A Systematic Literature Review of Multi-agent Pathfinding for Maze Research

Semuil Tjiharjadi[1,2], Sazalinsyah Razali[2], and Hamzah Asyrani Sulaiman[2]
[1] Computer System Dept., Faculty of Engineering, Maranatha Christian University, Bandung, Indonesia
[2] Centre for Robotics and Industrial Automation, Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia
Email: semuiltj@gmail.com, {sazalinsyah, asyrani}@utem.edu.my

***Abstract*—Multi-agent Pathfinding, also known as MAPF, is an Artificial Intelligence problem-solving. The aim is to direct each agent to find its path to reach its target, both individually and in groups. Of course, this path allows agents to move without colliding with each other. This MAPF application is implemented in many areas that require the movement of various agents, such as warehouse robots, autonomous cars, video games, traffic control, Unmanned Aerial Vehicles (UAV), Search and Rescue (SAR), many others. The use of multi-agent in exploring often assumes all areas to be explored are free of obstructions. However, the use of MAPF to achieve their goals often faces static barriers, and even other agents can also be considered dynamic barriers. Because it requires some constraints in the program, such as agents cannot collide with each other. The use of single-agent can find the shortest path through exploration. Still, multi-agent cooperation should shorten the time to find a target location, especially if there is more than one target. This paper explains the Systematic Literature Review (SLR) method to review research on various multi-agent pathfinding. The contribution of this paper is the analysis of multi-agent pathfinding and its potential application in solving maze problems based on an SLR.**

***Index Terms*—systematic review, multi-agent, pathfinding, maze**

I. Introduction

The maze is a travel puzzle with complex branching passages where the explorer must find a route. The mission of this travel puzzle is to find the fastest route to reach the destination. The objective can be a way out or a location in the maze to find. Since the route in the labyrinth is unknown, explorers must find a way to reach their destination. The destination location can be known or unknown, and different characteristics are needed in developing search algorithms for explorers. This maze problem is often contested, such as the micro mouse competition held every year in various countries. Participants develop an algorithm that can guide their robot to find a route in the maze in the fastest time. This maze algorithm application has been developed for various uses, such as find ing the shortest route for travel maps, developing warehouse applications, and even being used in the medical world.

This paper explains the SLR process on various multi-agent pathfinding researches used for maze exploration. Maze exploration algorithms have been studied since the second half of the nineteenth century and are closely related to Graph Theory. The maze exploration usually uses a single agent that roams the maze. However, as technology develops, Multi-agent Systems (MAS) have also begun to develop autonomous robot systems. The MAS used in finding a path to reach a destination is known as Multi-agent Pathfinding (MAPF). They must find the target without colliding with each other in the shortest time or the shortest distance.

Distributed Artificial Intelligence (DAI) technology has increased along with the need to solve complex computational problems. Based on its ability to complete tasks, Table I shows that the DAI algorithm can be grouped into three classifications: Parallel AI, Distributed Problem Solving (DPS), and Multi-agent System (MAS).

MAS has agents as autonomous entities. In DPS, agents make independent decisions based on their learning capability. They get the information based on their interaction with the environments and their collaborative information. Then agents can make the best calculation to achieve their tasks. Its high flexibility makes MAS suitable for solving problems [1]. Based on its fundamental capabilities and features, agents are entities in the environment and perceive any parameters and use them to make decisions. Fig. 1 shows the characters of a single agent as a part of MAS.

TABLE I. Three Classifications of Distributed Artificial Intelligence

Parallel AI	Distributed Problem Solving	Multi-Agent Systems
The use of parallelism tasks to increase the efficiency of the AI algorithm	Utilization of compute nodes that work together to share knowledge and resources to complete shared tasks.	MAS is more flexible due to autonomous agents who can learn, make independent decisions, and work collaboratively to complete tasks.

Based on many articles, agent characters can be mapped in Fig. 1, while Table II shows the seven important MAS features and their feature categorizations.

Manuscript received February 8, 2022; revised June 10, 2022.

doi: 10.12720/jait.13.4.358-367

There are two main objectives in using multi-agents in robots: the first is cooperation and coordination between robots, then the second is planning the trajectory of the robots. The robot uses its sensors to detect environmental conditions; It is programmed to function as a decision-maker, path planning, task management, and communication agent.

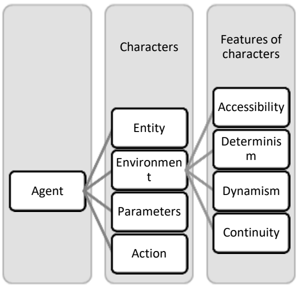

Figure 1. Agent's characters.

TABLE II. MAS FEATURES AND CATEGORIZATION [1]

Features	Categories	References
Leadership	Leader-follow Leaderless	[2] [3]
Decision function	Linear Non-linear	[4], [5] [6], [7]
Heterogeneity	Heterogeneous Homogenise	[8], [9] [6]
Agreement parameters	First-order Second-order High order	[7] [5] [5]
Delay	With delay No delay	[10], [11] [6]
Topology	Static Dynamic	[9] [11]
Data transmission	Triggered by time Triggered by event	[12]
Mobility	Static agents Mobile agents	[13]

One of MAS development is Multi-agent Pathfinding (MAPF), as a fundamental computation. It has been developed in many implementations such as computer games, warehousing, robotics, traffic optimization, vehicle automation, and many others. MAPF can be solved optimally using These algorithms:

1) Reduction-based solvers simplify MAPF into well-known algorithms, such as SAT [14], linear integer programming, and programming of response sets.
2) Search-based solvers, like Flood Fill algorithm [15], Dijkstra's algorithm [16], A* algorithm [17], Pledge algorithm [18], Genetic algorithm [19], Trees and Ant colony optimization [20].

A search-based solvers algorithm for pathfinding intends to reduce costs from start to target. A* algorithm is often used in Artificial Intelligence that follows the path because it can reduce the heuristic cost as low as possible. Apart from A *, there is also an ant colony algorithm that mimics the behavior of the ants. The algorithm initially randomly traverses nodes while updating the cost of each node. Then we can get a route that has the optimal time. The drawback is that it requires many iterations. There is also a Flood Fill algorithm often used for maze exploration. Each cell in Flood Fill has a value in its distance from the target. Of course, this requires constant updates. [21]

Table III shows the advantages and limitations of pathfinding maze algorithms. These algorithms are implemented as a single agent in the static maze environment. A* and Flood Fill are the most often used to implement a maze-solving robot.

TABLE III. COMPARISON OF PATHFINDING MAZE ALGORITHM

Algorithm	Advantages	Limitation
A*	Cost-efficient	Requires a large amount of memory
Ant colony algorithm	Optimal time	More number of iterations
Genetic algorithm	Multiple solutions (beneficial when input is enormous)	Inconstant optimization response times
Depth First Search	Simple to implement	Large computing power
Breadth-First Search	When memory is not a problem	Large space complexity
Flood Fill	Optimal	High-cost updates

Overall, the structure of the paper consists of an introduction as the first section. The second section explains the review process in a systematic review. The third section presents the findings, then conclusions and study of possible works in the fourth section.

