

3D Piping Route Design Including Branch and Elbow Using Improvements for Dijkstra's Algorithm

Hai Nguyen¹, Dong-Joon Kim^{2,*} and Jianke Gao¹

¹BK21+ MADEC, Pukyong National University, Busan, South Korea

² Department of naval architecture and marine system engineering, Pukyong National University, South Korea

*Corresponding author

Abstract—In ship design problem, piping route design is the most complex stage because the internal spaces inside the ship are limited and the ship, in general, has many systems and subsystems that consist a large number of pipes. In order to optimize the piping cost as well as the construction time, so many methods have been considered and applied. But, most of methods just consider to problem of shortest path while the piping cost and the construction time also depend on the branch and number of bend & elbow. Dijkstra's algorithm is one of the most famous method of deterministic approach methods because it is always for the best solution. But it takes a long calculation time, occupies much memory and just consider to problem of shortest path. To fix that disadvantage, in this paper, the Dijkstra's algorithm with some improvements is used to reduce the calculation time, occupied memory and the shortest paths which are exported by this method will have a minimum number of bend & elbow. An example in a 3D simple engine room was used to show the results what are exported by using our method.

Keywords-piping route design; Dijkstra's algorithm; optimum design; shortest paths

I. INTRODUCTION

A ship, in general, usually consists of a large number of pipes. For each area, the pipes have certain tasks and specifications therefore the pipes are also designed, arranged and installed differently for each area. Because of that, piping route design is a very complex and difficult task for piping engineers. Follow Asmara [1], a pipe which had already routed will be claimed to have good quality if the following conditions are satisfied: (1) the functional requirements are satisfied; the pipes are connecting pieces of equipment perfectly without having excessive length, unnecessary bends and collisions, (2) it complies with maritime rules and regulations as imposed by classification societies, and (3) it fulfills the subjective values of the pipes designer who has routed those pipes. Normally, the two last conditions above cannot change, so we just consider to condition (1) while (2) and (3) are default that it is satisfied.

Piping cost, in general, includes three parts that are production cost, installation cost, and operational cost; all of them, depending on the results of piping route stage. For example, by using a shorter length pipe or a pipe which has a smaller number of bends, the production cost is reduces. In addition by using a clever arrangement of supports, the installation cost as well as construction time are also decreased. This way can be achieved by having pipes to be routed in parallel; the valves must be placed to be easy to control, repair as well as substitute to reduce the operational cost; etc.

In order to optimize the piping cost, so many optimum methods have been considered and applied. In general, the optimum methods can be categorized into two directions: (1) deterministic approaches, and (2) meta heuristic methods. The first direction includes so many methods such as Lee's algorithm, escape algorithm, Dijkstra's algorithm, etc. The advantage of this direction that it is always for the best solution (if it exists), but it takes along calculation time and occupies much memory RAM. Some algorithms in this direction are difficult to apply for environments that have many obstacles and to apply it in a supported software for piping route design.

Meta heuristic methods, in recent years, becoming the best choice to solve problem of finding shortest paths. This methods use the stochastic process to find the results. Thus, it is fast to have the result and, of course, does not occupy much memory RAM. Some methods of this direction such as genetic algorithm (GA); ant colony algorithm (ACO); particle swarm optimization (PSO), etc. The methods in this direction are easy to use, to apply in supported software. But, as mentioned above, this direction uses the stochastic process to find the results, therefore, sometimes it may not for the best result (even if it exists), especially in the environments that are complex and have many obstacles.

Nowadays, the comparison of the performance between two directions above is impossible, it means that we don't know which is better. But both of two directions have the same disadvantages that are: (1) Most of optimum methods only consider the problem of the shortest paths while the problem of minimum number of bend & elbow is also an important factor in piping route design [2], and - as mentioned above - (2) it is difficult to get the results if the environment is very complex and have many obstacles.

In this paper, some improvements applied for one of deterministic approaches named Dijkstra's algorithm are introduced. We used Dijkstra's algorithm – one of deterministic approach methods – because this method, of course, is always used for the best solutions and this method is easy to study and apply in supported software. Moreover, this method is also a suitable method to solve problem of branches and collisions in problem of piping route design. By using the improvements for Dijkstra's algorithm, the results are always the shortest paths that have a minimum number of bend & elbow, the environment to apply method will be divided as cell decompositions. Problems of branch, calculation time, occupied memory RAM and collision are also solved by using the improvement.

