AUTOMATIC PIPE ROUTING TO AVOID AIR POCKETS

Y Ando and **H Kimura,** Naval Architecture and Marine Systems Engineering Course of Kyushu University, JP

SUMMARY

Pipe arrangement is one of the most time-consuming works in ship production because the process requires designers to decide the optimum pipe routes. Previous works focused on finding preferable routes by applying optimization methods, but these methods have not considered the effect of gravity in obtained pipe routes. This paper presents an automatic pipe routing method that avoids air pockets. We call vertical U-shaped pipes "air pockets". In this paper, the pipe routing problem is considered as a routing problem in a directed and weighted graph. Dijkstra's method is used in the routing process for generating candidates of optimum routes. In order to avoid making air pockets in the obtained routes, we try to use a new cost function. The performance of this method is shown in several demonstrations.

1. INTRODUCTION

Pipe arrangement is considered one of the most time-consuming works in the production of vessels. Designers have to make right decisions in order to generate optimum piping routes and suitable positions for equipment during the process. This paper presents a system to support the designers in decision making for piping layouts. Additionally, the proposed system has an ability to avoid making U-shaped pipes called "air pockets" in the obtained routes.

In a general piping design, pipers have to get information from PID and set equipment including pumps, valves, branches, pipe routes and others into the best position respectively. Nowadays, experienced engineers play an important role for designing the piping routes. The aim of our automatic piping system is to design the piping routes in less time while considering their heuristic methods. Additionally, the system takes account of pipe-racks, aisles, bending piping parts, and unfavourable routes such as detours and air pockets. The air pockets often cause serious damage to pipelines, so designers have to set drain traps at the vertical U-shaped points. In this paper, we focus on how to find the best routes for the pipeline while avoiding air pockets.

2. DESIGN OF PIPING LAYOUT

The piping layout problem includes two major tasks for the pipers. One task is to set each item in the right position respectively. The pipers have already decided locations of some equipment like pumps or electric generators in the early stage of the design. Therefore, these positions are not movable in the current design stage. However, the designers have to decide positions of other equipment like valves and branches. These positions strongly affect the quality of the piping layout because the ideal positions of movable equipment generate short piping routes and better operability of valves for crews. In previous works of the automatic design system, Burdorf et al. [1] proposed CAPD (= Computer-Aided Plant Design) in order to figure out a good layout of a chemical plant. The CAPD system is able to place equipment at the location following general requirements made in the system.

Another task for designers is to find the best piping routes connecting the equipment. The best route should be short and pass through pipe-racks. Moreover, the designers should be careful not to set the routes in aisle spaces. In the practical design, the pipers usually design the optimum piping routes one by one. In this process, designers start to find the best route, starting with the pipe with the largest diameter and finishing with the one with the smallest. The system also applies the order of routing that is used by the designers.

Many previous works have been done to solve the pipe routing problem. Ito [2] and Paulo et al. [3] used a genetic algorithm to find the best piping route. Both of their systems can deal with space like pipe-racks where piping routes are preferred to be set. Park et al. [4] also considered pipe-racks. Additionally, they proposed a new cell decomposition method to reduce the complexity of the computation. Asmara et al. [5, 6] proposed the DelftRoute system. The system used Dijkstra's method to generate the shortest piping routes. Similarly, our system also applies Dijkstra's method because it guarantees to find the optimum route in a network.

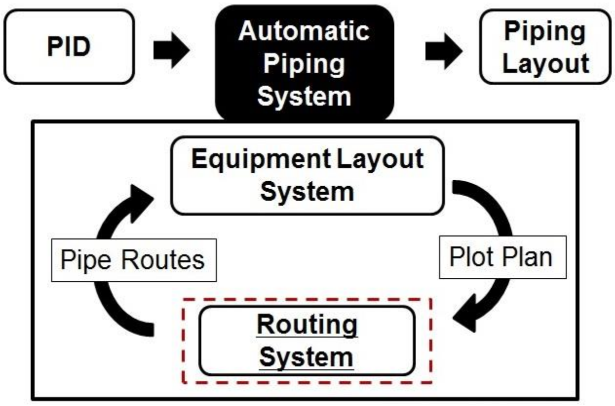

Figure 1: Flowchart of the automatic piping system

Figure 1 shows the process of piping design in the automatic system. We plan to set two modules in the automatic piping system in accordance with the practical design tasks. One is an equipment layout system that finds the right positions of movable equipment. Another is a routing system that finds the best route for each pipe. In our current stage, we have developed the routing system. In this paper, we present features of the system and algorithm to find practical routes in the piping problem.

