

Optimization of Auto Mesh Generation Using Genetic Algorithm

Rina Yamamoto, Yuki Takemoto, Yu Ishikawa, Kazuki Joe
Nara Women's University, Japan

In the manufacturing industry in recent years, CAE analysis is actively utilized as a method for evaluating product performance and reliability in advance. In order to perform CAE analysis, it is necessary to divide 3DCAD data into small rectangular elements. This element is called a mesh. It is difficult to automate because generation of meshes requires expert knowledge and craftsmanship based on empirical rules. So far mesh generation is done manually by experts. In order to optimize the mesh generation procedure and to generate meshes automatically, we propose a method to optimize using genetic algorithm.

1 Introduction

In the manufacturing industry, CAE (Computer Aided Engineering) that evaluates the performance and reliability of products in advance is proactively promoted. CAE analysis is a technique for evaluating and improving performance by an analysis method such as the finite element method (FEM) before problems occur. With the help of CAE, calculations that were impossible with conventional methods become possible, and more accurate performance prediction and reduction in calculation time are expected. As a work occupying the majority of the CAE analysis, we have mesh generation. Mesh generation is a work required to perform numerical analysis of 3D CAD data imitating the shape of a product. It is to generate data divided into small elements. This element is called a mesh. In order to perform mesh generation, various expert knowledges on CAE analysis such as analysis methods, finite element methods and the use of analysis software are required. The quality of the mesh is evaluated from the shape, flow, number and density distribution. Since the quality of generated mesh affects the analysis result, high quality mesh generation is required. Several methods [1] [2] [3] [4] have been studied for generating mesh automatically along regions, but manual correction is still indispensable for actual analysis. In order to correct the automatically generated meshes to get more appropriate meshes, the experience and intuition of a technician with expert knowledge are required. For these reasons, mesh generation requires huge cost. This is a major factor that affects the cost of the product. Therefore, development of a technique for automatically generating a optimized mesh that can be used for analysis is required.

In this paper, we use machine learning to optimize parameters of operations required for mesh modification. In addition, optimal operation procedures and optimal parameters are acquired for each operation, and better quality mesh is automatically generated. To generate the mesh, we use HyperMesh [1] of Altair Corporation, which is a specialized mesh generation software actually used in the CAE industry. For machine learning, we use a genetic algorithm (GA) that can find an optimal solution by giving appropriate evaluation.

The structure of this paper is as follows. In section 2, mesh quality, mesh generation method, and mesh generation software are described as basic knowledge of mesh generation. Section 3 describes coding methods of genes, mesh evaluation methods and application methods of GA. Section 4 describes experiments and their results to optimize each operation and operation procedures. Section 5 discusses the results.

2 Basic knowledge on mesh

2.1 Mesh quality

In the finite element method, it is well known that the larger the number of meshes, the higher the accuracy of analysis. However, the larger the data amount, the longer the analysis time becomes, which is not suitable for CAE analysis. In order to maintain the analysis accuracy while suppressing the data amount of mesh, it is necessary to generate high quality meshes. As the criteria for measuring the quality of the mesh, we know the number of meshes, the quality of the mesh shapes, the density distribution of the meshes, and so on.

First, the number of meshes is described. As the total mesh number increases, the analysis accuracy increases. But if it exceeds a certain number, the analysis result is saturated. Since the computation cost increases as the number of meshes increases, it is sometimes impossible to perform the analysis itself. The number of meshes must be determined in consideration of both the computation cost and the analysis accuracy required by the user.

Next, the quality of the mesh shape is described. As the criteria for measuring the quality of the shape, the angle of distortion and the aspect ratio are mentioned. We explain the calculation method defined by of a mesh generation software, HyperMesh [6]. The Angle of distortion from the angle

