Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/132/2024.20625

Path Planning for Robots in Fire Scenarios based on Dijkstra Algorithm and Genetic Algorithm

Yingwei Song[1,4], **Qihong Duan**[2,5], **Juntian Liu**[3,6]

[1]Xidian University, Xi'an, China
[2]Beijing Jiaotong Unitversity, Beijing, China
[3]Beijing No.2 highschool, Beijing, China

[4]songyingwei@stu.xidian.edu.cn
[5]22211141@bjtu.edu.cn
[6]m1a1rptc@gmail.com

Abstract. Robots have emerged as a pivotal element in enhancing firefighting operations, offering a blend of efficiency and safety to human responders. This paper delves into the development of a path planning strategy for robots navigating through fire scenarios. Particularly we fused the Dijkstra Algorithm and Genetic Algorithm (GA). Our methodology commences with a simplified yet comprehensive definition of the fire environment, incorporating factors such as obstacle height, surface roughness, and fire sources. The environment and surroundings are represented by a 2.5-dimensional grid map. On the other hand, the robot's traversal and ascent capabilities modeled to reflect varying velocities across different terrains. The Dijkstra Algorithm is subsequently utilized to identify the optimal path from the starting point to the destination, ensuring a balance between minimal traversal time and reduced thermal exposure. Our results, demonstrated through MATLAB simulations, reveal a marked improvement in path planning when GA optimization is applied. The comparative analysis across three scenarios underscores the versatility and effectiveness of our approach, showcasing a significant reduction in both traversal time and thermal exposure. Index Terms—Robots, Firefighting, Path planning, Dijkstra, GA

Keywords: Robots, Firefighting, Path planning, Dijkstra, GA

1. Introduction

Fires pose a significant threat to the safety of people's lives and property. According to statistics from the National Interagency Fire Center (NIFC), there were 56,580 wildfires across the United States in 2023, burning 2.7 million acres of land [1]. As a result, the application of robots in firefighting has become a critical research area within the robotics industry. These robots not only enhance firefighting efficiency but also contribute to the safety of firefighters. This paper presents a method for optimizing and constructing a cost map using Genetic Algorithm (GA) and Dijkstra Algorithm under known fire conditions. Comparative verification demonstrates the superiority, feasibility, and versatility of this approach [2].

DOI: 10.54254/2755-2721/132/2024.20625

2. METHODS AND METHODOLOGY

2.1. Definition of fire environment

To mitigate the computational complexity, this paper implements the following simplifications for the fire scenario and the robot:

(i) The fire environment map only includes height of obstacle, surface roughness, and fire sources.

(ii) The robotic system is endowed with the capability to ascend obstacles of a certain height. However, it exhibits varying velocities when traversing surfaces of different roughness and ascending obstacles of varying heights.

(iii) A 2.5-dimensional grid map is utilized to represent the fire environment, concurrently mapping the robot as a singular point within the map.

For instance: The value of $map(x, y).h$ represents the height of obstacle in $cell(x, y)$; the value of $map(x, y).temp$ represent the temperature of $cell(x, y)$ and the value of $map(x, y).r$ represent the roughness of $celll(x, y)$.

Considering the robot exhibits different velocities when ascending the edge of an obstacle and moving along the surface of the same obstacle, despite the identical elevation values at these times, this paper use the variance of height within the grid map to represent terrain fluctuations. Let denote the height at position in the height map. The local mean and local variance can then be calculated as follows:

2.1.1. Local Mean

$$\mu(x, y) = \frac{1}{4}\sum_{i=0}^{1}\sum_{j=0}^{1} map(x + i, y + j).h \tag{1}$$

Where, $\mu(x, y)$ represents the average of the height value at position (x, y) and all the height values within its 2×2 neighborhood.

2.1.2. Local Mean of Squared Heights

$$E\Big[map^2(x, y).h\Big] = \frac{1}{4}\sum_{i=0}^{1}\sum_{j=0}^{1} map^2(x + i, y + j).h \tag{2}$$

Where, $E\Big[h^2(x, y)\Big]$ denotes the average of the squared height values at position (x, y) and within its 2×2 neighborhood.