2. AUTOMATIC ROUTING SYSTEM

2.1 ROUTING ALGORITHM

In solving the pipe routing problem, the designers have to find the most practical and suitable pipe route between start and goal points. The obtained piping route is needed to avoid interfering with obstacles or other pipes. On the other hand, the obtained route should pass through pipe-racks unless it makes an extreme detour in the route. In order to find the practical piping route quickly, we consider the pipe routing problem as the routing problem in a directed and weighted graph. Figure 2 shows the steps of our system to find the pipe route.

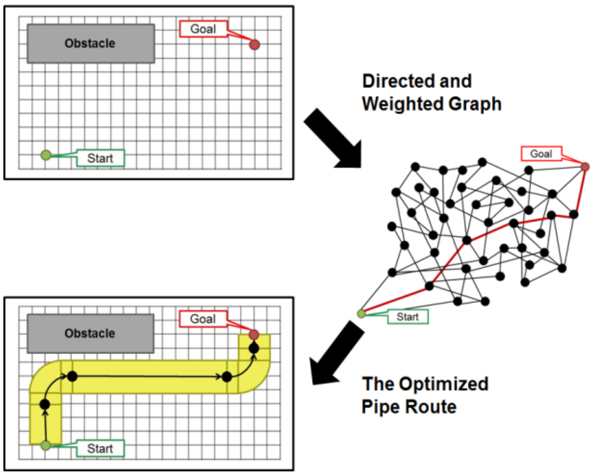

Figure 2: Routing steps in the system

As the first step, the system divides a design area into meshes. The size of mesh has to be inputted before beginning the search. When start and goal points are not on the mesh, the system adds extra meshes for those points. In the next step, the system makes a network with directed and weighted edges. The weight of edge is determined by the distance between two nodes. Additionally the weight also depends on whether pipe-racks or aisles exist in the space or not. The total weight of the shortest path represents features of the piping route. Details of how to set the weight are described in the next section. While making the network, the system also searches for the shortest path by using Dijkstra's method. Dijkstra's method is a graph search algorithm that solves the routing problem in a network. This algorithm is guaranteed to find the shortest path from a start to a goal point. The amount of computation is $O(n^2)$ at worst, where n means description length. After completion of the search, the final step is to change the shortest path in the network to the piping routes in the design area.

2.2 GRAPHING OF ITEMS

Routing patterns and items in the system are described in this section. The automatic routing system considers pipe pieces including straight pipes, elbows, and bending parts. The system also takes into account obstacles, pipe-racks, and aisles in the design area. More detailed information was described in our previous works [7, 8].

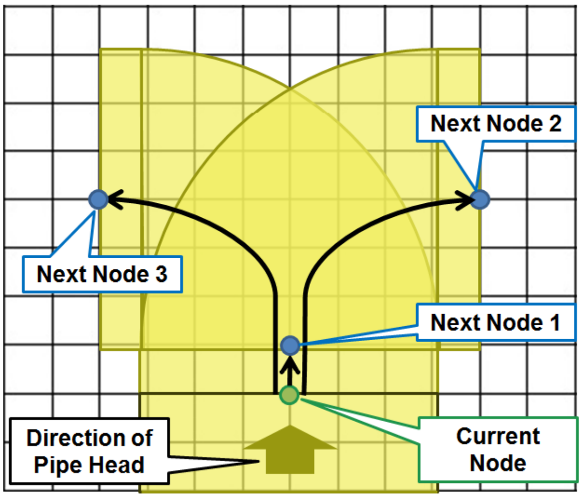

Figure 3: Transitions of nodes in a straight pipe and elbows

A straight pipe is the most general pipe piece in pipelines. During the graph search, the proposed system searches a next node from the current position along with the direction of the pipe. Next node 1 in Figure 3 indicates a candidate of a next node obtained by using the straight pipe. In this case, the weight of the edge in the network is the product of the length of the nodes and the diameter.

