to the active state between the parent and the child node, that is, one joint information should be added in the set of the active state joints F. In order to reduce the repeated calculation, the parts groups set of the child nodes can be obtained from the parent node through the merge operation. We combine the two connected groups into a new parts group according to the newly added joint. For each new parts group, it is necessary to optimize the joint parameters to achieve its minimum volume.

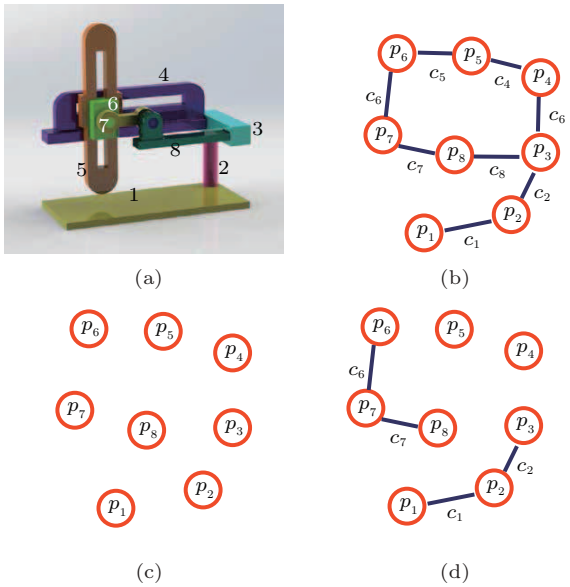


Fig.2. Splitting the model based on the joints. (a) 3D mechanical model. (b) Origin joint graph. (c) $F = \emptyset$. (d) $F = \{c_1, c_2, c_6, c_7\}$.

When the set of parts groups have been optimized, the bottom of the packing container needs to be pre-set according to the volume parameters of all the component groups. Then, a "rotation-translation" approach is used to test the transformations of each group to find the placement of the group leading to a minimized volume. When all of the groups are packed, we need to judge whether the space utilization target of the container is satisfied, and then decide whether the depth traversal is performed. Therefore, this algorithm can be divided into two steps.

1) *Merging the Group*: merging the old part groups according to the set of the active state joints F to form a new group, and then optimizing the joint parameters of the new group to reach its minimum volume.

2) *Packing the Group*: packing the set of the part groups to a container, and calculating the space utilization.

4 Search Tree Construction

We construct the search tree using the bottom-up strategy. In our setting, the root node represents all of the joints in the failure state, which means the set of the active state joint is empty, i.e., $F = \emptyset$. In this case, all the joints are split, and all the parts in the mechanism model are independent of one another. Thus, each part group at the root node only contains a single part. Our algorithm then switches the joint state to create the children nodes through the root node. Specifically, we can select one joint at the root node and change its state to be active, which might merge two parts groups connected by the joint into a group with two parts. Since each joint state change leads to a child node of the root node, the tree construction algorithm creates the number of child nodes corresponding to the number of joints. This step is recursively performed to create a search tree that can cover the complete solution space. In Fig.3, letter "x" indicates the default state, which means the connection relationship has not been established, i.e., the joint is in the failure state.

In the constructed search tree, each node corresponds to a split scheme of the mechanical model. While performing the depth-first traversal policy, if the space utilization of the node's split scheme is larger than the target space utilization, the split scheme is recorded and returned to the parent node to visit the other branches; otherwise, we should continue to visit its child nodes.

In order to reduce the computational load, we store the smallest number of the parts groups in the split schemes that has met the target space utilization so far, which is indicated by $N_{\min}$. If the group number in the current node is larger than $N_{\min}$, our algorithm will continue to visit its child nodes, but not pack or calculate the space utilization until the group number is less than or equal to $N_{\min}$. If the difference between the number of the group in the current node and the number of inactive joints is larger than $N_{\min}$, it indicates that all nodes in the sub-tree of the current node cannot meet the minimum group requirement. We then prune the sub-tree directly.