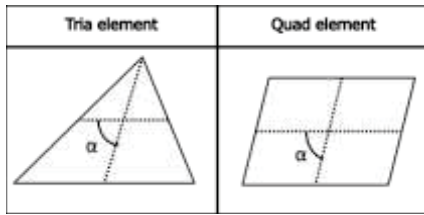


Figure 1 Calculate the Angle of distortion

shown in Figure 1 is calculated. Both triangle and square mesh are defined as better quality when α approaches 90° gradually. The aspect ratio is the ratio of the largest side against the smallest side, and the aspect ratio of 1:1 is considered as ideal. A mesh with a large difference between the maximum side and the minimum side is defined as poor quality. Although there are many other criteria for measuring the quality of the mesh shape, and their calculation methods are different, the ideal shape in all cases is the regular quadrangle or the equilateral triangle. Furthermore, in the finite element analysis, it is known that the solution accuracy and the quality are higher when the quadrilateral mesh is adopted rather than the triangular mesh. When generating quadrilateral meshes that satisfy the above-mentioned quality, it is difficult to fill square meshes in all the mesh generation regions and triangle meshes sometimes coexist. The smaller the proportion of the triangular meshes is included, the better the meshes are.

Finally, the density distribution of the mesh is described. High-quality meshes are not always uniform in density as a whole. In the portion where the stress concentrates or the shape is fine, meshes should be generated finely locally in order to obtain accurate calculation results. In some cases it is necessary to generate a mesh considering the density distribution of the meshes.

2.2 Mesh generation method

Many research results on mesh generation techniques have been reported [1] [2] [3] [4]. As main types of meshes, the shell mesh for generating meshes on the surface and the solid mesh for generating meshes in a solid are well known.

The bubble mesh method [3] and the advancing front method [4] are known as methods for generating meshes in the region. The bubble mesh method is a method of filling a spherical object in an area and connecting its center points to automatically generate a square mesh. The advancing front method is a method of arranging lattice points from the edge of the area toward the center, interpolating with meshes of quadrilaterals or triangles, and generating a mesh to move forward. There is also a technique to improve the quality of the generated mesh. The technique includes a method of finely dividing meshes such as Delaunay methods [5] and a method of reducing the number of meshes such as Edge Collapse [6].

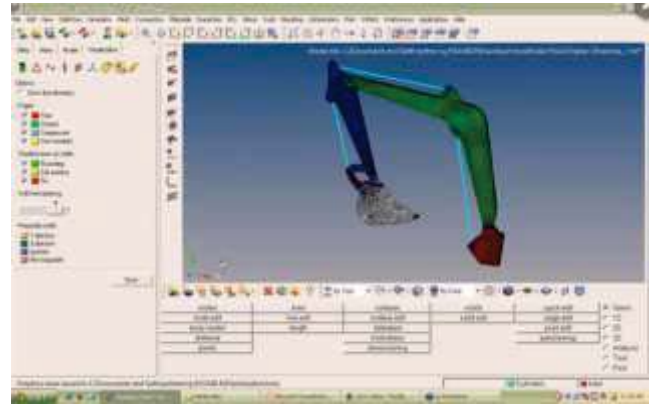


Figure 2 The operation screen of HyperMesh

In addition, software systems containing these mesh generation methods have been developed and it is possible to automatically generate meshes. However, the manual correction is indispensable in order to use the meshes for analysis.

2.3 Mesh generation software

HyperMesh is actually used in the manufacturing industry as automatic mesh generating software and is one of FEM analysis modeling software. It is a sophisticated finite element modeling preprocessor for streamlining the CAE workflow in product development. Figure 2 shows an operation screen of HyperMesh. Usually functions are selected and applied using the mouse or the keyboard to generate a mesh. HyperMesh provides a powerful shell and a solid mesh generation technique. Mesh generation is performed by reading 3DCAD data and applying various methods by full automatic or manual control. HyperMesh also provides operations using APIs, and VisualStudio is one of the environments to be used in conjunction.

A shell mesh generation operation for generating a mesh on the surface is described. There is a function Automesh for automatically generating meshes in the selected range. As a result, meshes are generated in the selected range by using the provided mesh generation method. However, conditions for mesh quality vary from user to user, so it is necessary to modify the meshes by manual control. There is a function QualityIndex for checking the quality of the whole or partial meshes. Meshes with poor quality such as mesh aspect ratio and angle of distortion are detected. In order to manually modify poor quality meshes, functions DeleteEntity, CreateNodes, EditElement and ReplaceNodes are available. An example of shell mesh modification is shown in Figure 3. In (1) to (3), a poor quality mesh is selected to be deleted. In (4) to (6), the required *node* is created and 4 *nodes* are selected to repeat the mesh creation. By operating in this way it is possible to modify the mesh by manual control. It is also possible to thin out the number of meshes using the function of combining nodes.