2.1.3. Local Variance

$$\sigma^2(x, y) = E\Big[map^2(x, y).h\Big] - \mu^2(x, y) \tag{3}$$

Where, $\sigma^2(x, y)$ represents the local variance of the height values at position (x, y). It is obtained by calculating the difference between the mean of the squared heights and the square of the local mean.

For better readability of mathematical symbols, in the following text, we will use $h_{\text{var}}(x, y)$ to represent $\sigma^2(x, y)$. The physical significance of $h_{\text{var}}(x, y)$ is the degree of height variation at $cell(x, y)$.

2.2. Construction of costmap

In the pursuit of facilitating our research within the relatively straightforward MATLAB environment and to diminish the computational time required to attain accurate experimental outcomes, this paper innovatively introduces the variable $time$, which signifies the duration the robot expends traversing each cell within the map.

$$time(x, y) = \alpha \cdot \frac{map(x, y).r}{v_f} + \beta \cdot \frac{1 + h_{\text{var}}(x, y)}{v_h} \tag{4}$$

DOI: 10.54254/2755-2721/132/2024.20625

Where, vf denotes the robot's velocity performance on surfaces with varying degrees of roughness, while vh signifies the robot's velocity performance when ascending obstacles of differing heights. The coefficients α and β serve as weights that can be manually adjusted through simulation experiments.

By manipulating the robot's traversal over surfaces with diverse roughness levels and its ascent of obstacles with varying heights, and subsequently measuring the requisite time, these values can be estimated through an analysis of the simulation data.

Following manual simulation experiments, this paper has derived the specific values for the parameters α, β, v_f, and v_h, which are 1, 4, 1, and 1, respectively. Consequently, the expression for $time$ is as follows:

$$time(x, y) = map(x, y).r + 4 \cdot (1 + h_{\text{var}}(x, y)) \tag{5}$$

Ultimately, within the costmap, the cost value of each cell can be succinctly defined by the following formulation:

$$cost(x, y) = w_1 \cdot time(x, y) + w_2 \cdot map(x, y).temp \tag{6}$$

Where, $w1$ and $w2$ symbolize the relative significance of time and temperature, respectively, on the costmap's cost value. These weights can be optimized through the Genetic Algorithm (GA) to be discussed later, which aims to achieve an optimal balance between the effects of time and temperature on the costmap. This optimization process is designed to assist the robot in planning a path that not only minimizes the time taken from the starting point to the destination but also maximally reduces the thermal exposure experienced by the robot during the traversal.

2.3. Dijkstra algorithm for path planning

The Dijkstra algorithm is a greedy algorithm used to solve the single-source shortest path problem. It is capable of finding the shortest path from a starting node to all other nodes in a graph [3]. The main principle of the algorithm is to maintain a set of unprocessed nodes and gradually expand the known shortest paths until the shortest path to the target node is found [4]. After obtaining the cost map, we use the Dijkstra algorithm to find the optimal path. Compared to common used algorithms like PRM, this algorithm returns a concrete result which is good for optimization. The pseudo code is shown in followings [5]:

2.4. D.Application of Genetic Algorithm (GA)

The Genetic Algorithm (GA) is a search algorithm that simulates the principles of natural selection and genetics to solve optimization and search problems [6, 7]. The alternative or potential solutions can be improved by GA through a process could mimicking biological evolution, which, in particular, includes inheritance, mutation, selection, crossover and so on. GA performs a global search across the entire solution space rather than being confined to local regions, and it can evaluate the fitness of multiple individuals in parallel, which facilitates the discovery of a global optimum or near-optimal solution [8].

