Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230640

The characters and advantages of different algorithms in multi-robot path planning

Liwen Wang[1]

[1]Jinan Foreign Language School,Jinan,Shandong,250000,China

20148191@stu.sicau.edu.cn

Abstract. Multi-agent path finding (MAPF) issue has recently gained popularity, with the goal of planning for several agents who can all follow instructions at the same time without colliding. It is a common difficulty when attempting to identify pathways to several agents while ensuring that each agent reaches its destination without colliding and confusion. Real-world MAPF applications such as warehouse management, food, and beverage distribution, and aviation management are paid a number of attention from people in recent years. This article discusses centralized and decentralized techniques in MAPF challenges. Several traditional algorithms will be quoted and introduced through the help of some past papers and experiments. Finally, it was determined that decentralized strategies generally provide more powerful performance, whereas centralized strategies are generally capable of computing optimal solution sequences with minimal coupling and solving complex problems, ranging from relatively few robot problems with high coupling to many robot problems with low coupling.

Keywords: Machine learning, multi-agent path planning, centralized, decentralized.

1. Introduction

The Multi-Agent Path finding (MAPF) problem is now a trendy topic, with the objective of planning for several agents who can all follow directions at the same time without colliding.

MAPF has a range of contemporary applications related to it, including automated warehouses, autonomous vehicles, and robotics. Therefore, this problem has attracted the attention of various research groups and academics in recent years.

Path planning for multiple robots means finding a path for each robot in the same workspace and ensuring that there is no collision between robots and robots at every moment, and there is no collision between robots and the environment, which involves the coordination and collision of roads between robots [1] [2] [3]. Therefore, MAPF 's algorithm is particularly important.

Many different MAPF research papers consider and study the possibilities and adaptability of various algorithms to apply to this problem, and give complex, difficult to read, and difficult to adjust conclusions. This makes it difficult for beginners and even practitioners to navigate and understand the existing literature and try to apply these algorithms to their research. Therefore, in this article, the advantages, disadvantages, and uses of the specific two different types of algorithms in solving multi-agent path planning problems will be discussed and compared, so that readers can better understand these algorithms, improve the efficiency and speed of problem-solving in this field, and more simply, directly and efficiently select the appropriate algorithm. At present, there are various algorithms for solving multi-agent planning path problems, such as traditional multi-agent collision algorithms, speed

DOI: 10.54254/2755-2721/5/20230640

barrier methods, and mesh rules based on artificial potential field methods [4] [5] [6]. From the perspective of planners and planning time, the above methods can be roughly divided into centralized planning and decentralized planning, which have their advantages and disadvantages and are usually combined in practical applications [7].

The process that can be using centralized search algorithms, such as A* [8]. However, in the multi-agent problem, the number of state and branching factors grows exponentially in the number of moving units. So while the integrity and solution optimization of centralized A* searches are guaranteed, multi-agent path planning problems have little practical value, even for relatively small atlases and mobile units, which are difficult to solve [9].

By decomposing the initial problem into a series of searches through a decentralized approach, it is indeed possible to extend the larger problem. Traditional decentralized approaches, such as FAR and WHCA*, are problem-based decomposition, faster, and more scalable. However, these methods are incomplete and do not guarantee uptime or solution quality. They may not be able to tell in a reasonable amount of time whether they will be able to successfully find a solution for a given instance [10].

In this paper, the literature on different centralized and decentralized algorithms will be reviewed and browsed, the total time it takes for multiple robots to reach the target in real simulation, the complexity of calculations and parallelism, and the three aspects will be compared and try to come up with final answers - the advantages and disadvantages of these algorithms used in multi-robot path planning.

2. Problem formation

In this paper, ɑ agents is defined by G, s, and t is cited[11] and G, s, and t are defined for which:

A. $G = (V, E)$is a conceptive diagram that the vertices in it representing hypothetical sites that the agent might occupy. Each edge (n, n) shows that the agent can go from vertex to vertex without touching any other vertex.