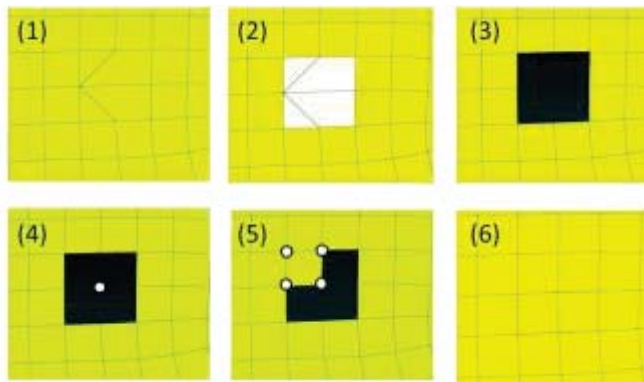


Figure 3 An example of shell mesh modification

In this way, HyperMesh has many functions to modify the mesh, but the functions described above are just a part. It is extremely difficult for a novice user to generate appropriate meshes. To understand the necessary functions and their effects to operate accurately, much time and effort are required.

3 Optimization of Auto Mesh Generation Using Genetic Algorithm

3.1 Coding method

HyperMesh is usually operated on the screen with a mouse and/or a keyboard. Since optimization of HyperMesh operation procedures should be controlled by programs, the operation procedures must be given as command lines rather than interactive operations using a mouse and a keyboard. During the interactive HyperMesh operation, the history is kept to be saved in a file `command.tcl`. From this history, the syntax for each operation is obtainable to create command lines to send the operation procedure to HyperMesh. By using the command lines as input and the history file as output, we get mesh data generated according to the procedure.

The operations for optimization in this paper are Remesh and ReplaceNode. We first construct two kinds of GAs: one for Remesh and another for ReplaceNode. Hereafter, the GA that optimizes each operation is called RemeshGA and ReplaceNodeGA. They are collectively referred to as operation GA. Furthermore, using the operation GAs, the operation procedure of mesh generation is optimized by another GA; it outputs optimum operation procedures and optimum parameters for each operationGA. Hereafter, the third GA is called integrationGA. The functions used in each GA and the coding method is described below.

Remesh is a function to reapply the Automesh function to the selected range. Depending on the selection range, the quality might be further worse than before in some cases. As a method of selecting the range, selecting elements or nodes with a mouse is used. The meshes selected by the method are recorded in the operation history as mesh IDs of the selected range, and Remesh is applied based on the mesh IDs. The mesh ID is the number assigned to each mesh at the time to generate meshes at the beginning. When the mesh is updated with Remesh etc., the mesh ID is also updated. Multiple mesh

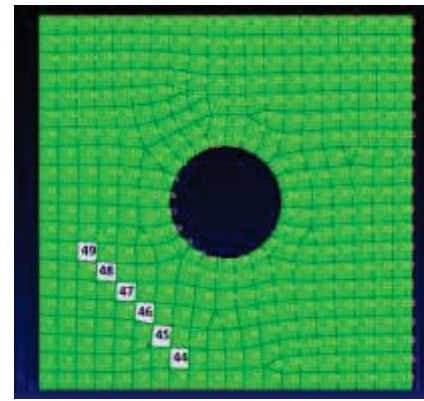


Figure 4 Selecting "44-49" meshes

IDs are selected with multiple numbers and spaces such as "24 8 78 9". Writing "-" between two numbers without writing spaces like "1-5", it is treated as the notation "1 2 3 4 5". Figure 4 shows the screen when selecting meshes with IDs of "44-49" in HyperMesh. We specify the selection range of Remesh by generating mesh IDs as random numbers for an initial individual of the RemeshGA.