The fitness function plays a crucial role in GA, serving as the metric for evaluating the quality of individuals (chromosomes), thereby determining the direction of the search and the effectiveness of optimization [9, 10]. In this project, we define the fitness function based on the cost map and the optimal path as follows:

We calculate the optimal path P from the starting point to the endpoint using the Dijkstra algorithm. The path consists of a series of coordinate points (ri, ci):

$$P = \{(r_1, c_1), (r_2, c_2), ..., (r_n, c_n)\} \tag{7}$$

The fitness value is calculated along the path P as the sum of the temperature and time values at each point, As shown in the following formula:

$$fitness = \sum_{i=1}^{n} \left[\lambda_1 \cdot map(r_i, c_i).temp + \lambda_2 \cdot time(r_i, c_i)\right] \tag{8}$$

DOI: 10.54254/2755-2721/132/2024.20625

Algorithm 1 Dijkstra's Algorithm

Require: Graph, source
Ensure: path, distance[target]

```
for each node in Graph do
    distance[node] ← ∞
    previous[node] ← UNDEFINED
end for
distance[source] ← 0
priority_queue ← {source}
while priority_queue is not empty do
    current_node ← node with smallest distance
    if distance[current_node] = ∞ then
        break
    end if
    for each neighbor of current_node do
        new_distance ← distance[current_node] + edge_weight(current_node, neighbor)
        if new_distance < distance[neighbor] then
            distance[neighbor] ← new_distance
            previous[neighbor] ← current_node
            priority_queue.add(neighbor)
        end if
    end for
    priority_queue.remove(current_node)
end while
path ← [ ]
current_node ← target
while previous[current_node] is defined do
    path.prepend(current_node)
    current_node ← previous[current_node]
end while
return path, distance[target]
```

Where, λ_1 and λ_2 are weight coefficients. Considering the difference in the order of magnitude of time and temperature in the above formula, this paper sets the values to 1 and 9 respectively [11].

As shown in the Figure 1 , we visualize the genetic algorithm using a flowchart.

3. Result

To clearly demonstrate the advantages of applying GA and the rationality of the cost map definition, we compare the results under three different scenarios:

- **Case 1:** Cost map defined solely based on roughness and height variance, without GA optimization.
- **Case 2:** Cost map established based on temperature, roughness and height variance, w1 and w2 initialized to 1, without GA optimization.
- **Case 3:** Cost map established based on temperature, roughness, and height variance, with GA optimization.

Using MATLAB to simulate these three scenarios, we obtain different outcomes:

Initially, MATLAB was utilized to generate visual representations of the map's roughness, temperature, and elevation, as depicted in Figures 2, 3, and 4, respectively.