B. Define the variation that records an agent returns to its original position as s.

C. Define the variation that records an agent to its goal position as t.

In decentralized, time is segmented into pieces, and each time fragment is termed a time step, as each agent has two states: waiting and acting. The wait action causes the agent to stay at the current time step, whereas the move operation leads the agent's position in change.

It may alternatively be considered as decoupling for the centralized kind, in which the robot is treated as a single unified robot. Each robot's course of action will be planned independently under this technique, and the planner will then integrate and plan a collision-free trajectory for all robots, simultaneously or in priority while avoiding the path position of the planned robot, which is regarded as an impediment to movement.

3. Methods

Algorithm 1 [11]	Search Algorithm A*
1	Make s “open” and calculate f(s)
2	Select the open node n whose value of f is the smallest. Resolve ties arbitrarily, but always in favor of any node n ∈ T
3	If n ∈ T, mark n “closed” and terminate the algorithm
4	Otherwise, mark n closed and apply the successor operator T to n. Calculate f for each successor of n and mark as open each successor not already marked closed. Remark as open any closed node $n_i$which is a successor of n and for which $f(n_i)$ is smaller that it was when n_i was marked closed. Go to Step 2.

The essential principle of search algorithm A*.

3.1. Decentralized

DOI: 10.54254/2755-2721/5/20230640

The A* algorithm heuristic search[12] is a popular method for solving the shortest path issue in decentralized schemes.

Heuristic approaches are frequently employed in the issue area depicted by the diagram[11] (for example, multi-agent path planning) and are primarily utilized to increase the computing efficiency of such search problem solutions.

The essential principle of the A* algorithm may be viewed in algorithm 1 for the reader's convenience. [12] demonstrates that the A* search algorithm is the optimum option.

The writers used the A* algorithm in air systems, as a way to plan multi-agent actions. In this research, three goals were set:

(1) The progress of the path for an agent is desired.

(2) From the current step to the next time step, the location is reachable.

(3) The progress of the paths for the other agents is desirable.

During the A* calculation, other agents are handled as barriers during the A* computation, yielding three degrees of freedom (x,y,z) as the PID controller's set point. The emulator works at a rate of one hertz (Hz). The movements of the agent define the search for the experiment, and its present state may be deduced based on model feedback paired with the observed state of the agent, which is an essential conclusion that can be reached from this implementation of A*. By using a model. In order to identify the feasible actions in this manner, real agents are matched appropriately and their movements are restricted (agents can only eliminate these motions at specified times).

Table 1. Maximum performance of aerial agents [12].

	Velocity Max (m/s)	Velocity 70% (m/s)	Acceleration Time (s)	Distance
Forward	1.0	0.7	2.0	1.4
Backward	-1.0	-0.7	3.0	2.1
Right Lateral	0.5	0.35	5.5	1.9
Left Lateral	-0.4	-0.3	7.0	2.1
Up	0.6	0.4	2.0	0.8
Down	-0.3	-0.2	0.5	0.1

3.2. Centralized

Multi-robot motion issues are frequently seen in this light as collapsing into low-dimensional sub-problems with priority.

As a function, define the execution sequence as the order in which n robots split the robot:

$R_1 \cup \ldots \cup R_k = \{r_1, \ldots, r_n\}$ and $R_i \cap R_j = \emptyset$ for i ≠ j.

S is known as the solution sequence, if it is an equivalent class sequence that solves the coupling of collision-free pathways.

Assuming the CR planner's planner is a black box and complete, the feasibility of the composite robot R in a defined workspace may be assessed. As a result, the black box is applied to the instance, and the other robots r are for obstacles, with R serving as the beginning point. Choose whether or not to pick the destination location. Each viable path discovered adds its potential limitations to the DNF statement.

Finally, the constraint produced from the constraint expression E is kept, and E is represented as an analytical norm equal to the analytical binding, i.e. as the conjunction J_i of the analyte $\mathrm{E} = \mathrm{J}_1 \cup \mathrm{J}_2 \cup \cdots$.