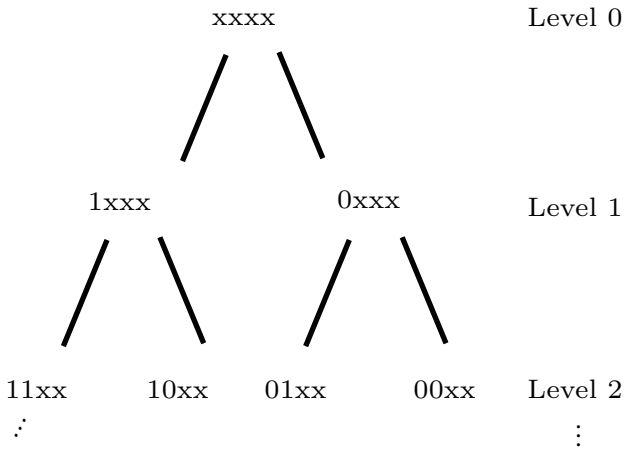


Fig.3. Search tree construction.

5 Minimizing the Packing Volume

Most of the commonly used mechanical models are a combination of planar institutions. Therefore, this paper also investigates planar mechanisms. In the plane mechanism, the kinematic pair is defined to be two parts when the joints are in different groups. The kinematic pair is classified into the lower pair and the higher pair. The lower pair is the contact between the surfaces which belong to two parts. As shown in Fig.4, the revolute pair and the translational pair belong to the low pair. The higher pair is connected by points or lines of two parts, such as the gear pair and the cam pair in Fig.4. Since the contact area of the higher pair is small and not so stable as that of the lower pair, the higher pair is not considered in this paper. Therefore, only the revolute pair and the translational pair are considered when performing the joint parameter optimization.

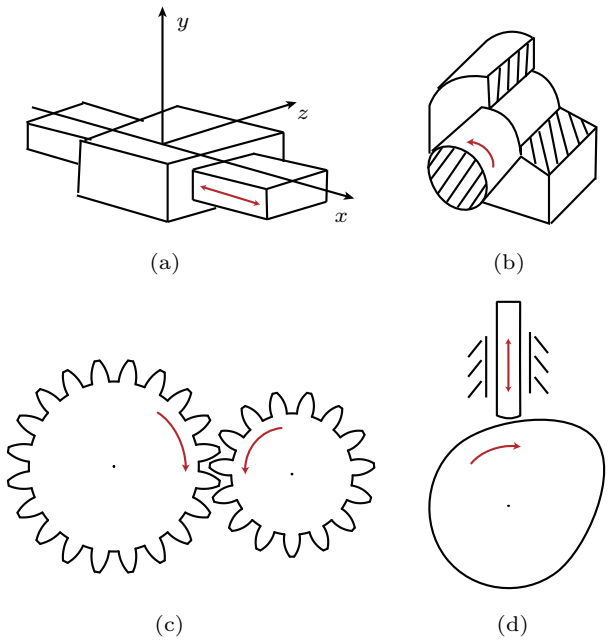


Fig.4. Joint type of planar institution. (a) Translational pair. (b) Revolution pair. (c) High gear pair. (d) Cam high pair.

We optimize the joint parameters to change the relative position of the parts in the group, to make the group bounding box the smallest. We use the L-BFGS optimization algorithm[16] to optimize the volume of the parts group to be the minimum. In the implementation of the algorithm, the parts group bounding box volume is set as the optimization objective function. The gradient function is realized by numerical derivation, and the joint parameters in the parts group are used to construct the gradient vector. The optimization algorithm constructs the matrix based on the previous iteration results to compute the next optimization vector after each iteration. The objective function can be expressed as:

$$F = Vol(\sum_{i=1}^{n} Rot(\theta_i) + \sum_{j=1}^{m} Trans(\tau_j)),$$

where $Rot(\theta_i)$ represents the rotate θ angle of the i-th revolution pair relative to the original position, $Trans(\tau_j)$ represents the translate distance τ of the j-th translational pair relative to the original position, and $Vol()$ represents the size of the bounding box of the group after translation or rotation of the joint. The objective function is not a monotonic function since the rotation function exists. It may fall into the local minimum when using a single L-BFGS optimization algorithm. Because the result of the optimization algorithm has a great relationship with the initial values, we use multiple iterations to obtain the minimum value. That is to say, in order to get the minimum volume, we adjust the initial value of the joint parameters in the group, and use the L-BFGS algorithm to optimize several times. Fig.5 shows that the parts group's bounding box volume is optimized to the minimum by adjusting the joint parameters.