An elbow is a pipe piece that is useful for changing a direction of a route. The elbow usually changes the direction at right angles. In order to get enough space for the elbow, spatial requirements are generated. Therefore, vertical and horizontal gaps of the current and next node are larger than the radius of the pipe. Next, node 2 and 3 in Figure 3 are candidates of the next nodes when using the elbows. In considering the elbow, the weight of the edge is the product of Manhattan distance of the current and next node and the diameter of the pipe.

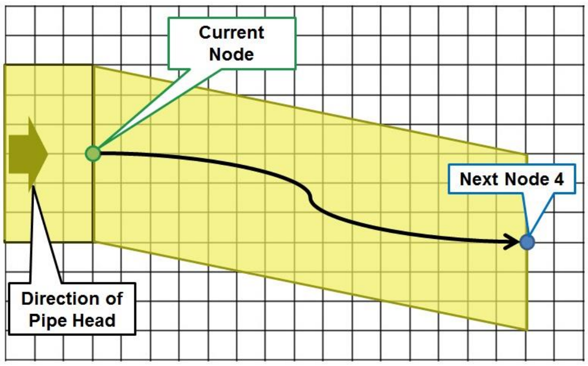

Figure 4: Transition of nodes in a bending part

The system also considers bending parts during the routing search. The bending part is the pipe piece running diagonally in the local design space. The part is useful for avoiding an extreme detour in the route. Figure 4 shows the transition of nodes when using the bending part. The cost of the edge is the product of the Manhattan distance and the diameter of the route.

Structural models like ship hulls, walls, and other pipes are simplified to reduce the complexity of the problem. These obstacles are represented as boxes and triangles in the system. If a transition of nodes interferes with any of the obstacles during the routing search, the system stops generating edges. In this way, the proposed system checks interferences whenever making edges, so that obtained piping routes are guaranteed not to interfere with obstacles.

In order to make obtained routes more practical, the system needs to consider pipe-racks and aisles in piping routes because these spaces strongly affect the maintainability and the safety for passengers. Pipe-racks are space where piping routes are preferred to be set. Designers usually set some routes in the space because it is easy for workers to fix and maintain the pipes. Additionally, the pipe-racks are also useful to reduce supporters of piping routes. In the proposed system, pipe-racks are represented as boxes. If a part of a piping route is set in the pipe-rack, the routing cost of a corresponding part is reduced in accordance with the discount value. In other words, the routing cost of the edge in the network is decreased when the corresponding piping piece is completely included in the pipe-rack. Therefore, an obtained piping route passes through pipe-racks unless it makes an extreme detour. Figure 5 shows a simple simulation of a pipe-rack. It is clear that the obtained routes are making detours in order to pass through the pipe-rack that is located at the bottom of the design space.

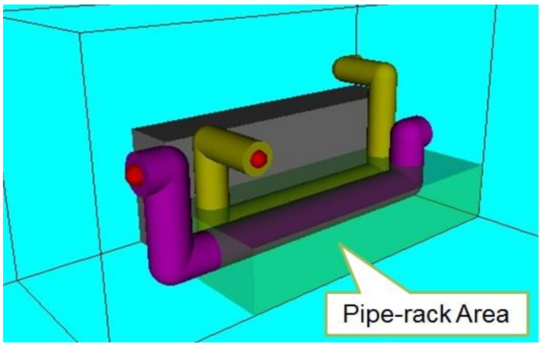

Figure 5: A simulation of a pipe-rack

Aisles are also important items in a design space when considering the safety for passengers. The aisles are represented as boxes in the same way as pipe-racks. However, when dealing with aisles, the routing cost of a piping piece is increased if the routing part interferes with aisles in accordance with the extra value. Therefore, the system searches the route that avoids aisles unless the routing costs are extremely high.

