Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Multi-Agent Pathfinding in Real-World Scenarios: Challenges and Solutions

Siddharth Gupta, Dr. Riddhi Garg*

Associate Professor, Department of Mathematics, School of Sciences, IFTM University, India
Email: Riddhigarg5@gmail.com

Multi-agent pathfinding is a crucial task that deals with efficiently coordinating the movement of multiple agents, ensuring they can reach their respective targets without any collisions. Although there have been notable advancements in MAPF algorithms, their practical implementation poses unique challenges that require innovative methodologies and approaches. Conventional MAPF algorithms typically work with simplified assumptions, such as discrete environments, holonomic agents, and full knowledge of the world state. When it comes to addressing real-world challenges, it is crucial to develop solutions that take into account the practical limitations of robots, including their physical attributes like shape, size, and kinematic capabilities such as differential drive and car-like steering. In addition, real-world environments seldom provide complete information. MAPF algorithms need to be able to handle uncertainty caused by factors such as sensor noise, incomplete maps, or dynamic changes in the environment. In practical scenarios, it is common to have a multitude of agents working in intricate environments. This calls for scalable solutions that can handle the complexities involved, surpassing the capabilities of centralized, optimal solvers. This work offers a thorough examination of the latest research developments aimed at addressing the fundamental obstacles of MAPF in practical situations. We discuss the integration of Kino dynamic constraints, emphasizing techniques that guarantee feasible trajectories for different types of robots using realistic motion models. We explore various methods for managing uncertainty in the environment or agent sensing capabilities, including replanning strategies and probabilistic or partially observable planning approaches. For tackling the scalability challenge, we delve into decentralized MAPF techniques, prioritization-based methods, and hierarchical solutions that break down the problem into more manageable sub-parts. Finally, we discuss the latest trends in the field, including the combination of MAPF with task assignment for multi-robot teams, and the use of learning-based techniques to improve pathfinding efficiency and adaptability in real-world scenarios. This article provides a valuable resource for researchers and practitioners who want to gain a deeper understanding of the current state of real-world MAPF and the ongoing research in this field.

Keywords: Multi-Agent Pathfinding (MAPF), Kinodynamic Constraints, Machine Learning, Replanning, Probabilistic Planning, Decentralization.

1. Introduction

Multi-agent pathfinding (MAPF) is essential for coordinating intelligent systems that operate in shared settings. The issue at hand pertains to the calculation of collision-free paths for several entities in order to ensure their secure movement from their starting positions to their individual desired destinations [1]. MAPF is very relevant in a wide range of practical applications. Warehouse automation involves the employment of autonomous robots to navigate through intricate and possibly crowded layouts, which makes it a very interesting application. Similarly, the optimization of the coordination of self-driving cars in urban settings or the coordinated movement of groups of robots are domains where Multi-Agent Path Finding (MAPF) algorithms lead to enhancements in efficiency and safety. The potential consequences of effective Multi-Agent Path Finding (MAPF) solutions are revolutionary. They include increased efficiency in logistics, decreased traffic congestion, and improved capabilities for collaborative robots. These are only a few instances of the extensive advantages [2]. The field of theoretical multi-agent path finding (MAPF) has generated a plethora of techniques and frameworks. Conventional methods usually make use of simplifying assumptions in order to make the issue easier to solve using computers. Agents are often represented as point entities that move on discrete grids, and it is believed that the environment is completely understood. Centralized solvers aim to find the best or limited suboptimal solutions, ensuring that they are complete if viable alternatives are available. The classical approaches provide a robust theoretical basis for MAPF [3]. Nevertheless, the transition from these abstract algorithms to their implementation in real-life situations poses a distinct and complex array of obstacles. This research article explores the difficulties associated with real-world multi-agent navigation, highlighting the intricacies involved. It also examines recent improvements that aim to connect theoretical concepts with practical implementation. A key disparity between traditional MAPF models and real-world systems is the way robotic agents are shown. Physical robots in the real world exhibit many forms, dimensions, and limitations on movement. Unlike hypothetical point agents positioned on a grid, they are unable to immediately alter their course; their motion is regulated by kinodynamic models [4]. For instance, a differential drive robot lacks the ability to move sideways straight and instead must adhere to curved paths. Robots that resemble cars bring forth additional intricacies, such as the need to consider steering angles and minimum turning radii. Conventional Multi-Agent Path Finding (MAPF) algorithms that disregard these limitations run the risk of producing pathways that are impractical or dangerous for robots to carry out [5]. The use of accurate robot kinematics in pathfinding greatly enhances the intricacy of the issue, necessitating the use of specialist solutions. An oversimplification often seen in a significant portion of the literature on Multi-Agent Path Finding (MAPF) is the assumption of possessing comprehensive and deterministic information about the environment. Real-world surroundings are naturally characterized by constant change and are prone to unpredictability. Noises from sensors, maps that are not correct, unexpected barriers, or the behaviors of other agents that cannot be predicted (such as people in a shared environment) might make pre-calculated plans no longer useful [6]. Algorithms intended for practical implementation must be resilient to these uncertainties, showcasing the capacity to dynamically reevaluate or modify paths in response to evolving circumstances.

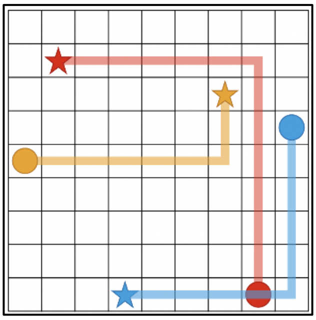

Fig 1. MAPF [4]

Approaches such as online replanning, probabilistic approaches, and planning under partial observability are necessary for effectively managing this inherent uncertainty as shown in Fig 1. Ultimately, the problem of scalability presents a tangible obstacle for Multi-Agent Path Finding (MAPF) in real-world scenarios. Numerous fascinating scenarios include a significant number of agents functioning within intricate and vast contexts. Contemplate the complex coordination necessary for several or even numerous robots inside a bustling warehouse. Centralized, optimum algorithms for Multi-Agent Path Finding (MAPF) may become computationally intractable in such situations. Hence, it is crucial to develop strategies to manage extensive Multi-Agent Path Finding (MAPF) scenarios for practical use [7]. This study explores many techniques to address the challenge, including decentralized planning, where agents cooperate on a local level, priority-based methods that strategically prioritize agent planning, and hierarchical approaches that break down the problem into smaller, more manageable sub-problems. This article provides a thorough examination of the latest progress in multi-agent pathfinding, specifically addressing the difficulties presented by real-world situations. We discuss the integration of kinodynamic restrictions, presenting an overview of techniques that may provide viable trajectories while adhering to robot motion models. We explore ways for effectively navigating unpredictable and ever-changing settings, ranging from tactics for adapting plans to approaches based on the use of probabilistic models. In addition, we discuss strategies for efficiently addressing real-world multi-agent path finding (MAPF) problems, such as decentralized methods, prioritizing, and hierarchical approaches [8]. This study also emphasizes the increasing incorporation of Multi-Agent Path Finding (MAPF) with job allocation for teams of many robots, as well as the capacity of machine learning methods to improve flexibility and optimization in practical MAPF solutions. The primary objective of this study is to provide a helpful resource for scholars and practitioners. It intends to enhance comprehension of the difficulties and the latest advancements that influence the effective implementation of Multi-Agent Path Finding (MAPF) in real-world situations that are dynamic, complicated, and unpredictable.

2. Background of Classical MAPF

The theoretical underpinnings of multi-agent pathfinding (MAPF) are based on a number of formal definitions, essential assumptions, and algorithmic techniques that have been the driving force behind a significant amount of research within this field. The purpose of this part is to present an overview of these conventional MAPF principles, with the goal of building a foundational knowledge before moving on to the issues that are faced in real-world applications.

2.1. Problem Definition

The foundation of classical multi-agent pathfinding (MAPF) is a rigorous mathematical framework that accurately establishes the issue and the limitations under which solutions are sought [9]. The central focus of this description is the primary goal: to calculate paths that avoid collisions for many entities as they move from their starting points to their individual destinations in a common environment [10]. In order to comprehend the functioning of MAPF algorithms, it is essential to deconstruct the fundamental components and parameters of this issue specification.

