Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230732

Review on common algorithms in path panning and improvements

Xinyi Li

Department of Physics and Astronomy, University College London, Gower Street, London, England, WC1E 6BT

zcaplio@ucl.ac.uk

Abstract: Based on the cooperation of artificial intelligence (AI) and unmanned driving technology, finding the best path from the starting node to the target node in the shortest time is a research hotspot. The required path planning algorithms can therefore be classified according to their different approaches to solving the problem. This paper focuses on Dijkstra's algorithm, A* algorithm, Ant colony optimization, and genetic algorithm. It also discusses the present problems of these algorithms and some improvements made by researchers focusing on automated guided vehicle path planning.

Keywords: AVG, Path planning, A* algorithm, Ant colony optimisation, Genetic algorithm.

1. Introduction

We live in an era where technology and information permeate every aspect of people's lives. In the field of artificial intelligence (AI), one of the most widely used and researched areas is automated guided vehicles, the core challenge of which is path planning. Path planning has long been established through global navigation systems and smart warehouses, and will be the basis for future driverless cars. Different algorithms used in path planning lead to large differences in efficiency. Therefore, the algorithm directly affects the ability of automated guided vehicles, while finding the best path from the starting node to the desired node in the shortest time becomes the current research focus. Path planning algorithms can be classified according to their different approaches to the problem. Some well-known graph-based algorithms, including Dijkstra's algorithm and A*, and others, including simulated annealing, ant colony optimization, genetic algorithm, and particle swarm optimization, are often classified as nature-inspired algorithms.

This paper will be reviewing four of the common global path planning algorithm with their improvement published by researchers in recent years. It is no surprise that the algorithm merging ideas from several different traditional algorithms outperforms other kinds of improvement. With the increasing complications of infield practice, new algorithms should be focusing on combining existent algorithms to fill individual disadvantages rather than developing completely new approaches.

2. Graph based algorithms

2.1. Dijkstra's algorithm

© 2023 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/6/20230732

Dijkstra's algorithm was first brought up by Dijkstra in 1956, and implemented by an adjacency matrix to calculate the shortest path between nodes in weighted graphs. The method using adjacency matrix has to go through all the nodes, giving the time complexity of $O(n^2)$. The square relation makes the cost of time grow enormously when the number of nodes exceeds a certain amount. Luckily, computer scientists have invented many abstract data types that can be used to improve known algorithms.

A common way to improve Dijkstra's algorithm is to use binary heap to replace the adjacency matrix. Liang et al [1] examined the efficiency of Dijkstra's algorithm using different data types in Pentinum double core CPU and analysed time and space complexity. Test data was plotted in Figure 1, apparently, the time complexity improvement as in percentage goes down as the number of nodes increases, but still keeps a very impressive amount. By the time number of nodes becomes 1000000 and the number of vertices gets to 500000, the time complexity compare to using adjacency matrix still increased by 76.28%.

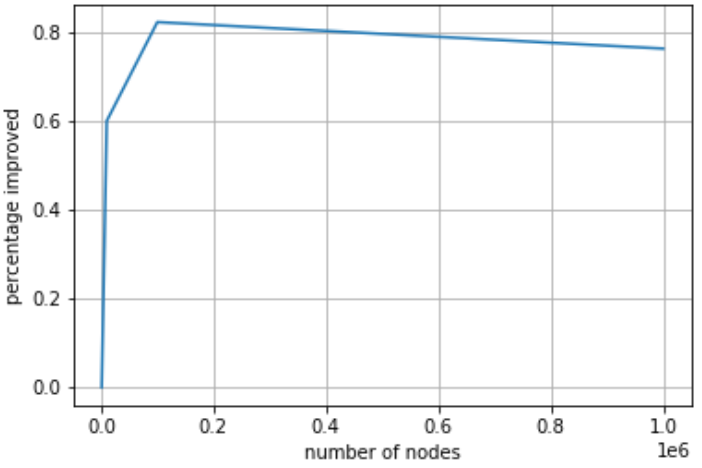

Figure 1. Test data.