As described in the above sections, the routing costs include not only the length of a pipe and the cost of each pipe piece but also the rewards of pipe-racks and penalties of aisles. The system considers the shortest path in a network as the best piping route in the design space, which means the route with the lowest routing costs.

2.3 ORDER OF ROUTING

In a general piping design, pipers usually design piping routes one by one, starting from the pipe with the largest diameter. In order to generate practical piping routes automatically, the system also adopts the order of routing that is used by general designers. Additionally, when there are several pipes with the same diameter in a design space, it chooses one pipe randomly from them. As a result of the approach, the order of routing becomes one of the most important factors of the automatic design system because final piping routes definitely depend on it. We could verify the strong influence of the order through simulations. Details of the influence are discussed in section 4 of this paper.

3. ALGORITHM TO AVOID POCKETS

3.1 AIR POCKETS

During the practical design process, designers have to be careful of locations of piping routes and the effects of the routes as well. Some kinds of pipes possibly cause serious damage to pipelines. An air pocket is one of those undesirable piping routes. In this paper, we call vertically U-shaped pipes air pockets. The air pockets usually cause a malfunction of connected equipment because inner fluids like gasses or liquids often settle at the U-shaped points. Therefore, piping designers have to check whether or not air pockets exist in designed routes. Moreover, they also have to set drain traps at the corresponding parts in order to remove the settled fluids when they find pockets in the routes.

To generate practical piping routes, the automatic routing system includes two modules to search a piping route that avoids air pockets. After loading the necessary information, the system starts the routing search by using the restriction method. The restriction method searches a piping route without any pockets. When the system cannot find any route by using the restriction method, the system switches the routing algorithm to the penalty method. The penalty method finds a route that avoids air pockets unless the route makes an extreme detour. Details of the methods are described in the next sections.

3.2 RESTRICTION METHOD

The system starts to search piping routes with the restriction method after loading information about locations of equipment. This method restricts the vertical direction of the routing search in order to generate a piping route without any air pockets. For example, if a start point is located at a higher place than a goal point, the system searches nodes located at only horizontal and lower places than the start point. Additionally, the system also deletes any candidate of a piping route that makes a vertical U-shape. Through the approach, the restriction method decreases the number of node transitions. In other words, the size of the network is reduced by the restriction method. As a result of the method, an obtained route is guaranteed not to include any air pockets.

Another positive effect of this method is that the search time is decreased because the method restricts the size of the network. We could verify the decrease of the search time through several simulations. Figure 6 indicates the positive effects of the restriction method. As shown in the left side of the figure, the old system found the route with a pocket. On the other hand, it is clear that the obtained route of the new system succeeded to avoid the pocket. In addition, the new system could find the route in less time than the search time of the old system.

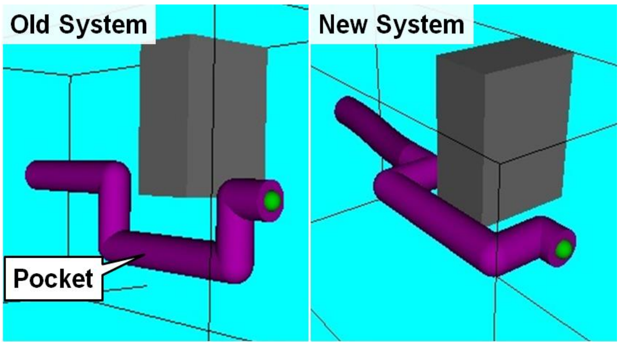

Figure 6: Piping routes obtained by the old system and the new system with the restriction method

However, as a negative point of the restriction method, it cannot deal with a piping route including air pockets. In a practical design of piping routes, there are often piping routes that cannot avoid air pockets. In such cases, the system changes the routing algorithm to the penalty method from the restriction method.

3.2 PENALTY METHOD

The penalty method is operated after the restriction method failed to find a piping route connecting its start and goal points. The failure of the restriction method means that the best piping route must include at least one pocket. In order to find the best route including at least one pocket, the penalty method needs to deal with air pockets in a different way from the restriction method.