ReplaceNode is a function of selecting two nodes to move the nodes to the center position of the two nodes or the second selected node position and combining them. It is used to delete unnecessary meshes. Like the above mesh IDs, a node ID is assigned to a node each time a mesh is generated. Two node IDs are selected by delimiting them with a space like "289 314". Like Remesh, node IDs are generated as random numbers for an initial individual of the ReplaceNodeGA. Since ReplaceNode usually corrects meshes to be applied multiple times, we assume that the operation is performed 10 times. However, in order to arrange a mesh by ReplaceNode, it is necessary to select a pair of adjacent nodes. Since we need to eliminate pairs of nodes not necessary for the learning out of all pairs randomly selected from all nodes, we use a mesh evaluation acquisition function Qualityindex described in section 3.2. We execute the ReplaceNode with a pair of randomly generated nodes and acquire the evaluation value (CompoundQ.I.) of the entire mesh with Qualityindex. This evaluation value represents the quality of the mesh and the quality of the mesh is worse as the value is larger. By this value, a pair of nodes which markedly deteriorates the quality of the mesh is eliminated and a pair of nodes to be used for the learning is selected. From the pair of nodes generated by this method, a pair of nodes is randomly selected for an initial individual to optimize with ReplaceNode. When overlapped nodes are selected for an individual the shape of the mesh is destroyed. To avoid this problem, the evaluation values of the individuals in which node overlapping has occurred are set to 0 so that mesh generation is not performed.

The integrationGA controls operationGAs. Remesh, ReplaceNode and "do nothing" are given arbitrary numbers, and these three are randomly selected to be represented as a one-dimensional array. An operation corresponding to the order of the array is executed one by one, the mesh to be obtained last is evaluated by the method described in section 3.2. When executing Remesh and ReplaceNode, evolutionary operations from the above coding array are selected to be

#	Criterion	Failure criteria	% failed	Worst	Q.I.	Worst elemId1	Worst elemId2	Worst elemId3
1	min size	2.50	0.00	3.33	0.07			
2	max length	7.50	0.54	8.00	8.08	144	161	
3	aspect ratio	3.00	0.00	1.92	0.00			
4	warpage	15.00	0.00	0.00	0.00			
5	max angle quad	140.00	0.00	125.28	0.01			
6	min angle quad	40.00	0.00	57.36	0.01			
7	max angle tria	120.00	0.00	88.25	0.01			
8	min angle tria	30.00	0.00	40.57	0.09			
9	skew	40.00						
10	Jacobian	0.60						
11	chordal dev	1.00						
12	taper	0.60						
13	% of trias	15.00						
compound Q.I. =		6.03						
# of quads =		362						
# of trias =		11						
% trias =		2.95						
# of failed =		2						

Figure 5 Text data by QualityIndex function

executed with optimal parameters. In other words, while performing evolutionary operations with integrationGA, operationGAs on individual genes are also executed. When changing a mesh with a function in Hypermesh, the mesh IDs and node IDs of the changed meshes are updated to other IDs. After updating IDs, if the next operation is performed in the same way, inconsistency may occur in the result. Therefore, it is necessary to reassign the IDs using Renumber function. Renumber is a function to reassign a numerical value designated by the user to an element with an ID in Hypermesh such as an element or a node of a selected range. Before executing operations for individual genes, the procedure of reassigning IDs with Renumber prevents inconsistency in results.

3.2 Mesh evaluation acquisition method

In our system, we obtain evaluation values of meshes using QualityIndex function in HyperMesh. This is a function to calculate whether each mesh satisfies the criterion based on the default or the quality index specified by the user in advance. It is possible to calculate the values representing various qualities and store them in a text file. Figure 5 shows an example of preserved text data. In the text data, the evaluation values of the entire meshes are written. It describes the aspect ratio of the meshes and the angle of distortion, the maximum interior angle and the minimum interior angle of meshes, the ratio of triangle and square meshes etc. In general, in the case of square dominant meshes, smaller percentage of triangular meshes is the good mesh. In this paper, we set the evaluation value of each individual to the ratio of triangular meshes. The number of triangular meshes is written in the text data. When the number is acquired, they are recalculated so that the evaluation value becomes higher as the number is smaller, and the individual is evaluated.