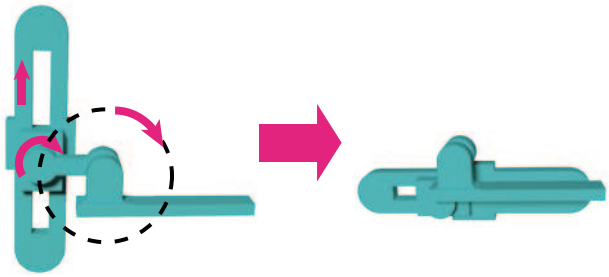
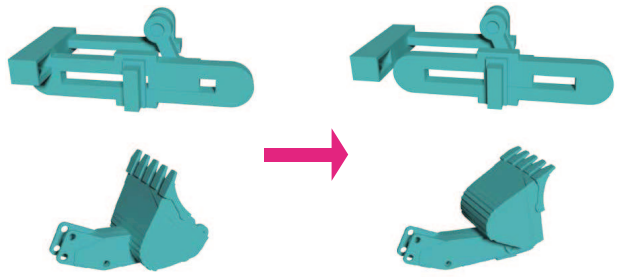


Fig.5. Group optimization process.

However, if only the joint constraints are taken into account in the optimization process, erroneous collisions may occur between the parts in the group at the minimum volume. As shown in Fig.6(a), due to that the geometric constraint of the part is not considered in the volume optimization stage, there is an erroneous collision in the optimization result of the parts group. In order to avoid the collision between parts, this paper adopts the collision detection algorithm[17] to carry out the collision detection of the parts in the optimization, and adds the collision penalty strategy in the objective function to ensure the optimization result is correct. The collision detection is performed after each volume optimization. If a false collision condition is detected, the volume of the collision is calculated and the volume is used as the basis for the collision penalty in the objective function. If no collision has occurred, the optimization is continued. When the collision occurs, the objective function value becomes larger and the optimization function is optimized in the opposite direction. Therefore, the optimization function of the parts group can be expressed as:

$$F = Vol(\sum_{i=1}^{n} Rot(\theta_i) + \sum_{j=1}^{m} Trans(\tau_j)) + \lambda Vol_{\text{collision}}(G),$$

where λ is a constant. In order to ensure the sensitivity of collision detection, λ is set to a very large value, i.e., $\lambda = 1e + 5$ is the default. $Vol_{\text{collision}}(G)$ represents the collision volume. Fig.6(b) shows the correct optimization result after adding the collision detection constraint.



(a) (b)

Fig.6. (a) Optimization without the geometric constraint. (b) Optimization with the geometric constraint.

6 Packing Groups into a Container

In the packing process, the objective function is the space utilization of the packing container. The calculation formula of space utilization is: $E = \frac{\sum_{i=1}^{n} Vol_{Part_i}}{Vol_{\text{packing}}}$, where $\sum_{i=1}^{n} Vol_{Part_i}$ indicates the volume of all the parts of the model. Vol_{packing} indicates the packing container volume V. Since the total volume of all the parts of the model is fixed, the space utilization E is inversely proportional to the packing container volume.

The packing problem is a kind of NP-hard problem. In order to make the NP problem be a solvable problem, we set the size of the bottom of the packing container in advance, and then increase the height to the minimum. Further we transform the minimum volume problem into the minimum height problem.

When we traverse the search tree in depth, it is necessary to pack the groups for the node that meet the packaging conditions and test their space utilization. The size of the bottom of the container is pre-specified according to the maximum length and width of each parts group in the groups set so that the height of the container can be minimized. When packing groups into the container, their packing order also affects the final packing space utilization. In [18], it is found that if the parts were placed in a random order, the difference of the space utilization in different packing orders would be as high as 70%. If parts are placed in descending order by size, the space utilization difference is only 20%. Therefore, before packing the groups set, we need to sort the groups by their volumes.

6.1 Packing Individual Parts Group into the Packing Container

When packing a group g_i into the container, we find the least costly position by "rotation-translation" method. The rotation matrix is represented by a quaternion, denoted by $\boldsymbol{Q}$, and the translation vector is represented by a three-dimensional vector $\boldsymbol{V}$. In the outer loop, we use the quaternion to rotate the parts group to an orientation, and then use the translation vector to find the least costly position in the inner loop. When traversing the container space to get the translation vector of the least costly position, our strategy is as follows: finding out whether there is a position where the height h of the packing volume is not increased:

- if such a position exists, packing the group g_i to the position;
- if such a position does not exist, selecting the position where the increase of height h is the smallest;
- if there is more than one position where the increased heights are the same, selecting the position where the free space is the smallest under the group.