II. RELATED WORKS

A. Dijkstra's Algorithm

Dijkstra's algorithm is one of the deterministic approaches. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later [3]. For a given origin node (source) in the graph (environment), this method considers to all of other vertex and also finds the shortest paths between that node and all of other vertex. Dijkstra's algorithm is one of the most famous that can be categorized as a Best-first search technique [1]. In practice, Dijkstra's algorithm is usually applied to find the shortest road in the map. Figure 1 illustrates for the results what is exported by using Dijkstra's algorithm.

Dijkstra's algorithm is always for the best solutions if it exists. This method considers to all of other vertex from the same origin point, so it is very suitable and useful for problem of branch or the cases when we need to connect several points from the same source, the cases that commonly encountered in practice. That's why Dijkstra's algorithm is chosen to apply for our researches. But, similar to most of other methods, Dijkstra's algorithm just considers the problem of the shortest paths while the problem of number of bend & elbow and parallel are also important, too. Moreover, this method also takes a long calculation time and occupies much memory RAM.

In order to solve that problems, the algorithm needs to be made some improvements to have a better result (excepting problem of shortest paths) as well as reducing the calculation time and occupied memory RAM, there are some improvements applied for Dijkstra's algorithm, it is mentioned one by one in subsections below.

B. Problem of Minimum Number of Bend & Elbow

As mentioned above, most of optimum methods just consider to the problem of the shortest paths while problem of minimum number of bend & elbow is also important; therefore, the first thing needed to solve that is problem of minimum number of bend & elbow.

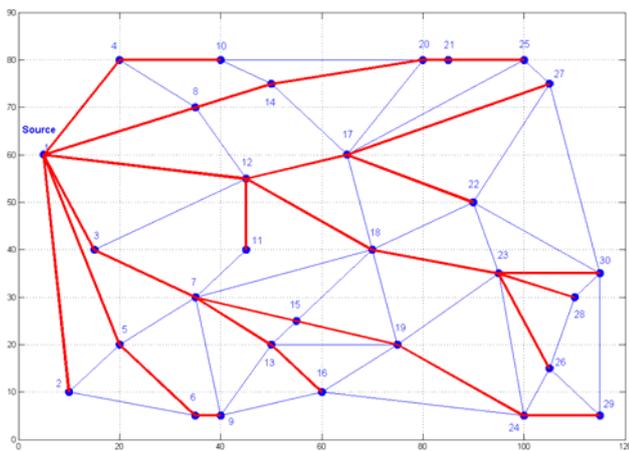


FIGURE I. ILLUSTRATED RESULT OF DIJKSTRA'S ALGORITHM

In practical, there are so many to connect two point in a graph with the same length even if the graph has so many obstacle. In

these lines, each line has a certain number of bend & elbow, and, of course, we always want to route and install a pipe which has number of bend & elbow as low as possible. It means that applying a good optimum method for piping route design must consider the problem of minimum number of bend & elbow. In this paper, with the support of MATLAB program, some algorithms are applied to make sure that the results (the shortest paths) exported by using Dijkstra's algorithm always have a minimum number of bend & elbow.

Figure 2 shows three lines with the same length, the dark-green line (A-D-F-B) has the minimum number of bend & elbow (two vs four of magenta line A-C-K-E-F-B and five of blue lines A-G-H-I-E-F-B), so, applying our algorithm, the Dijkstra's algorithm will choose the dark-green line as the exported result. Also, in order to reduce the piping cost, piping route in parallel is one more problem that is considered in this paper. By this way, the piping cost and construction time are decreased.

C. Problem of Piping Route in Parallel

After considering the problem of minimum number of bend & elbow, in order to reduce the construction time as well as the installation cost for pipe supports, every pipe must be routed close to the steel construction or the pipes that run in the same direction and lie close together should be routed in parallel, by that way, the pipe support installation is easier. In this paper, an example which has no steel constructions is used, so we only consider the problem the pipes that run in the same direction and lie close together should be routed in parallel.

