Abstract—In this paper, we propose an algorithm to generate optimal trajectory path considering the complex operating environment of excavator front part composed of the boom, arm, and bucket by using genetic algorithm. In order to express motion in space, we propose a method of coordinate plane space of grid cell, and define the fitness value by path distance. After generating chromosome candidates for each motion unit based on the polygonal structure of the front part of the excavator, we calculate the fitness value about each chromosome. The crossover and mutation operations between the chromosomes selected through roulette wheel of top 20% are repeatedly performed to generate paths with optimal fitness values. This paper verifies the structural analysis of the front part of excavator and the utility of the genetic algorithm to optimize the path in the grid space.

Keywords—optimal path, excavator, complex structure, genetic algorithm, evolutionary computation.

I. INTRODUCTION

Recently, unmanned and automated excavators have been required in the industrial field due to the aging and unskilled operators of excavator drivers. In this respect, studies have been continuously carried out to automate the front part, which accounts for more than 90% of excavator work. Therefore, the optimal path generation of the front part of the excavator becomes an important issue in order to maximize task efficiency (safety, work speed, fuel efficiency, etc.). Also, an unmanned excavator must have the capability to produce optimal motions in a reliable and efficient manner.

Many previous studies are performed on the use of a central server to calculate strategy and robot behavior [1][2]. However, there is a limitation that the central control method cannot maximize the efficiency of the work because it has difficulties in real time response such as delay due to communication and data processing time. Therefore, recent studies have begun to transfer more intelligence to machines, and building standalone systems with intelligence is of interest in many areas. This paper aims to calculate the motion of the multi-joint machine to determine the optimal path to avoid the collision and reach the desired destination efficiently within the available route. The path planning algorithm is simply to find the shortest path while avoiding obstacles between the start and end points.

Also, from an implementation standpoint, there is a constraint that the task must be performed within the limited resources of the machine-mounted microcontroller. Discrete path planning algorithms such as potential fields, grid-based algorithms, splines, and tangent find require too much CPU performance or do not meet memory constraints [3][4][5]. The genetic algorithm (GA) presented in this paper is one of the solutions to overcome the limitation of the algorithms described above. GA is a way to cover a large search space and use a relatively small amount of memory and CPU resources. In previous studies, the movement of the joints was limited. On the contrary, in this paper, we propose an environment and algorithm that considers motions with a multi-joint connection structure using GA, and define the fitness value using energy cost for efficient management of power resources distributed to each part. On the other hand, we focus on the motions that are fast, high power-efficient, and smooth movement with a global optimal path. Our contributions lie in 1) generating the fastest possible efficient movement for digging a path; 2) maximum fuel efficiency (minimum torque/time); 3) shortest movement to target point; 4) considering limited CPU resource; and 5) implementation to the embedded board.

II. RELATED WORKS

Many previous studies have described a given environment as a grid-based space; dividing the given environment into squared cells. We restrict the movements because each point always moves to the center of eight adjacent cells (up, down, right, left, and four diagonal cells). This was done for simplicity in several studies as shown in Figure 1, but it did not represent a complex path of movements [6-8]. In order to solve this problem, we propose a new algorithm that can be applied to A* algorithm [9], fuzzy theory [10], simulated annealing [11], artificial latent field (ALF) [12], Dubin theorem [13], Voronoi diagram, evolutionary algorithm, visibility graph [14, 15], random extraction [16-19], and natural induction algorithm [20-22].

Visibility graphs have been proposed to avoid collisions with obstacles in 1979, which was modified to take advantage of the constituent spatial approach [14], and [15] used visibility graphs in their studies where the mobile agent and obstacles were assumed to have a cycloid shape. They chose the edge of each obstacle as a node. They created a visibility tree by arranging the linked nodes until they contain the tree target node.

However, this research lacks research on structures that are coupled with the dynamics of machines such as excavators. Another important factor is energy cost. Most of the previous studies have not taken into consideration the resources that
operate. This paper proposes a GA considering the above
dynamics and energy cost.

III. BACKGROUNDS

A. Structure and operations of the excavator

The excavator front is composed of the boom, arm, and
bucket, and we express as a point on the x, y plane where the
moment acting on the system is zero. Figure 2 shows the
specification of front movement of the actual excavator. We
simplify the structure and digitize it by connecting three points
and a straight line.

Figure 3 defines the connection point $F$ between the boom
and arm at the center $O$ of the movement and the connection
point $P$ with the bucket when expressed on the x, y plane. The
boom and arm lengths are represented as $l_2$ and $l_1$, respectively.
As we want to verify the algorithm except for the points at the
end of the bucket, it is assumed that the $P$ arrives at the final
target point.