The traditional Multi-Agent Path Finding (MAPF) issue may be formally represented as a tuple (A, G, S, E), where: Agents (A) is a set A = {a₁, a₂, ..., a_k} that specifies the 'k' agents engaged in the pathfinding problem. Every agent 'a_i' is an individual entity with a distinct initial and target position. Agents are often represented as basic entities inside the environment and are the main focus of the planning process. The term "environment" (G) refers to the specific conditions and surroundings in which the agents carry out their tasks. Traditional Multi-Agent Path Finding (MAPF) often represents the environment as a graph composed of vertices (nodes) and edges. Vertices indicate allowable positions that agents may occupy, whereas edges provide the connection between these positions, suggesting potential movement transitions for agents. A graph may be classified as either undirected, allowing movement in both directions along an edge, or directed, restricting movement to a specified direction. It should be noted that 'G' may include specified barrier sites that agents are required to avoid. The start positions (S) are defined as S = {s₁, s₂, ..., s_k}, which represents the first vertex locations inhabited by each agent inside the environment graph 'G'. Each agent, denoted as 'a_i', is associated with a certain beginning vertex, represented as 's_i'. The term "Goal Positions (E)" refers to the desired or intended locations that need to be reached or achieved. The set E = {e₁, e₂, ..., e_k} represents the collection of target vertex positions in the environment that each agent strives to achieve. Each agent, denoted as 'a_i', is assigned a specific objective vertex, represented as 'e_i'.

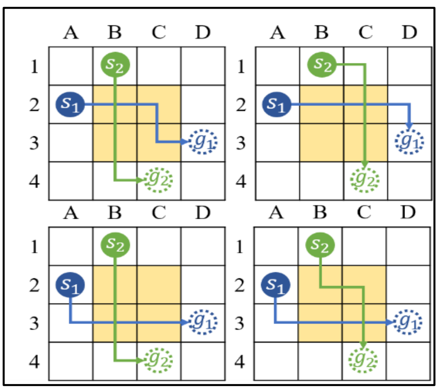

Fig 2. Pathfinding Problem Definition

Time discretization in classical multi-agent path finding (MAPF) involves dividing time into discrete units. At each time step, an agent has the option to either Wait, which implies staying at its present vertex. Alternatively, move down an edge of the graph 'G' to a neighboring vertex that is not currently occupied [11]. In order to provide safe navigation for agents without any conflicts, MAPF categorizes collisions into two basic kinds as in Fig 2. Vertex collisions occur when several agents try to occupy the same vertex simultaneously. Edge collisions happen when two agents attempt to cross a single edge in opposing directions at the same time. The main objective of MAPF algorithms is to generate a collection of pathways (one for each agent) that satisfy the following requirements, Free from collisions, The design of paths should ensure that there are no collisions between vertices or edges throughout the whole execution of the plan. This is necessary for achieving completeness. Every agent is required to effectively achieve its assigned objective vertex from its initial position. In addition, conventional Multi-Agent Path Finding (MAPF) algorithms often strive to find solutions that meet optimality requirements, generally by minimizing the Makespan [12]. The overall duration for all agents to achieve their objectives (i.e., the number of time intervals until the last agent meets its objective) and the cumulative costs. The total lengths of the pathways taken by all agents combined. To summarize, the traditional Multi-Agent Path Finding (MAPF) problem description offers a meticulous mathematical representation that includes crucial components such as agents, environment, spatial and temporal discretization, collision detection, and desirable solution features [13]. This formal description provides a basis for creating and examining solution algorithms in the field of classical Multi-Agent Path Finding (MAPF).

The theoretical underpinnings of multi-agent pathfinding (MAPF) are based on a set of formal definitions, fundamental assumptions, and algorithmic methods that have spurred substantial study in this field. This part presents a comprehensive introduction to the classical notions of Multi-Agent Path Finding (MAPF), aiming to create a basic comprehension before delving into the difficulties faced in real-world scenarios.

2.2. Assumptions of Classical MAPF

Classical Multi-Agent Path Finding (MAPF) methods, while they provide a strong theoretical foundation for coordinating many agents, sometimes depend on a series of simplifying assumptions to guarantee manageable computing complexity [14]. While these assumptions are useful for first defining the issue and deriving a solution, they may create a substantial gap when implementing MAPF algorithms in real-world contexts. An essential assumption in traditional Multi-Agent Path Finding (MAPF) is the division of time and space into discrete units [15]. Time is split into regular, distinct periods or time increments. The discretized representation enables systematic exploration of the state-space, which forms the foundation for many search-based MAPF algorithms. Similarly, the environment is often represented as a graph or grid. Agents are present as individual entities in this discretized space, limited to occupying just one vertex or grid cell at a time [16]. During each time step, the agent is only allowed to move to a neighboring vertex or stay in its present position if the desired spot is already occupied. Discretization simplifies the issue representation but creates a discrepancy with real-world situations where time is continuous and robots have dimensions that need considering partial cell occupancy in motion planning. Traditional MAPF algorithms often neglect the physical attributes, movement limitations, and orientations of agents in the environment. Agents are conceptually represented as points that may travel instantly and in any direction between neighboring cells or vertices [17]. This oversimplification disregards the intricacies involved in the navigation of robots in real-world scenarios. Robots are unable to change directions immediately, and their ability to turn may be restricted by factors such as differential drive types, car-like steering, or non-circular footprints. The assumption of holonomic agents poses the possibility of developing courses that are either physically impossible or very wasteful for robots to carry out in real-world environments [18]. An underlying assumption in conventional Multi-Agent Path Finding (MAPF) is that agents have comprehensive awareness of the whole environment. Agents are assumed to possess accurate knowledge of the map's configuration, including stationary obstructions, as well as the positions of other agents at all times. Maps, in actuality, often lack comprehensive information, necessitating agents to construct and revise representations of the world as they traverse it. In addition, sensors contribute both noise and ambiguity into the agent's perception of its surroundings, as well as the positions of other agents or unanticipated barriers. Classical approaches for Multi-Agent Path Finding (MAPF) often presume full knowledge, which creates a disconnect between theory and real-world application. In actuality, choices sometimes need to be made based on partial or ambiguous information about the state of the environment [19]. Traditional Multi-Agent Path Finding (MAPF) often represents the environment as fixed and completely predictable. The map structure, beginning agent placements, and environmental circumstances remain constant throughout the execution of the plan. In contrast, real-world surroundings are seldom stationary.

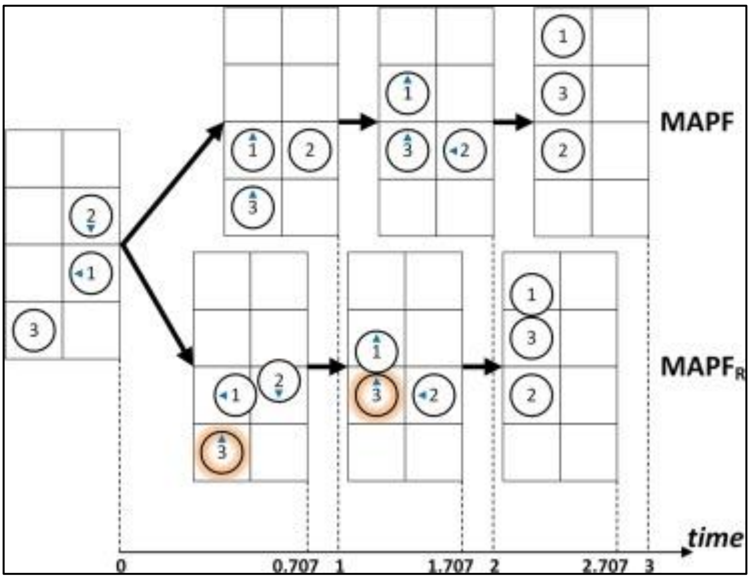

Fig 3. Problems on Continuous Time [13]

Agents may face dynamic impediments, unanticipated changes to the landscape, or unexpected actions of other things (such as people) coexisting in the same environment. The use of environmental determinism oversimplifies the computing challenge while disregarding the need for resilient algorithms that can adjust their strategies in response to unforeseen alterations [20]. Several conventional Multi-Agent Path Finding (MAPF) methods, especially those emphasizing optimum solutions, assume the presence of a central planner who has full access to information on all agents and the environment Fig 3. This one organization oversees and manages all planning decisions [21]. However, in situations involving large-scale robotics or systems with intermittent connectivity, centralized planning might create a significant bottleneck. When aiming for scalability and robustness in real-world applications, there is a growing emphasis on using distributed and decentralized Multi-Agent Path Finding (MAPF) solutions. It is essential to comprehend the constraints imposed by these assumptions in order to move MAPF from a theoretical realm to its effective implementation in dynamic, real-world settings. Identifying these inconsistencies motivates research focused on resolving practical intricacies, which includes developing techniques that integrate robot movements and creating algorithms that can make plans while accounting for uncertainty and adapt trajectories in response to changes in the environment and the agent's perception over time [22]. Addressing the disparity between conventional MAPF theory and its actual implementation is an important research subject. This study has the potential to unlock the practical advantages of multi-agent pathfinding in several areas.