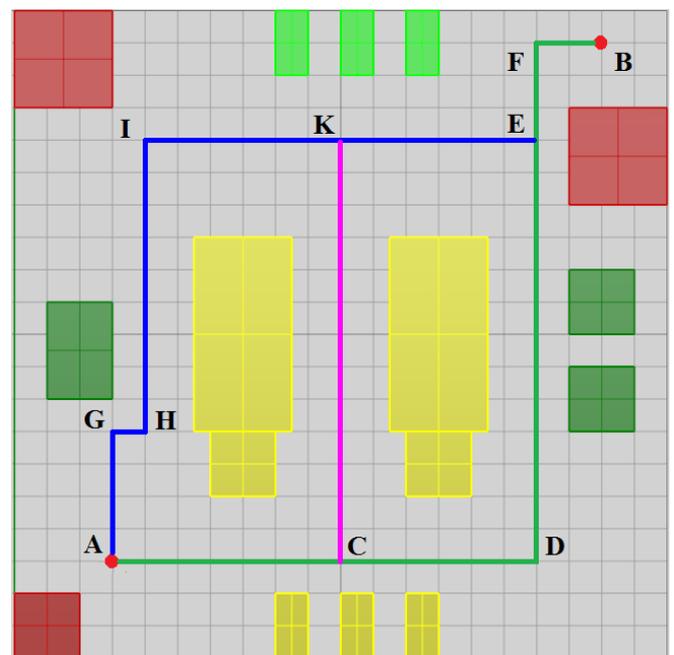


FIGURE II. PROBLEM OF MINIMUM NUMBER OF BEND & ELBOW

D. Problem of Branch

As mentioned, one of the advantages of Dijkstra's algorithm that is the method considers all of other vertex, so, this method is suitable and useful for problem of branch. In our research, we

solved problem of branch as follow: From the source, the algorithm considers all of the other vertices - including all of vertex on the basic line - and by using our improvement, the algorithm just considers all of vertex on the basic line and only chooses one vertex which is the nearest vertex with the basic line. After that connect that point with the source and the pipe will be routed. The problem of minimum number of bend & elbow and the problem of piping route in parallel are also applied for this problem.

E. Problem of Cross and Collision

In order to avoid the cross as well as the collision between two or more pipes, in our algorithm, when the line is routed (including branch), it will be updated to become the obstacle apply for the next line. Because all of the vertex which is in or on the obstacles won't be considered, there are no line can be routed through these vertex. By that way, problem of collisions will be avoided.

F. Reducing Calculation Time and Occupied Memory RAM

Taking a long calculation time and occupies much memory RAM are the main disadvantage of Dijkstra's algorithm. To solve that problem, the only way we can do that is reducing the considered space of graph. It means that after reducing the number of vertex in the graph (environment), and, of course, the calculation time occupied memory RAM are also reduced, too. This is also the principle applied for problem of cross and collision.

III. STUDY CASE

An example with the task that is routes the pipes for a 3D engine room is used. The engine room, with total 18 lines in 4 different systems, includes fuel supply system, air pressure supply systems, cooling system, and system of waste oil. The position and the location of the equipment are shown in figure 5. To simulate the geometry and apply the algorithm, a code was written by using the support of MATLAB program.

A. Modeling the Geometry

In practice, the 3D models used in a ship design process have a very high level of detail, so it is too difficult task to simulate the geometry that looks like 3D actual model. In order to simulate the 3D actual model easier, the Model Simplification Method, which was mentioned by Andi Asmara [1] is used. By this way, the equipment will be represented by boxes with the difference of size, so that the geometries simulated by the method look like the 3D actual model. However, the level of detail is lower. Depending on the requirement that the level of detail is chosen suitably.

In our research, to decrease the calculation and simulation time, the low level of details are chosen to simulate the model. Figure 3 shows the 3D actual model (above) and the simulated model (below) which is created by using model simplification method in order to represent for 3D actual model.