Other than using different data types to improve the performance of the algorithm, Dijkstra's can also be altered into a heuristic algorithm. Liang[2] focuses on the path planning problem of automated guided vehicles(AGV) in an unknown environment, introducing estimation and heuristic cost so that heuristic cost guides the algorithm into the target state. The estimation function $F(n) = G(n) + H(n)$ is the estimation cost from the initial state n0 to the target state via n states. With G(n) being the cost of actual path calculated using the accumulated cost from the starting node, and H(n) being the cost of best path calculated using Euclidean distance between two points. Comparing two algorithms in Matlab simulation, the planned path is reduced by 22%, and the responding time is reduced by 21.6%. By analysing the graph planned by the algorithm, it is found that the algorithm does not go through all the nodes like traditional Dijkstra's algorithm does, since the improved algorithm uses the heuristic cost to guide the algorithm to target status.

2.2. A algorithm*

A* algorithm is an intelligent heuristic algorithm developed based on Dijkstra's algorithm. In addition to the estimated cost of target node, it guides the search direction toward the target. Compared with other graph based algorithms, it requires less calculation, gives better paths planned and has many other advantages. But the heuristic function takes few parameters to calculate, leaving many redundant grids when used in a grid environment.

Zhang et al [3] introduced a new method of targeted expansion to prioritise specific adjacent nodes when adding to the open-list. If at least one of the adjacent nodes in the target direction is viable, then expand only the nodes in that direction. Otherwise if not viable, then expand in four directions. Also

DOI: 10.54254/2755-2721/6/20230732

improved the algorithm by multiplying the estimated cost function by an estimated influence coefficient to improve the evaluation function in the traditional A* algorithm. The estimation cost function is then altered with additional turning costs. When the algorithm is used in path planning for AGV, taking turns forces the vehicle to slow down which means the shortest path might not be the best part. Therefore, reducing the number of turns can effectively increase the efficiency of the path. The improved algorithm is then tested to prove reducing the number of expansion nodes during searching, and yield a smoother resulting path.

Qi et al [4] also uses targeted expansion to improve the traditional A* algorithm, in addition to target visibility judgement, a new heuristic cost evaluation, and bringing the idea of simulated annealing in the selection strategy. Target visibility judgement serves the function of breaking the heuristic process when the previous node and target node have no visible obstacle in the line connected. And the new heuristic cost is

$$f(n) = g(n) + D \times (h(n) + h(p) + h(p^2) + h(p^3) + \ldots + h(p^k)) \quad (1)$$

instead of traditional

$$f(n) = g(n) + h(n) \quad (2)$$

Where D is the heuristic cost coefficient, p is the parent node of node n, and p^k is the k-th generation parent node of node n. Adding the estimated cost of k-th generation parents reduces the probability of local optimum. At the same time, the heuristic function is taken as the energy parameter in simulated annealing. Even if the heuristic function increases, the path will not be discarded immediately. The probability of accepting the path will be calculated by the principle of Metropolis, further reducing the probability of local optimum. The improved A* algorithm is then tested against the traditional one, giving a running time of 67.06% less under the circumstances that the fluctuation range of optimized length is ±0. 6%. And the number of grids involved in the calculation decreased by 73.53%.

Chen et al [5] further minimize the difference between the estimated cost and actual cost by changing the distance used to calculate the heuristic function into an arc. So that the arc gives a value that's between the value calculated using Euclidean distance and Manhattan distance. This improved heuristic function effectively increased the efficiency of initial trajectory at the beginning of searching. Moreover, the Bessel function is used to replace the original polynomial optimization. Facilitate by the convex hull of B-spline, a new set of constraining calculations is designed to improve the trajectory. When tested against the original algorithm, the improved algorithm has an efficiency increase of 22.5%. And the Bessel function specifically improved the optimization process by 10.5% less time.