2.3. Search Based Approaches

One common approach in classical Multi-Agent Path Finding (MAPF) algorithms is to use a search-based methodology. This involves systematically exploring the solution space to find collision-free trajectories for all agents. The foundation of this approach lies in the notion of a search space, which serves as an abstract representation of every conceivable configuration or state of the problem at hand [23]. In Multi-Agent Path Finding (MAPF), a state typically represents the positions of all agents at a specific time step. Search-based algorithms are utilized to construct a graph-like structure. This structure connects different states based on the actions that agents can take [24]. These actions include moving to an unoccupied adjacent location or waiting for a time step. The objective of the search is to identify a sequence of states, along with their corresponding actions, that facilitate the transition of the agents from their initial starting configuration to a goal configuration [25]. In the goal configuration, all agents will have successfully reached their intended destinations without any instances of collision.

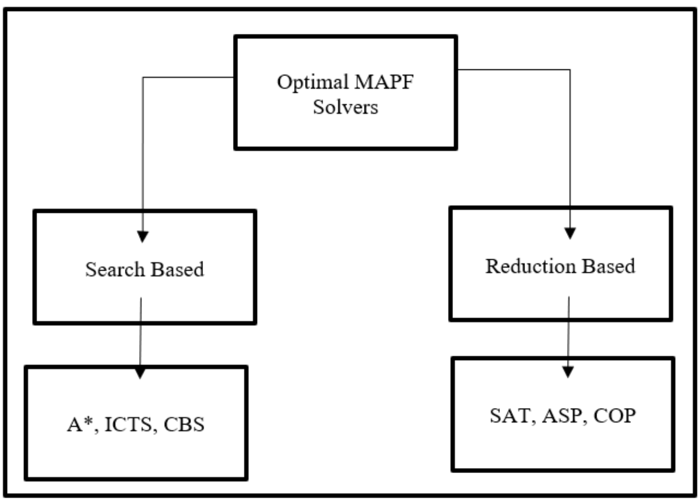

Fig 4. Main Approaches

The concept of state representation refers to the way in which the state of a system or object is described or modelled Fig 4. It involves capturing the relevant information and variables that define the The state representation in Multi-Agent Path Finding (MAPF) typically includes the current vertex locations of all agents within the graph G [26]. Moreover, it is possible to enhance solution efficiency or tackle specific problem variants by integrating supplementary information into the state representation. The parameters that can be included are time step, remaining agent path distances, and data structures that encode pathfinding history to assist in conflict resolution [27]. The process of constructing a search space graph involves... The search algorithms employ an iterative process to construct a graph or tree-like structure that represents the search space. The nodes in this structure represent various states, while the edges represent actions that facilitate the transition of the system from one state to another [28]. The typical actions performed by an agent include either moving to a neighboring vertex or staying in its current position. The efficient expansion of the search space is crucial for optimizing the performance of the algorithm. The following is a discussion on heuristic functions for optimality. Heuristic functions are essential in guiding the search process, particularly when the goal is to find optimal solutions, such as the shortest path length. A heuristic function is utilized to provide an estimation of the cost or distance between a specific state and the goal state [29]. The heuristic in Multi-Agent Path Finding (MAPF) must take into account all agents involved, which can be done by either aggregating individual agent cost estimates or directly calculating a joint heuristic. The Breadth-First Search (BFS) algorithm is a methodical approach that explores the state space in a layer-by-layer manner. The computational efficiency of solving large Multi-Agent Path Finding (MAPF) problems is often suboptimal. The Depth-First Search (DFS) algorithm is used to explore a single branch of the search space in a deep manner before backtracking. The memory in Multi-Agent Path Finding (MAPF) can become limited as the length of paths increases with the number of agents and the size of the environment. The user has entered the text "A Search * ," which appears to be a search query.The A* algorithm is a widely used informed search algorithm that incorporates both the current known cost from the start state and a heuristic estimate of the remaining cost to the goal state. The A* algorithm is commonly used as a preferred method for finding optimal solutions in classical Multi-Agent Path Finding (MAPF) problems. Algorithms such as Anytime A* (and its derivatives) or Weighted A* provide different variations that offer trade-offs between solution quality, optimality, and runtime bounds. The identification and resolution of conflicts, specifically vertex or edge collisions, pose a fundamental challenge in the field of Multi-Agent Path Finding (MAPF) search [30]. Commonly employed strategies involve the utilization of backtracking search mechanisms alongside pruning or reordering of agent movement priorities in order to systematically investigate alternative paths. Search-based Multi-Agent Path Finding (MAPF) algorithms often prioritize completeness, ensuring that a feasible solution is found if one exists. Various search variations are also focused on achieving optimality by seeking trajectories that minimize the makespan or the sum-of-costs metrics.

Search-based algorithms are commonly used in classical Multi-Agent Path Finding (MAPF) problems [31]. However, they have certain limitations when applied to real-world scenarios. The size of the search space increases exponentially as the number of agents and the size of the environment increase. The computational challenges arise when applying centralized search-based Multi-Agent Path Finding (MAPF) algorithms to large-scale problems. The omission of robot kinematics in traditional search-based approaches can result in impractical paths that cannot be executed by real robots. The effectiveness of classical search-based Multi-Agent Path Finding (MAPF) methods is limited in real-world situations with uncertainty and unforeseen changes due to their reliance on perfect information and static environments.

The current research in multi-agent pathfinding (MAPF) aims to address the gap between the theoretical foundations laid by search-based methods and the practical difficulties of implementing MAPF solutions in dynamic and complex real-world environments [32]. The transformation encompasses a range of advancements that focus on integrating real-world complexities, managing uncertainties, and addressing scalability limitations that arise in search-based classical Multi-Agent Path Finding (MAPF) algorithms.

The incorporation of kinodynamic constraints is a fundamental area of focus in robotics. These constraints are designed to reflect the physical limitations and motion characteristics of real robotic systems [33]. The classical search-based Multi-Agent Path Finding (MAPF) approach, which represents agents as holonomic points, has the potential to produce trajectories that are either infeasible or excessively inefficient for real-world robots. The technique being discussed is Discrete Search with Kinodynamic Post-Processing. Search algorithms function within a discrete representation, similar to classical MAPF (Multi-Agent Path Finding). The resultant paths are transformed into smooth, feasible trajectories that respect robot kinematics through a post-processing stage. The techniques used in this context involve the utilization of interpolating curves to connect discrete waypoints or the application of trajectory optimization methods that incorporate kinodynamic constraints. In the context of exploring robot kinematics, a more efficient approach is to utilize search-based methods that directly operate within a Kinodynamically-Aware State Space, rather than relying on post-processing techniques. The process typically entails augmenting the state representation to incorporate factors such as robot orientation or velocity, and customizing the expansion of the search space to generate actions that align with the dynamics of the robot [34]. Hybrid approaches utilize a combination of discrete high-level search methods and continuous, local trajectory optimization techniques. This allows for the generation of smooth, collision-free, and kinodynamically feasible movement within shorter time horizons.

The classical Multi-Agent Path Finding (MAPF) paradigm is based on the strict assumption of complete and deterministic knowledge. However, this assumption does not align with real-world environments, where uncertainty is prevalent due to factors such as sensor noise, imperfect world models, and unforeseen changes [35]. In order to tackle this issue, research in Multi-Agent Path Finding (MAPF) explores various prominent techniques such as The process of replanning entails the periodic recalculation of agent paths in response to the availability of new information. The utilization of this adaptive approach allows agents to effectively respond to alterations in the environment or perceived deviations from the initial plan. There are different variations of replanning methods, including time-bounded replanning which involves periodic replanning within a specified time horizon, and event-triggered replanning which is based on the detection of specific occurrencesv[36]. Probabilistic approaches offer a framework for addressing incomplete and noisy information when uncertainty is present in sensing. The MAPF algorithms have the capability to include uncertainty in environment maps, as well as the positions of agents and obstacles. This allows for the modeling of beliefs about the state of the environment and updating actions accordingly. The utilization of Partially Observable Markov Decision Process (POMDP) frameworks and associated methodologies is prevalent in the modeling of scenarios where agents possess restricted observation capabilities. One possible approach is to maintain and update probabilistic belief states. Another option is to focus planning efforts on reachable and observable regions of the environment.