3.3 Application of GA

We obtain evaluation values of the mesh from the text data using QualityIndex function. The evaluation values are given to individuals and each evolutionary operation is performed to

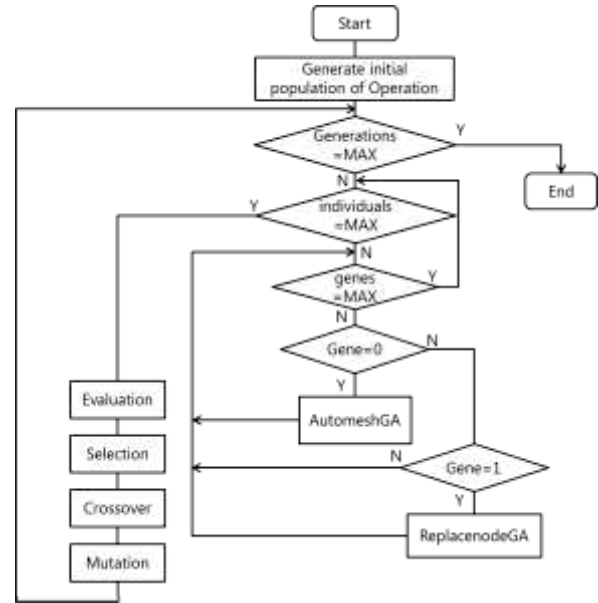


Figure 6 IntegrationGA Flowchart

optimize the operation procedure. The procedure of GA is as follows.

- step 1. Individual generation
- step 2. Evaluation of individual
- step 3. Parent individual selection
- step 4. Crossover of genes
- step 5. Gene mutation

In step1, each GA has different gene expression. In RemeshGA, each gene shows the mesh IDs to be selected. Remesh is performed on the selected mesh IDs. In ReplaceNodeGA, each gene represents the number of the sequences in which the adjacent node pair is stored. When an adjacent node pair is acquired from the corresponding number ReplaceNode is executed. The number of genes is the number of times to execute ReplaceNode. The procedure of the GAs after Step 2 is the same. In step 2, mesh generation is performed one by one, evaluation values are calculated from the acquired evaluation data, and the evaluation values are given to each individual. In step 3, two individuals to become parents are selected from the population. In step 4, gene recombination is carried out with single point crossover from the two parent individuals selected in step 3. In step 5, genes are mutated with a certain probability. This process is executed to prevent the learning from falling into a local solution. With the above procedure, individuals of the next generation are created. After that, evolution operation is performed by repeating steps 2 to 5 as many times as the number of generations.

We explain the procedure of the integrationGA. Figure 6 shows the flow chart of integrationGA. The integrationGA optimizes the operating procedure in Hypermesh. In order to express the mesh generation operation, genes are randomly generated from the number calculated by adding 1 to the number of operations to be selected. In this paper, Remesh is set to 0, ReplaceNode is set to 1, and the operation that “does

nothing” is set to 2. Therefore, an array with the size of gene lengths where individuals consist of 0, 1 and 2 is taken as an (integrated) individual. We explain the evaluation procedure of individuals. It is necessary to execute the corresponding operation in order from the beginning of the array with GA optimized parameters. When the gene is 0, RemeshGA is executed: Remesh with optimized parameters. The executed mesh data is saved to read the next gene. When the gene is 1, ReplaceNodeGA is executed, mesh generation is performed as in the case of 0, and data is saved to read the next gene. When the gene is 2, do nothing and move on to the next gene. When a mesh is changed by any operation, the mesh ID and node IDs of the changed part are updated and the IDs had to be reassigned before the operation of the next gene is executed. After completing the manipulation of the genetic component of one individual, the last stored mesh is evaluated in the same way as in operation GAs and it is used as the evaluation value of the individual. Hereafter, from the evaluation to the mutation of the gene, it is the same as the procedure of the operation GA. Evolutionary operation is done by repeating this for each generation. By recording the genes of the individuals from the final generation and the optimized parameters, we acquire the optimized operation procedure and the corresponding parameters.

4 Experiments

4.1 Experiment method

In order to optimize parameters for the operation in Hypermesh, we perform experiments of RemeshGA and ReplaceNodeGA. In addition, we perform an integrationGA experiment to optimize operation procedures. Mesh data used as input data in each experiment is described. It is CAD data of a plane with a circular hole opened in the square as shown in Figure 4. This data was created using Automesh to generate the entire mesh, and then QuickEdit was used to generate meshes around the circular hole. For the meshes around the hole, their mesh IDs and node IDs are assigned that do not change during the experiments. In the experiments, meshes except this part are optimized.