DOI: 10.54254/2755-2721/132/2024.20625

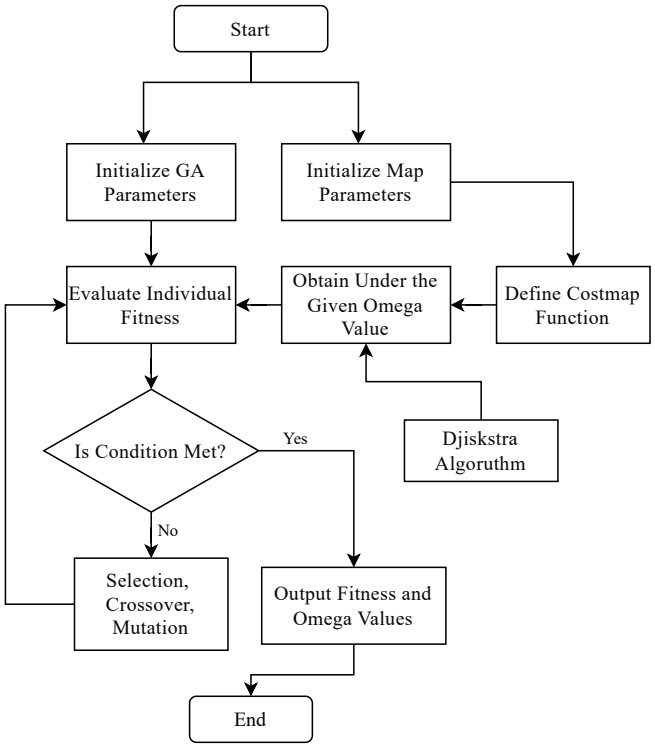

Figure 1. Genetic Algorithm Flowchart

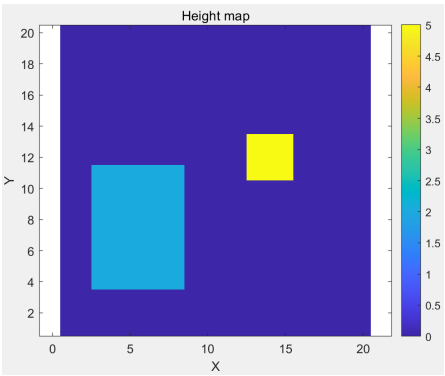

Figure 2. Height map

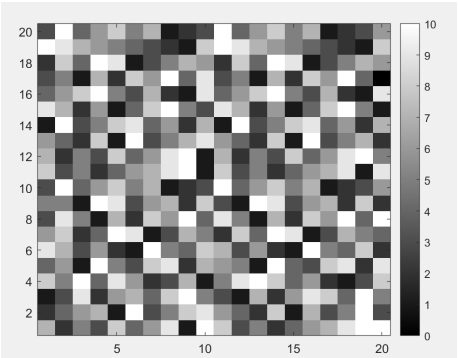

Figure 3. Roughness map

DOI: 10.54254/2755-2721/132/2024.20625

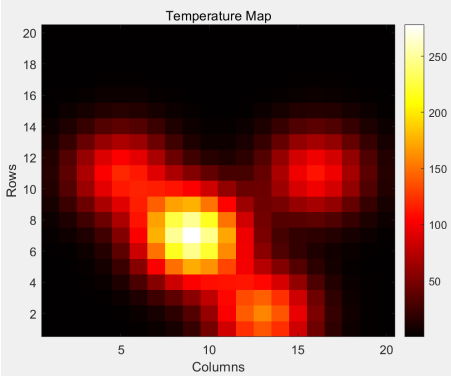

Figure 4. Temperature map

Finally, three distinct paths corresponding to different costmaps are revealed in Figures 5, 6, and 7, as the executing result of case 1, 2, and 3. By integrating the visual maps to observe the characteristics of the paths, it can be discerned that in case 1 (Figure 5), the robot considered only the height variance, resulting in a path with the minimum total time. However, this path directly traverses the high-temperature center, which is not the desired outcome in a fire scene; in case 2 (Figure 6), since the Genetic Algorithm (GA) was not employed to optimize the weights ω_1 and ω_2 in the cost value, the constructed costmap closely resembles the temperature map, largely neglecting the impact of height variance. Consequently, the path planned by the robot completely circumvents the high-temperature area, leading to a significantly increased time along the entire path, which is also not the desired outcome; in case 3 (Figure 7), with the optimization by the GA, it is evident that the constructed costmap effectively integrates the influence of height variance and temperature on the cost value. The planned path avoids the high-temperature areas and crosses the obstacles with low height values at the same time. This not only significantly reduces the cumulative temperature exposure along the entire path for the robot but also substantially shortens the time required to complete the total path.

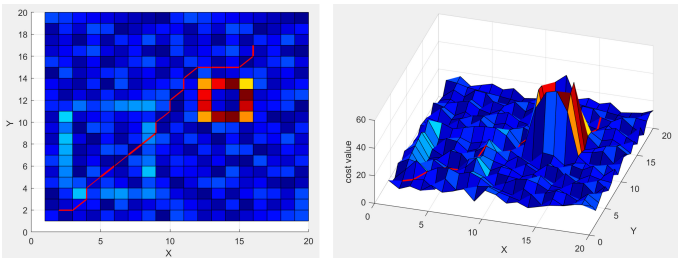

Figure 5. Path on costmap Case 1

By examining a table (Table 1), the variance in the final value (hereinafter denoted as 'fval') for the robot upon completion of the paths corresponding to the three distinct cases becomes more apparent. It is evident that the fval for case 3 is markedly less than that of case 2, and the fval for case 2 is also less than that of case 1. This observation directly substantiates the viability and rationality of the approach we have proposed within the context of a simplified fire environment scenario. Specifically, it demonstrates the capability of the method to enable the robot to circumvent the high-temperature central zone while concurrently reducing the overall time required to traverse the designated path to the greatest extent possible.