According to the strategy above, the cost of each position can be calculated by the following formula:

$$Cost(g_i) = \Delta_h B + \int_{\Gamma} (\lfloor g_i(x) \rfloor - \lceil P(x) \rceil) \partial x,$$

where Δ_h indicates the increased height, and B indicates the container volume. $\lceil P(x) \rceil$ is the upper point of intersection of the packed groups P and the vertical line defined by $X = < x, y >$. $\int_{\Gamma} (\lfloor g_i(x) \rfloor - \lceil P(x) \rceil) \partial x$ indicates the free space volume below group g_i, where Γ is the 2D projection area on the bottom of the container, and $\lfloor g_i(x) \rfloor$ is the minimum height where the vertical $X = < x, y >$ intersects the group g_i. In order to more easily calculate the free space, we convert parts and packaging containers into voxels, using a set of uniform discrete points to represent the geometry of the part.

6.2 Packing Algorithm

When the packing algorithm is executed on a mechanical model that has been split into multiple parts groups, the bottom of the packing container should be first calculated by the bounding box for the minimum volume of all groups. Then we can pack the sorted parts groups into the packaging container in turn. For each parts group operation, the group is rotated first in the outer loop, and the rotated group is translated to find the least expensive position in the inner loop. When all the groups are placed, the maximum height of the groups' top surface $\lceil P(\mathrm{x}) \rceil$ is the height of the container. Finally, the space utilization of the packing result is calculated based on the volume of the container.

Algorithm 1 shows the process of packing groups into a container. It packs the mechanical model that has been split into multiple parts groups into the container with the minimum volume. We use R to represent the pre-generated quaternion list, and use $R_q(g_i)$ to represent the rotated group g_i. In Algorithm 1, the rotation matrix is represented by a quaternion, denoted by $\boldsymbol{Q}$, the translation vector is represented by a three-dimensional vector $\boldsymbol{V}$, I is a rasterized 3D image for container B, and H is the added height.

Algorithm 1. Packing Groups into a Container

```
Calculate the bottom of the packing container by the axis-aligned bounding box for the minimum volume of all groups
Create a rasterized 3D image for container B
Put the sorted groups into the packing queue Q
While Q is not empty do
  Pop the first part from Q (let it be g_i )
  For all q ∈ R do
    For all v ∈ ⌈P(x)⌉ do
      v̂ = v + s, where s is minimum vertical translation that makes R_q(g_i) rasterized
      If a valid s exist then
        Compute the cost
      End if
    End for
  End for
  If one or more positions could be found then
    Actually roto-translate using Q and V that minimize the cost
  Else
    Terminate with failure
  End if
  Rasterize in I and update H
End while
Compute the space utilization of the packing box
```