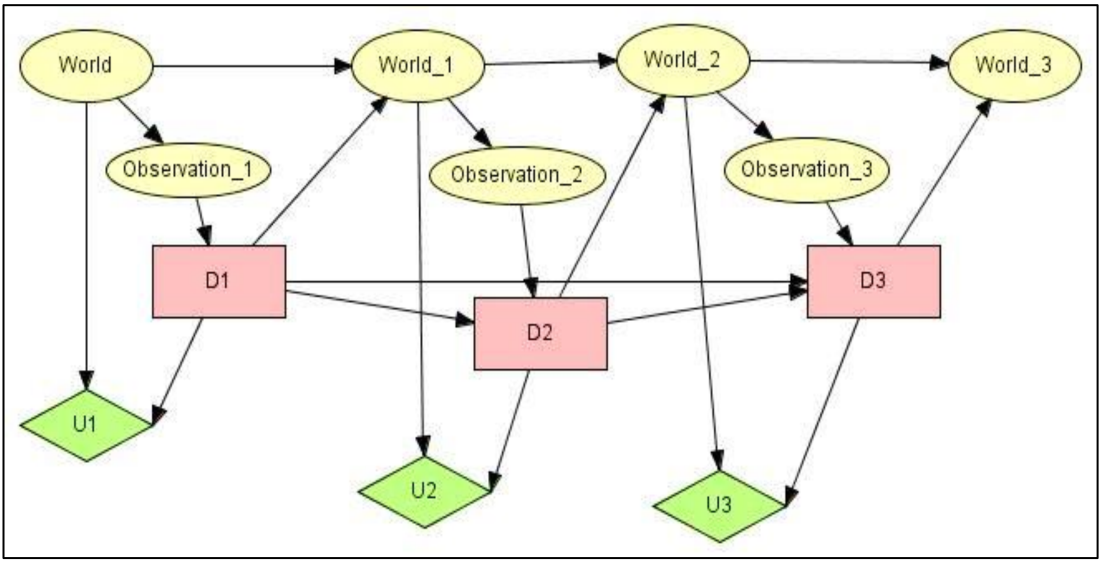

Fig 5. POMDP

The computational burden imposed by centralized search-based Multi-Agent Path Finding (MAPF) solutions becomes a significant bottleneck for real-world deployment as the number of agents or the size of the environment increases. The concept of Decentralized Planning refers to the decentralized Multi-Agent Path Finding (MAPF) approach, where individual agents calculate paths within a restricted range Fig 5. These agents also collaborate with nearby agents to resolve any potential conflicts that may arise. The process involves the decomposition of the computationally demanding central planning task and its subsequent distribution, resulting in enhanced scalability for systems of significant scale [38]. The process of prioritization involves determining the relative importance or urgency of tasks, activities, or items. It is a method used to The assignment of order or priority to agents within the planning process is referred to as techniques. The search complexity is reduced by employing a structured approach that involves sequencing agent path calculations and integrating conflict resolution heuristics. The Hierarchical Approaches employ a decomposition strategy to break down the Multi-Agent Path Finding (MAPF) problem into multiple levels of abstraction. The process of high-level planning involves the generation of broad paths or sub-goals, whereas local-level planning is concerned with executing detailed, collision-free paths within smaller subregions [39]. The objective of this research is to leverage the progress made in kinodynamic modeling, uncertainty handling, and scalable search algorithms to enhance the applicability of Multi-Agent Path Finding (MAPF) in practical domains. Research continues to address these challenges, fostering the development of MAPF solutions that are robust, adaptable, and efficient enough to guide robotic systems within complex and unpredictable environments.

2.4. Reduction Based Approaches

Reduction-based approaches provide a different approach in conventional MAPF compared to the search-based exploration of agent configurations. This class of algorithms focuses on transforming the MAPF issue into a distinct (and often well researched) problem type, in order to use established solution methods from other fields to effectively identify feasible pathways for agents. The concept of issue transformation is central to reduction-based techniques. Instead of doing a direct search for pathways in the original MAPF representation, which is a graph with changing agent locations over time, these approaches transform the MAPF problem into an alternative problem structure [40]. Common problem classes used in reductions include Boolean Satisfiability (SAT). MAPF instances may be represented as a collection of logical constraints, with propositional variables indicating the presence or absence of agents at specified locations and time steps. A solution that meets these requirements corresponds to a collection of valid trajectories for the agents in the original Multi-Agent Path Finding (MAPF) issue. Subsequently, SAT solvers may be used to quickly discover solutions, capitalizing on extensive research in solver optimization spanning many decades. Constraint Satisfaction Problems (CSPs) may be used to model Multi-Agent Path Finding (MAPF), where agents are treated as variables and constraints are used to enforce collision avoidance principles. Constraint satisfaction problem (CSP) approaches, including constraint propagation and search algorithms, may be used to resolve conflicts and provide viable agent assignments that correspond to collision-free pathways. Flow networks. MAPF with discrete time and space may be converted into network flow issues in certain formulations. Agents are shown as "flow" moving across a graph-like network that is built according to environmental restrictions and allowable behaviors over time intervals. Algorithms specifically developed to determine the most efficient or viable routes may be used to calculate potential paths.

2.4.1. Conflict-Based Search (CBS)

CBS functions by doing a search to identify potential conflicts. In this sense, a conflict is a scenario where two or more agents clash either at a vertex or along an edge. The method begins by separately calculating a distinct route for each agent, disregarding any possible interactions with other agents [41]. Conflicts are then recognized and handled in a methodical manner. Resolution entails imposing limitations on the trajectories of agents, such as preventing agent 'a1' from occupying vertex 'v5' at time step 't3', and then recalculating the plans for the agents that are impacted. CBS employs a search structure similar to a binary tree to arrange conflict resolution, gradually refining limitations until a solution without conflicts is discovered.

2.4.2. Auction-based approaches

Auction-based methodologies represent Multi-Agent Path Finding (MAPF) as a problem of allocating resources, where agents compete by placing bids on specific time and space positions. Agents iteratively calculate pathways that minimize their individual costs, often using efficient single-agent pathfinding techniques [42]. An auctioneer oversees the process, settling issues by considering bids from participants and assigning places or time periods. This might include the reallocation of sites that experience high levels of competition or the dynamic adjustment of pricing. Different iterations of auction-based algorithms vary in terms of their pricing methods, auction structures, and dispute resolution procedures. MAPF may benefit from reduction-based techniques by using the existing research and optimized tools in fields such as SAT, CSPs, and network optimization. Nevertheless, there are significant factors to take into account: Translating the original MAPF issue into a new problem domain might result in increased computing burden. While some reduction-based techniques may provide assurances of completeness, others may depend on heuristics and repeated refinement, which might result in scenarios where a solution is not discovered, even if one theoretically exists. Reduction-based procedures provide a convincing alternative to search-based methods in conventional Multi-Agent Path Finding (MAPF). The selection of a reduction strategy and its particular implementation have a substantial impact on the effectiveness and appropriateness of these techniques for various instances of Multi-Agent Path Finding (MAPF).

3. Challenges of Real-World MAPF

Although conventional MAPF provides a solid theoretical framework, moving these algorithms from idealized problem formulations to real-world application poses a distinct set of difficulties. The reason for these gaps between theory and practice is that simplistic presumptions falter when faced with the uncertainty and complexity of the actual world. Let us examine the three main types of obstacles encountered in the implementation of MAPF in real-world scenarios:

3.1. Kinodynamic Constraints

By introducing the physical limits and motion characteristics of actual robots into the planning process, kinodynamic constraints provide another level of complexity to multi-agent pathfinding (MAPF). Kinodynamic constraints require taking into account elements such as robot body dimensions, actuation mechanisms, and limitations in acceleration, velocity, and turning capabilities. This is in contrast to classical MAPF, which treats agents as holonomic points capable of instantaneous movement between adjacent locations. In order to create viable and effective courses for actual robots to follow inside an environment, it is essential to comprehend these limits.