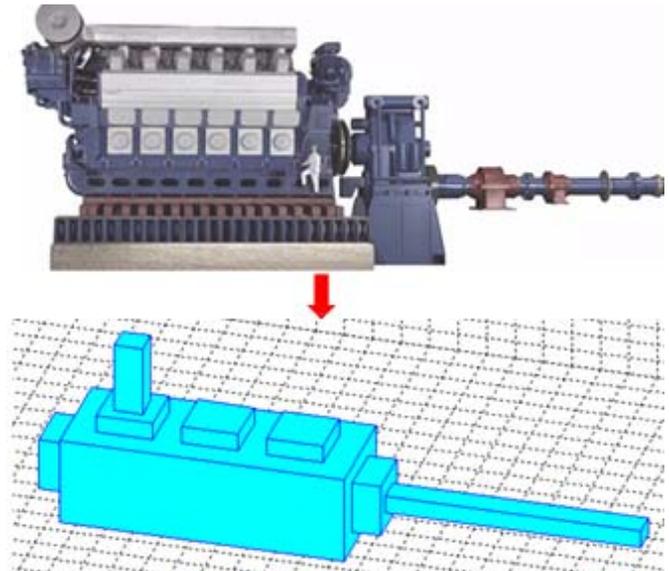


FIGURE III. MODEL SIMPLIFICATION OF AN ENGINE

B. The Graph Is Created as Cell Decomposition

Before applying the method, the graph, of course, must be created. There are so many ways that can create the graph, but the common method normally used is grid decomposition (or cell decomposition). By using this method, the space is decomposed into grid of cells. The cell is represented by a cube which has the size x by y by z (unit of length). Normally, the grid is just divided for unobstructed space, but, in order to create the graph easily as well as reduce the time for creating the graph, all of workspace is divided into the cell decomposition and then, when we define the obstacles, all of the cells in or on the obstacles will be removed as describing in previous section.

Each cell in the graph has a certain parameter, but, in the same location, most of them has the same size. The size of the cells affects to the quality of the results. The small size improves the quality of the routed pipes, but using the small size, the number of cell is increased and the calculation time as well as occupied memory RAM are also increased, especially in 3D workspaces. Noticeably, in all case, the size of cell must be greater than the maximum of diameter of the biggest pipe (in that region).

C. Priority

In practice, each case of design has a certain priority for problem of shortest path, bend & elbow, parallel or the pipe must be route along a steel construction side (space factor) [4]. For example, as show in figure 4, with the same length, if the space factor is more important than the bending factor, the pipe will be routed by line A-C-D-E; otherwise, the line A-B-D-E (less number of bend than line A-C-D-E) will be chosen.

In our study case, because there are no steel construction in the graph, so the priority is chosen as follow:

- The shortest path
- Problem of minimum number of bend & elbow
- Problem of piping route in parallel

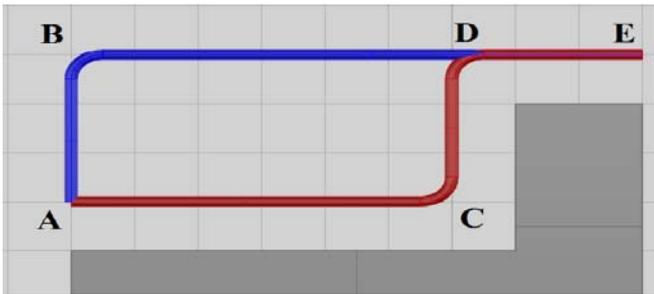


FIGURE IV. SPACE FACTOR AND BENDING FACTOR

D. Results and Discussion

As described above, this algorithm is applied to route some piping system in a 3D simple engine room, the workspaces including the equipment and obstacles are simulated by using model simplification at low level of detail. The engine room is assumed that it is placed in a region that has the size in unit of length is 30x30x15. The engine room is divided by different size cells to be suitable with piping systems, in particular, the pipes around engines have small diameter, so the size of cells around two engines and the generator are smaller than other regions. The total number of cells in this case is 38,808 cells.

Applying the improvements for Dijkstra's algorithm, the result is exported with the support of MATLAB program as show in figure 6. From the result, the total length of lines in unit of length is 303.5, the number of branch is 5 and the number of bend is 35. All of the results would be changed if the size of cell is changed. So that choosing the size for cells is the very important step to get a good result.

From the result and based on the references, the Dijkstra's algorithm applied our improvements to evaluated that the method is suitable, useful and give a good solution. Our improvement has been solved the problem of minimum number of bend & elbow as well as the problem of cross and collision. In the future, problem of application of this method for more complex engine room (including steel construction) to re-check the ability of our improvement in problem of piping route in parallel will be performed.