II. REVIEW PROCESS

The purpose of this paper is designed to find out studies that are relevant to the use of multi-agent pathfinding in the maze using SLR. This section involves search terms and selecting literature, which is required for the next stage of the search process. Understanding how some researchers conducted previous research and their contribution is necessary. It will enhance research in minimizing the waste of effort and duration. This Literature Review will review previous research to summarize its publication, but it is not a complete review of all studies.

The preliminary findings from the systematic literature review undertaken to analyze the domain described are carried out in this section. It aims to explore in this regard:

- As an intelligent multi-agent scheme, the essential requirements of MAPF.
- The methods, methodologies, and algorithms used to construct the MAPF field and various integrated approaches to reasoning.
- Strength and weaknesses of the approaches proposed.
- Also, all issues were found, including future trends and developments.

Systematic Literature Review (SLR) is a systematic mapping study that aims to gather and classify based on specific criteria (Petersen, Vakkalanka, and Kuzniarz, 2015 [22]). This study uses Kitchenham's concept of the systematic literature review [23]. The systematic review process is shown in Fig. 2.

It is possible to collect, evaluate, and interpret various studies relevant to the research through the SLR. So that can be presented an appropriate evaluation of different research topics using a reliable methodology, with the following reasons: The need to determine and summarize the current outcome of technology research and recognize technology gaps that can produce future research topics. In Fig. 2, the review process has three levels: planning, implementing, and reporting the study. The remainder of this section addresses our chosen strategy and all its main components in detail.

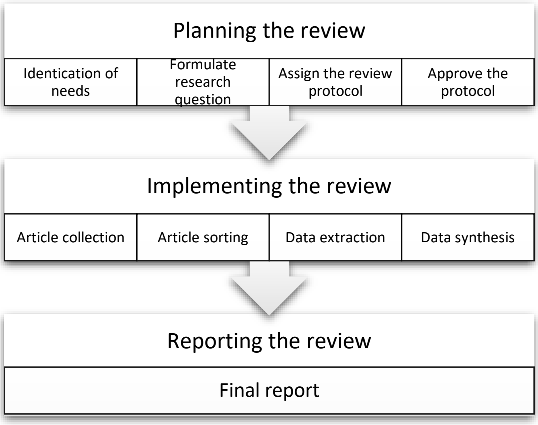

Figure 2. The systematic review process [24].

TABLE IV. RESEARCH QUESTION ON LITERATURE REVIEW

ID	Research Questions
RQ1	What are the multi-agent pathfinding application areas?
RQ2	What are the central purposes of the MAPF system?
RQ3	What is the kind of environment for a multi-agent-based maze pathfinding system?
RQ4	What techniques are appropriate in the navigation environment for pathfinding?
RQ5	What are the algorithms used to develop intelligent maze pathfinding? Furthermore, which strategies are appropriate to use to give agents the ability to make the decision?
RQ6	Which of the proposed system of multi-agent pathfinding is a complex maze environment being studied or incorporated?
RQ7	What are the advantages, weaknesses, and assumptions of the reasoning frameworks proposed for MAPF?
RQ8	In the literature, what are the main challenges and potential future research?

Fig. 3 shows a basic mini-map of a systematic literature review. This systematic literature review's primary purpose is to identify multi-agent pathfinding in a dynamic maze, model, and dataset methods.

A. *Planning the Review*

This process describes and discusses current systematic MAPF literature reviews and ensures the need for a systematic review. The outcome of these pre-review activities specifies the research questions.

1) *Formulating research questions*

Using the Goal-Question-Metrics approach, the purpose of this analysis is as follows:

- Goal: Search, Study, and Compare
- Topic: Consideration Systems
- Object: a MAPF in the maze
- Point of view: The perspective of the researcher.

Eight Research Questions (RQ) were formulated based on the objectives to identify the problems of MAPF in the maze and seek various contributions from the literature. These RQs are the standard for sorting and evaluating all reference articles. Table IV lists literature review research questions sorted from the most basic.

2) *Specifies the protocol for analysis*

Kitchenham [23] outlined the need to use a review protocol to determine the method used for systematic reviews. SLR activity is determined through a review protocol such as steps to select the main study, applied criteria, assessing the quality of the study, a strategy to be used for data collection and dissemination.

B. *Implementing the Review*

This phase has several phases: article collection, sorting, extraction, and synthesis.

1) *Article collection*

The collection of various articles has been carried out from October 2019 to January 2021. Using repeated search techniques, the SLR aims to find as much of the primary research as possible that is relevant to the research problem. Therefore, a comprehensive search for preliminary studies is carried out, so it is necessary to determine a strategy in search and its application. The article submission process is divided into the following two main sub-activities:

a. Search queries use MAPF, pathfinding algorithm, multi-agent system, maze solver, search, and rescue.
b. Scopus, SpringerLink, IEEExplore, ACMDigital Library, Google Scholar, and Semantic Scholar conduct searches using the following electronic databases.

A search from these electronic databases returned 4983 article titles. However, the results obtained may occur that the articles have the exact title duplication. Those are why the following selections were made.

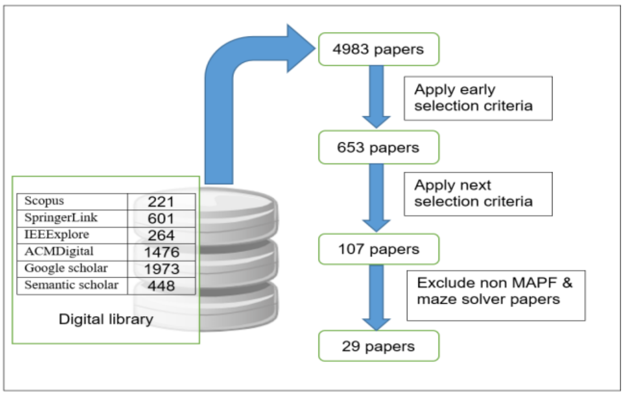

Figure 3. The process of collecting and selecting articles.

2) *Article sorting*

Fig. 3 shows the process of article selection. In order to sort suitable studies select appropriate studies for review, two groups of inclusion and exclusion criteria were determined:

a. Early Selection: So many articles are collected for sorting, then all of them are sorted by abstract, title, and keyword list. For this reason, inclusion and exclusion criteria are applied at this stage. Articles that meet these criteria will be included in the next stage.

