The modification of A* pathfinding algorithm for building Mechanical, Electronic and Plumbing (MEP) path

Wonjun Choi¹, Cheekyeong Kim¹, Seokjae Heo¹, and Seunguk Na¹

¹Department of Architectural Engineering, Dankook University, 16890, Republic of Korea

Corresponding author: Seunguk Na (e-mail: drseunguk@dankook.ac.kr).

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government Ministry of Education (No. NRF-2018R1A6A1A07025819 and NRF-2020R1C1C1005406).

ABSTRACT This paper introduces a modified A* pathfinding algorithm that can be used in building Mechanical Electronic Plumbing (MEP) path design by revising nodes selection process and post-processing. The pathfinding algorithm is used when a computer calculates the optimal path in a given space by algorithmizing how humans intuitively calculate the optimal path. As construction technology is gradually advancing, buildings with large and complex internal structures are increasing, so there is a need to automatically optimize existing design methods that rely on human intuition for a more efficient design. In the case of building MEP design, it is time and money consuming to design paths since they are complexly arranged throughout the building, and designs are frequently changed in response to the nature of the construction industry, where construction errors are frequent. Therefore, an MEP path design optimization module, MEPAutoroute, was developed by implementing a modified A* pathfinding algorithm to solve these problems. Algorithm was applied to seven different exemplary structures with MEP equipment, and the results are analyzed to determine its efficiency.

INDEX TERMS Pathfinding, optimization, building MEP, design, automation

I. INTRODUCTION

Everyone has at least once thought about which path or method should be used to reach their desired destination. Main considerations are roads with the shortest distance, roads with the least required time, and roads that can pass through specific locations. However, for paths that are large or complex, it is difficult for humans to intuitively identify an optimal path without the help of electronic devices that have high processing speeds. In response to these situations, attempts to find a desired path using electronic devices have been made widely across a number of industrial fields, which has led to the creation and introduction of various pathfinding algorithms.

A pathfinding algorithm is an algorithm that calculates alternative paths that meets specific conditions when the existing path is no longer optimal in response to unexpected conditions [1]. The more complicated the path or the more complex the conditions to be satisfied in selecting the optimal path, the more advantageous it is to search for an optimal path in real time and to draw conclusions that are difficult for humans to find in a short period. This algorithm, first introduced in 1950, has been applied and improved in various fields as electronic devices have developed over time [2].

The development of electronic devices that have contributed to the application of pathfinding algorithms is accelerating by the fourth industrial revolution in the modern era. At the Davos Forum in 2016, it was emphasized that the fourth industrial revolution is currently underway, and the digital, physical, and biological boundaries of technologies such as artificial intelligence, robots, Internet of Things (IoT), and self-driving vehicles are disappearing. For the application and utilization of these technologies, information in the real world must be accurately implemented in the virtual world. Various attempts are being made to close the gap between the real and virtual world across all industries [3]. This trend is equally applicable in the construction sector. To automate and efficiently manage building design, construction, and maintenance, the construction industry is gradually progressing from acquiring information based on blueprints to...
using Building Information Modeling (BIM) [4], a 3D digital modeling tool that contains information on buildings. These changes in the construction industry are accelerating in line with the government's support policies [5]. The construction industry is a labor-intensive industry in which a number of people are engaged in the entire process of design, construction, and operation & maintenance, regardless of field. In the case of Lotte Tower, a skyscraper in South Korea, the number of people directly and indirectly involved until completion reached around 4 million [6]. However, developed countries or countries in the final stage of land development are experiencing problems such as a stagnant construction economy, high wages, and difficulties in the supply and demand of human resources in response to the avoidance of the 3D industry. Therefore, governments of each country tend to focus on low-cost and high-efficiency policies to replace the shortage of human resources and maintain competitiveness in the global market. For example, in the case of South Korea, the Ministry of Land, Infrastructure, and Transport aims to increase the productivity of the construction industry by 25% by investing 169 million dollars in intelligent construction technologies through digitalization by 2025 [5], and the US General Service Administration made the BIM model compulsory to submit in the event of a public building design contract [7]. Considering these points, the digitalization of the construction industry could be considered as a global trend, suggesting that continuous investment at the government level will continue.

As the information on buildings with complex structures has been digitized, it has become possible to use pathfinding algorithms in various fields. Representative examples include evacuation route design in case of fire, navigation, robot route planning, drone pathfinding, and ship MEP route design. The larger the building, the more effective the pathfinding algorithm can be. In this study, the optimal path was calculated according to the specific conditions given in example structures by selecting and fusing various pathfinding algorithms currently commercially available to automate the design of the MEP route for buildings, and the effectiveness was then analyzed.

In order to quantitatively analyze the effectiveness, the pathfinding algorithm was applied to seven example structures. Afterward, for each example, the number of clicks and the amount of time spent was compared among manually designing the route, using the internal pathfinding function of the 3D BIM tool REVIT, and applying a developed pathfinding algorithm, MEPAutoroute.

In response to the nature of the construction industry, the interests of various companies are intertwined, which causes confusion in process control and supervision as well as frequent design changes [8]. In particular, construction equipment has a high frequency of design changes in response to design changes in the previous processes, and the connection relationship could be complicated depending on the shape of the structure [9]. In addition, since high professionalism is required to understand the MEP structure only with blueprints, installation is often different from the original design when a non-specialist participates in the work. The construction industry has tried to minimize frequent design changes and errors during construction using 3D BIM models, which has turned out to be effective to some degree. However, since creating a 3D BIM model from MEP blueprints is all done manually, high costs are still incurred. Moreover, when design changes are made, all remodeling must be done manually as well. Therefore, to reduce time and costs in response to redesign and to examine the number of cases and draw optimal conclusions beyond the existing optimal path search method that relies on human intuition, the pathfinding algorithm, MEPAutoroute, was introduced.

In the remainder of this paper, Section 1 explains changes in the construction industry and government policy trends. After that, Section 2 introduces previous studies on pathfinding methods in other fields such as evacuation, navigation, robots, drones, and ship MEP design. Section 3 discusses conventional pathfinding algorithms to analyze the strengths and weaknesses of each algorithm to find suitable algorithms for this research. In Section 4, pathfinding algorithms are combined and modified in this research to automate MEP route design. Moreover, Section 5 presents an efficiency analysis according to the application results and considers future research directions. The conclusion of the research is stated in Section 6.