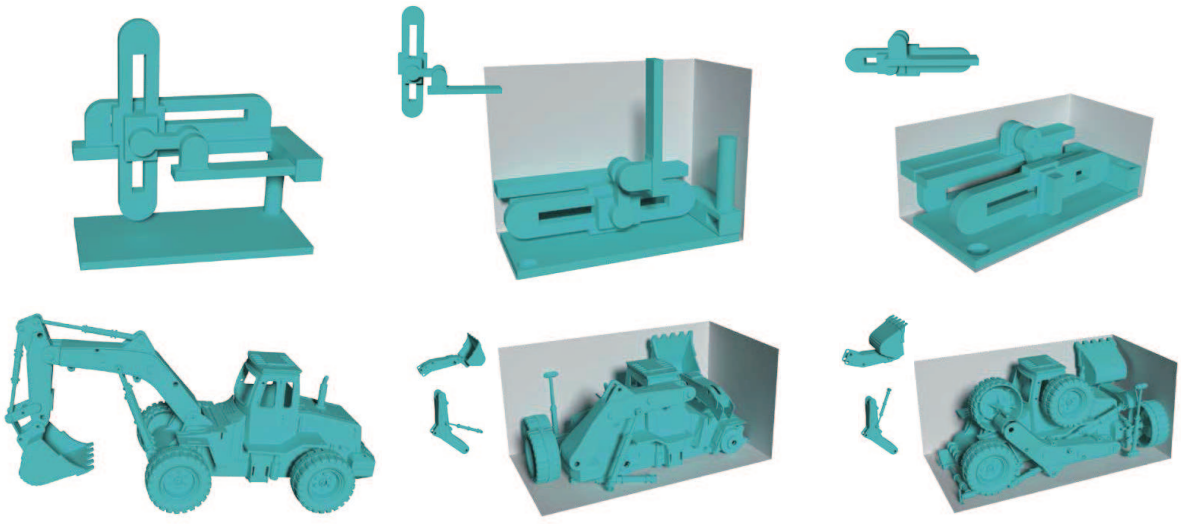
7 Results and Discussion

We implemented a prototype program to validate our algorithm. Experiments were run on a desktop PC with Intel® Xeon® E3-1231 v3, 3.40 GHz CPU with 8 G RAM. We selected four mechanical models to test our algorithm. In these experiments, the validity of the algorithm is verified by adjusting the parameters in the algorithm. Table 1 lists the parameters of the mechanical models.

Table 1. Parameter Information of the Experimental Mechanical Models

Model	Number of Parts	Number of Joints	Space Utilization Without Split (%)
Crank slider	9	9	13.6
Excavator	32	37	6.2
Robot arm	18	21	9.4
Motorcycle	29	28	5.8

Joint Parameter Optimization in Parts Group Volume Minimization. Joint parameter optimization can minimize the volume of the split parts group, and avoid wasting space in the packing phase. We used the L-BFGS optimization algorithm to minimize the group volume, and used multiple sampling iterations to optimize the group volume, which can avoid local optimal result. In this experiment, 10 iterations are enough to obtain optimal results. As shown in Fig.7, in order to verify the effectiveness of the joint parameter optimization, we tested two sets of mechanical models in the



(a) (b) (c)

Fig.7. Packing results of Attene's algorithm[5] and ours. (a) Input mechanical models. (b) Packing results with Attene's algorithm. (c) Packing results with our algorithm.

same way of separation and the same container bottom area, and compared our algorithm with the packing algorithm of Attene[5]. Table 2 lists the packing efficiency comparison of the two packing algorithms. The experimental results show that the packing algorithm of optimizing the joint parameters is superior to packing results with unmodified joints.

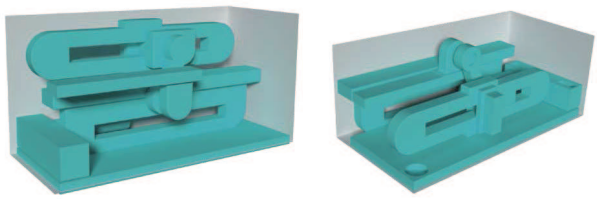
Table 2. Comparison of Packing Result Between Attene's Algorithm and Ours

Model	Crank Slider(%)	Excavator(%)
Attene's Algorithm[5]	11.95	9.1
Ours	25.10	10.7

Quaternion List. When packing a group into the container, the minimum position is found by "rotation-translation" method, and the quaternion of the rotation matrix is pre-defined. In the algorithm proposed by Attene[5], the quaternion list is pre-defined by a random method. However, because the geometric shape of the mechanical model is regular, when rotating the model, we set the three axes of the mechanical model as the rotating axes, and the rotation angles are selected to be 90, 180 and 270 degrees to make the rotated axes be still on the same line of the original axes. It avoids the tilt situation. Moreover, the more the quaternion lists, the longer the program running time.

Bottom Parameter Adjustment. In this algorithm, in order to approximate the NP-hard packing problem, we used the pre-set bottom size and then minimized the packaging height. The setting of the bottom size also affects the space utilization of the packing result. Therefore, we needed to find a suitable way to set the bottom. In the algorithm of Attene[5], the container bottom size is calculated according to the bottom of the axis-aligned bounding box of the model, and the bottom size is set to be the stretching of the bounding box bottom with stretch ratios of 0%, 25%, 50%, and −25%.

In this paper, using the axis-aligned bounding box of the mechanical model is not a suitable way because of the presence of the rotation or translation joints. Thus, we set the bottom size of the container based on the groups that the joint parameters have been optimized, and telescoped the bottom size using stretch ratios of 0% and 25%. As shown in Fig.8, the packing space utilization with 0% expansion of the bottom is 23.3%, while that with 25% expansion is 25.1%. Finally, we selected the latter scheme as the best experimental results.