TABLE V. SUMMARY OF SLR SOLUTIONS TO ANSWER RESEARCH QUESTIONS

ID	Research Questions	Result
RQ1	What are the multi-agent pathfinding application areas?	Transportation, mining, planetary exploration, autonomous management, search and rescue, monitoring and surveillance, data transfer, computer games
RQ2	What are the central purposes of the MAPF system?	Finding the shortest and fastest path using multi-agent
RQ3	What is the kind of environment for a multi-agent-based maze pathfinding system?	Static, dynamic, real-time environment, 2D/3D, unknown environment
RQ4	What techniques are appropriate in the navigation environment for pathfinding?	Skeletonization, cell decomposition
RQ5	What are the algorithms used to develop intelligent maze pathfinding? Furthermore, which strategies are appropriate to give agents the ability to make a decision?	Flood Fill, A*, Dijkstra's, Pledge, Trémaux, Breadth-First Search, Depth First Search, Ant colony optimization. Flood Fill is the most using algorithm for maze pathfinding.
RQ6	Which of the proposed system of MAPF is a complex maze environment being studied or incorporated?	All maze solver algorithm is for a single agent in a static maze environment, so an improved Flood Fill is proposed to be tested
RQ7	What are the advantages, weaknesses, and assumptions of the reasoning frameworks proposed for MAPF?	Autonomy, distribution, adaptation, interoperability are the advantages, and uncertainty, inconsistency, and conflicting rules are weaknesses of the proposed MAPF system.
RQ8	What are the advantages, weaknesses, and assumptions of the reasoning frameworks proposed for MAPF?	Pattern recognition, integration of existing application domains, intelligent environments into systems, increasingly complex environments with multiple inhabitants are the significant challenges and future research direction set out in the literature.

Inclusion criteria:

- The Papers have a title related to reasoning systems in the maze environment.
- The keywords of the papers have a match the specified search keywords.
- Many applications are used in maze environments.
- Besides, articles related to the introduction, learning, and monitoring of the maze environment are also included as potentially relevant articles for consideration because the techniques or methods may be helpful.
- Including research on multi-agent, pathfinding, and path planning is included when the research is applied to a maze environment.

Exclusion criteria:

1) Multiple publications from the same research, under these conditions, only the latest version was used.
2) Publications that do not use English.
3) Different fields focus such as research on brain tissue, medicine, and other disciplines.
4) Other studies focus on brain tissue, medicine, and other disciplines.
5) Grey literature.

b. Following Selection: in this stage, the articles that can be main studies are assessed by skimming the full text of their contributions. Articles must meet quality assessment criteria such as context and contribution.

3) Data extraction

Data Extraction obtains the necessary data and information to build a research base, and the potential studies are then studied in more detail. Several fields were extracted based on: authors, year of publication, targets, article abstraction, MAPF technology (i.e., techniques and frameworks), reasoning (i.e., approaches, methodologies, and tools), assumptions, advantages, weaknesses, works, and future challenges.

4) Data synthesis

The selected data must be synthesized during the SLR to answer the problems contributing to answering the research questions related to this review. Table V is the summary of SLR solutions to answer Research Questions.

C. Results of Research Questions

This chapter presents an overview of related literature in the field of MAS, especially multi-agent pathfinding. This literature aims to investigate the methods used to reach the target location in the maze by the multi-agents in research published between 2005 and June 2020. MAPF is a part of a MAS that is overgrowing today. Its essential role in developing the robotic world has received significant attention from researchers who develop various algorithms for various environmental conditions.

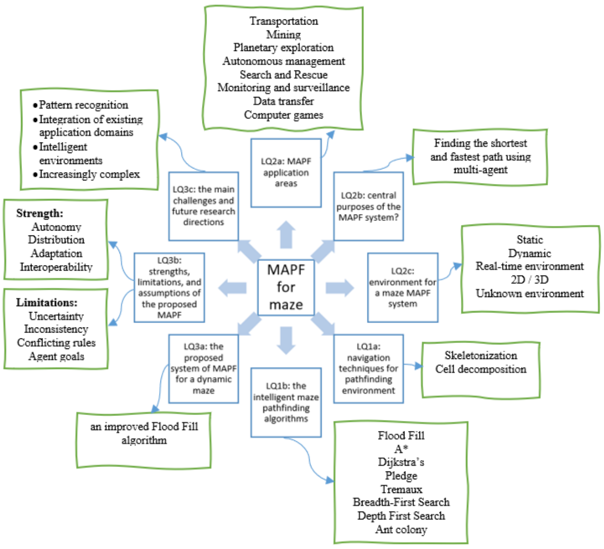

Figure 4. Mind map of an SLR on MAPF in the dynamic maze.

Fig. 4 shows the mind map of an SLR on MAPF in the dynamic maze.

Many multi-agent pathfinding applications have been developed in various fields, such as transportation, mining, planetary exploration, autonomous management, search and rescue, monitoring and surveillance, data transfer, computer games. In simple terms, The central purpose of a MAPF system is to reach the target location using the shortest and fastest path.

Of course, various types of environments will affect the effectiveness of multi-agent maze pathfinding. The environment is: the most common is the static environment like a static maze, dynamic environment, real-time environment for real robot applications, 2D/3D environment, and unknown environment. For this reason, two techniques are used that are appropriate in the navigation environment for pathfinding: skeletonization and cell decomposition. Two types of irregular grids are produced by Skeletonization techniques: a visibility or waypoint graph. Another technique that decomposes the traversable space in a sustainable environment becomes cell decomposition.

Many algorithms have been developed to develop intelligent maze pathfinding, such as the Flood Fill algorithm, A* algorithm, Dijkstra's algorithm, Pledge algorithm, Trémaux algorithm, Breadth-First Search algorithm, Depth First Search algorithm, Ant colonization algorithm, and many others. Are these strategies used to give agents the ability to make a decision? These maze solver algorithms are built for a single agent in a static maze environment, so an improved Flood Fill is proposed to be tested. The Flood Fill algorithm is the most implemented maze pathfinding algorithm. The Flood Fill algorithm has been implemented to win the micro mouse competition. So Flood Fill algorithm is chosen to test or integrated as the proposed system of multi-agent pathfinding for the dynamic maze environment.

Some strengths of MAPF systems are autonomy, distribution, adaptation, and interoperability. Nevertheless, it also has limitations like uncertainty, inconsistency, conflicting rules, and agent goals.

The significant challenges and future research direction set out in the research publication are developing agent behavior pattern recognition, integrating existing application domains and intelligent environments into systems, increasingly complex environments with multiple inhabitants.

III. Discussion

Maze exploration has long been a constantly evolving field of research. Its use in the gaming industry, robots, autonomous vehicles, autonomous warehousing, and others has involved researchers from all corners of the world. Overall, the maze problems can be grouped into three scenarios, namely: (1) The target location is unknown but has a maze schema; (2) The Labyrinth Schematic is unknown but has a target location; (3) The location of the target and the labyrinth scheme is unknown. Various approaches and ways of thinking have been developed to produce new methods and algorithms better for exploring the maze. [25]

The maze problem already has a maze scheme in the first scenario, but the target's location is unknown. For this reason, the robot or agent needs to search at the target location. This scenario is often used in game development. For example, the game Ms. Pac-man has been around for a long time and is a game where we explore a famous maze and have to dodge the ghosts that are chasing us while trying to hit our target (pills, fruits, and ghosts). [26]

In the second scenario, the maze problem determines the location to go to, then the robot or agent must reach the target location by exploring the maze whose path scheme is unknown. The robot tries to get to the destination as fast as possible in terms of time or distance. The robot is also expected to return to its starting position while exploring the maze to get complete maze schematic information so that it has better data to find the best path. The target location can be in the middle or on the edge of the maze. The target position on the edge is used for the robot to find a way out of the maze.