II. RELATED WORK

Research applying the pathfinding algorithm to the digitalized world has been conducted by a number of scholars and is being done in various industrial fields, but research applying it to buildings, especially the automation of MEP route design, has not progressed to the same degree. Therefore, an analysis of formal studies mainly in fields other than architecture, such as disaster evacuation plan establishment, navigation, robot route planning, drone pathfinding, and ship MEP design, was conducted, and essential research cases among them will be introduced.

The pathfinding algorithm is frequently implemented when establishing a disaster evacuation plan. Buildings are designed to withstand the building's self-weight and external load for people's safety. However, there was a need to design an evacuation path for users to quickly evacuate in disasters and accidents such as earthquakes and fires. For example, in the event of a fire, the “golden hour” is 5 min, and every minute is essential. Therefore, the path from each room to the exit and the safest and fastest path by which a rescue team can reach a person in need should be calculated using the pathfinding algorithm to shorten the time required to perform the rescue. In the case of [10], the potential evacuation path was simplified to a node network to reduce calculation time and remove unnecessary nodes, and the Dijkstra pathfinding algorithm was then applied. Similarly, in [11], the authors tried to avoid dangerous areas and speed up the algorithm by...
removing unnecessary nodes when searching for an evacuation route via filtering with sensors. Moreover, to calculate a safe evacuation route by reflecting the characteristics of the construction site, an attempt has been made to identify obstacles using a 4D BIM model that includes construction progress information [12]. In [13], a virtual environment was built on BIM for safety education in the case of a fire, and a VR game engine was then created. Moreover, in [14] and [15], the authors tried to minimize computation time while considering the maximum capacity of each evacuation route. Finally, in [16], the authors used the BIM model to simulate an evacuation route in an emergency for each construction process. They then suggested various construction methods based on the time it takes to complete the evacuation. Based on these studies, it is confirmed that pathfinding algorithms used for the establishment of a disaster evacuation plan tend to place importance on the accuracy and operation time of the algorithm.

The pathfinding algorithm implemented in navigation that selects the movement path of people and vehicles is somewhat different from the disaster evacuation path plan in that it has to select a path depending on the situation or the number of destinations, rather than the shortest path. It is mainly used to calculate the shortest distance, but depending on the situation, it may be necessary to go through an area away from the shortest path or present various options for user convenience while only using available routes; thus, the pathfinding algorithm in navigation needs to be fast when performing frequent recalculations and considering safety. Various attempts have been made to meet these requirements, and representative examples are as follows: First, in NavMesh, which implements the shape of the terrain, performs a visibility test to determine whether or not a mesh is near an obstacle to filter and select nodes that could be used for the algorithm [17]. Second, the weight to be used when selecting the best path for the trip is calculated after analyzing time, traffic, and other characteristics that may affect the choice [18]. Third, nodes within a specific range are reviewed based on the start and end point locations [19]. Fourth, non-structural elements, such as parking lines, are recognized as obstacles during pathfinding for improved safety and a precise path to follow [20]. As can be seen from these cases, research has mainly focused on shortening the time required for the pathfinding operation by reducing nodes to be reviewed among countless nodes on the map based on specific conditions.

In the case of robot route planning, sensors are used to determine and digitalize the environment where robots have to operate. Based on the information of digitalized environments, various pathfinding algorithms have been applied to find optimal paths for robots. For example, the authors in [21] introduced path planning for mobile robot navigation by creating a node network from the surrounding environment with a Voronoi diagram and used the fast-marching method to find the optimal path. Similarly, the authors in [22] created a contoured figure of the worktable for robotic drills and used particle swarm optimization (PSO) to shorten the time required for the drill to move to different locations. Furthermore, the authors in [23], [24], and [25] suggested a new route planning method using the artificial potential field to avoid obstacles while minimizing the repulsion caused when robots change the direction of movement. However, there is research focusing more on the pathfinding algorithm itself than on the creation of various methods to digitalize the environment, such as the implementation of the A* algorithm on a grid composed of cells [26] and the implementation of a modified ant colony optimization algorithm along with a fuzzy interference system, tuned with a Simple Tuning Algorithm to create smooth paths that robots could use [27].

Drones and robots share a number of common properties when applying pathfinding methods. Both drones and robots have to detect nearby environments using sensors and apply pathfinding algorithms, which ultimately redounded similar research in both fields of industry. For example, drones use or create a three-dimensional model of obstacles with sensors and use Euclidean distance transformation along with the A* algorithm to find an optimal path that is a certain distance away from obstacles [28]. In detail, the expression pathway of a three-dimensional model of obstacles may vary depending on situations, and one of them is the point cloud model [29]. Furthermore, sensors, such as float charge stations on the sea, could be used to create environment information as a grid for the pathfinding algorithm [30]. Similarly, the authors in [31] transformed nearby environments into meshes for drone paths and used the particle swarm method to find an optimal path. However, there are some differences between drones and robots. The significant difference is that robots move on a two-dimensional plain, while drones operate in three-dimensional space. It may not look considerably different, but two-dimensional and three-dimensional movements have enormous differences in usage. Unlike robots, drones require a relatively vast amount of energy to fly and cannot carry heavy batteries, which is why the authors in [32] focused on finding the shortest path using the A* algorithm with the lowest energy consumption rate. Moreover, drones can use their transportation truck as a moving platform and can execute multiple transport operations at the same time by using the K-means clustering method in complex areas, such as cities [33]. Furthermore, the authors in [34] tried to use the ant-colony method for highly crowded drones, enough to be called swarms, for a safer and more efficient ant-colony method.

Unlike the pathfinding algorithm applied in navigation, robots, and drones, ship MEP design is not a task that needs to be done in real-time or within a short time. The primary purpose of ship MEP design is to find an adequate path with an algorithm with a lower operation time, a lower cost, and easier maintenance. Moreover, like building MEP equipment, ship MEP equipment should not interfere with the space where people traverse, meaning that the MEP design method of the ship is
similar to the building. In [35], the candidate path for the optimal path was created according to specific conditions. The path with the lowest installation cost was selected as the optimal path, and it included accessory information. Similarly, the authors in [36] created a 3D grid for pathfinding based on 3D space and structural information. On the other hand, the authors in [19] attempted to optimize the ship pipe path by introducing a genetic algorithm reflecting the decision-making process of ship MEP engineers to create a model that suggests a path similar to that of the engineers. In addition, since the MEP design depends heavily on engineers’ or customary practices, the authors in [37] implemented a geometric modeling kernel and expert system shell to automate the design process while considering practices from former projects. Furthermore, there have been attempts to design ship piping using the pheromone theory of ant-colony optimization algorithm [38, 39]. Lastly, the authors in [40] developed an algorithm more optimized for pipe route design by modifying the node selection rules in the JPS algorithm, which is similar to the algorithm developed in this study. It filters nodes to be used in the pathfinding algorithm.