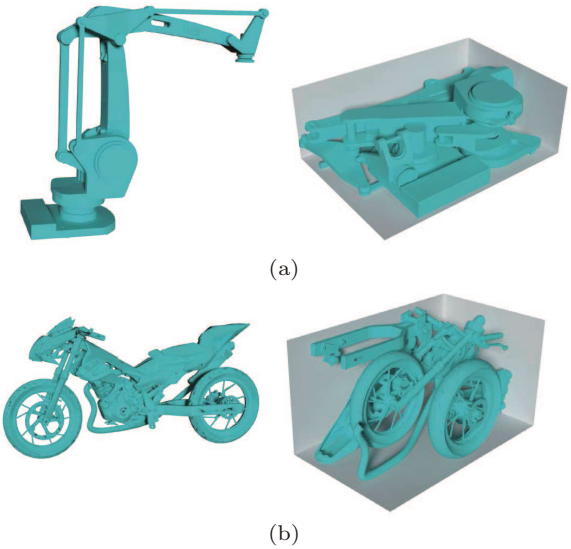
(a) (b)

Fig.8. Comparison of the packing results with different bottom sizes. (a) Bottom without extension. (b) Bottom with extension of 25%.

Space Utilization. Space utilization is the global optimization objective function for solving the packing problem. The larger the target space utilization setting, the more the times we need to split. When traversing the search tree in depth, if a node's packing space utilization is greater than the target space utilization, we continued to visit its child node; otherwise, we recorded the node's split program, and returned to the parent node to visit other branches. In the experiment, we adjusted the target space utilization to test the impact on the number of splits and the volume of the package. We experimented with the robotic arm model and the motorcycle model. The target space utilization and the final package space utilization were shown in Table 3. Fig.9 shows the packing results.

Table 3. Target Space Utilization and Final Package Space Utilization

Model	Target Space Utilization (%)	Number of Original Parts	Number of Final Parts	Final Space Utilization (%)
Robot arm	20	18	6	24.00
Motorcycle	10	29	5	10.14



(a)

(b)

Fig.9. Packing results. (a) Robot. (b) Motorcycle.

Running Time. The running time of our algorithm is related to the number of parts of the model, the number of joints, the pre-set quaternion list, and the size of the bottom. Table 4 is the running time for the four models in the experiments. The number of sampled quaternion is 10, and the bottom size is based on the maximum length and the maximum width of the group set. Usually, 0% and 25% stretch of the maximum length and width are used respectively to be the bottom size in the algorithm respectively.

Table 4. Running Time List

Model	Number of Original Parts	Number of Final Parts	Packing Efficiency (%)	Running Time (s)
Crank slider	9	5	38.00	12
Excavator	32	13	11.50	577
Robot arm	18	6	24.30	102
Motorcycle	29	5	10.14	111

8 Conclusions

In this paper, we proposed a mechanism packing algorithm which considers the joint DOFs. The algorithm is able to obtain the optimal packing efficiency with a small number of splits. Experiments showed that the algorithm can be applied to various kinds of mechanical models packing problems.

However, we just took the space utilization as the optimization objective. The force and the joint weight of the model are not considered in the algorithm. In future, we will analyze the joint force of the mechanical model in order to avoid the small joints in the separated parts and make the split groups more stable.

References

[1] Martello S, Pisinger D, Vigo D. The three-dimensional bin packing problem. *Operations Research*, 2000, 48(2): 256-267.

[2] Bansal N, Han X, Iwama K, Sviridenko M, Zhang G. A harmonic algorithm for the 3D strip packing problem. *SIAM Journal on Computing*, 2013, 42(2): 579-592.

[3] Fanslau T, Bortfeldt A. A tree search algorithm for solving the container loading problem. *INFORMS J. Computing*, 2010, 22(2): 222-235.

[4] Bortfeldt A. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 2012, 39(9): 2248-2257.

[5] Attene M. Shapes in a box: Disassembling 3D objects for efficient packing and fabrication. *Computer Graphics Forum*, 2015, 34(8): 64-76.