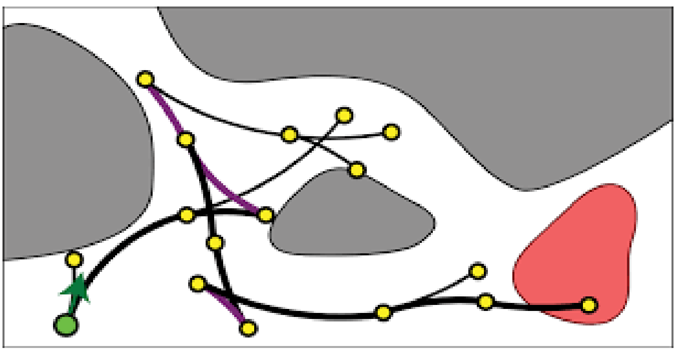

Fig 6. Motion Models Pathways

Actual robots move in accordance with a variety of motion models that specify what is allowed. Popular models include of Two wheels on each of these robots are driven individually. Although they are highly maneuverable, their mobility is limited to forward, backward, and curved trajectories because of the intrinsic non-holonomic restriction that prevents them from translating straight sideways [43]. These robots' steering mechanism's physical limitations give them a minimal turning radius, much as automobile steering systems Fig 6. This restricts their capacity to make abrupt turns, which affects the viability of routes intended for holonomic agents. Omnidirectional wheels may allow robots to travel in any direction. But even these models have limits when it comes to maximum speeds, rotational restrictions, and acceleration that need to be taken into account when designing a route. The pathfinding issue in conventional MAPF is greatly simplified by the assumption of holonomic agents. Ignoring kinodynamic limitations might result in the creation of potentially, Routes that need abrupt direction changes or quick turns on a small radius might go against the robot's turning limitations. Real robots may choose inefficient paths that involve a lot of backtracking or pointless accelerations because of their limited ability to accelerate. It is possible for paths that ignore the robot's physical dimensions to collide with objects or other agents that are not taken into consideration in a purely geometric description. Resolving Kinodynamic Limitations Path planning algorithms need to take robot motion models and their constraints into account directly in order to overcome these difficulties. Motion Primitives, which represent essential robot movements (forward, backward, and turning) as the cornerstones for creating workable routes, are often used in this context [44]. The Method of Sampling Robot configuration space may be effectively explored using methods such as Rapidly-exploring Random Trees (RRTs), which take into account the robot's kinematic restrictions and look for pathways that avoid collisions. Strategies: Trajectory optimization techniques may improve candidate pathways produced by other algorithms, making sure they respect kinematic restrictions and reduce variables like as trip time or path length. One of the most important steps in creating practical and workable designs for robot navigation in the actual world is to include kinodynamic restrictions into MAPF. These methods open up the possibility of deploying multi-agent systems in complicated situations safely and effectively by taking into account the physical constraints of robots.

3.2. Uncertainty

In traditional Multi-Agent Pathfinding (MAPF), the assumption of possessing comprehensive and flawless information about the environment and the states of agents, often known as "certainty," is significantly different from real-world situations. Uncertainty stems from several factors that make the planning process more complex and need strong algorithms for effective implementation of Multi-Agent Path Finding (MAPF). Sensor noise is a significant factor contributing to uncertainty. Cameras, lidars, and other robot perception devices unavoidably contribute flaws into the agent's comprehension of its environment [45]. This presents itself as sensor readings that are characterized by excessive noise, which may possibly result in the incorrect identification of impediments, imprecise estimations of distances, or inaccuracies in determining the positions of other entities. The disparities between the real environment and the perception produced from sensors may have a substantial influence on judgments related to pathfinding. In addition, real-world ecosystems are seldom well charted or unchanging. Another problem arises from maps that are not fully completed. Maps may provide an incomplete depiction of the landscape, since they may include unknown regions or inaccurately portray obstacles. Likewise, the environment itself might be subject to change. Unanticipated hindrances such as items that have fallen, changes in the landscape caused by weather events, or the erratic motions of other entities (e.g., people) inject unanticipated intricacies into the process of planning. In order to tackle these uncertainties, many technological methods may be used in MAPF algorithms. Replanning encompasses a single core approach. Replanning is the process of frequently recalculating pathways for the robot to adapt to the current circumstances as it navigates the area and gets fresh sensor data. Replanning may be initiated either at regular time intervals, known as time-bounded replanning, or in response to particular events, such as the identification of an impediment that was not previously recognized, known as event-triggered replanning. In addition to replanning, probabilistic approaches provide a structure to explicitly include uncertainty into MAPF algorithms [46]. Probabilistic techniques, as opposed to deterministic representations, use probability distributions to model robot locations, map characteristics, and the presence of barriers. Pathfinding algorithms may use these probabilities to make calculated judgments in situations of ambiguity. For example, an algorithm may aim to find routes that have the highest probability of attaining the desired outcome while also reducing the chance of coming across unexpected barriers, using the probability distributions for where these barriers could be located. POMDPs are useful for circumstances when there are limitations in sensory capacity. POMDP-based techniques include agents maintaining a "belief state," which is simply a probability distribution that represents the potential true states of the world. This includes the positions of other agents and barriers. The belief state is revised as further observations are obtained via sensors. Planning is the process of choosing activities that will result in the highest possible benefits, taking into account the present level of knowledge or beliefs. By integrating uncertainty-handling strategies, MAPF algorithms may surpass the constraints of traditional methods, enabling the development of resilient and flexible solutions that can effectively navigate the intricacies of real-world situations.

3.3. Scalability

Scalability in traditional Multi-Agent Pathfinding (MAPF) algorithms pertains to their capacity to effectively manage situations involving a substantial number of agents moving across intricate landscapes. Nevertheless, the computing requirements of centralized search-based Multi-Agent Path Finding (MAPF) solutions, which strive to identify optimum or viable routes for all agents at once, present a substantial obstacle as the problem's scale increases. The fundamental problem comes in the rapid and significant increase in the number of possible search options. As the quantity of agents (n) and the magnitude of the environment (shown by the number of vertices or locations, m) escalate, the quantity of potential arrangements or conditions that the system may assume expands exponentially (often estimated as O(mn^k), where k is a constant factor) [47]. The exponential increase in the complexity of the search space results in a significant rise in the processing time needed for centralized search algorithms to examine every potential route and find solutions that are free from collisions. Consider a basic setting consisting of 100 vertices and 5 agents as an example. The search space now includes an astounding 1005 (10 billion) possible combinations. When the environment size or the number of agents increase to more realistic levels, centralized algorithms that rely on exhaustive exploration approaches struggle to handle the search space. Decentralized techniques include distributing the planning process instead of relying on a single central planner to coordinate all actors. Agents strategically choose their routes by considering just the information available within their communication range. They may use negotiation protocols or priority-based processes to effectively address any possible conflicts that may arise with nearby agents. By dispersing the planning responsibilities, the search space load on any one entity is greatly reduced. Enhancing scalability may be achieved by giving priority to agent planning. Agents here strategically choose their courses in a predetermined sequence. This enables agents with greater priority to set their pathways first, hence possibly streamlining pathfinding for agents with lower priority, who may then adjust their plans based on the previously defined paths of higher-priority agents. It is crucial to meticulously construct prioritization mechanisms to guarantee equity and prevent scenarios in which low-priority agents are indefinitely stuck because of the activities of higher-priority actors [48]. Hierarchical planning involves breaking down the issue into many layers of abstraction. High-level planning involves creating general courses or waypoints for each agent, often using simplified models of the environment or agent interactions. Subsequent to the initial planning, a more specific and precise planning process is used. This process involves refining the initial pathways into more detailed trajectories that ensure there are no collisions. Additionally, these trajectories are designed to adhere to the specific movements and mechanics of the robot inside smaller sections of the environment. The use of this stratified method aids in the management of the complexity of the search space, as the use of high-level planning prevents being overwhelmed by the detailed specifics of low-level collision avoidance. MAPF algorithms may enhance scalability by using decentralized, prioritized, or hierarchical techniques. This allows them to effectively manage bigger agent populations and more intricate settings, resulting in better efficiency when compared to centralized search-based methods.

4. Recent Advancement in Real World MAPF

The rise of recent developments in MAPF research has been driven by the need to tackle the obstacles posed by real-world applications. These developments aim to close the disparity between theoretical MAPF algorithms and their actual use in areas such as warehouse automation, autonomous cars, and swarm robots. Now, we will examine important progress made in recent research, carefully evaluating their advantages, constraints, and possible influence on practical situations.