Although the introduced pathfinding algorithm is mainly for evacuation, navigation, robots, drones, and ships, it does not apply only to those fields. There has been research in a number of other fields, e.g., involving the calculation of oil pipeline paths under the ocean to find an optimal path using the Laplacian smoothing method [41] and an optimal path for tugboats to minimize fuel consumption and travel distances based on a navigation mesh [42]. Furthermore, attempts to decrease computation time have been made by using the line of sight to skip cells on the grid when there are no obstacles in between [43] and by actively using information from former pathfinding cases for similar search conditions [44].

The purpose of the MEP route design is to connect the MEP equipment set as the starting point and the end point with an optimal path. Unlike ship MEP design, errors occur frequently in response to dissonance between design and construction caused by the lack of digitalization and the relatively low expertise of workers at construction sites. Frequent changes in the original building design ultimately result in the frequent change of the MEP route design. Frequent design changes imply that it is necessary to reduce the operation time of the pathfinding algorithm so that it can respond to a large number of design changes within a limited time. However, the time required for the pathfinding algorithm increases significantly as the building size or the number of MEP types that need to be modeled increases. Therefore, it could be said that it is necessary to develop an automation algorithm that can quickly calculate the optimal path for MEP route design based on the pathfinding algorithms analyzed in disaster evacuation plan establishment, navigation, robot route planning, drone pathfinding, and ship MEP design.

III. PATHFINDING ALGORITHMS

As for pathfinding algorithms, Breadth-First Search (BFS), Depth First Search (DFS), Dijkstra, and Jump Point Search (JPS) are representative and are the most used algorithms due to their high effectiveness [45]. In this paper, the characteristics of each pathfinding algorithm were analyzed for effective building MEP route design, and the strengths were collected and applied.

A. BREADTH FIRST SEARCH (BFS)

The Breadth-First Search algorithm searches for all adjacent nodes in the order of nearest distance. When searching for neighboring nodes in close order, the nodes that have been searched for appear to gradually expand as if drawing large concentric circles starting from the start point.

\[
O(|V| + |E|) \tag{1}
\]

In this formula, \(|V|\) is the total number of nodes, and \(|E|\) is the total number of branches. In other words, the total time required in this algorithm is the sum of time spent to review each node and all branch points before reaching the destination. In more detail, the BFS searches for a path for all adjacent nodes first, and when the search is completed, the calculation is repeated until the destination is reached by additionally searching for nodes branching from each node that was searched for. Since each node has its parent node and child node information, it is possible to determine which nodes and branches were visited to reach the current node. This algorithm always guarantees the shortest path, but computation dramatically increases as the number of nodes increases, which is unrealistic for actual usage in the field. However, it is possible to utilize the advantages of BFS in MEP design.

In the case of MEP route design, the main path is designated based on two main-MEP equipment for each MEP system, and the sub-MEP route is designed to be connected with the main-MEP route. Therefore, as the structure of the building becomes more complex, it is not clear which part of the main-MEP route the sub-MEP route should be connected to, so it is advantageous to use the BFS algorithm. The BFS algorithm searches for neighboring nodes until the condition is met, which is the destination in most cases; thus, it can be used when the destination is not confirmed. The main-MEP path node contacted first by the BFS algorithm can designate the destination for sub-MEP routes.

B. DEPTH FIRST SEARCH (DFS)

Depth First Search and Breadth-First Search are similar in that adjacent nodes are sequentially examined. However, unlike the Breadth-First Search, the Depth First Search algorithm searches for all branch points before moving to the next node, as shown in Error! Reference source not found.. Although there is a disadvantage that the operation time is longer than the BFS algorithm since it reviews all branches, the number of required fittings caused by bending in each connection of MEP routes could be minimized by examining the possible branches that could be used as avoidance routes.
C. DIJKSTRA

The Dijkstra Algorithm searches for the shortest path by sequentially examining the nodes between the starting point and the end point in a network composed of nodes based on the sum of costs required to move from the starting point to the corresponding node, respectively. For example, in the case of Error! Reference source not found., each node in the Dijkstra algorithm is added to the priority queue indicating the shortest path candidate node and sequentially reviewed. Since the cost required to reach a node is calculated and compared for all possible paths, the shortest path could always be identified. However, it cannot be used if the path contains negative weights, and it is not easy to use in response to a large amount of computation in a large-scale space because the node must be taken out or updated from the priority queue every time the node is reviewed. The time taken to compute the Dijkstra algorithm could be expressed as the following equation:

\[ O((|E| + |V|) \log |V|) \]  

Like the above-mentioned BFS algorithm equation, \(|V|\) is the total number of nodes, and \(|E|\) is the total number of branch points. The time required to find the shortest path from the starting point to each node for which the shortest distance has not yet been calculated is \(O(V \log V)\), and the time taken to calculate the shortest distance between each current node and the neighboring node is \(O(E \log V)\). These two equations are combined as Equation (2) to express the total time needed to complete the algorithm.