The approach of the penalty method is to add penalties on edges connecting two nodes vertically. Unlike the approach of the restriction method, the penalty method takes into account all candidates of piping routes even if a candidate contains a vertical U-shaped pipe. As a result of the approach, the shortest path in a network usually avoids the edges with penalties unless the total weight of the route becomes extremely high. Therefore, the system can generate a piping route that avoids moving vertically unless it makes an extreme detour. The obtained shortest path is, in other words, a piping route that avoids air pockets as much as possible. Figure 7 shows an example of piping routes obtained by the penalty method.

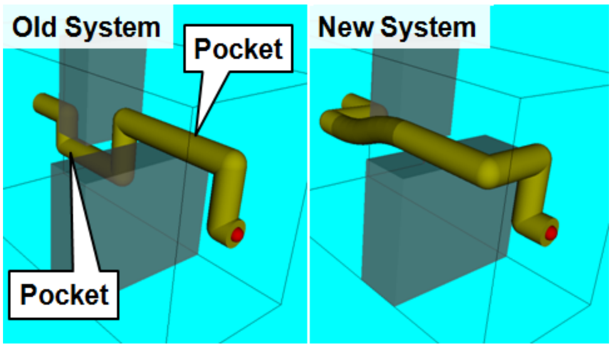

Figure 7: Piping routes obtained by the old system and the new system with the penalty method

As shown in the left side of Figure 7, the old system formed a piping route which included two pockets. Thus, designers have to design two drain traps in this case. On the other hand, the new system searched the route with one pocket. In addition, the air pocket is not an extremely but a gently curving pipe. In general, gentle curving pipe pieces are likely not to prevent inner fluid. Through this test, it was confirmed that the new system with the penalty method can generate desirable piping routes.

However, the penalty method has a poor search time. As described before, the penalty method deals with all candidates during the routing search. Moreover the method searches candidates linking horizontally as its priority, because the local shortest path usually avoids vertical connecting edges with added penalties. Therefore, the search time often becomes longer than that of the old system, especially in cases where start and goal points are separated in a vertical direction.

4. SIMULATIONS

4.1 TEST MODEL

The automatic routing system has been tested to verify its performance. The design space extends 6.0 [m] in each direction. The mesh size is 0.15 [m] in each direction. The system searches 13 piping routes while dealing with three obstacles, three aisles, and two pipe-racks. The diameters and number of piping routes are 0.9[m] x 1, 0.6[m] x 2, 0.4[m] x 4, 0.3[m] x 6. The discount value of the pipe-racks is 0.3 and the extra value of aisles is 2.0. When the system uses the penalty method as the routing method, the weight of a vertical connecting edge is doubled. Additionally, the system starts to search from the largest pipe and chooses one pipe with the same diameters randomly during the routing search. As the computing environment, we used Windows7 with Intel Core i7 3.4 GHz and 8.0 GB memory. Java version 1.6 was used as the programming language.

4.2 RESULTS

Figure 8 shows the result of the simulation. The piping routes in the diameters of 0.9[m], 0.6[m], 0.4[m] and 0.3[m] are coloured red, yellow, blue and green respectively. The blue boxes indicate aisles and the green boxes are pipe-racks. It took about 30 minutes for the automatic routing system to complete all routing searches. In the following sections, we discuss the following topics: a comparison of the old and the new systems, a comparison of different mesh sizes, the influence of the order of routing, and the validity of routing costs.