4.1. Algorithms for Handling Kinodynamic Constraints

Classical Multi-Agent Pathfinding (MAPF) algorithms often consider agents as point-like entities, disregarding the physical constraints of actual robots in the real environment. Recent progress has tackled this issue by integrating kinodynamic restrictions, which define a robot's ability to move, into pathfinding algorithms. Perform a search using the Kinodynamic algorithm [49]. Post-processing refers to the steps taken after an initial process or task has been completed. Search methods function inside a distinct state space, but a further phase of post-processing is used to refine the trajectories and guarantee adherence to kinodynamic restrictions. This might include using methods such as interpolating curves (e.g., splines) to connect discrete waypoints or utilizing optimization algorithms that explicitly integrate kinodynamic restrictions. This technique utilizes preexisting search-based algorithms that operate inside a discrete state space, such as grids or graphs. The search algorithm identifies a consecutive series of states that reflect the intended route for each agent. Nevertheless, these trajectories may not be immediately implementable by physical robots owing to their intrinsic constraints. In order to close this divide, a post-processing phase converts these distinct pathways into seamless, achievable trajectories. Some of the methods used include Spline interpolation involves using spline functions, which are mathematical curves that accurately pass through a series of waypoints or discrete states created by a search process. The use of these splines guarantees the preservation of a consistent and seamless trajectory, enabling uninterrupted movement of the robot. Kinodynamic trajectory optimization utilizes optimization methods especially tailored for robotic motion models. These algorithms use the discrete route and the kinematic restrictions of the robot, such as velocity and acceleration limitations, as input. They then construct a trajectory that complies with these constraints while minimizing a cost function, such as travel time or energy consumption. Explore Kinodynamic State Spaces, The search space is enlarged to include parameters that describe the robot's orientation, velocity, or other elements of its kinodynamic state. Search algorithms subsequently go over this state space that takes into account the dynamics of the system. These methods combine discrete high-level search or coarse-level planning with continuous trajectory optimization techniques. The advanced search algorithm may provide intermediate points, and local trajectory optimization techniques further improve these points to create smooth pathways that are possible in terms of both motion and dynamics within shorter time periods. Conventional search algorithms work with a state space that represents the locations of agents. However, this technique broadens the state space to incorporate factors specifically linked to the movement of robots [48][49]. For robots that have non-holonomic motion, such as differential drive robots, orientation becomes an important state variable. The search algorithm examines various combinations of locations and orientations throughout the planning phase to ensure that the resulting route adheres to the robot's constraints on turning radius and maneuverability. Velocity may be included as a state variable, enabling the search algorithm to take into account acceleration limitations and provide pathways with realistic velocity profiles. This guarantees that the robot can physically execute the intended movement without beyond its constraints. Algorithms specifically built with kinodynamics in consideration provide pathways that can be directly executed by robots in the actual world, therefore avoiding the possible problems that arise when these limitations are ignored. There are also Hybrid techniques available that combine the advantages of discrete high-level search with continuous, local trajectory optimization. At a macroscopic level, a search algorithm might determine general routes or intermediate places for each agent while taking into account the prevention of collisions between agents. Afterwards, these rough pathways are sent to local trajectory optimization modules that enhance them into intricate, kinodynamically viable trajectories within shorter time periods. This hybrid method combines the advantages of both approaches, allowing for effective high-level planning and assuring compliance with kinematic limitations at a later stage. The inclusion of kinodynamic restrictions often leads to a rise in computational complexity, which in turn affects the time required to find a solution. Discretization in these techniques might result in a balance between solution correctness and computational practicality. By including these techniques for managing kinodynamic restrictions, MAPF algorithms may produce trajectories that are both free of collisions and directly executable by physical robots, hence improving their practicality and efficacy in real-world scenarios. This is crucial for practical situations when robots with differential drive, car-like steering, or other motion limitations need to work together, such as groups of warehouse robots or self-driving automobiles.

4.2. Methods for Planning Under Uncertainty

Classical MAPF algorithms often assume perfect knowledge of the environment and agent states. However, real-world scenarios are inherently uncertain due to limitations in sensor capabilities, dynamic environments, and unforeseen events. To address this, recent advancements in MAPF research have introduced several methods for planning under uncertainty that is, A fundamental approach is replanning. Here, the robot operates with an initial plan based on the best available information. However, as the robot navigates and gathers new sensor data, discrepancies between the expected and observed states of the environment become evident. Replanning algorithms allow the robot to react to these discrepancies by dynamically recomputing its path in real-time. Two common variations exist: time-bound replanning, where the robot replans periodically (e.g., every 10 seconds), and event-triggered replanning, where replanning is triggered by specific events like encountering an obstacle not present in the map. When sensor noise or incomplete information is a major concern, probabilistic methods offer a powerful framework. Instead of relying on deterministic representations of the environment and robot states, these methods model uncertainty using probability distributions. For instance, the location of an obstacle might be represented as a probability distribution over a certain area, reflecting the sensor's potential for error. Similarly, the robot's position itself can be modeled probabilistically, accounting for potential odometry drift or localization errors. Pathfinding algorithms are then adapted to incorporate these probabilistic representations. Some approaches utilize Bayesian inference techniques to update these probability distributions as new sensor data is received, while others leverage sampling-based methods like Monte Carlo simulations to explore a range of possible scenarios and identify robust plans that minimize risk or maximize the probability of success under uncertainty [50]. Partially Observable Markov Decision Processes (POMDPs) provide a sophisticated framework for situations where the robot's ability to observe the world is limited. POMDPs model the environment as a set of states, with the robot possessing an incomplete belief state (a probability distribution) about which state the world is actually in. As the robot moves and receives observations, it updates its belief state using Bayes' rule. Planning within a POMDP framework involves selecting actions that maximize the expected long-term reward, considering the current belief state and the potential outcomes of different actions under various world states. While computationally more demanding than other methods, POMDPs offer a powerful approach for planning under significant limitations in observability. The choice of method for handling uncertainty depends on the specific characteristics of the environment, sensor capabilities, and computational resources available. Replanning offers a practical approach for adapting to unexpected changes, while probabilistic methods provide a robust framework for modeling and reasoning under sensor noise or incomplete information. POMDPs represent the state-of-the-art for planning under partial observability, but their computational complexity necessitates careful consideration for real-world deployment.

Classical Multi-Agent Path Finding (MAPF) methods often rely on the assumption of having complete and accurate information about the environment and the states of the agents. Nevertheless, real-life situations are intrinsically unpredictable as a result of constraints in sensor capabilities, ever-changing settings, and unexpected occurrences. To tackle this issue, recent progress in Multi-Agent Path Finding (MAPF) research has brought forward many techniques for devising plans in situations when ambiguity is present. Replanning is a key method. Here, the robot functions using an initial strategy that is established based on the most accurate and up-to-date information. Nevertheless, when the robot moves about and collects fresh sensor data, inconsistencies between the anticipated and actual conditions of the surroundings become apparent. Replanning methods enable the robot to respond to these inconsistencies by recalculating its course in real-time. There are two typical variations: time-bound replanning, which involves the robot replanning at regular intervals (e.g., every 10 seconds), and event-triggered replanning, which occurs when certain events, such as hitting a barrier that is not on the map, happen. Probabilistic approaches provide a robust foundation for dealing with sensor noise or partial information. Instead of depending on fixed representations of the environment and robot states, these techniques use probability distributions to account for uncertainty [51]. For example, the position of a barrier may be expressed as a probability distribution throughout a certain region, indicating the sensor's margin of error. Likewise, the location of the robot might be represented probabilistically, taking into consideration any deviations in odometry or inaccuracies in localization. Pathfinding algorithms are modified to include these probabilistic representations. Certain methodologies employ Bayesian inference techniques to update these probability distributions when new sensor data is obtained, while others utilize sampling-based methods such as Monte Carlo simulations to investigate various scenarios and determine resilient plans that minimize risk or maximize the likelihood of success in uncertain situations. Partially Observable Markov Decision Processes (POMDPs) provide a comprehensive framework for scenarios in which the robot's capacity to see the environment is restricted. POMDPs represent the environment as a collection of states, while the robot has an uncertain belief state (expressed as a probability distribution) on the true state of the world. As the robot navigates and collects data, it modifies its belief state by using Bayes' rule. Planning in a POMDP paradigm entails choosing activities that optimize the anticipated long-term payoff, taking into account the present belief state and the possible consequences of alternative actions in other world situations. Although POMDPs need more processing resources compared to other approaches, they provide a robust strategy for planning in situations with substantial constraints in observability. The selection of a strategy for managing uncertainty is contingent upon the particular attributes of the surroundings, the capabilities of the sensors, and the computing resources at hand. Replanning provides a pragmatic strategy for adjusting to unforeseen alterations, whereas probabilistic approaches give a resilient structure for modeling and logical thinking in the presence of sensor noise or inadequate information. POMDPs are currently the most advanced method for planning in situations when there is little information available. However, their high computational complexity means that they need to be carefully evaluated before being used in real-world applications.