D. JUMP POINT SEARCH (JPS)

Since the pathfinding algorithm usually takes a relatively long time to review all the costs of countless nodes, the JPS algorithm tried to reduce the computation time by filtering and selecting the nodes for the pathfinding search based on a few rules. The pathfinding algorithm used in JPS calculates the total cost of each node by adding the distances from the starting point and the ending point and uses the A* pathfinding algorithm that searches for the path. The formula used to compare costs is expressed as follows:

\[ f(x) = g(x) + h(x) \]  

In Equation 3, \(g(x)\) is the cost of the current node, \(h(x)\) is the cost to be added by calculating the distance between the start and end points when moving to the next node, and \(f(x)\) is the cost of the next node. The JPS algorithm calculates the cost of nodes and selects nodes with the lowest cost until the destination is reached. Furthermore, since applying a pathfinding algorithm directly to a grid composed of countless nodes is challenging, the JPS algorithm uses nodes that only meet the condition of specific rules and ignores all nodes that fail to meet those conditions. Although this node exclusion process takes some time, it has the advantage of searching for the shortest route at a fast speed once the exclusion is complete. Therefore, the idea of selecting the necessary nodes of the JPS algorithm was applied to the pathfinding algorithm developed in this study.

| Algorithm        | Diagram Description                                                                 |
|------------------|-------------------------------------------------------------------------------------|
| **Breadth First Search (BFS)** | The nodes are indicated by numbers in reviewed order when the BFS algorithm, which reviews nodes from a closer distance, is applied to the node network. |
| **Depth First Search (DFS)**     | The nodes are indicated by numbers in reviewed order when the DFS algorithm, which reviews all branches before reviewing the next node, is applied to the node network. |
| **Dijkstra**                                                                 | The cost for each node is displayed at each stage of the pathfinding process of the Dijkstra algorithm. |
When running the JPS algorithm, nodes that could be used for the pathfinding algorithm are filtered and selected.

**FIGURE 1.** BFS, DFS, Dijkstra and JPS algorithm diagram comparison chart

**IV. RESEARCH METHOD**

**A. ALGORITHM PROCESS**

Since this study aims to modify the A* algorithm to optimize MEP path design, it is assumed that information on the building structure, MEP equipment, and MEP system is provided. In addition, to increase the practical suitability of the algorithm and search for an effective avoidance path, information about the building structure and MEP model is obtained from the REVIT program, a BIM tool mainly used for MEP design in the construction industry.

The overall MEP route design process starts from obtaining the BIM structure and MEP model as specified in Error! Reference source not found.. Afterward, before applying the pathfinding algorithm, the nodes used for the pathfinding algorithm are extracted from the structural information and added to the node list. The extracted nodes in the node list are used in the pathfinding algorithm to calculate the optimal path between MEP equipment. However, since the initially calculated optimal path selects nodes even if the path is diagonal, the path should be recalculated in an orthogonal direction while minimizing the bending at each branch point of the path for practical use. After completing the calculation of the orthogonal path, the algorithm ends by placing the MEP route model, such as duct and piping, in the BIM program based on the type of start and end equipment along the calculated path.
FIGURE 2. MEP route design automation process based on commercial BIM tool using a path-finding algorithm

B. OPERATION PROCESS

In order to run MEPAutoroute, the pathfinding algorithm developed in this study, the user must input some necessary information, as shown in Error! Reference source not found., before the algorithm starts. Since the main equipment where the main ducts or pipe should be connected to in each system is not indicated, the user has to manually designate the main equipment for each system before applying the pathfinding algorithm. In Error! Reference source not found., the Supply Air MEP system and the Return Air MEP system are shown as an example, and the two systems share the same supply/return air MEP equipment. In this example, MEPAutoroute first starts by selecting the main equipment for the main route in the Supply Air system. When a selection is made, the main route connecting the main equipment is modeled, and the sub-route from each sub-equipment is then connected to the main route. After the MEP route design of the supply air system is finished, users then select the main-MEP equipment for the return air system, and the process after selection is the same as the supply air system.
C. ALGORITHM SEQUENCE

To use MEPAutoroute, the user must input in each system the information on which equipment is the start or end equipment that the main route will be connected to. Aside from designating main equipment, importing the extracted node from the library, calculating the path, and modeling MEP routes are automated. The detailed algorithm is shown in a flow chart in Figure 4.

When MEPAutoroute is executed in a situation where structure and MEP equipment BIM model information is given, the node is extracted from structural geometry information in the first step. The nodes extracted in this step are the base nodes used for the pathfinding algorithm and are stored in an external library in EXCEL format. After that, the user has to manually designate some MEP equipment as the start and end equipment in each MEP system to which the main-MEP route will be connected. Unselected remaining MEP equipment is automatically set as sub-MEP equipment.

After the main-MEP route is created, the pathfinding algorithm for automatically designated sub-MEP equipment is executed. First, the calculated sub-MEP path is modified to move only in an orthogonal direction. Second, whether it overlaps with the other MEP route elements is checked. If there is an overlapped location, the destination is changed. The sub-MEP pathfinding algorithm is finished after placing the sub-MEP route model element at the overlapped location.
However, if there is no overlapped location, the sub-MEP pathfinding algorithm performs like the main-MEP pathfinding algorithm. It calculates the original path containing diagonal paths and then finds paths that only move in an orthogonal direction.

**Composition of MEPautoroute**

### 1. Node Extractor

In the first step, the Node Extraction step of Error! Reference source not found., data are extracted from the BIM structure geometry information, and nodes that the MEP routes can pass through are placed. Extracted nodes are divided into normal nodes and preferred nodes. Normal nodes are the most common and basic nodes used for pathfinding, and preferred nodes are used to make routes required to pass through specific locations in terms of maintenance and industrial practice, even if it is not the shortest route. Normal and preferred nodes are composed of the same Node class. As shown in Error! Reference source not found., all types of nodes have common variables, such as XYZ absolute coordinates, the cost at the start point, the cost at the end point, the total cost, calculation status, start point location, end point location, preferred status, and final path node status. The criterion to indicate whether a node is a regular or preferred node depends on the mustUse variable. If the mustUse variable is set to true, it is a preferred node; if false, it is a normal node.

![FIGURE 5. Composition of the Node class](image)

The 3D model of the structural elements in the BIM tool is expressed as a geometry object composed of faces. Each face is defined by its surrounding curves, and each curve consists of two vertices. The algorithm shown in Error! Reference source not found., is an algorithm that extracts faces, lines, and vertices from the geometry object of the structure to create nodes that will be used for the pathfinding algorithm and then store them as an Excel file. This algorithm is iteratively applied to all structural elements. However, when the coordinates of vertices composing the lines extracted from each face overlap, only one of the coordinates are used to create nodes.