Various methods and algorithms have been developed to solve different maze environmental conditions in the second scenario. Each algorithm has its advantages and disadvantages. A* is used to choose the path that has the lowest cost to reach the destination [27], while Flood Fill is used to find the closest distance to reach the target location in the middle of the maze [28]-[32]. Dijkstra's algorithm has low complexity [16], [33]. Pledge algorithm that can solve circular obstacles [34]. These methods are used to reach the target location with their respective advantages and limitations.

In the last scenario, the scheme and the target's location are unknown, so the agent needs to explore the unknown maze to find the target location. One algorithm that fits this scenario is Trémaux. Robots use the Trémaux algorithm to find a way out of the maze. It uses the Depth-First Search algorithm, which allows it to find a way out of the maze, only this algorithm is not guaranteed to have the shortest route solution [35].

Now the development of maze pathfinding has entered a new stage, namely the use of multi-agents to explore the maze environment. It is hoped that multi-agent complex maze exploration can be more easily completed through multi-agent. It requires coordination and cooperation so that complex maze exploration can be accelerated.

Usually, in in-game development, the location of a fixed wall is predetermined. Then players just run their avatars, and if they find other players or other Non-player Characters (NPC), then the interaction can be done. This condition is undoubtedly different from Real Robot, especially in an unknown labyrinth. It is a challenge that robot needs to distinguish walls from other moving objects. However, the development of multi-robots is expected to make the robots a team that can work together, supporting shared goals. So that the level of effectiveness, speed, and efficiency to explore the unknown maze can be improved.

The use of robots in exploring the maze is usually equipped with various sensors such as ultra-sonic and infrared to detect obstacles so that the robot looks for a path that can be passed. The problem that often arises with the use of multi-robots is that the presence of other robots is considered an obstacle. Moreover, the condition of a constantly moving robot can be confusing. For this reason, it is necessary to develop a method that robots can use to distinguish between fixed obstacles and dynamic obstacles due to the movement of other robots.

This research on the use of this multi-agent continues to proliferate. Biswas has introduced time-efficient cooperation for the path planning model [36]. It explains how multi-agent systems deal with uncertainty together with high-efficiency planning for assignment problems. Using two frameworks: See models and Nearby-Neighbor-Search models, they have investigated cooperative path planning combined with assigning MAS in a dynamic environment. Both frameworks can control several autonomous agents to do many tasks in any location. Besides, both models can handle proactively changing environments.

Multi-agent also needs coordination, so communicating and managing information from The Actuation Map with Multi-agent Planning to solve the problem of multi-robot path planning that more efficiently skips estimates of costs during preparation is becoming essential [37]. A* promising technique for extending wireless connections to terrestrial users who have an open Radio Access Network (RAN) is made by Shi [38]. It improved the user's reasonableness and network performance. Shi designed 3D trajectories of several DBS on Drone-Assisted Radio Access Networks (DARAN), where DBS flies over related areas of interest and conveys communication between the parent station (BS) and users in the field-related interests. Other researchers, Wang and Guo, introduced simple language for MAS to coordinate them, which helps determine best plans and give more ability in behavior and reducing communication costs [39].

On the other hand, multi-agent path planning also needs formation control and obstacle avoidance. Xue introduced $H\infty$ to characterize the obstacle avoidance performance, and then the hybrid learning system will be implemented with a mission set to overcome various obstacles. Solving the problem of planning multi-robot movements with complex constraints is also introduced by Imeson and smith [40], that proposed an algorithm by combining the sophisticated Boolean Satisfaction Problem and Traveling Salesman Problem. It can implement task allocation and handle complicated constraints such as battery age limits, robot carrier capacity, and robot task mismatch.

Multi-agent is also developing in the Unmanned Aerial Vehicle (UAV) application. Coordination between multi-agents in finding path planning routes for UAVs in a 3D environment is interesting to study its application in maze applications with limited mobility. Causa [41] improves path planning using various techniques for finding route planning for multiple UAVs in a 3D heterogeneous environment characterized by variable coverage of GNSS satellites. A path planning approach is proposed to maximize efficiency in task assignments by distributing targets among UAVs. The problem of routing and scheduling of multi-drone assisted by vehicles is discussed by Hu [42], that introduced an efficient algorithm developed by Vehicle-assisted Scheduling and Multi-UAV Scheduling (VURA).

Simultaneous Localization and Mapping (SLAM) and route planning are essential research in robotics [43]. Exploration of unknown environments efficaciously is severe trouble for intelligent cell robots. Dezert-Smarandache Theory is used to handle conflicting and uncertain data.

Multi-agent applications are also developed for rescue missions. If there is a disaster in a big city, it is essential to have proper task management and preparation of paths in rescue operations. Shi introduced ISODATA-K with different ad hoc multi-agents to coordinate mission and execution, although predicaments can hinder it, such as obstacles due to building collapse [44]. Another researcher, Schurr, introduced DEFACTO as a large-scale prototype, which currently focuses on describing the potential for future disaster response to disasters that might arise because of massive terrorist attacks. Working with multi-agent teams in disaster rescue is beneficial, so they must have a good plan to choose the best strategy in the upcoming situation [45].

A. Concept of Pathfinding

Pathfinding is a path searching problem to determine the optimal path from a start node to a target node through interconnected nodes. Pathfinding algorithm is also an algorithm to solve many connection problems such as the shortest telephone line, public transport navigation routes environment, and the path selection problem in an autonomous agent [46].

One of the essential research fields is autonomous navigation systems. It enables one to reach the destination without a human operator. Arriving at the destination with the best time is the aim of the researchers, which leads to the race for the best technical development [47].

Pathfinding can be applied in any environment as a fundamental technology that affects various essential fields such as Global Positioning systems, games, robotics, warehousing, and crowd simulation. Many developments have improved pathfinding techniques, but it still attracts a great deal of research. Different pathfinding problems include single-agent pathfinding search, MAPF, changing environment, diverse terrain, cellular units, and incomplete information. All of them have different applications in unique implementation. Generally, pathfinding has two parts: graphing and pathfinding algorithms [48].

B. Significance of Multi-agent Pathfinding

This study will use the maze as an area used for conducting research simulations. In rescuing victims trapped in collapsed buildings, the speed factor in finding victims and rescuing them is the crucial factor being chased. The robot can extend the victim's critical time by bringing a survival kit such as water, food, lighting, and communication equipment when the victim is found. The speed of finding and reaching a target in the maze is the primary goal, and then a multi-agent is then proposed.

MAPF has attracted much attention in the AI community. Fig. 5 shows more detail about the MAPF, including agent, application, features, aims, and solver. There are many challenging problems of MAPF with practical implementations such as warehousing and gaming. Often, abstract versions of the problem are solved, in which the graph defines the possible nodes and the edges of the agents and the agents moving in sync. However, no more than one agent can stay in the same node so that the plans implemented are collision-free and can be executed freely [49].