4.3. Approaches for Scalable MAPF

The emerging field of real-world Multi-Agent Path Finding (MAPF) requires scalable systems capable of efficiently managing a high volume of agents exploring complex surroundings. Centralized search-based algorithms, albeit providing assurances of optimality, often encounter difficulties in terms of scalability since the search space expands exponentially as the number of agents (n) and the size of the environment (m) grow. In this article, we examine three primary strategies that improve the scalability of Multi-Agent Path Finding (MAPF) algorithms:

Decentralized techniques, in contrast to centralized systems, allocate the responsibility of planning among several individuals rather than relying on a single entity. Every agent devises its trajectory by considering just the information available within its communication range. Conflicts are addressed via negotiation protocols or methods based on priority [52]. This greatly lowers the workload of searching for any one entity, since each agent simply devises strategies for itself and its close neighbors. Effective communication is essential in decentralized Multi-Agent Path Finding (MAPF). Agents may share information on their planned routes, enabling them to detect any problems and work together to resolve them. Agents may construct negotiation procedures to engage in bargaining over space and time slots, with the ultimate goal of reaching agreements to prevent collisions. Alternatively, one may use prioritizing algorithms to allocate agents a certain sequence for planning their pathways. The use of a sequential strategy streamlines the process of resolving conflicts and decreases the intricate nature of the issue. Nevertheless, the task of creating efficient prioritizing systems continues to be a difficult one, since it is crucial to ensure that lower-priority agents are not indefinitely paralyzed by the activities of higher-priority agents.

Introducing an order into the planning process addresses scalability by prioritizing agent planning. Agents are allocated priority according to criteria such as their level of urgency, intended destination, or the relevance of their mission. Agents with greater priority prioritize their route planning, providing a reference point for agents with lower priority to adjust their own pathways based on the previously established trajectories of the higher-priority agents. Existing pathfinding algorithms may be used via prioritization strategies, giving higher-priority agents the first opportunity to use them. In this context, meticulous planning is essential to guarantee equity and avoid scenarios where less important agents are consistently hindered by the activities of more important ones. Furthermore, the concept of dynamic prioritizing may be used, wherein the priorities of agents are modified over time in response to evolving conditions.

Hierarchical techniques include breaking down the Multi-Agent Path Finding (MAPF) issue into many layers of abstraction. High-level planning involves creating general courses or waypoints for each agent, often using simplified models of the environment or agent interactions. This strategic strategy takes into account the overarching objectives and avoids being too focused on the detailed specifics of low-level accident prevention. Subsequent to the initial planning, lower-level planning takes control and further refines the broad pathways by creating intricate, collision-free trajectories that adhere to the specific movements of the robot inside smaller sections of the environment. The use of a tiered approach aids in the management of search space complexity [51][52]. High-level planning is responsible for addressing the overall scope, while the lower level handles the specific tasks of generating collision-free paths. Nevertheless, the effectiveness of hierarchical systems depends on the quality of the high-level strategy. Errors or inconsistencies at this level may have a domino effect and adversely affect the practicality of the ultimate, detailed paths. MAPF algorithms may offer increased scalability by using decentralized, prioritized, or hierarchical techniques. These systems are capable of effectively managing situations on a vast scale, including many participants and intricate surroundings. This makes them suitable for practical usage in areas such as automating warehouses, coordinating autonomous vehicles, and implementing swarm robots.

5. Discussion and Future Directions

This study examined the basic difficulties in transferring traditional Multi-Agent Pathfinding (MAPF) methods from theoretical settings to real-world deployments in complicated situations. We reviewed notable differences between classical MAPF assumptions and the conditions that robotic systems must operate in, with an emphasis on scaling problems, kinodynamic limitations, and environmental unpredictability. A critical review of recent developments revealed strategies that tackle these issues: robot kinematics-based algorithms, uncertainty-handling methods such as probabilistic modeling or replanning, and decentralization, prioritization, and hierarchical planning strategies that improve MAPF scalability [51][53]. Now let us summarize the key results, point out unresolved issues, and consider possible directions for further study as well as the wider implications of this work.

One major obstacle to the use of conventional MAPF is the disregard for the physical constraints of real-world robots. techniques that include kinodynamic models directly into search procedures, via post-processing, or via hybrid techniques, have opened the door to producing practical and optimal paths in practical situations. Because real-world situations are inherently unpredictable, strong and flexible MAPF solutions are required. Techniques based on partial observability frameworks, probabilistic techniques, and replacement strategies provide means of reasoning in the face of uncertain world situations. These techniques enable agents to respond to shifting surroundings or imprecise perception. The more agents and larger the environment, the more computationally demanding MAPF becomes. Large-scale MAPF challenges need the use of hierarchical approaches, decentralized planning, and prioritizing. By allocating the computing burden, concentrating planning, or permitting efficient search space decomposition, these strategies control complexity.

5.1. Open Challenges and Future Research Directions

Even though individual difficulties have been addressed with some progress, kinodynamics, uncertainty, and scalability elements still need to be addressed together in integrated solutions. Present approaches often tackle these issues separately, which restricts their efficacy in complex real-world situations with several, concurrent difficulties. A lot of existing techniques assume static kinodynamic models, ignoring the possibility that they might vary as a result of robot wear and tear or variations in the payload. Robustness and long-term dependability would be improved by developing MAPF techniques that can adjust to changing robot capabilities while in use. Well-calibrated uncertainty models are a common prerequisite for probabilistic MAPF techniques [52]. The robustness of these methods may be greatly increased by creating methods for acquiring, honing, and measuring uncertainty from actual data. Coordination, negotiating, and guaranteeing the quality of the global solution become more difficult when using decentralized MAPF, even if it reduces computational bottlenecks. Research on bargaining tactics, communication protocol optimization, and ensuring equity in multi-agent prioritizing is still ongoing. Scalable MAPF that integrates learning-based techniques is a new field. More research might focus on developing effective heuristics for search-based methods, figuring out coordination in decentralized systems, or developing uncertainty models from practical application.

5.2. Potential for Broader Real-World Impact

Achieving success in addressing these problems would unleash the potential for revolutionary applications of Multi-Agent Path Finding (MAPF), transforming several industries such as Logistics and Warehousing. The optimized coordination of extensive fleets of warehouse robots has the potential to significantly enhance the efficiency, throughput, and scalability of warehouses and distribution centers. Self-driving vehicles, The secure and effective synchronization of independent automobiles, automated delivery vehicles, or groups of flying drones has the potential to transform the future of transportation, logistics, and city mobility. The integration of Human-Robot cooperation, together with developments in Multi-Agent Path Finding (MAPF) and job allocation approaches, may facilitate efficient cooperation between people and robots in shared workplaces [53]. This can lead to the development of innovative models for industrial assembly lines and service robotics. This study assessment emphasizes the substantial progress achieved in the development of practical Multi-Agent Path Finding (MAPF) and highlights the promising and influential areas of research that are still to be explored. Ongoing research in this domain has the potential to uncover a future where cooperative robotic systems execute intricate tasks with high efficiency, safety, and resilience in dynamic and unpredictable real-world settings.

6. Conclusion

This research study has explored the core difficulties associated with implementing Multi-Agent Pathfinding (MAPF) algorithms in practical settings. Classical Multi-Agent Path Finding (MAPF) is built upon a solid theoretical foundation. However, its assumptions of holonomic actors, perfect information, and unchanging surroundings typically deviate significantly from the limits and uncertainties faced in actual circumstances. The examination emphasized the intricacies brought about by robot kinematics, the have to strategize amidst fluctuating degrees of uncertainty, and the difficulties in scaling up to solve real-world problems of significant magnitude. Additionally, a comprehensive review was conducted on the latest progress in Multi-Agent Path Finding (MAPF), which encompasses the integration of kinodynamic restrictions, strategies for managing uncertainty, and the creation of scalable algorithms using decentralized, prioritized, and hierarchical methods. This study highlights the crucial significance of connecting theoretical MAPF models with the complexities of real-world implementation. To fully harness the capabilities of Multi-Agent Path Finding (MAPF) in applications like warehouse robots, autonomous vehicle coordination, and swarm systems, it is crucial to tackle kinodynamic restrictions, navigate effectively in unpredictable settings, and provide scalable solutions.

References

1. Stern, R. (2019). Multi-agent path finding–an overview. Artificial Intelligence: 5th RAAI Summer School, Dolgoprudny, Russia, July 4–7, 2019, Tutorial Lectures, 96-115.
2. Semiz, F., & Polat, F. (2021). Incremental multi-agent path finding. Future Generation Computer Systems, 116, 220-233.