DOI: 10.54254/2755-2721/132/2024.20625

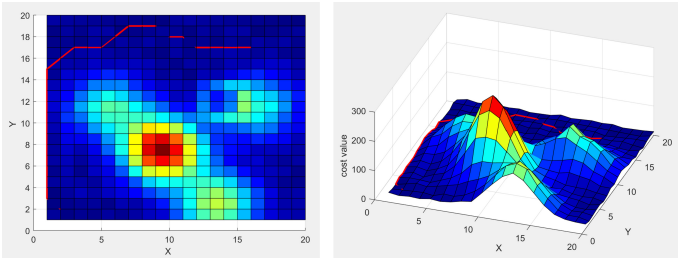

Figure 6. Path on costmap Case 2

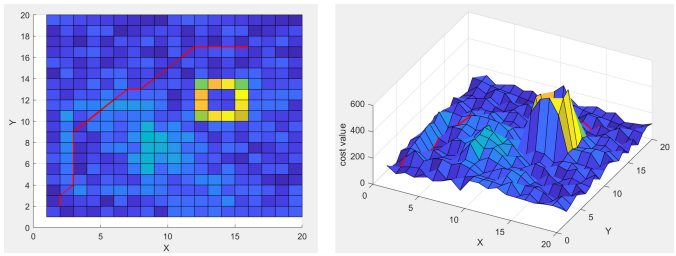

Figure 7. Path on costmap Case 3

Table 1. Final value of Case1,2,3

	Case 1	Case 2	Case 3
fval	2306.33	2281.98	2042.69

4. CONCLUSION

This project raises a method to construct the cost map based ib a grid map incorporating height, roughness, and fire sources, and optimized it using a Genetic Algorithm (GA). By analyzing and calculating the variance in terrain height and the impact of temperature, we effectively planned an optimal path for the quadruped robot within a simulated fire environment [12]. Our projects verified the effectiveness of combining height and heat based cost map and GA algorithm successfully, specifically in fire environment. This lays a solid foundation for future research and practical applications.

5. DISCUSSION

The improving points of our paper are the lack of simulation and map information.To the first point, basic ROS simulation has been achieved, which is able to calculate the path based on given algorithm, follow given way points and record the time from the start to the end [13]. The full simulation can show in the following paper. To the second point, more featured maps are being built to improve the effectiveness of the GA algorithm [14].

To deply our algorithm on the robot, we use simulink and ROS to simulate the running situation and test our algorithm efficiency on different map, as shown in figure 8, 9.

However, we get in trouble with combing ROS and GA algorithm in simulink. Therefore, we decide to rewrite our program in python code, so that we can do GA based on ROS-simulation time.

Acknowledgment

Yingwei Song and Qihong Duan contributed equally to this work and should be considered co-first authors. We would like to express our sincere gratitude to Neoschool for bringing together individuals