The excavator is a structure that moves the boom, arm, and
bucket using the hydraulic pressure from the main pump and can
be adjusted by the joystick of driver. In addition, the combined
simultaneous adjustment of the joystick is a system in which the
main pump's hydraulic pressure is proportionally distributed.
Therefore, when moving at the same time, the velocity changes
exponentially rather than linearly, and consequently the overall
speed gets lowered. This is caused by physical factors such as
the length of the cylinder in which the hydraulic pressure is
formed, and the driver needs a strategic selection for effective
movement.

B. Definition of problem

Previous researchers investigated the optimal path
generation for various applications, but they did not consider the
dynamic environment where joint limitations exist such as the
excavator front. We aim to define the starting and target points
in arbitrary space and create the efficient path of the excavator
front which is a jointed structure for the digging operation that
occupies most of the excavator work.

We construct a grid-based space used in many studies to
construct an environment for path generation. In digging
operation, we define the structural operation range of the
excavator front as shown in Figure 4.

We create a jointed structure object as shown in Figure 5 by
expressing the above working range and front movement in the
grid cell. The value of each cell is set according to the x, y
coordinate plane. In this paper, the range of each axis is defined
from -10 to 10. Also, we represent the front structure looks like
shoulder, elbow, and wrist. In this paper, the wrist is not itself a
joint, but we can consider it as our end-effector. If we constrain
the shoulder to the origin, we can write the rotating kinematics
for the elbow and the wrist in Figure 6.

We calculate to determine whether the structure is available
or not. The calculation of the position of the point of the wrist
follows Equations (1) and (2) for $x$ and $y$, respectively. In this
paper, because the grid space value is used, $\theta_\theta, \theta_\psi$, which can
calculate the current attitude of the structure based on $x$ and $y$
coordinate, they can be calculated according to Equations (3)
and (4), respectively. We also follow Equations (5), (6) and (7) for the elbow as each part performs a complex or single action. As a result, the calculated value is used as a factor for determining the next moving point, and also as a discriminant value for the single or combinational operation.

\[
x_w = l_0 \cos(\theta_0) + l_1 \cos(\theta_0 + \theta_1)
\]
\[
y_w = l_0 \sin(\theta_0) + l_1 \sin(\theta_0 + \theta_1)
\]
\[
\theta_0 = \tan^{-1} \left( \frac{y_w}{x_w} \right) - \tan^{-1} \left( \frac{l_1 \sin(\theta_1)}{l_0 + l_1 \cos(\theta_1)} \right)
\]
\[
\theta_1 = \cos^{-1} \left( \frac{x_w^2 + y_w^2 - l_0^2 - l_1^2}{2l_0 l_1} \right)
\]
\[
x_e = l_0 \cos(\theta_0)
\]
\[
y_e = l_0 \sin(\theta_0)
\]
\[
\theta_0 = \tan^{-1} \left( \frac{y_e}{x_e} \right)
\]

The starting point is expressed as \(P_n\) and the target point as \(P_m\). As our goal is to generate the minimum trajectory path distance from \(P_n\) to \(P_m\), it can be calculated as the following Equation (10).

\[
\min f(P_n, P_m) = d(P_n, P_m) + \sum_{i=1}^{n-1} d(P_i, P_{i+1}) + d(P_n, P_m)
\]

C. Energy cost

Based on the operating mechanism, the boom and the arm are connected to each other so that two choices can be made: moving two parts at the same time or moving one part at a time. When two operations are performed at the same time, the hydraulic pressure is proportionally distributed and the overall motion gets slowed down. Therefore, we define the energy cost as one of the factors for generating the optimal path by proportionally dividing the motion of the point \(P\) and point \(P\) as Equation (11).

\[
\text{energy cost}(t) = \begin{cases} \frac{E}{P} & d(F) < d(P) \\ \frac{F}{E} & d(F) > d(P) \\ 1 & \text{else} \end{cases}
\]

Since the fitness value of the GA for path planning is basically proportional to the moving distance of the point, we calculate the optimal path by minimizing the travel distance as shown in the following Equation (12). Euclidean distance is used for the distance calculation and Equation (11) with energy cost is defined for the fitness value.

\[
\min f(P_n, P_m) = d(P_n, P_m) + \sum_{i=1}^{n-1} d(P_i, P_{i+1}) + d(P_n, P_m)
\]