Other approaches are Less-than-optimal rule-based algorithms, with the main assumption that the underlying graph is not directed. It works in the navigation domain, such as one-way entrances, escalators, bridges, and highways. It suggests an approach in which unidirectional traffic is deliberately imposed on the game map to avoid head-to-head collisions between mobile agents [50].

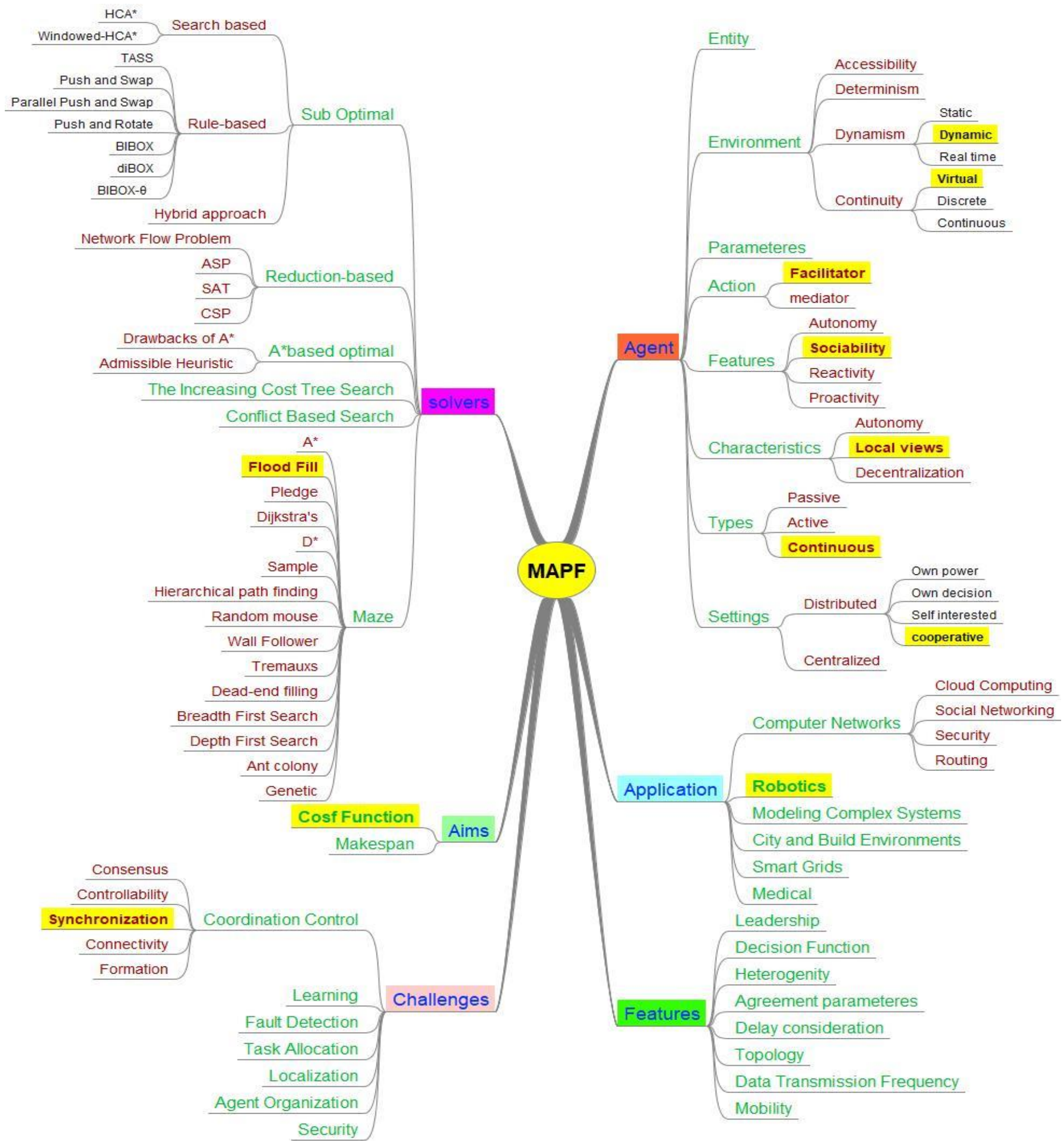

Figure 5. Multi-agent Pathfinding (MAPF) mind mapping.

Much other research on the MAPF problem and its variations includes MAPF with simultaneous moves and variants of independent detection of SAT-based [51], MAPF with capacity constraint [51]. Actual robot kinematics is the reason for studying multi-robot path discovery, Autonomous cinematography [52], Clustering-based task coordination in search and rescue teamwork [44], and much other research.

C. *Algorithms*

The maze is used as a simulation tool and algorithm testing. Based on the literature review, it is found that various kinds of algorithms are commonly used in running pathfinding in the maze. Flood Fill is one maze pathfinding algorithm widely used to solve the robot maze problem [21]. The Flood Fill algorithm gives the value that estimates the node's distance from the center. It floods the maze when it reaches a new cell; thus, it requires high-cost updates. Each cell in the maze assigns a value showing its distance from the target. Flood Fill is used to find walls and get the best route available [47].

Besides Flood Fill, the A* search algorithm and Dijkstra method are used as other maze pathfinding algorithms. Murata and Mitani also compare the A* and Djikstra algorithm, proving that the A* search algorithm outperformed Dijkstra to solve the maze in terms of search time. It had been improved from Dijkstra's method, and it uses a cost function of searching paths in the A* search algorithm [53].

Trémaux's algorithm is an efficient depth algorithm applicable to finding the way out of a maze. Trémaux may not provide the shortest route solution but will find a solution for all mazes by marking all visited pathways, so the way to return to the source is recorded, and all visited pathways would not be revisited. This algorithm will never pick the same marked pathway twice [35]. Pledge algorithm is also a maze pathfinding algorithm designed for circular obstacles and has an initial direction to move forward. The robot will run in the main direction until it finds obstacles. When the robot finds an obstacle, the robot will use a wall follower search method and avoid obstacles by prioritizing the right or left side. It will calculate the total turn and try to return to the initial direction (total turn count is "0") [18].

Not many studies have used Multi-agent Pathfinding (MAPF) in mazes. All these algorithms are designed for a single maze pathfinding algorithm. A single agent starts searching in the maze to target and returns to the starting point via another path that is probably shorter than before. The pathfinding algorithm in this maze is certainly not suitable for MAPF, which requires communication and coordination to work together to achieve the target position.

Generally, the MAPF algorithm can be grouped into two categories. The first is Reduction-based solvers that can reduce MAPF to another known problem such as SAT, integer programming, and answer set programming. The second is Search-based solvers that look for some of the constraints imposed on the agent [49]. Pavel Surynek introduced the first SAT-solver for the sum of costs variant of MAPF. It uses the amount of cost as the lower bound and makespan as the upper bound, which can have a reasonable number of variables in SAT coding [14].