DOI: 10.54254/2755-2721/132/2024.20625

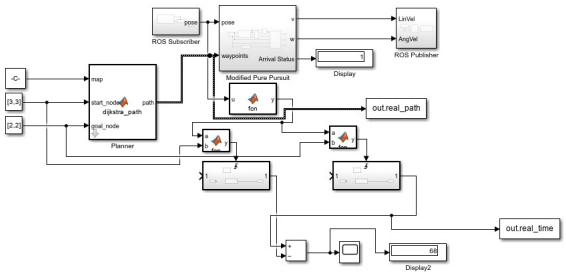

Figure 8. Simulation in Simulink

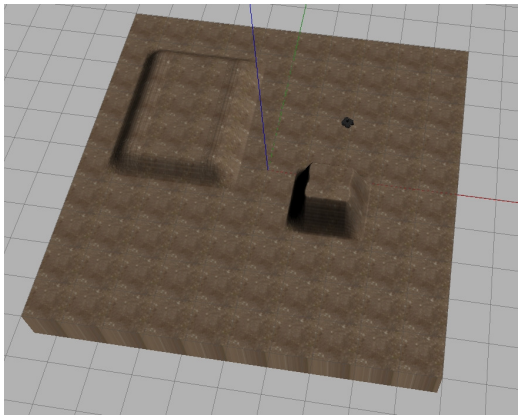

Figure 9. Simulation in ROS

from all corners of the world. Special thanks go to Professor Quan and the Teaching Assistants for their tireless guidance and instruction.

References

[1] G. M. Kruijff, et al.: "Rescue Robots at Earthquake-Hit Mirandola, Italy: A field report," Proc. of the 10th IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR2012), pp.1–8, 2012.

[2] Bogue, R. (2021), "The role of robots in firefighting", Industrial Robot, Vol. 48 No. 2, pp. 174-178. https://doi.org/10.1108/IR-10-2020-0222

[3] Ruinan Chen, Jie Hu et al. "An RRT-Dijkstra-Based Path Planning Strategy for Autonomous Vehicles." Applied Sciences (2022). Ruinan Chen; Jie Hu; Wencai Xu.

[4] DongKai Fan and Ping Shi. "Improvement of Dijkstra's algorithm and its application in route planning." 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (2010). 1901-1904.. DongKai Fan; Ping Shi.

[5] Alican Bozyigit, Gazihan Alankus et al. "Public transport route planning: Modified dijkstra's algorithm." 2017 International Conference on Computer Science and Engineering (UBMK) (2017). 502-505.. Alican Bozyigit; Gazihan Alankus; E. Nasiboğlu.

[6] W. Dougherty and R. D. Blanton. "Using regression analysis for GA-based ATPG parameter optimization." Proceedings International Conference on Computer Design. VLSI in Computers and Processors (Cat. No.98CB36273) (1998). 516-521.. W. Dougherty; R. D. Blanton.

[7] Wenrui Zhao and Borui Wu. "Improved GA and Its Application in Performance Optimization of Electronic Components." Advances in Multimedia (2022).. Wenrui Zhao; Borui Wu.

[8] R. Ngamtawee and P. Wardkein. "Simplified Genetic Algorithm: Simplify and Improve RGA for Parameter Optimizations." (2014). 55-64.. R. Ngamtawee; P. Wardkein.

[9] V. Cicirello and Stephen F. Smith. "Modeling GA Performance for Control Parameter Optimization." Annual Conference on Genetic and Evolutionary Computation (2000).. V. Cicirello; Stephen F. Smith.

[10] Lim Wei Jer, Gerald Lee Jun Xiong et al. "GA-based Optimization for Circuit Design Assistance." 2012 Third

DOI: 10.54254/2755-2721/132/2024.20625

International Conference on Intelligent Systems Modelling and Simulation (2012). 732-736.. Lim Wei Jer; Gerald Lee Jun Xiong; S. Neoh; Arjuna Marzuki.

[11] N. Mohan and N. Kasabov. “Transductive modeling with GA parameter optimization.” Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. (2005). 839-844 vol. 2.. N. Mohan; N. Kasabov.

[12] Nathan P. Koenig, A. Howard. “Design and use paradigms for Gazebo, an open-source multi-robot simulator.” IEEE/RJS International Conference on Intelligent Robots and Systems (2004).. Nathan P. Koenig; A. Howard.

[13] David Whitney, Eric Rosen et al. “ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots.” IEEE/RJS International Conference on Intelligent Robots and Systems (2018). 1-9.. David Whitney; Eric Rosen; D. Ullman; Elizabeth Phillips; Stefanie Tellex.

[14] Brian P. Gerkey, R. Vaughan et al. “The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems.” (2003).. Brian P. Gerkey; R. Vaughan; A. Howard.