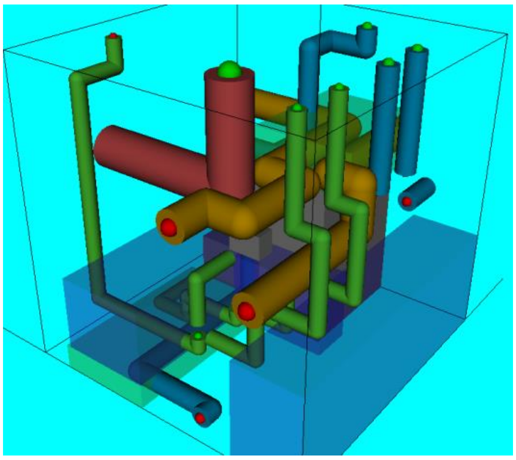

Figure 8: The result of the automatic routing search

4.3 DISCUSSIONS

4.3 (a) COMPARISON OF OLD & NEW SYSTEMS

Compared to the old system, the new system could find the piping routes in less time. In the test case, the old system needed about 45 minutes to find the result; however, the search time of the new system took almost 30 minutes to generate all piping routes. Through the simulation, the restriction method succeeded to find 11 piping routes for 13 pipes. The remaining two piping routes were searched by the penalty method. As described before, the restriction method can reduce the size of the network, so that the method can decrease the search time. This is the reason why the new system could find the result faster than the old system. Moreover, the number of pockets in the results of the new system was less than that of the old system. This is a favourable result because the first purpose of the new system is to avoid making pockets in an obtained piping route.

4.3 (b) INFLUENCE OF MESH SIZE

We tested the influence of different two mesh sizes. In the first case, the mesh size was 0.15 [m] and Figure 8 shows the result. The search time of the first case was almost 30 minutes. As the second case, we set the mesh size to 0.25[m]. In this case, the system needed almost 10 minutes to find all piping routes. However, the result was not good enough because the total length was longer than that for 0.15[m]. Additionally, the result in the second case included three more pockets as compared to that of the first case. From the two cases, it is confirmed that large mesh size reduces the search time and generates rough drawing of piping routes. On the other hand, when the mesh size is small, the system needs a long search time but it generates accurate piping routes. We need further investigations to find the best mesh size. As our future plan, we plan to add extra meshes around items already located before the routing search. Those extra meshes can help the system to search an accurate piping route in less search time.

4.3 (c) INFLUENCE OF LOCATED PIPING ROUTE

The system uses the order of routing that is currently applied by piping designers. It searches piping routes from the thickest to the thinnest. When there are several pipes with the same diameters, the system selects one pipe randomly from them. Therefore, this approach generates multiple different orders during the routing search. Figure 9 shows two different piping routes obtained by different orders. The numbers 1 to 3 indicate the order of routing. As shown in the figure, the obtained routes are quite different. Obviously, the obtained design on the right side is not preferred because the third searched piping route makes a detour. Through the simulation, we could verify that the orders of routing have a significant effect on the final piping routes because a different order generates different initial locations for the next routing. We need to find more practical rules of routing orders to reduce the random selections of the system.

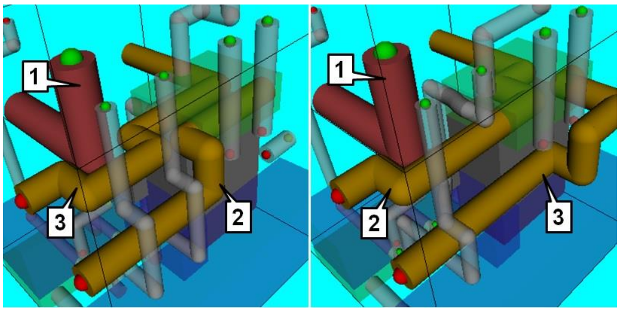

Figure 9: A comparison of different orders

Two results showing the strong influence of located pipes are displayed in Figure 10. The figure indicates that a small difference in a located pipe can generate an undesirable detour. In the test case of the figure, the difference of the right and left front yellow pipes is the position of the elbows. The routing costs of the two yellow routes are the same during the routing search although the two piping routes are actually different. When there are multiple routes holding the same routing costs like the test case, the system selects one route randomly from the candidates. However, the random choice often has a serious effect on the remaining routing searches as shown in the figure. Therefore, we have to develop a new approach to distinguish multiple routes holding the same routing costs while considering the remaining routings.