Figure G(J) represents each conjunction J, and it is worth noting that Figure G(J) includes a constraint diagram. Each robot is a node, and all nodes form a full constraint graph with a set of directed edges that limit the order of precedence of the robots. That is, on the constraint graph, there is always an edge that links to the node to constrain any atoms in J. To identify answers, all functioning robots must be monitored and governed as composite robots. If there are pathways in the diagram from r_i to r_j and r_j to r_i, the two nodes r_i and r_j are tightly related.Let GSCC (J) be the component graph of G(J), which comprises each strongly connected component of G(J) as well as all directed edges from node R to node

DOI: 10.54254/2755-2721/5/20230640

R. Each node in GSCC(J) represents a composite robot that is part of a network of tightly coupled components.

See Algorithm2 for detailed information.

Algorithm 2[10]	P(R)	
Let	$s, g \in RM(R)$ be the start and goal configuration of R	
	for	all vertices x in $RM(R)$ **do**
		$P(x) \leftarrow \perp$
	$P(s) \leftarrow \wedge_{r_i, r_j \in R}\, r_i \sim r_j$	
	$Q \leftarrow$ {s}	
While not	(priority) queue Q is empty **do**	
	Pop front vertex x from Q	
If not		$P(s) \rightarrow P(g)$ **then**
	for all	edges (x, x') in $RM(R)$ **do**
		$C \leftarrow P(x)$
	for all	robot $r_i \notin R$ **do**
	if	robot r_i configured at s_i “blocks” edge (x, x') **then**
		$C \leftarrow \wedge\ r_i \prec R$
	if	robot r_i configured at g_i “blocks” edge (x, x') **then**
		$C \leftarrow \wedge\ R \prec r_i$
	if not	$C \rightarrow P(x)$ **then**
		$P(x') \leftarrow P(x) \wedge$ C
	if	$x' \notin Q$ **then**
		$Q \leftarrow Q \cup$ {x’}
	$P(g)$	

4. Results

According to the preceding discussion, the centralized strategy has certain limitations in the plan from the beginning to the goal compared to the decentralized strategy, for example, it simply calculates the complete sequence of the robot plan, while the more coupling required for some problems is not considered. For example, if numerous robots block the middle of a long hallway at the same time and another robot tries to pass through, the robots will all be linked, even if the robots that are not directly coupled can go elsewhere and be moved by another robot.

For centralized, the x, y, and z motion, real-world agents are bench-marked. Send the customer service to a waypoint 5m away, and then calculate the maximum speed and acceleration in each direction, as shown in Table 1.

The A* Path Planner sends time-synchronized way points in a preset order of unique alphabetic order way points, while additional way points are delivered once all cars have reached the same vertex number. If one vehicle deviates from the target or does not move efficiently to the vertex, the path planner also slows down the trajectory of other vehicles.

The discreteness within each time step in decentralized is an abstraction of the time continuum, in reality, an abstraction that sometimes leads to an incomplete order of precedence in the application of MAFF. In principle, if the agent does not need to wait, this problem does not need to be treated seriously: after all, the duration of the move action is determined by the time it takes to cross the associated edge. When an agent is in a space where time has not been discretized and has to wait, the waiting action on each vertex has an infinite number of alternatives in theory.

In the decentralized experiment, Three situations were tested and applied to their reduction in this investigation.

Scenario (a) robots are scarce, yet they have a significant potential for coupling. In comparison to the (c) robot theory, but with a lesser degree of linkage. There are 16 robots (a) in the circumstances with very restricted maneuver space, Traditional multi-robot planners struggle to tackle this challenge

DOI: 10.54254/2755-2721/5/20230640

since the robots are the robots of the entire room and each robot must switch positions with other robots.

In scenario (b), To quantitatively evaluate how the decentralized algorithm works in these diverse settings, the quantity of robots n and the degree of connectedness are adjusted.