Finally, we define the fitness value by calculating the Euclidean distance between \(i\)th and \((i+1)\)th with the energy cost.

\[
\text{fitness value} = \min f(P_n, P_m) \ast \text{cost}(t)
\]

IV. THE PROPOSED METHOD

In this paper, we propose a trajectory path generation method by applying the modified GA from chromosome generation to
genetic operations. Unlike many previous studies that used random extraction to generate viable chromosomes, this paper considers the energy efficiency of the actual equipment and considers the energy cost, and the initialization, crossover, and mutation perform the operations. Chromosomes are reproduced according to the order of the fitness values and the diversity is achieved through random operations. The details of the proposed GA are as follows.

1) Set the environment using grid-based space
2) Generate a feasible path and a gene encoding
3) Reproduce chromosomes
4) Apply crossover
5) Apply mutation

A. Grid-based space

As many studies have represented given environments as grid-based spaces, the given environments are divided into squared cells, we assume that our environment is consisted as 10 to 10 squared cells in Figure 8. The shaded part in the figure reflects the working range of Figure 4 and indicates the starting point, the target point, and the environmental factor when determining the path and the path of obstacles.

B. Generate feasible path and gene encoding

In this paper, all GA chromosomes are expressed as an array of free or active cells with starting and target cells in which the front moves. A free cell is a movable cell, meaning cells in which a structure can move. Because there are a large number of free cells in a given environment, it is very difficult to determine how many free cells should be selected. Moreover, the generated path must also be executable. This paper proposes a new path planning method for effective paths for GA initialization. We generate an individual that reflects various paths and then encode it as follows.

1) Initialize the environment
2) Check and update the current position
3) Deduce the moving direction
4) Check collide with any obstacle or goal
5) Check whether any effective free cells
6) Selection of effective free cells
7) Move to random effective free cell
8) Repeat 2) process

The overall process can be defined as shown in Figure 9. Each individual is implemented with a feasible path-finding algorithm and randomly generated for all forward or backward approaches as shown in Figure 10. The generated path can be coordinated and expressed as a one-dimensional vector.

C. Reproduction

After evaluating the fitness value of each chromosome, the chromosomes are ranked in order of fitness values. Figure 11 shows the reproduction process. Top 20 chromosomes are copied directly to the next generation, and the remaining 40 chromosomes are reproduced. Half of the bottom 40 chromosomes are reproduced via crossover, and the other half is generated via mutation. The redundant chromosomes are removed.

D. Crossover

Next, we perform a crossover operation and each parent chromosome is randomly selected from the top 20 chromosomes. Any gene of the mother chromosome, that is, an arbitrary point is selected as an exchange point, and the gene is transferred from the (i+1)th to the ith selected gene to generate a descendant chromosome. A child has the gene of mixed gene from parent 1 and 2. Figure 12 shows the crossover process.

In the case of crossover, some of the generated children may have a return route that must be returned to the way they came, and sometimes the inefficiency may occur, such as a large value between them. Also, when the children are produced via the crossover, there is a chance that infeasible paths are created. Therefore, if infeasible paths are created (e.g., collapse), the corresponding chromosome is discarded, and if it is not infeasible, the path to the intermediate route must be recalculated.

Therefore, we can see that the child computed above is a sub-problem for the new target point as $P_{t,s}$ starting from the swapping point. Thus, recalculating the new feasible path can result in undesirable results, such as when the path is rotated as
shown in Figure 14. Also, this can vary depending on the swapping point of the crossover. This is the part where the GA can deal with. As the algorithm is repeated, only the chromosome with the optimal fitness value is left.

![Diagram showing chromosomes](image)

(a) Parent 1 
(b) Parent 2 
(c) Chromosomes

Fig. 10. Shortest path with different direction limitations

![Diagram showing reproduction of chromosomes](image)

E. Mutation

In GA, the mutation can be regarded as an important operation in order to maintain the diversity of chromosomes and eliminate the local optimal problem.

In this paper, random rth genes are selected from the selected chromosomes, adopting a random mutation in the roulette wheel selection among the top 20% chromosomes.

Mutant chromosomes inherit the genes of the chosen chromosomes before rth gene and it then generates a random path; the path is generated from a new start point. In Figure 15, the chromosome will fetch the first three genes from the parent and the remaining chromosomes will be filled with the new chromosome, creating a new chromosome.