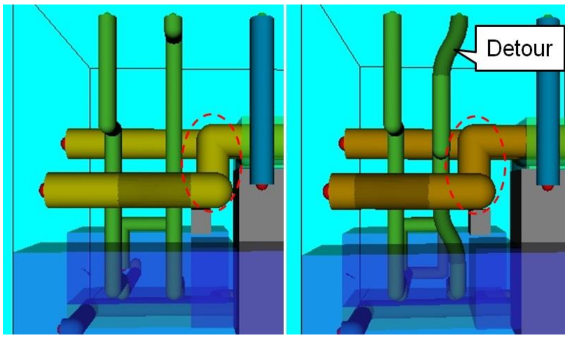

Figure 10: A detour caused by the located pipe

4.3 (d) VALIDITY OF ROUTING COSTS

The automatic system selects the best piping route in terms of routing costs. In the early stage of this work, we planned to use the price of pipe pieces as routing costs in order to represent a piping route correctly in a network. However, we think that the discount value of pipe-racks and the extra value of aisles can have a negative effect on the validity of the routing costs. Moreover, the penalty method makes the validity more doubtful because it adds penalty value to vertical connecting edges. Therefore, further investigations into cost function are needed in the future to arrive at optimum routing costs.

5. CONCLUSIONS

A new automatic pipe routing system was proposed in this paper. The obtained piping routes from the system are short and not winding to reduce the number of bending points. The obtained routes also avoid aisles and passage through pipe-racks as much as possible. In addition, the system considers avoiding air pockets during the routing search. Therefore, the obtained route does not include air pockets unless it makes an extreme detour.

The graph search in a network is applied in order to find the best piping route. In the network, the weight of each edge represents the length and features of a pipe piece, so that the shortest path in the network becomes a practical piping route in a design field. In addition, the system uses two routing algorithms to avoid making air pockets. The first algorithm is the restriction method. The method deletes candidates making U-shapes. Therefore, the obtained route by this method does not include any pockets. Another method is the penalty method. The penalty method adds penalties to edges connecting in a vertical direction; this generates a piping route that avoids air pockets as much as possible.

Through several simulations, it was verified that the new system is able to generate desirable piping routes automatically. In future tasks, it is necessary to improve on many practical aspects of the automatic piping system such as the order of routing, selecting a suitable mesh size, and developing an equipment placing system.

6. ACKNOWLEDGEMENTS

This paper is the result of work sponsored by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (B) 23360388. This support is gratefully acknowledged.

7. REFERENCES

1. A. BURDORF, B. KAMPCZYK, M. Lederhose, and H. SCHMIDT-TRAUB ‘CAPD-computer-aided plant design’, *Computers and Chemical Engineering*, 2004.

2. T. ITO and S. FUKUDA, ‘A genetic algorithm approach to piping route path planning’, *Journal of Intelligent Manufacturing*, 1999.

3. T. M. PAULO, and V. J. A. S. LOBO, ‘A Tool for Automatic Routing of Auxiliary Circuits in Ships’, *Portuguese Conference on Artificial Intelligence (EPIA)*, 2009.

4. J. H. PARK, and R. L. STORCH, ‘Pipe-routing Algorithm Development: Case Study of a Ship Engine Room Design’, *Journal of Expert Systems with Application*, 2002.

5. ASMARA, A. and U. NIENHUIS ‘Automatic Piping System in Ship’, *International Conference on Computer and IT Application (COMPIT)*, 2006.

6. ASMARA, A. and U. NIENHUIS ‘Automatic piping system Implementation’, *International Conference on Computer and IT Application (COMPIT)*, 2007.

7. Y. ANDO and H. KIMURA ‘An Automatic Piping Algorithm including Elbows and Bends’, *International Conference on Computer Application (ICCAS)*, 2011.

8. Y. ANDO and H. KIMURA ‘An Automatic Pipe Arrangement Algorithm Considering Elbows and Bends’, *International Marine Design Conference (IMDC)*, 2012.

8. AUTHORS BIOGRAPHY

Yuto Ando a Ph.D. candidate of the Department of Maritime Engineering, Graduate School of Engineering, Kyushu University.

Hajime Kimura holds the current position of Associate Professor at the Department of Maritime Engineering, Kyushu University. His previous experience includes optimization in shipbuilding.