![FIGURE 6. Creation of nodes based on the BIM structure model](image)

Unlike normal nodes, MEP routes must pass through preferred nodes even if it is not the shortest path. The difference between preferred nodes and normal nodes is a “true” or “false” value for the mustUse variable, as shown in Error! Reference source not found., within the Node class, and preferred nodes are created and placed manually to fulfill the user’s unique design style and needs. Error! Reference source not found. shows the difference when only normal nodes exist and when preferred nodes are included in the pathfinding algorithm.

```csharp
foreach element in structuralElements
    if element.Category.Name == Structural Framing
        faces = extractFaces(element.solid)
        foreach faces
            curves = extractCurve(faces)
            points = extractPoint(curves)
            addPointsToPt(points)
        else..
        foreach point in Points
            Nodes = createNode(point)
            addNodeToXLWorksheet(Nodes)
            SaveExcelWorkbook(xlWorksheet)
```

![FIGURE 7. C# algorithm for node extraction based on the shape of the structural BIM model](image)

Placing a node grid in a tile format in all spaces in a building requires much calculation, so it takes a long time to review even a short MEP route for a small building. Therefore, nodes are extracted from the structural geometry information to minimize unnecessary nodes, as shown in Error! Reference source not found.. Since buildings without walls, columns, or beams do not exist inside, it is possible to secure enough nodes for the pathfinding algorithm by extracting nodes from the structural shape. The nodes extracted from the shape are normal nodes by default, and the preferred nodes could be directly placed at a location desired by the user later.
2. Main-MEP equipment selection

After node extraction in the previous stage is finished, the main equipment is designated to create and define the main-MEP route, as shown in the second stage in Error! Reference source not found.. Creating the main-MEP route and connecting it with sub-MEP routes reduces unnecessary expenses by minimizing the number of pipes and ducts for MEP system. Therefore, it could be said that selecting the main-MEP equipment as the starting point and the end point is an inevitable process. When the main equipment is selected, the MEP route connecting two pieces of equipment is first modeled as the main-MEP route, and sub-MEP routes are then connected to the main route as shown in Error! Reference source not found..

3. Pathfinding for main-MEP equipment

After selecting the main-MEP equipment where the main-MEP route will be connected, the pathfinding algorithm is executed based on the grid extracted in Step 1 of Error! Reference source not found.. The pathfinding algorithm at the current stage examines the paths regardless of their direction, including diagonal paths, for computational efficiency.

1) MODIFIED START/END LOCATION

When the pathfinding algorithm is executed, the coordinates of the starting and end points required for an algorithm are first calculated and defined. This is to calculate the precise location of the starting and end points where the MEP routes will be connected, which is different from the location of the MEP equipment. However, there is the problem that it becomes more difficult for the user to manually designate exactly where the connection is to be made in the main-MEP equipment for all MEP routes. Therefore, it is necessary to find an appropriate location and connect it automatically. Each piece of MEP equipment has connectors to connect MEP routes such as pipes and ducts. In addition, MEPs connected in the same system share a common type of connector if the equipment has all the necessary information in the BIM program. Therefore, MEPAutoroute calculates the precise positions of the main-MEP path's start point and end point by comparing the connector types. This means that one connector is set as the start point, and the other connector is set as the end point when the same type of connector exists among the two pieces of equipment designated as the main-MEP equipment. Since the MEP equipment mainly discussed in this paper is duct equipment and its routes have to move along the ceiling, additional MEP routes from the start and end points extracted from the connector to the ceiling are placed in Error! Reference source not found.. In detail, the specific length of the MEP route is protruded from the connector to its pointed direction, and a vertical MEP route from the tip of the
A protruded route to the ceiling is placed. Afterward, Modified Start and End Location is created as the actual start and end point for the pathfinding algorithm.

**FIGURE 10.** Connector, extruded MEP route model and modified start/end location

2) PATHFINDING

The pathfinding algorithm used in the automatic MEP route design is the A* algorithm. The A* algorithm is an algorithm that finds a path by prioritizing the search using a heuristic estimation of the available nodes from the start point to the destination point. The heuristic is a technique used to guess when time or information is insufficient. In the A* algorithm, the absolute distance from the destination and the total distance traveled by visiting various nodes from the starting point is used to determine the search priority. The search continues until the destination is reached. When the destination is reached, it goes back to the start point and visits previously visited nodes in the lowest cost order and selects the final path on the way back. A schematic diagram of the A* algorithm applied to this study is as follows:

**FIGURE 11.** Process of the path-finding algorithm applied in the research

First, in Step 1 of Error! Reference source not found., nodes that could be connected without obstacles are identified from the starting point. The total cost is calculated and allocated by adding the distance from the start point and the distance from the end point for each node. After that, the Cost 22 node is visited since it has the lowest cost from the start point. In Step 2, the total cost is identified for neighboring nodes that could be connected to the currently visited node, which is the Cost 22 node from Step 1. There are three nodes that can be
connected to a Cost 22 node, and each has the cost of Costs 29, 33, and 25. The node with the lowest cost among the three nodes is the Cost 25 node, and it is visited for further search. However, since there are no nodes around the Cost 25 node anymore, the Cost 29 node is visited because it is the node with the next lowest total cost. The process is repeated until the visited node has the end node as its neighbor node. In Step 3, the Cost 33 node could directly connect to the end node, and the initial stage of the pathfinding algorithm is completed. The final path calculation is performed by sequentially selecting the lowest cost among the visited nodes in the reverse order from the end node to the start node. For example, the Cost 33 node is selected first from the end node. The Cost 32 node has a lower cost than the Cost 33 node, but it is not selected since only visited nodes can be selected when selecting the final path node by going back in reverse order. At the Cost 33 node, the Cost 22 node has the lowest cost among the visited nodes, so it is selected as the next final path node. Finally, since the Cost 22 node could be directly connected to the start node, the pathfinding algorithm ends.

Among a number of visited nodes, the nodes selected as the final path nodes are the Cost 22 node and the Cost 33 node. The pathfinding algorithm is executed only when the connector type of the start and end points in the same MEP system is identical. After that, a modified start location and a modified end location are created from the connector, as shown in Error! Reference source not found., and nearby nodes are searched to calculate the cost and saved in the neighbor node list to find the next node to visit. This process of calculating and selecting paths for the MEP routes is shown in Error! Reference source not found..

The nodes in the neighbor list are assigned with a total cost value, and when the calculation is finished, the node with the minimum cost in the neighbor node list is selected as the next node to visit. Visited nodes have IsUsed variable set to true, and if there are several nodes with the same minimum cost, the node with the shortest distance from the starting node is visited first. The process is repeated until the destination is reached. When the destination, the modified end location, is reached, the final route is searched in the reverse order from the modified end point to the modified start point. The final path nodes are used in Step 4, the MEP route modeling, as shown in Error! Reference source not found..