3. Nature-inspired algorithm

3.1. Ant Colony Optimisation

Ant Colony Optimisation(ACO) is inspired by the phenomenon of ant foraging. During the foraging process, the ant leaves pheromone on the path travelled and enables the following ants to follow the subsequent path with greater pheromone. As the ants search for food and carry it back to the nest, the pheromone accumulates along the paths. Based on this positive feedback mechanism, the best path can be found with increasing iterations. The resulting path has great robustness. But due to the even distribution of pheromones at the early stage, it is more likely to stuck in local optimum; it also increases the number of iterations in the beginning, wasting unnecessary time.

To ameliorate this shortcoming, Jiang et al[6] proposed an improved algorithm combing four different changes. First find the rectangular area formed by the starting point and target point, as this area is flavoured in the global search. Then increase the pheromone of this area, therefore decreasing the probability of blind search at the early stage. Another problem in the early stage of searching is that it generates a number of cross paths, making deadlocks inevitable. This can be improved by adding obstacle parameters to reduce the risk of deadlocks during iterations. At the same time, the pheromone updating protocol is changed. After each iteration, only update and amplify the pheromone on the path

DOI: 10.54254/2755-2721/6/20230732

where ants successfully reach the target point. Reduce the pheromone left by the deficient ants and discard the ants that go into deadlocks. A second path planning strategy is also proposed in this paper. It is carried out to smoothen the path found by the first run. Connect the points where turns are taken and no obstacles are set in between, reducing the number of turns taken and therefore saving more time. The simulation result shows great improvement in global searching efficiency, also giving faster convergence. And when combining four strategies, the new algorithm yield 85% less iterations compare with the original ACO.

3.2. Genetic Algorithm

Being also a famous nature-inspired algorithm, genetic algorithm as its name utilises the processes of gene selection, crossover, and mutation to find the best solution. When used in path panning in grid environments, each grid is given a code. All possible paths are stored in sequences. And the length of sequence is calculated and compared, with the shortest sequence being the best path. A problem with this algorithm is local optimum just as many other algorithms.

To increase the global search ability, Sun et al[7] put forward a modified self-adaptive genetic algorithm. When the path has the least value of objective function, introduce the idea of simulated annealing into the selection process to ensure population diversity. Simulated annealing using the principle of Metropolis determines the probability of transfer from the current solution to the new solution, therefore improving global search ability. When tested, the modified algorithm reduced the search time by 25.5% compared with the traditional genetic algorithm on the map with 25 obstacles. With 66 obstacles on the map, the search time is reduced by 34.8%. With 374 obstacles on the map, the search time is reduced by 22.0%. The probability of global optimum is also above 90% for all three tests.

Wang et al[8] bring a new improvement idea, to add both length of the path and number of turning point as parameters in the fitness evaluation.

The new fitness function becomes

$$\mathrm{E_{fitness}} = \frac{\mathrm{r}}{\mathrm{mL+nT}} \tag{3}$$

where r is a coefficient based on distribution and number of the grid, L is the length of the path using $\mathrm{L} = \sum_{\mathrm{i=0}}^{\mathrm{N-1}} \sqrt{(\mathrm{x_{i+1}} - \mathrm{x_i})^2 + (\mathrm{y_{i+1}} - \mathrm{y_i})^2}$; and T is the number of turning points. At the same time signing m and n different values to give weight to L and T. Under complex environment, value of m should be increased.And when there are less turning points, increase the value of n. To ensure excellent individual sequences are reserved in the later stage, introduce Sigmoid function to modify the probability of crossover and mutation.

$$P_c = P_{c\_min} + (P_{c\_max - P_{c\_min}}) \frac{1}{1+e^{-(f_{iave} - f_i)}} \tag{4}$$

$$P_m = P_{m\_min} + (P_{m\_max - P_{m\_min}}) \frac{1}{1+e^{-(f_{iave} - f_i)}} \tag{5}$$

When tested in the simulation environment of 20×20 grid, the improved algorithm uses only 12 iterations to achieve the same fitness value the conventional algorithm would take 40 iterations to reach. Proving great improvement in convergence efficiency. And the result also shows that the maximum fitness value has increased by 28.93% in average.