Botea introduced the first tractability analysis of MAPF on directed graphs. Its focus is on strongly biconnected digraphs where the undirected graphs obtained by ignoring the edge orientations have no-cut vertices [50]. MAPF assumed that planning is centralized and the agents' goals are known, but Nebel designed agents to achieve their respective goals using an implicitly coordinated plan without communication [54].

Hang Ma studied several MAPF problems in the actual situation. There are two general concerns in categories:

1) The method is not fast enough, so it is necessary to formulate a new formula.
2) MAPF is not just a combinatorial optimization problem because it needs to be run. [55].

One key feature of pathfinding is making the right decision to find the path in the maze. Many AI techniques have been applied to the Maze environment, so it is hoped that the MAS architecture can explore dynamic labyrinths by developing autonomous intelligent agent interactions. The characteristics of a multi-agent as an intelligent agent are as follows: [24]

- Autonomy: agents can interact directly independently.
- Sociability: MAS can interact with each other.
- Reactivity: agents can provide appropriate responses to changing situations.
- Pro-activeness: agents can take the initiative to respond to their environment.

Table VI shows various studies on maze exploration and multi-agents, the main research sources on Multi-Agent Pathfinding for unknown environments, schemes, and target locations.

TABLE VI. MAIN SOURCES OF RESEARCH ON MULTI-AGENT PATHFINDING

No	Author - Year	Title	Env.	single /multi	Task Allocation
1	V. Rahmani, et al. 2020	Multi-Agent Parallel Hierarchical Path finding In Navigation Meshes (MA-HNA*)	Static - known	Multi	No
2	R. Semenas, R. Bausys 2020	Modelling Of Autonomous Search And Rescue Missions By Interval-Valued Neutrosophic WASPAS Framework	Static - known	single	No
3	R. Barták, et al. 2019	Multi Agent Path finding On Real Robots	Static - known	Multi	No
4	A. Bogatarkan, et al. 2019	A Declarative Method For Dynamic Multi-Agent Path Finding	Dynamic - known	Multi	No
5	P.M. Chu, et al. 2019	Flood Fill Based Object Segmentation And Trackting For Intelligent Vehicles	Static - known	single	No
6	Y. Liu, et al. 2019	Formation Control And Collision Avoidance For A Class Of Multi Agent Systems	Static - known	Multi	No
7	K. Shetty, P. Kanani. 2019	Drivable Road Corridor Detection Using Flood Fill Road Detection Algorithm	Static - known	single	No
8	H. Shi, et al. 2019	Clustering Based Task Coordination To SAR Teamwork Of Multiple Agents	Static - known	Multi	No
9	A. Botea, et al. 2018	Solving Multi-Agent Path finding On Strongly Biconnected Digraphs	Static - known	Multi	No
10	D. R. Robinson et al. 2018	An Efficient Algorithm For Optimal Trajectory Generation For Heterogeneous Multi - Agent Systems In Non Convex Environments	Static - known	Multi	No
11	C.H. Wu, et al. 2018	A Method For Finding The Routes Of Mazes	Static - known	single	No
12	M.S. Gade, et al. 2017	Design And Implementation Of Swam Robotics Using Flood Fill Algorithm	Static - known	Multi	No
13	H. Ma, et al. 2017	Overview Generalizations Of Multi-Agent Path finding To Real-World Scenarios	Static - known	Multi	Yes
14	A. Ambeskar, et al. 2016	Path finding Robot Using Image Processing	Static - known	single	No
15	A.M. Jabbar 2016	Autonomous Navigation Of Mobile Robot Based On Flood Fill Algorithm	Static - known	single	No
16	P. Surynek, et al. 2016	Efficient SAT Approach To Multi Agent Path finding Under The Sum Of Costs Objective	Static - known	Multi	No
17	T. Andre, C. Bettstetter 2015	Collaboration In Multi Robot Exploration	Static - unknown	Multi	No
18	A. Ansari, et al. 2015	An Optimized Hybrid Approach For Path-Finding	Static - known	single	No
19	S. Ozturk, et al. 2015	Optimal Bid Valuation Using Path finding For Multi Robot Task Allocation	Static - known	Multi	Yes
20	G. Sharon, et al. 2015	Conflict Based Search For Optimal Multi Agent Path finding	Static - known	Multi	No
21	A. Singh 2015	A New Shortest First Path finding Algorithm For A Maze Solving Robot	Static - known	single	No
22	G. Law 2013	Quantitative Comparison Of Flood Fill And Modified Flood Fill Algorithms	Static - known	single	No
23	H. Reddy 2013	Path finding - Dijkstra's And A Star Algorithms	Static - known	single	No
24	D. Xue, et al. 2013	Formation Control And Obstacle Avoidance For Hybrid Multi Agent Systems	Static - known	Multi	No
25	A. G. Chauhan, et al. 2012	Evaluation Of Modified Flood Fill Algorithm For Shortest Path Navigation In Robotics	Static - known	single	No
26	I. Elshamarka, A.B.S. Saman 2012	Design And Implementation Of A Robot For Maze Solving Using Flood Fill Algorithm	Static - known	single	No
27	N. Zheng, et al. 2011	Recursive Path Planning In A Dynamic Maze With Modified Tremaux's Algorithm	Static - known	single	No
28	Jin H.J., et al. 2010	Multi Robot Path finding With Wireless Multihop Communications	Static - known	Multi	No
29	N. Schurr, et al. 2005	The Future Of Disaster Response Humans Working With Multiagent Teams Using DEFACTO	Static - unknown	Multi	No

Table VI shows that most of the algorithms used have a static environment and a known target position. Only one article explores the dynamic environment with the known target position, and two articles have an unknown target position in the static environment. Meanwhile, research using multi-agent compared to single-agent is almost balanced, namely 16 multi-agents and 13 single-agents. Most agents have fixed tasks, and only one study allows agents to exchange tasks. It looks that research characteristics to be carried out have their characteristics compared to the research that has been done. The Improved Flood Fill algorithm can work in dynamic environments where the target location is unknown by using a multi-agent that can swap tasks more effectively.

IV. Conclusion

Based on various literature research conducted from October 2019 to January 2021, various articles were obtained describing the research results of researchers on multi-agent systems, various maze solver algorithms, and especially multi-agent pathfinding. Meanwhile, most MAPF applications are applied to static environments and targets with known locations. The use of multi-agents also mainly uses agents with their own goals to have distributed decisions and interests. A small proportion of research demonstrates the ability to swap environmental topologies, tasks, and more. Table VI shows the comparison of single and multi-agent pathfinding applications.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S. Tjiharjadi conducted the research; S. Tjiharjadi and S. Razali analyzed the data; S. Razali and H.A. Sulaiman supervised the research; S. Tjiharjadi wrote the paper; all authors had approved the final version.