The experiment method of each GA is described. In RemeshGA, the number of genes of an individual is 10 and each individual is numbered in ascending order beginning from 1. A pair out of the ten numbers (genes) is selected to acquire the sequential numbers between the pair. For example, when the pair of "1" and "5" is selected, the sequential numbers of "1, 2, 3, 4, 5" are acquired. When we generate an initial individual with randomly numbered genes that are not arranged sequentially like "7 9 3 4 ...", the gathered selection range hardly appears. Therefore, no change occurs in the mesh and appears in the evaluation. Selecting genes with sequential numbers, it is easier for meshes to be gathered in a selection range. Using this method, we have more chance to increase the change in the quality of the mesh. The number of individuals is 100, the number of generations is 30, the choice of individuals is roulette selection, the crossover rate is 0.95, and the mutation rate is 0.05.

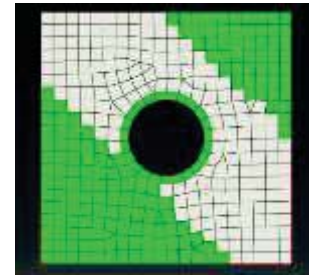


Figure 7 the selection range by RemeshGA

In ReplaceNodeGA, pairs of adjacent nodes are generated as described in section 3.1 before generating the initial individual. We generate 100 pairs of adjacent nodes. Each individual contains 10 genes and each gene contains 10 pairs of adjacent nodes out of the 100 pairs. The number of individuals is 100 and the generation number is 30, 100, and 200 for three cases. The selection method, the crossover rate and the mutation rate are the same as RemeshGA.

In integrationGA, the number of genes per individual is 10. As described in section 3.3, ten randomly selected genes with operation number 0, 1, or 2 are used, and operations of ten steps are expressed per individual. The RemeshGA executed in the integrationGA has 10 individuals and 10 generations. 100 pairs of adjacent nodes of ReplaceNodeGA are created. The the ReplaceNodeGA in the integrationGA has 10 individuals and 10 generations. In the integrationGA, the number of individuals is 10, and the number of generations is 10. The selection method, the crossover rate, and the mutation rate are the same as RemeshGA.

4.2 Experimental results

The experimental results of RemeshGA, ReplaceNodeGA, and integrationGA are described.

Figure 7 shows the result of the selection range output by RemeshGA. In the selection range, the white part is output from the RemeshGA. Figure 8 shows the changes before and after RemeshGA application to the selection range shown in Figure 7. As shown in Figure 8(a), the data before RemeshGA contains eleven triangular meshes while as shown in Figure 8(b), the applied mesh has seven quadrilateral meshes. The RemeshGA succeeds in generating quadrilateral meshes with a natural flow.

As for the results of ReplaceNodeGA, Table 1 shows the number of individuals, the number of generations, and the evaluation values of the experiments. We set the upper limit of the evaluation value 100 and the evaluation value is calculated by subtracting the number of triangular meshes from the upper limit. The evaluation value of the data before ReplaceNodeGA is 89. Running ReplaceNode on a pair of neighboring nodes in a lattice mesh increases the mesh of triangles by two. For this reason, the number of triangular meshes is increased from the data before the application of ReplaceNode to result in lowering the evaluation value for a while. However, As the generation progresses, the evaluation value is slightly improved. This is because there are more places to select pairs of adjacent nodes each time the

Table 1 Result of ReplaceNodeGA

individuals	generations	evaluation
100	30	74
100	100	75
100	200	76

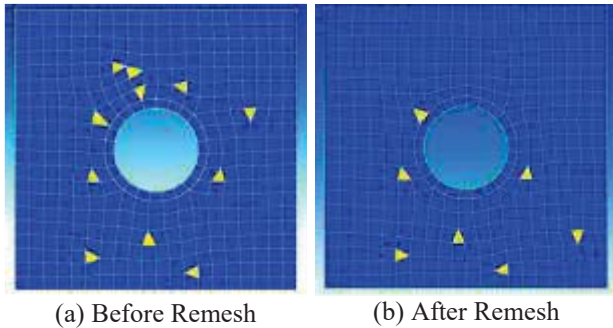


Figure 8 The changes before and after remesh

generation progresses. An example is shown in Figure 9. Figure 9(a) is the result of executing ReplaceNode on randomly selected node pairs. Two triangular meshes are added for each pair. Figure 9(b) is the result of executing ReplaceNode with selecting adjacent node pairs. The quadrangular mesh between them has been deleted. ReplaceNode is executed twice but the number of triangular meshes decreases by two.