The Genetic algorithm does not scale well with time complexity, and the quantitative method to analyse genetic algorithm is not yet complete[9]. So the upcoming researches should focus on combing genetic algorithms with other methods to achieve the greatest efficiency.

4. Conclusion

As the research on path planning proceeds, it is no wonder combining different algorithms makes the best choice when improving conventional algorithms. With evolving hardware capabilities, evolutionary algorithms will play a core role in the area[10]. When implementing algorithms in complex

DOI: 10.54254/2755-2721/6/20230732

environments, knowledge from other fields should be also applied to utilise advantages for greater results. Another point to make is that although researchers focus on 2-d path planning predominate the field, new studies on three-dimensional path finding shouldn't be neglected as they may inspire new ideas that benefit both 2-d and 3-d path planning.

References

[1] Liang B., Yang XM., (2020). Study on a Path Planning Method Based on the Improved Dijkstra Algorithm, Informatization Research, 46(2), pp. 13-16, http://www.cqvip.com/qk/91231a/202002/7102460649.html.

[2] Liang Y., (2020). Improved Dijkstra algorithm on Automated Guided Vehicle path planning, Science and Technology & Innovation, 24, pp. 159-161, https://www.cnki.com.cn/Article/CJFDTotal-KJYX202024061.htm.

[3] Zhang JG, Zhang F., Chen LG, Jiang Q, Zhu W. (2022). Path planning of automatic guided vehicle based on improved A* algorithm, Foreign Electronic Measurement Technology, 41(1),pp.123-128,https://ie.cnki.net/kmobile/Journal/detail/XSDN_XNYJ/GWCL202201021.

[4] Qi XX., Huang JJ., Cao JA., (2020). Path planning for unmanned vehicle based on improved A* algorithm, Journal of Computer Applications, 40(7), pp. 2021-2027, http://www.joca.cn/CN/10.11772/j.issn.1001-9081.2019112016.

[5] Chen TF., Deng ZL., Zhang ZH., (2021). Path Planning of Mobile Robot Based on Improved A* algorithm, China Satellite Navigation Conference. Jiangxi, Nanchang, China, 21 May, https://cpfd.cnki.com.cn/Article/CPFDTOTAL-WXDH202105009004.htm .

[6] Jiang M., Wang F., Ge Y., Sun LL., (2019). Research on path planning of mobile robot based on improved ant colony algorithm. Chinese Journal of Scientific Instrument, 40(2), pp. 113-121, https://www.cnki.com.cn/Article/CJFDTotal-YQXB201902013.htm.

[7] Sun B., Jiang P., Zhou GR., Lu YT., (2019). Application of Improved Genetic Algorithm in Path Planning of Mobile Robots. Computer Engineering and Applications, 55(17), pp. 162-168, https://www.cnki.com.cn/Article/CJFDTOTAL-JSGG201917026.htm.

[8] Wang H., Zhao XJ., Yuan Xiujiu. (2022). Robot Path Planning Based on Improved Adaptive Genetic Algorithm. Electronics Optics & Control, 29(5), pp. 72-76, https://www.cnki.com.cn/Article/CJFDTOTAL-DGKQ202205014.htm.

[9] Li Y., Yuan HY., Yu JQ., Zhang GW., Liu KP., (2019). A literature review of the Application of Genetic Algorithms in Optimization application. Shandong Industrial Technology, 12, pp. 242-243, http://www.cqvip.com/qk/70459x/201912/836871745048495749505049 49.html.

[10] Liang XH., Mu YH., Wu BH., Jiang Y., (2020). Summary of Path Planning Algorithms.Value Engineering, 3, pp. 295-299, http://www.cnki.com.cn/Article/CJFDTotal-JZGC202003117.htm.