Acknowledgment

The authors wish to thank the Centre for Robotics and Industrial Automation (CeRIA) and Fakulti Teknologi Maklumat dan Komunikasi (FTMK) at Universiti Teknikal Malaysia Melaka (UTeM) for supporting the research and facilities. This work is supported in part by a scholarship from Maranatha Christian University.

References

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, “Multi-agent systems: A survey,” *IEEE Access*, vol. 6, pp. 28573-28593, 2018.

[2] J. Fu and J. Wang, “Adaptive coordinated tracking of multi-agent systems with quantized information,” *Syst. Control Lett.*, vol. 74, pp. 115-125, 2014.

[3] S. Li, H. Du, and X. Lin, “Finite-Time consensus algorithm for multi-agent systems with double-integrator dynamics,” *Automatica*, vol. 47, no. 8, pp. 1706-1712, 2011.

[4] D. M. Zhang, L. Meng, X. G. Wang, and L. L. Ou, “Linear quadratic regulator control of multi-agent systems,” *Optim. Control Appl. Methods*, vol. 36, no. 1, pp. 45-59, 2015.

[5] G. Wen, G. Hu, W. Yu, J. Cao, and G. Chen, “Consensus tracking for higher-order multi-agent systems with switching directed topologies and occasionally missing control inputs,” *Syst. Control Lett.*, vol. 62, no. 12, pp. 1151-1158, 2013.

[6] H. Du, Y. He, and Y. Cheng, “Finite-Time synchronization of a class of second-order non-linear multi-agent systems using output feedback control,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 61, no. 6, pp. 1778-1788, 2014.

[7] G. Miao and Q. Ma, “Group consensus of the first-order multi-agent systems with non-linear input constraints,” *Neurocomputing*, vol. 161, pp. 113-119, 2015.

[8] Y. Kim and E. T. Matson, “A realistic decision making for task allocation in heterogeneous multi-agent systems,” *Procedia Comput. Sci.*, vol. 94, pp. 386-391, 2016.

[9] Z. Liu, X. You, H. Yang, and L. Zhao, “Leader-Following consensus of heterogeneous multi-agent systems with packet dropout,” *Int. J. Control. Autom. Syst.*, vol. 13, no. 5, pp. 1067-1075, 2015.

[10] L. Gao, X. Liao, H. Li, and G. Chen, “Event-Triggered control for multi-agent systems with general directed topology and time delays,” *Asian J. Control*, vol. 18, no. 3, pp. 945-953, 2016.

[11] B. Liu, H. Su, R. Li, D. Sun, and W. Hu, “Switching controllability of discrete-time multi-agent systems with multiple leaders and time-delays,” *Appl. Math. Comput.*, vol. 228, pp. 571-588, 2014.

[12] G. Guo, L. Ding, and Q. L. Han, “A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems,” *Automatica*, vol. 50, no. 5, pp. 1489-1496, 2014.

[13] C. Wang, G. Xie, and M. Cao, “Controlling anonymous mobile agents with unidirectional locomotion to form formations on a circle,” *Automatica*, vol. 50, no. 4, pp. 1100-1108, 2014.

[14] P. Surynek, A. Felner, R. Stern, and E. Boyarski, “Efficient SAT approach to multi-agent pathfinding under the sum of costs objective,” *Front. Artif. Intell. Appl.*, vol. 285, pp. 810-818, 2016.

[15] P. M. Chu, S. Cho, K. Huang, and K. Cho, “Flood-Fill-Based object segmentation and tracking for intelligent vehicles,” *Int. J. Adv. Robot. Syst.*, vol. 16, no. 6, pp. 1-11, 2019.

[16] H. Reddy, “Path finding - Dijkstra’s and A* algorithm’s,” *Int. J. IT Eng.*, pp. 1-15, 2013.

[17] S. Tjiharjadi, M. C. Wijaya, and E. Setiawan, “Optimization maze robot using A* and flood fill algorithm,” *Int. J. Mech. Eng. Robot. Res.*, vol. 6, no. 5, 2017.

[18] S. Tjiharjadi, “Design and implementation of flood fill and pledge algorithm for maze robot,” *Int. J. Mech. Eng. Robot. Res.*, vol. 8, no. 4, 2019.

[19] R. Kala, “Multi-robot path planning using co-evolutionary genetic programming,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3817-3831, 2012.

[20] A. Viseras, R. O. Losada, and L. Merino, “Planning with ants,” *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, pp. 1-16, 2016.

[21] A. Ansari, M. A. Sayyed, K. Ratlamwala, and P. Shaikh, “An optimized hybrid approach for path finding,” *Int. J. Found. Comput. Sci. Technol.*, vol. 5, no. 2, pp. 47-58, 2015.

[22] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Inf. Softw. Technol.*, vol. 64, pp. 1-18, 2015.

[23] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering - A systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7-15, 2009.

[24] D. N. Mekuria, P. Sernani, N. Falcionelli, and A. F. Dragoni, *Reasoning in Multi-agent Based SMART homes: A Systematic Literature Review*, Springer International Publishing, 2018, vol. 544.

[25] C. M. Wu, D. C. Liaw, and H. T. Lee, “A method for finding the routes of mazes,” in *Proc. Int. Autom. Control Conf.*, 2018, pp. 1-4.

[26] G. Foderaro, A. Swingler, and S. Ferrari, “A model-based approach to optimizing Ms. Pac-Man game strategies in real time,” *IEEE Trans. Comput. Intell. AI Games*, vol. 9, no. 2, pp. 153-165, 2017.

[27] E. Todt, G. Raush, and R. Sukez, “Analysis and classification of multiple robot coordination methods,” in *Proc. International Conference on Robotics and Automation. Symposia Proceedings*, 2000.

[28] K. Shetty and P. Kanani, “Drivable road corridor detection using flood fill road detection algorithm,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, pp. 5011-5014, 2019.

[29] T. Andre, *et al.*, “Evaluation of modified flood fill algorithm for shortest path navigation in robotics,” *J. Intell. Manuf.*, vol. 7109, no. 7, pp. 126-132, 2015.

[30] M. S. Gade, G. GAjitha, and D. S, “Design and implementation of swarm robotics using flood fill algorithm,” *IAES Int. J. Robot. Autom.*, vol. 6, no. 4, p. 269, 2017.

[31] S. Tjiharjadi and E. Setiawan, “Design and implementation of a pathfinding robot using flood fill algorithm,” *Int. J. Mech. Eng. Robot. Res.*, vol. 5, no. 3, 2016.

[32] H. Dang, J. Song, and Q. Guo, "An efficient algorithm for robot maze-solving," in *Proc. 2nd Int. Conf. Intell. Human-Machine Syst. Cybern*, 2010, vol. 2, pp. 79-82.

[33] S. Öztürk and A. E. Kuzucuoğlu, "Optimal bid valuation using pathfinding for multi-robot task allocation," *J. Intell. Manuf.*, vol. 26, no. 5, pp. 1049-1062, 2015.

[34] S. Tjiharjadi, "Performance comparison robot pathfinding uses flood fill - Wall follower algorithm and flood fill - Pledge algorithm," *Int. J. Mech. Eng. Robot. Res.*, vol. 9, no. 6, pp. 857-864, 2020.