[6] Chen X, Zhang H, Lin J, Hu R, Lu L, Huang Q, Benes B, Cohen-Or D, Chen B. Dapper: Decompose-and-pack for 3D printing. *ACM Trans. Graph.*, 2015, 34(6): 213:1-213:12.

[7] Vanek J, Galicia J A, Benes B *et al.* PackMerger: A 3D print volume optimizer. *Computer Graphics Forum*, 2015, 33(6): 322-332.

[8] Kim K Y, Manley D G, Yang H. Ontology-based assembly design and information sharing for collaborative product development. *Comput. Aided Des.*, 2006, 38(12): 1233-1250.

[9] Mitra N J, Yang Y L, Yan D M, Li M, Agrawala M. Illustrating how mechanical assemblies work. *Commun. ACM*, 2013, 56(1): 106-114.

[10] Xu M, Li M, Xu W, Deng Z, Yang Y, Zhou K. Interactive mechanism modeling from multi-view images. *ACM Trans. Graph.*, 2016, 35(6): 236:1-236:13.

[11] Coros S, Thomaszewski B, Noris G, Sueda S, Forberg M, Sumner M, Matusik W, Bickel B. Computational design of mechanical characters. *ACM Trans. Graph.*, 2013, 32(4): 83:1-83:12.

[12] Egeblad J, Pisinger D. Heuristic approaches for the two- and three-dimensional knapsack packing problem. *Computers & Operations Research*, 2009, 36(4): 1026-1049.

[13] Wu Y, Li W, Goh M, de Souza R. Three-dimensional bin packing problem with variable bin height. *European Journal of Operational Research*, 2010, 202(2): 347-355.

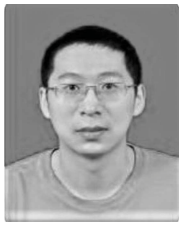
[14] Gomes A M, Oliveira J F. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 2006, 171(3): 811-829.

[15] Crainic T G, Perboli G, Tadei R. Extreme point-based heuristics for three-dimensional bin packing. *INFORMS J. Computing*, 2008, 20(3): 368-384.

[16] Morales J L, Nocedal J. Remark on algorithm 778: L-BFGS-B: Fortran subrou tines for large-scale bound constrained optimization. *ACM Trans. Math. Softw.*, 2011, 38(1): 7:1-7:4.

[17] Jiménez P, Thomas F, Torras C. 3D collision detection: A survey. *Computers & Graphics*, 2001, 25(2): 269-285.

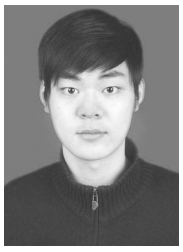
[18] Johnson D S. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 1974, 9(3): 256-278.



Ming-Liang Xu is a professor in the School of Information Engineering, Zhengzhou University, Zhengzhou, and currently is the director of CIISR (Center for Interdisciplinary Information Science Research). His research interests include computer graphics and computer vision. Xu got his Ph.D. degree in computer science and technology, Zhejiang University, Hangzhou, in 2011. He has now published more than 40 papers in international journals and conferences including ACM TOG, IEEE TPAMI, IEEE TIP, IEEE TCSVT, etc.

Ning-Bo Gu is a Master student in the School of Information Engineering, Zhengzhou University, Zhengzhou. His research interests include interactive mechanism modeling, mechanism disassembly and packing.

Wei-Wei Xu is a researcher in State Key Laboratory of Computer-Aided Design and Computer Graphics, Zhejiang University, Hangzhou, and is an Awardee of the NSFC Excellent Young Scholars Program in 2013. He has published around 60 papers on international graphics journals and conferences, including 16 papers on ACM TOG. Xu got his Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, in 2003.



Ming-Yuan Li is pursuing his Ph.D. degree in the School of Information Engineering, Zhengzhou University, Zhengzhou. His research interests include interactive mechanism modeling, mechanism disassembly and packing.

Jun-Xiao Xue is an associate professor in the School of Software, Zhengzhou University, Zhengzhou. His research interests include computer graphics and computer aided geometric design. He got his Ph.D. degree in computational mathematics from Dalian University of Technology, Dalian, in 2009.

Bing Zhou is a professor in the School of Information Engineering, Zhengzhou University, Zhengzhou. His research interests include computer vision and multimedia applications. Zhou got his Ph.D. degree in computer science and technology from Beihang University, Beijing, in 2003.