In the scene, randomly created settings featuring up to 65 robots that give entire composite constructions with 130-dimensional modeling space are tested (c). The centralized algorithm will yield a sequence of solutions involving only one robot, but the decentralized approach will discover a solution by designing 65 individual robots in a 2D configuration space. Despite the enormous number of robots, the coupling coefficient in this scenario is quite low. This example took 73 seconds to solve.

The experiment in this scenario shows how the decentralized algorithm operates for varying numbers of robots and coupling degrees. When the coupling degree is low, solving more tasks for the robot in a realistic operating period is very well proven. Because the runtime rises with the amount of vehicles and the polynomial, trials may involve up to 30 robots. However, when the number of robots exceeds or equals 30, a very modest exponential component (due to the combination) begins to dominate. Table 2 displays the results.

Table 2. results [10].

α=a		α=2a		α=3a		α=4a	
a	time (in seconds)	a	time (in seconds)	a	time (in seconds)	a	time (in seconds)
5	1.39	20	0.30	27	18.8	20	41.7
6	44.0	22	0.69	30	38.8	24	167
7	n/a	24	2.15	33	75.6	28	542
		26	7.69	36	146	32	1254
		28	42.4	39	287	36	3356
		30	261	42	672	40	7244

The results in real-world agents experiment.

5. Conclusion

In this paper, a simple introduction of the centralized and decentralized algorithm are given. In the part of decentralized multi-agent path planning, through bringing the concepts of the A* Algorithm Heuristic search and referring to the experiment. The conclusion that decentralized strategies often provide more powerful performance because in this field, stable policy gradients sometimes seem more important than stable value functions is given. While the other one, the centralized algorithm, in this part, through applying them into real life, it can be found that such algorithms are often able to compute the least coupled optimal solution sequences and solve complex problems, from relatively few robot problems and high coupling to many robot problems and low degrees of coupling.

References

[1] M. Gerke and H. Hoyer 1997 Planning of optimal paths for autonomous agents moving in homogeneous environments *The 8th International Conference on Advanced Robotics*-347.

[2] J. Xiao and Z. Michalewicz and L. Zhang, K. Trojanowski 1997 Adaptive evolutionary planner/navigator for mobile robots *IEEE Trans. Evolut. Comput*, 1 (1).

[3] Z. Bien and J. 1992 Lee A minimum-time trajectory planning method for two robots *IEEE Trans. Rob. Autom*, 8 (3) -443.

[4] Wagner G. and Choset H. 2015 Subdimensional expansion for multirobot path planning. Artificial Intelligence 219, 1–24

[5] Wang K H C and Botea 1870-1875 A tractable multi-agent path planning on grid maps[C] IJCAI. 2009, 9

[6] Wang K H C and Botea 2011 A MAPP: a scalable multi-agent path planning algorithm with

tractability and completeness guarantees[J]. *Journal of Artificial Intelligence Research*, 2011, 42- 55.

[7] Stern R., Sturevant N.R., Felner A., Keonig S., Ma H.,Walker T.T., Li J. Atzmon, D. Cohen, L. Kumar T.S. and Boyarski E. and Bartak R. 2019 Multi-agent pathfinding: Definitions, variants and benchmarks Symposium on Combinatorial Search(SoCS).

[8] Appi J M A 1966 A formal basis for the heuristic determination of minimum cost paths[R] *Tech.* Rep.

[9] Stern R Multi-agent path finding–an overview[J] *Artificial Intelligence*, **96-**115.

[10] Van Den Berg J and Snoeyink J and Lin M C 2009 Centralized path planning for multiple robots: Optimal decoupling into sequential plans[C] *Robotics: Science and systems* 2(2.5): 2.3.

[11] P. E. Hart, N. J. Nilsson and B. Raphael, 1968 A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics*, 4, 2, - 100.

[12] S. Cho, V. Mishra, Q. Tao, P. Vamell, M. King-Smith, A. Muni, and W. Smallwood and F. Zhang , 2017 Autopilot design for a class of miniature autonomous blimps, *IEEE Conf. Control Technology and Applications (CCTA) (Hawaii, USA, 2017)*,- 841.