In the integrationGA, Figure 10 shows the operation procedure and parameters of the execution results. "2200201002" arranged in order from top to bottom on the left end is an optimized operation procedure. Numerical values arranged to the right of numbers representing operations are parameters optimized for each operation. "0" indicates mesh IDs in selection ranges for Remesh, and "1" indicates pairs of adjacent node IDs for ReplaceNode. "2" indicates that nothing is executed and it is displayed as a dash sign. As a result, the final procedure is obtained: three times Remesh, one time ReplaceNode and two times Remesh. The evaluation value is 89. We perform the experiment several times under the same condition and find resultant procedures tend to contain a small number of ReplaceNode executions or even zero.

5 Discussion

5.1 RemeshGA

We presume that the evolutionary operation was performed efficiently by devising the individual generation method so as to increase the variation of the selection range and the difference in the number of triangle meshes set for the evaluation value also increases. So we succeed in the evolutionary operations in the GA and the experiment results show the success in optimizing the mesh generation operation. As for the shape of the selected range, the results show selected ranges often cover the area where triangular meshes

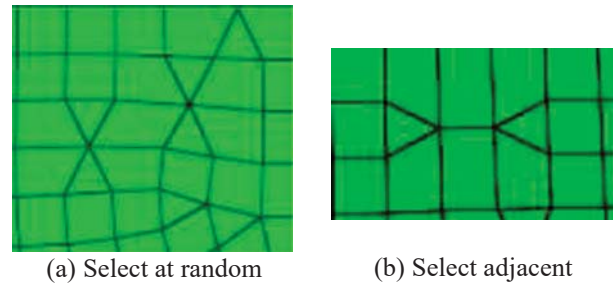


Figure 9 Change in result depending on selected part

```

2 -
2 -
0 33-301 317-243 312-264 229-244 280-47
0 123-309 173-288 201-34 328-143 172-294
2 -
0 229-334 320-56 84-181 235-189 171-150
1 242,243 63,64 18,19 146,147 154,155 281,282 8,9 2,3 239,240 309,310
0 157-311 263-195 114-287 17-191 231-171
0 289-171 321-100 142-75 217-15 141-224
2 -
    
```

Figure 10 The operation procedure and parameters of the execution result

are concentrated but also sometimes chose the area with meshes to be unnecessary to be modified. It seems that this is influenced by the arrangement of mesh IDs of input data and the selection method of genes. In input data, mesh IDs are allocated in the direction from the lower right to the upper left in order from the lower left mesh. As we generate individuals to select consecutive numbers, the result of the shape extending from the lower right to the upper left is obtained. In order to further optimize the operation, it is necessary to investigate more effective selection methods within any range not depending on the arrangement of mesh IDs.

In the experiment, we set the evaluation value to be higher quality as the number of triangular meshes is smaller. The Remesh sometimes increases triangle meshes according to the selection range. We presume that a wide range of results came out because in the experiment it was able to keep the quality with the wide range where meshes are intensively arranged. However, if data of another shape is used, even if succeeding in reducing the number of triangle meshes, there is a possibility that quadrilateral meshes with poor quality are generated. We think that it is possible to generate higher quality meshes by investigating new evaluation values for the quadrilateral mesh quality.

5.2 ReplaceNodeGA

We discuss the factors that greatly reduce the evaluation value. Applying the ReplaceNode once, the number of meshes of 0 to 2 triangles increases, and it is necessary to execute ReplaceNode at appropriate multiple positions in order to reduce the meshes. An example how to reduce the number of triangular meshes using ReplaceNode is shown in Figure 11. In Figure 11(a), the pairs of the nodes that are sequentially arranged from the triangular mesh toward the outside are selected. By executing ReplaceNode, it is possible to delete