3) PREFERRED NODE REPLACEMENT
When preferred nodes exist separately from the nodes extracted from the structure and their location is in the opposite direction to the destination, the MEP route passes through the nodes. However, when the MEP route model is placed, the preferred nodes are located inside the MEP route modeling elements, which can no longer be used as a pathfinding algorithm node. Therefore, additional preferred nodes are placed around the used preferred node, as shown in Error! Reference source not found., and the used preferred nodes are then removed to reduce the amount of unnecessary computation.

```csharp
foreach stC in edC:
    if stC.DuctSystemType == edC.DuctSystemType:
        currentNode = startLocation:
        do
            foreach point in Points:
                if pt.IsUsed == false & pt.IsCalculated == false:
                    Node = connectionPossible(point):
                    addToNeighbor(Node):
                    if currentNode == endLocation:
                        DestinationReached = true
                        while DestinationReached != true:
                            do
                                foreach point in Points:
                                    if pt.IsUsed == true:
                                        Node = connectionPossible(point):
                                        addToNeighbor(Node):
                                        if currentNode == neighbor:
                                            currentNode = findMinimumCostNode(neighbor):
                                            while currentNode == startLocation:

FIGURE 12. C# codes of the path-finding algorithm applied in the research
```
4. MEP Route Modeling

The optimal path selected after searching for the main-MEP route is most likely composed of diagonal nodes. The path was reviewed by determining whether nodes could be connected, even if the node is not in the orthogonal direction. However, Placing MEP routes diagonal is not recommended unless inevitable from management, construction, and manufacturing perspectives. Therefore, based on the optimal path calculated in Step 3 in Error! Reference source not found., the orthogonal path is created in Step 4 for each section of the optimal path.

The calculated final path consists of nodes. In Step 4, nodes are created at arbitrary orthogonal points, as shown in Error! Reference source not found., to find the orthogonal path between the start point and the end point for each section by dividing the calculated optimal path into small sections. After the nodes are created in the algorithm shown in Error! Reference source not found., the orthogonal path between the starting point and the ending point is calculated by limiting the connectable candidates to the orthogonal direction only and putting them in the neighbor list.

In the process of searching for the orthogonal path, there may be a case where the path search fails by colliding with the MEP routes of other MEP systems or the structural elements, as shown in Error! Reference source not found.. When the search for an orthogonal path only with nodes arbitrarily arranged at orthogonal points fails, a 3D grid is placed for a more precise and successful orthogonal path search, as shown in Error! Reference source not found.. After that, when the orthogonal MEP route calculation is performed for each section of the optimal path calculated in Step 3, the MEP route model and its fittings are placed, and the main-MEP route modeling process is completed.
5. Pathfinding for Sub-MEP Equipment
After the modeling of the main-MEP route is completed, the connection algorithm between the main-MEP route and the sub-MEP route is executed as Step 5 of Error! Reference source not found., and the algorithm is almost identical. The only difference is that, unlike the route connection between the main-MEP equipment, where the start point and the end point are clear, it is unclear which part of the main-MEP route is the optimal destination for the sub-MEP equipment. Therefore, the BFS algorithm that selects the destination among the main-MEP route is added to find the optimal destination, as shown in Error! Reference source not found..

![Image](A)

FIGURE 17. Algorithm for a connection between the main-MEP route and the sub-MEP equipment

V. Discussion
In response to the complexity of the structure of a building, it is difficult for a person to design and model MEP routes manually. Considering the overall project size of the construction site, the time and cost required for MEP design are not a critical problem for large construction companies unless it is a structure with a large amount of MEP, such as a hospital, plant, or factory. However, since MEP design offices mainly do MEP design work are small-scale companies subcontracted by construction companies, they are sensitive to the time and cost required for design, so that automation may have a significant effect. In addition, automating the MEP design process could reduce errors and construction delays caused by MEP design. Therefore, after analyzing and combining conventional pathfinding algorithms this study, suggests MEPAutoRoute, a modified pathfinding algorithm.

![Image](B)
As a result of applying and testing MEPAutoroute with seven different examples, as shown in Error! Reference source not found., operation time was shorter by an average of 33,800%. The average number of clicks was 1480% lower in all examples than that of manual MEP route modeling in conventional BIM tools. In addition, the larger and more complex the structure is, the more time is required for both the manual and automatic modeling of the MEP design function in conventional BIM tools. However, MEPAutoroute always requires the same effort to operate, regardless of building size. Furthermore, unlike previous studies that usually focus on connecting main-MEP equipment, the path between the sub-MEP and main-MEP belonging to the same system was also found and arranged according to the conditions in the developed MEP automation algorithm. In conclusion, it can be said that the modified A* pathfinding algorithm showed enough performance and potential to be applied in building MEP route design.

As shown in FIGURE 18, when a comparison is made with manual modeling, auto-route functions in a conventional BIM tool, and MEPAutoroute, the number of clicks and the complexity of the structure increase from (A) to (G), and the amount of equipment and system types are the same in all examples. In small-scale buildings, the distance between structural elements is narrow; thus, MEP routes being placed on the side of the path have no significant difference in terms of maintenance or aesthetic point of view. However, if the distance between structural elements is vast, MEP routes skewed to one side of the space could be a problem. To prevent skewing problem, preferred nodes were created and placed to require MEP routes to pass through specific locations, but the same problem occurs in areas where preferred nodes are not present. Therefore, there is a need for a function that automatically creates preferred nodes according to pre-set conditions rather than manually placing them.

Moreover, as the structure of a building becomes more complex, the connection point of the routes between the main-MEP route and the sub-MEP route becomes unclear. Thus, the BFS method was introduced, which sequentially searches for all adjacent nodes and finds the nearest path to the main-MEP route. However, the operation time required to search large and complex buildings is relatively long. In order to shorten the time required for the search, a method that prioritizes the search nodes in a specific area and direction first should be developed in future research.