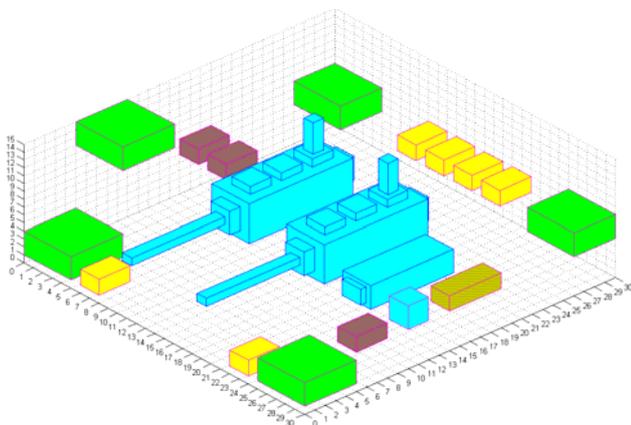


FIGURE V. 3D MODEL OF ENGINE ROOM SIMULATED BY USING MODEL SIMPLIFICATION

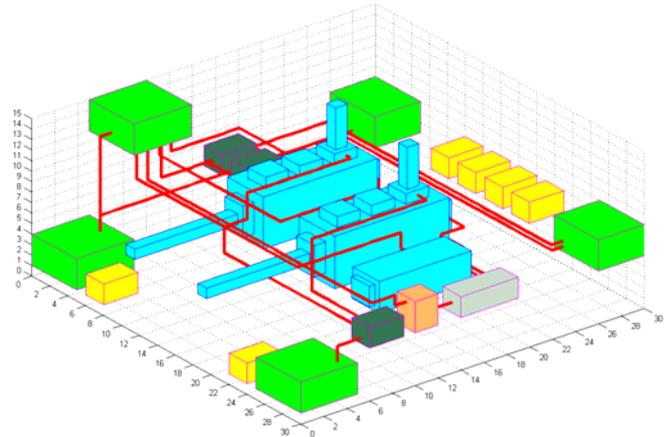


FIGURE VI. THE RESULT OF ALGORITHM

IV. CONCLUSION

Nowadays, with the development of computer industry, both memory and analyzed speed of computers are increased faster, so the problem of calculation time and occupied memory of the deterministic approach methods are not the biggest problem. Because of that, the exact of result will be considered as the biggest problem in piping route design, and Dijkstra's algorithm was chosen to be the main method for our research.

The combination among cell decomposition, model simplification and Dijkstra's algorithm are a new direction that can apply for problem of piping route design. The algorithm is always used for the good solution even if the graph is very complex and includes a large number of obstacles.

Acknowledgment

The research was supported by the BK21 plus MADEC human resource development group in 2015. Without them this paper could not be submitted.

REFERENCES

- [1] Andi Asmara, "Pipe routing framework for detailed ship design," Master of Science in automation and robotics, University Dortmund, Duitsland, Germany. ISBN 97890-6562-326-3, 2013.
- [2] Nguyen Hai, Dong-Joon Kim and Gao Jianke, "An improvement of Dijkstra's algorithm for piping route design," 6th International Symposium on Advaned Engineering (ISAE), pp. 76-79, October 2015.
- [3] https://wikipedia.org/wiki/Dijkstra's_algorithm, October 28, 2015
- [4] Shin-Hyung Kim, Won-Sun Ruy and Beom Seon Jang, "The development of a practical pipe auto-routing system in a shipbuilding CAD environment using network optimization," International Journal of Naval Architecture and Ocean Engineering 5:468~477, 2013.
- [5] TERUAKI ITO, "A genetic algorithm approach to piping route path planning," Journal of Intelligent Manufacturing, pp. 103-114. October 1999.
- [6] Kimura, H., "Automatic designing system for piping and instruments arrangement including branches of pipes," International Conference on Computer Applications in Shipbuilding (ICCAS2011), Vol3, pp. 93-99, Trieste, Italy, 20-22 September 2011.
- [7] Y Ando and H Kimura, "An Automatic Piping Algorithm Including Elbows and Bends," International Conference on Computer Applications in Shipbuilding (ICCAS2011), Vol3, pp. 153-158, Trieste, Italy, 20-22 September 2011.
- [8] Andi Asmara and Ubald Nienhuis, "Automatic Piping System in Ship," Delft University of Technology, Delft, Netherlands, 2006.