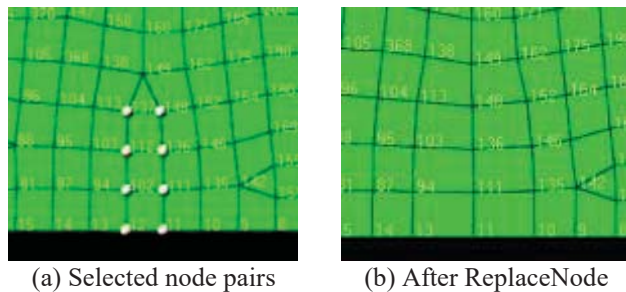


Figure 11 An example of ReplaceNode

the meshes between outer meshes and delete the triangular mesh as shown in Figure 11(b). In the process of GA, when an individual succeeds to make the selection as in this example, the evaluation value is just increased by 1 in the current evaluation method. Moreover, since another gene increases triangular meshes by 0 to 2, it is estimated that the desired evaluation value cannot be given and the learning efficiency is poor. It is necessary to efficiently improve the evaluation value when pairs of consecutive nodes are selected and to investigate the coding method so that pairs of consecutive nodes are easily selected.

5.3 IntegrationGA

By adding an operation of “do nothing” as an option, it is possible to freely change the number of operations without limiting it to the number of times corresponding to the number of genes. As a possible improvement to avoid bad operations, for example, a method of giving high evaluation values when a mesh can be efficiently generated with fewer times should be investigated.

We discuss the reasons why the unbalanced number of genes, namely a very small number of ReplaceNodeGAs, is found in the results. In this experiment, ReplaceNodeGA is the element that lowers the evaluation value, so it becomes difficult to appear as a gene in integrationGA. We think that it becomes possible to generate meshes with better quality by optimizing the operation to be selected as a gene.

In the experiment, there are only 2 operations for the option in the integrationGA. Nevertheless, sufficiently useful results are obtained. We think that by increasing the number of choices, a more diverse mesh generation procedure is obtained.

6 Conclusions

In this paper, we proposed an optimized mesh generation method using GA. In order to obtain highly accurate results in CAE analysis, it is necessary to generate high quality meshes. Many methods of automatically generating meshes have been studied, but manual correction is indispensable. In order to manipulate the mesh generation software to generate appropriate meshes, it is necessary to have an intuition based on empirical rules. Therefore, we proposed the operation GA which optimizes the parameters of mesh generation operations. In addition, we proposed the integrationGA that generates optimum parameters for each operation and the optimum operation procedure. In the experiment Hypermesh is used for

actual mesh generation and the operation is automated with visualstudio. As operation GAs, we proposed RemeshGA and ReplaceNodeGA that optimize the parameters of the Remesh function and the ReplaceNode function of Hypermesh. We have implemented the proposed methods, and perform experiments of the operationGAs and the integrationGA. In RemeshGA, we succeeded in optimizing mesh generation by a method where the target range selects closely arranged individuals. In ReplacenodeGA, we succeeded in selecting pairs of adjacent nodes but the results did not converge well. It is necessary to improve. In integrationGA, we succeeded in freely changing the number of operations by adding the operations of “do nothing” as an option. From this result, it was confirmed that optimization of mesh generation is possible by devising the representation method for individuals and the way to give evaluation values.

In this paper, only the number of triangular meshes is defined as the evaluation value of GA. It does not include the criteria for measuring other mesh quality as described in section 2.1. In the future work, we think that it is possible to generate higher quality meshes by improving a method of setting evaluation values including other criteria such as mesh shape quality.

7 References

- [1] K.Shimada, Physically-based mesh generation : automated triangulation of surfaces and volumes via bubble packing, Massachusetts Institute of Technology,1993.
- [2] Parikh,P.,Pirzadeh,S.,Loehner,R., A Package for 3-D Unstructured Grid Generation, Finite-Element Flow Solution and Flow Field Visualization, NASA CR-182090, 1990.
- [3] Ho-Le K., Finite Element Mesh Generation Methods: a Review and Classification, Computer Aided Design, Vol.20,No.1,pp.27-38,1988.
- [4] M Onodera, I Nishigaki, Y Hiro, C Kongo, A Mesh Morphing Technique Involving Addition and Deletion of Features for Tetrahedral Meshes, Journal of Environment and Engineering vol.6 ,No.3, pp.595-603, 2011
- [5] (2018) Large Model Finite Element Preprocessing Altair HyperMesh [online].Available <https://altairhyperworks.com/product/HyperMesh>,
- [6] "Altair HyperMesh Training nyūmon Course"[Altair HyperMesh Training Introductory Course]. Altair Engineering, Inc.,2017. (in Japanese)

Acknowledgment

We would like to thank Integral Technology Co., Ltd for their technical support and important comments.