This study was carried out as part of the development of an automation program for the design of a building MEP route that can be used in practice. In response to the nature of building MEP, where there are countless types, and the connection method differs greatly depending on the type, further research is required to realize the automation of MEP route design. Therefore, the MEP route design automation study was divided into three stages. In the first stage, we studied and developed the program that defines the route between MEP systems by using a pathfinding algorithm, MEPAutoroute, for building MEP systems, and that searches for an alternative route in the case of interference with other MEP systems. This paper corresponds to the first stage of the three stages and is the foundation for further stages. In the second stage, the pathfinding algorithm developed in the first stage will be modified for the MEP types that need to be buried inside the structure. In the third stage, the automation of MEP capacity optimization and the arrangement of the accessories necessary to install the MEP route, such as ducts and piping, will be studied.

### VII. Conclusion

In this study, a grid-based method composed of regularly arranged cells, a network node method with irregular locations and connection relationship information, and the strength of conventional pathfinding algorithms such as JPS, BFS, and DFS were converged and modified A* algorithm for the MEP route design.

1. A module that extracts nodes from the geometrical information of the structure and places them in an empty space where MEP routes can pass through was developed.
2) A module that calculates the path with the shortest distance between the start point and the end point through the extracted nodes according to the set conditions was developed.
3) A module that modifies the optimal path to move only in the orthogonal direction and finds an alternative path in the case of a collision with MEP equipment of other systems was developed.
4) For efficiency and optimization, the pathfinding algorithm was performed using a network node-based method, and the avoidance path in the case of a collision with other elements was calculated based on a temporarily created grid.
5) As a result of applying MEPAutoroute to seven example structures, it was confirmed that the performance was superior to that of the existing manual modeling method and the automatic path design function in conventional commercial BIM tools in terms of the time, cost, and effort required to use it. Therefore, if node creation from the structure and path selection according to the situation is further developed to reflect the actual MEP design standard, replacing the existing MEP design work is possible.

Furthermore, when the start point, end point, and nodes extracted from the structure are placed inside the structural elements, MEPAutoroute can be used to design the path of a buried MEP route type that only needs to move inside the structure. In addition, when the structure's external and internal MEP route modeling is completed, it will be possible to automate the MEP route design to a level that could suggest a detailed option, such as capacity optimization and the automatic arrangement of accessories based on MEP route model information. Additionally, since building information, including the MEP, has become digitalized, artificial intelligence (AI) can be integrated for a more precise design and the improved optimization of various features of buildings in future research [46-49].

Further research is required to solve the problems that were encountered in this study, namely, MEP routes following near-wall lines instead of the center of isles, manually placing preferred nodes, and the operation time of searching for a path between main-MEP routes and the sub-MEP equipment.

REFERENCES

[1] Lester, P.J.o.G.W.h.w.g.n.r.a.a.a., A* pathfinding for beginners. 2005.
[2] Zou, H., et al. Optimized application and practice of A* algorithm in game map path-finding. in 2010 10th IEEE International Conference on Computer and Information Technology. 2010. IEEE.
[3] El Jazzar, M., M. Piskernik, and H. Nassereddine. Digital twin in construction: an empirical analysis, in EG-ICE 2020 Proceedings: Workshop on Intelligent Computing in Engineering. 2020.
[4] Kim, K. and Sung, Park, Comparative analysis of the BIM status in the UK and US for improving the efficiency of construction project management process in Korea. 2012. 2(2): p. 1-16.
[5] Korea Institute of Construction Engineering and Management, Construction trend briefing – construction policy and management trend.
[6] Journal of Korea Institute for Structural Maintenance and Inspection , Kim,J., Core-element technology for high-rise building construction(focused on Lotte World Tower) 2016. 16(2): p. 18-22.
[7] 7. Park, J. and C.-H.J.J.o.K. Yeom, A Study on Priority of BIM Introduction Policy-Focusing on Overseas Cases and Analytic Hierarchy Process Analysis. 2021. 11(2): p. 17-23.
[8] Aljohani, A., et al., Construction projects cost overrun: What does the literature tell us? 2017. 8(2): p. 137.
[9] Kim, S.-H., et al., The development of a practical pipe auto-routing system in a shipbuilding CAD environment using network optimization. 2013. 5(3): p. 468-477.
[10] Phyo, K.-z. and M.M. Sein. Effective evacuation route strategy during natural disaster. in Proceedings of the APAN–Research Workshop. 2017.
[11] Gelenbe, E., F.-H.C. Wu, and M.W. Applications, Large scale simulation for human evacuation and rescue. 2012. 64(12): p. 3869-3880.
[12] Kim, K. and Y.-C.J.A.S. Lee, Automated generation of daily evacuation paths in 4D BIM. 2019. 9(9): p. 1789.
[13] Wang, B., et al., BIM based virtual environment for fire emergency evacuation. 2014.
[14] Kim, S., B. George, and S. Shekhar. Evacuation route planning: scalable heuristics, in Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems. 2007.
[15] Shekhar, S., et al., Experiences with evacuation route planning algorithms. 2012. 26(12): p. 2253-2265.
[16] Marzouk, M. and I.J.S.S. Al Daour, Planning labor evacuation for construction sites using BIM and agent-based simulation. 2018. 10(9): p. 174-185.
[17] Kim, H.-G., et al., Reducing the search space for pathfinding in navigation meshes by using visibility tests. 2011. 6(6): p. 867-873.
[18] Shahi, G.S., et al., A comparative study on efficient path finding algorithms for route planning in smart vehicular networks. 2020. 7(5): p. 157-166.
[19] Nazari, S., et al, An advanced algorithm for finding shortest path in car navigation system, in 2008 First International Conference on Intelligent Networks and Intelligent Systems. 2008. IEEE.
[20] Dolgov, D., et al., Path planning for autonomous vehicles in unknown semi-structured environments. 2010. 29(5): p. 485-501.
[21] Garrido, S., et al Path planning for mobile robot navigation using voronoi diagram and fast marching, in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006. IEEE.
[22] Adam, A., et al. A particle swarm optimization approach to Robotic Drill route optimization. in 2010 fourth Asia international conference on mathematical/analytical modelling and computer simulation. 2010. IEEE.
[23] Albinu, A., et al., A novel clustering based genetic algorithm for route optimization. 2016. 19(4): p. 2022-2034.
[24] Barraquand, J., et al., Numerical potential field techniques for robot path planning. 1992. 22(2): p. 224-241.
[25] Ge, S.S., Y.J.J.I.T.o.r. Cui, and Automation, New potential functions for mobile robot path planning. 2000. 16(5): p. 615-620.
[26] Loong, W.Y., L.Z. Long, and L.C. Hun, A star path following mobile robot, in 2011 4th International conference on mechatronics (ICOM). 2011. IEEE.
[27] Garcia, M.P., et al., Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. 2009. 9(3): p. 1102-1110.