V. EXPERIMENTAL RESULTS

We implement the algorithm in the python and implemented and verified the algorithm in the JETSON TX2 board platform, which is NVIDIA 's embedded board, to verify it for real excavator application. The specification of JETSON TX2 Board is as shown in Figure 16, and it is used as an actual embedded board for vehicles or drone, etc.

In addition, we perform an experiment on a two-link structure that does not take the bucket into consideration, and a three-link structure that takes into account buckets (the structure of the small radius at the front end). In this experiment, the obstacle is randomly generated for each experiment and two or three obstacles are added to represent obstacles that can occur in actual digging work.

Finally, we perform a comparison with the A* algorithm used in previous studies to verify the effectiveness of the proposed method. In the experiment, the iterative algorithm is designed to perform up to 400 times when the result value changes continuously, and it is designed to stop when the evolution is not over a certain value.

![Diagram showing crossover operation](image)

Fig. 12. Crossover operation

![Diagram showing children of crossover operation](image)

(a) Parent 1; (b) Parent 2; (c) Child 1; and (d) Child 2.
three obstacles. As can be seen in Figure 18, the path is more complicated than before and the travel distance is longer. Table II summarizes the time required for calculation and the fitness value. It can be confirmed that the time required for calculation increases rapidly with the joints, and the fitness value itself is also relatively large.

![Figure 14. Reproduced children of crossover operation. (a) Child 1; (b) Child 2; and encoded chromosomes for child.](image)

![Figure 15. Mutation operation](image)

![Figure 16. JETSON TX 2 Embedded board](image)

A. Two links

In the experiment, the length of each part can be defined by the user, and the experimental result is the movement of the two links with the length of four. The obstacle was set to two, and fitness value has converged to 8.434355. Figure 17 shows the generated trajectory paths with fitness, and Table I shows the average and best values of fitness and CPU running time.

B. Three links

We also proceed with the experiment on the three links with the bucket. Each length was defined as 4, 4, and 2, and two or

<table>
<thead>
<tr>
<th>Category</th>
<th>Fitness Value (Average)</th>
<th>CPU Time (s) (Average)</th>
<th>Best Fitness Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two links, two obstacles</td>
<td>8.340809</td>
<td>95.23752189</td>
<td>8.434355</td>
</tr>
<tr>
<td>Two links, three obstacles</td>
<td>8.022782</td>
<td>96.12157893</td>
<td>8.552833</td>
</tr>
</tbody>
</table>

Finally, we prove the effectiveness of the proposed method by comparing the A* algorithm with the computation time and fitness, and the proposed method is more robust in limited hardware resources. The A* algorithm is widely known as a method of calculating the optimal path by searching all the paths in the grid space. As a result, as shown in Table III, it is confirmed that the CPU time is shorter and the fitness value is almost the same. We can conclude that GA is better and faster method.

VI. CONCLUSION

In this paper, we have proposed a GA approach that can perform path planning around the axis of rotation of the whole joint in excavator. We have configured the grid cells to take into account the movement of the boom, arm, and bucket on the connected axis for a particular axis. We have added the energy cost to the fitness value and calculated the optimal path for each route, taking energy consumption into account. Finally, we have
verified the effectiveness of the proposed method by considering various types of articulated systems.

In the future, we hope that the proposed method can be extended to other excavator route creation and planning for various operations. In addition, we will apply the function to the actual vehicle to verify and mass-produce the motion in the actual vehicle.

Fig. 18. Generated trajectory path and calculated fitness value with three links and 2 or 3 obstacles.

TABLE II. COMPARISON WITH A* ALGORITHM

<table>
<thead>
<tr>
<th>Category</th>
<th>Proposed Method CPU Time Average (sec)</th>
<th>Proposed Method Best Fitness Value</th>
<th>A* Algorithm CPU Time (sec)</th>
<th>A* Algorithm Best Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two links, two obstacles</td>
<td>95.237521</td>
<td>8.4343</td>
<td>115.2357</td>
<td>8.2123</td>
</tr>
<tr>
<td>Two links, three obstacles</td>
<td>96.1215</td>
<td>8.5528</td>
<td>132.1233</td>
<td>8.2231</td>
</tr>
<tr>
<td>Three links, two obstacles</td>
<td>932.3811</td>
<td>16.3385</td>
<td>1678.2173</td>
<td>15.4221</td>
</tr>
<tr>
<td>Three links, three obstacles</td>
<td>1612.2849</td>
<td>34.6711</td>
<td>2033.1647</td>
<td>35.2134</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENT

This work has been supported in part by the Doosan Infracore Co., Ltd.

REFERENCES