[35] N. Z. Yew, K. M. Tiong, and S. T. Yong, "Recursive pathfinding in a dynamic maze with modified tremaux's algorithm," *International Journal of Mathematical and Computational Sciences*, vol. 5, no. 12, pp. 2102-2104, 2011.

[36] S. Biswas, S. G. Anavatti, and M. A. Garratt, "A time-efficient co-operative path planning model combined with task assignment for multi-agent systems," *Robotics*, vol. 8, no. 2, p. 35, Apr. 2019.

[37] N. Luis, T. Pereira, S. Fernández, A. Moreira, D. Borrajo, and M. Veloso, "Using pre-computed knowledge for goal allocation in multi-agent planning," *J. Intell. Robot. Syst. Theory Appl.*, vol. 98, no. 1, pp. 165-190, 2020.

[38] W. Shi, *et al.*, "Multi-drone 3-D trajectory planning and scheduling in drone-assisted radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8145-8158, May 2019.

[39] L. Wang and Q. Guo, "Coordination of multiple autonomous agents using naturally generated languages in task planning," *Appl. Sci.*, vol. 9, no. 17, 2019.

[40] F. Imeson and S. L. Smith, "An SMT-based approach to motion planning for multiple robots with complex constraints," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 669-684, 2019.

[41] F. Causa, G. Fasano, and M. Grassi, "Multi-UAV path planning for autonomous missions in mixed GNSS coverage scenarios," *Sensors (Switzerland)*, vol. 18, no. 12, 2018.

[42] M. Hu, *et al.*, "Joint routing and scheduling for vehicle-assisted multi drone surveillance," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1781-1790, 2019.

[43] P. Li, X. Huang, S. Wang, and J. Dezert, "SLAM and path planning of mobile robot using DSmT," *J. Softw. Eng.*, vol. 7, no. 2, pp. 46-67, 2013.

[44] H. Shi, R. Zhang, G. Sun, and J. Chen, "Clustering-Based task coordination to search and rescue team work of multiple agents," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1-9, 2019.

[45] N. Schurr, J. Marecki, M. Tambe, P. Scerri, N. Kasinadhuni, and J. P. Lewis, "The future of disaster response: Humans working with multi-agent teams using DEFACTO," *AAAI Spring Symp. - Tech. Rep.*, vol. SS-05-01, pp. 9-16, 2005.

[46] J. C. Young, A. Suryadibrata, R. Luhulima, and U. M. Nusantara, "Review of various A * pathfinding implementations in game autonomous agent," *Int. J. New Media Technol.*, 2019.

[47] A. Jabbar, "Autonomous navigation of mobile robots based on flood fill algorithm," *Iraqi J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 79-84, 2016.

[48] Z. A. Algfoor, M. S. Sunar, and H. Kolivand, "A comprehensive study on pathfinding techniques for robotics and video games," *Int. J. Comput. Games Technol.*, vol. 2015, 2015.

[49] R. Barták, J. Švancara, V. Škopková, D. Nohejl, and I. Krasičenko, "Multi-agent path finding on real robots," *AI Commun.*, vol. 32, no. 3, pp. 175-189, 2019.

[50] A. Botea, D. Bonusi, and P. Surynek, "Solving multi-agent pathfinding on strongly biconnected digraphs," *J. Artif. Intell. Res.*, vol. 62, pp. 273-314, 2018.

[51] P. Surynek, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with capacity constraints," in *Proc. International Conference of the Italian Association for Artificial Intelligence*, 2019.

[52] A. Torres-González, J. Capitán, R. Cunha, A. Ollero, and I. Mademlis, "A multidrone approach for autonomous cinematography planning," *Adv. Intell. Syst. Comput.*, vol. 693, pp. 337-349, 2018.

[53] Y. Murata and Y. Mitani, "A fast and shorter path finding method for maze images by image processing techniques and graph theory," *J. Image Graph.*, vol. 2, no. 1, pp. 89-93, 2014.

[54] B. Nebel, T. Bolander, T. Engesser, and R. Mattmüller, "Implicitly coordinated multi-agent pathfinding under destination uncertainty: Success guarantees and computational complexity (extended abstract)," *Int. Jt. Conf. Artif. Intell.*, pp. 6372-6376, 2019.

[55] H. Ma, *et al.*, "Overview: Generalizations of multi-agent pathfinding to real-world scenarios," arXiv, 2017.

Semuil Tjiharjadi currently serves as an assistant professor in Computer Engineering Department. His primary research is on Robotics, Computer automation, control, and security. He has written several books, To Be a Great Effective Leader (Jogjakarta, Indonesia: Andi Offset, 2012), Multimedia Programming by SMIL (Jogjakarta, Indonesia: Andi Offset, 2008), Computer Business Application (Bandung, Indonesia: Informatics, 2006) and other. He contributed to the various academic bodies as Vice-Rector of human capital management, assets, and development, Head of Computer Engineering Department, Member: Senate of University, Member: APTIKOM, Member: MSDN Connection, Member: AAJI.

Sazalinsyah Razali has a Ph.D. in Computer Science, Masters in Computer Science, BSc. (Hons.) in Information Technology, a Certified Tester (Foundation Level - MSTB/ISTQB, 2013), and a Certified IT Professional (FE Module - MITPE/ITPEC, 2005). He joined Universiti Teknikal Malaysia Melaka in 2004 as a Lecturer at the Faculty of Information and Communication Technology and now a Senior Lecturer since 2014. He started his academic career in 2003 as a Lecturer at Kolej Universiti Kejuruteraan Utara Malaysia, now known as Universiti Malaysia Perlis. Before that, Dr. Sazalinsyah Razali was a Lead Web Developer in a local IT company and was involved in various web-based and e-commerce projects. He has professionally served as Program Committee and Reviewer in several international conferences & journals, and he is a Professional Member of several professional societies, such as IEEE, IEEE Robotics & Automation Society (IEEE-RAS), IEEE Computational Intelligence Society (IEEE-CIS), IEEE Systems, Man & Cybernetics Society (IEEE-SMC) and Malaysian Society for Engineering & Technology (MySET).

Hamzah Asyrani Bin Sulaiman received his Bachelor and Master degree from Universiti Teknologi Malaysia (UTM), Malaysia, in 2007 and 2010, respectively. He received the Doctor of Philosophy (Ph.D.) in Mathematics with Computer Graphics from Universiti Malaysia Sabah in 2015. His research interest lies in Computer Graphics and Visualization, Mathematics, Computer Games Technology, Data Computation, and Computer Engineering. He has over 700 citations, with Google Scholar at H Index at 12 and SCOPUS H Index at 8. Many of his publications concentrated on Computer Graphics and Visualization and various computer engineering fields. Currently, he holds the position as Head of Department in Interactive Media from 2020. He is actively involved in the management of the faculty. He is also a Professional Technologist from the Board of Technologist Malaysia (MBOT) recognition that held a title of Ts from 2018 until now