[28] Li, F., et al., Universal path planning for an indoor drone. 2018. 95: p. 275-283.

[29] Rodenberg, O., et al., Indoor A* pathfinding through an octree representation of a point cloud. 2016. 4: p. 249-255.

[30] Kilic, K.I. and L. Mostarda. Optimum Path Finding Framework for Drone Assisted Boat Rescue Missions. in International Conference on Advanced Information Networking and Applications. 2021. Springer.

[31] Habermann, M.J.S.S.U., Drone Path Planning. 2018.

[32] Li, F., et al., Universal path planning for an indoor drone. 2018. 95: p. 275-283.

[33] Chang, Y.S. and H.J.J.E.S.w.A. Lee, Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. 2018. 104: p. 307-317.

[34] Wu, Y., et al., Swarm-based 4D path planning for drone operations in urban environments. 2021. 70(8): p. 7464-7479.

[35] Park, J.-H. and R.L.J.E.S.w.A. Storch, Pipe-routing algorithm development: case study of a ship engine room design. 2002. 23(3): p. 299-309.

[36] Dong, Z. and X.J.I.A. Bian, Ship pipe route design using improved A* algorithm and genetic algorithm. 2020. 8: p. 153273-153296.

[37] Kang, S.-S., S. Myung, and S.-H.J.J.o.S.P. Han, A design expert system for auto-routing of ship pipes. 1999. 15(01): p. 1-9.

[38] Fan, X., Y. Lin, and Z. Ji. The ant colony optimization for ship pipe route design in 3D space. in 2006 6th World Congress on Intelligent Control and Automation. 2006. IEEE.

[39] Xiaoming, F., L. Yan, and J.J.J.o.S.P. Zhuoshang, Ship pipe routing design using the ACO with iterative pheromone updating. 2007. 23(01): p. 36-45.

[40] Min, J.-G., et al., Faster pipe auto-routing using improved jump point search. 2020. 12: p. 596-604.

[41] Kang, J.Y., B.S.J.I.J.o.N.A. Lee, and O. Engineering, Optimisation of pipeline route in the presence of obstacles based on a least cost path algorithm and laplacian smoothing. 2017. 9(5): p. 492-498.

[42] Anisyah, A.S., P.H. Rusmin, and H. Hindersah. Route optimization movement of tugboat with A* tactical pathfinding in SPIN 3D simulation. in 2015 4th International Conference on Interactive Digital Media (ICIDM). 2015. IEEE.

[43] Nash, A., et al. Theta*: Any-angle path planning on grids. in AAAI. 2007.

[44] Liu, B. Intelligent route finding: combining knowledge, cases and an efficient search algorithm. in ECAI. 1996. Citeseer.

[45] Cui, X., H.J.J.o.C.S. Shi, and N. Security, A*-based pathfinding in modern computer games. 2011. 11(1): p. 125-130.

[46] Heo, S., et al., Challenges of Data Refining Process during the Artificial Intelligence Development Projects in the Architecture, Engineering and Construction Industry. 2021. 11(22): p. 10919.

[47] Heo, S., et al., Flip Side of Artificial Intelligence Technologies: New Labor-Intensive Industry of the 21st Century. 2021. 34(5): p. 327-337.

[48] Na, S., S.-J. Heo, and S.J.A.S. Han, Construction Waste Reduction through Application of Different Structural Systems for the Slab in a Commercial Building: A South Korean Case. 2021. 11(13): p. 5870.

[49] Shin, Y., et al., An Image-Based Steel Rebar Size Estimation and Counting Method Using a Convolutional Neural Network Combined with Homography. 2021. 11(10): p. 463.
Wonjun Choi (M’93) Received the B.S. degree in architectural engineering in 2019. He is currently pursuing the Ph.D. degree in structural engineering at Dankook University, South Korea. Since 2018, he is a Research Assistant at Dankook University.

He is the author of 2 papers, and holds 1 patent and 9 software copyrights. His research interest includes Building Information Modeling (BIM), Digital Twin (DT), and software design and development.

Cheekyeong Kim (M’63) Received the B.S. degree in architectural engineering in 1986, M.S. degree in structural engineering in 1988, and doctorate degree in structural engineering in 1992 from Dankook University, South Korea. From 1992 to 1996, he was a team leader of the system engineering team from the structural consultant company Chang Minwoo Inc. From 1996 to 2014, he was a professor at the Sunmoon University, South Korea. Since 2015, he is a professor at Dankook University, South Korea.

He is the author of 46 papers, holds 2 patents, and 14 software copyrights. His research interest includes Building Information Modeling (BIM), Digital Transformation (DT) in construction, parametric design, Artificial Intelligence (AI), sensor, Digital Twin (DT), and software design and development.

Seokjae Heo (M) Received the B.S. degree in architectural engineering in 2007 and M.S. degree in construction material in 2009 from Dankook University, South Korea. In 2018, he received Ph.D. degree in Civil Structural Engineering from the Dankook University, South Korea. Since 2018, he was a Research Professor in Architectural Engineering Department at Dankook University, South Korea.

He is the author of 46 papers, holds 10 patents. Dr. Heo was a recipient of the Dankook University ‘Byeongjeong’ Prize for academic papers in 2009, Air Force Headquarters Commendation for Combat Command Inspection in 2011, and Korean Society of Mechanical Engineers Award in 2016 and 2017.

His research interest includes large-scale concrete structures, 3D structural analysis, Artificial Intelligence (AI) in construction, sensor, Digital Transformation (DT), and Digital Twin (DT).

Seunguk Na (M’78) Received the B.S. degree in architectural engineering in 1998 and M.S. degree in construction material in 2004 from Dankook University, South Korea. In 2010, he received MSc degree in management of projects in construction, and Ph.D. in project management in 2015 from the University of Manchester, UK.

From 2017 to 2018, he was a Research Professor in Architectural Engineering Department at Songwon University, South Korea. Since 2019, he is a Research Professor in Architectural Engineering Department at Dankook University, South Korea.

He is the author of 10 SCI/SCIE papers and 7 SCOPUS papers. His research interest includes Project Management (PM), Construction Management (CM), Building Information Modeling (BIM), and greenhouse gas emission in the construction industry.