3. Bai, Y. (2024). Synergistic Strategies in Multi-Robot Systems: Exploring Task Assignment and Multi-Agent Pathfinding (Doctoral dissertation, Luleå tekniska universitet).
4. Skrynnik, A., Andreychuk, A., Nesterova, M., Yakovlev, K., & Panov, A. (2024, March). Learn to follow: Decentralized lifelong multi-agent pathfinding via planning and learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 16, pp. 17541-17549).
5. Hudziak, M., Pozniak-Koszalka, I., Koszalka, L., & Kasprzak, A. (2017). Multi-agent pathfinding in the crowded environment with obstacles: algorithms and experimentation system. Journal of Intelligent & Fuzzy Systems, 32(2), 1561-1573.
6. Zhang, Y., Jiang, H., Bhatt, V., Nikolaidis, S., & Li, J. (2024). Guidance Graph Optimization for Lifelong Multi-Agent Path Finding. arXiv preprint arXiv:2402.01446.
7. Singh, K. Sequential Decision Making using Neurosymbolic AI.
8. Singh, K. Efficient Statistics usage in Biological Research Analysis Techniques of Data and Tools.
9. Tan, W. J., Tang, X., & Cai, W. (2024, March). Robust Multi-Agent Pathfinding with Continuous Time. In 34th International Conference on Automated Planning and Scheduling.
10. Surynek, P., Li, J., Zhang, H., Satish Kumar, T. K., & Koenig, S. (2021). Mutex propagation for SAT-based multi-agent path finding. In PRIMA 2020: Principles and Practice of Multi-Agent Systems: 23rd International Conference, Nagoya, Japan, November 18–20, 2020, Proceedings 23 (pp. 248-258). Springer International Publishing.
11. Merschformann, M., Xie, L., & Erdmann, D. (2018). Multi-agent path finding with kinematic constraints for robotic mobile fulfillment systems.
12. Salerno, M., E-Martín, Y., Fuentetaja, R., Gragera, A., Pozanco, A., & Borrajo, D. (2021). Train route planning as a multi-agent path finding problem. In Advances in Artificial Intelligence: 19th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2020/2021, Málaga, Spain, September 22–24, 2021, Proceedings 19 (pp. 237-246). Springer International Publishing.
13. Kent, T., Richards, A., & Johnson, A. (2022). Homogeneous agent behaviours for the multi-agent simultaneous searching and routing problem. Drones, 6(2), 51.
14. Selvek, R., & Surynek, P. (2020). Towards Smart Behavior of Agents in Evacuation Planning Based on Local Cooperative Path Finding. In Knowledge Discovery, Knowledge Engineering and Knowledge Management: 11th International Joint Conference, IC3K 2019, Vienna, Austria, September 17-19, 2019, Revised Selected Papers 11 (pp. 302-321). Springer International Publishing.
15. Ali, Z. A., & Yakovlev, K. (2021). Prioritized SIPP for multi-agent path finding with kinematic constraints. In Interactive Collaborative Robotics: 6th International Conference, ICR 2021, St. Petersburg, Russia, September 27–30, 2021, Proceedings 6 (pp. 1-13). Springer International Publishing.
16. Maoudj, A., & Christensen, A. L. (2023). Improved decentralized cooperative multi-agent path finding for robots with limited communication. Swarm Intelligence, 1-19.
17. Chen, W., Wang, Z., Li, J., Koenig, S., & Dilkina, B. (2024). No Panacea in Planning: Algorithm Selection for Suboptimal Multi-Agent Path Finding. arXiv preprint arXiv:2404.03554.
18. Wang, F., Zhang, H., Koenig, S., & Li, J. (2024, March). Efficient Approximate Search for Multi-Objective Multi-Agent Path Finding. In 34th International Conference on Automated Planning and Scheduling.
19. Tang, Y., Yu, Z., Zheng, Y., Kumar, T. K., Li, J., & Koenig, S. (2024). Caching-Augmented Lifelong Multi-Agent Path Finding. arXiv preprint arXiv:2403.13421.
20. Chen, W., Koenig, S., & Dilkina, B. (2024). Why solving multi-agent path finding with large language model has not succeeded yet. arXiv preprint arXiv:2401.03630.
21. Chung, J., Fayyad, J., Younes, Y. A., & Najjaran, H. (2024). Learning team-based navigation: a review of deep reinforcement learning techniques for multi-agent pathfinding. Artificial

Intelligence Review, 57(2), 41.
22. Surynek, P. (2020, February). Continuous Multi-agent Path Finding via Satisfiability Modulo Theories (SMT). In International Conference on Agents and Artificial Intelligence (pp. 399-420). Cham: Springer International Publishing.
23. Zhao, X., Yang, R., Zhong, L., & Hou, Z. (2024). Multi-UAV Path Planning and Following Based on Multi-Agent Reinforcement Learning. Drones, 8(1), 18.
24. Gao, J., Li, Y., Ye, Z., & Wu, X. (2024). PCE: Multi-Agent Path Finding via Priority-Aware Communication & Experience Learning. IEEE Transactions on Intelligent Vehicles.
25. Tang, H., Berto, F., & Park, J. (2024). Ensembling Prioritized Hybrid Policies for Multi-agent Pathfinding. arXiv preprint arXiv:2403.07559.
26. Chen, W., Koenig, S., & Dilkina, B. (2024). MARL-LNS: Cooperative Multi-agent Reinforcement Learning via Large Neighborhoods Search. arXiv preprint arXiv:2404.03101.
27. Zhang, Y., Deekshith, U., Wang, J., & Boedecker, J. (2024, March). LCPPO: An Efficient Multi-agent Reinforcement Learning Algorithm on Complex Railway Network. In 34th International Conference on Automated Planning and Scheduling.
28. Phan, T., Driscoll, J., Romberg, J., & Koenig, S. (2024). Confidence-Based Curriculum Learning for Multi-Agent Path Finding. arXiv preprint arXiv:2401.05860.
29. Zhang, Y., Chen, Z., Harabor, D., Le Bodic, P., & Stuckey, P. J. (2024, March). Planning and Execution in Multi-Agent Path Finding: Models and Algorithms. In 34th International Conference on Automated Planning and Scheduling.
30. Hanou, I., Thomas, D. W., Ruml, W., & de Weerdt, M. (2024, March). Replanning in Advance for Instant Delay Recovery in Multi-Agent Applications: Rerouting Trains in a Railway Hub. In 34th International Conference on Automated Planning and Scheduling.
31. Felner, A., Stern, R., Shimony, S., Boyarski, E., Goldenberg, M., Sharon, G., ... & Surynek, P. (2017). Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In Proceedings of the International Symposium on Combinatorial Search (Vol. 8, No. 1, pp. 29-37).
32. Phan, T., Huang, T., Dilkina, B., & Koenig, S. (2024, March). Adaptive Anytime Multi-Agent Path Finding Using Bandit-Based Large Neighborhood Search. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 16, pp. 17514-17522).
33. Wang, Q., Veerapaneni, R., Wu, Y., Li, J., & Likhachev, M. (2024, March). MAPF in 3D Warehouses: Dataset and Analysis. In 34th International Conference on Automated Planning and Scheduling.
34. Wang, Q., Veerapaneni, R., Wu, Y., Li, J., & Likhachev, M. (2024, March). MAPF in 3D Warehouses: Dataset and Analysis. In 34th International Conference on Automated Planning and Scheduling.
35. Shi, R., Steenkiste, P., & Veloso, M. M. (2019). SC-M*: A multi-agent path planning algorithm with soft-collision constraint on allocation of common resources. Applied Sciences, 9(19), 4037.
36. Yamauchi, T., Miyashita, Y., & Sugawara, T. (2021, June). Path and action planning in non-uniform environments for multi-agent pickup and delivery tasks. In European Conference on Multi-Agent Systems (pp. 37-54). Cham: Springer International Publishing.
37. Queiroz, A. C. L., Bernardino, H. S., Vieira, A. B., & Barbosa, H. J. (2020, October). Solving multi-agent pickup and delivery problems using a genetic algorithm. In Brazilian Conference on Intelligent Systems (pp. 140-153). Cham: Springer International Publishing.
38. Kuznetsov, A. V., Schumann, A., & Rataj, M. (2024). Continuous optimisation problem and game theory for multi-agent pathfinding. International Journal of Game Theory, 53(1), 1-41.
39. Chudý, J., Popov, N., & Surynek, P. (2020, November). Multi-agent Path Finding and Acting with Small Reflex-Based Mobile Robots. In International Conference on Robotics, Computer Vision and Intelligent Systems (pp. 51-75). Cham: Springer International Publishing.
40. Eran, C., Keskin, M. O., Cantürk, F., & Aydoğan, R. (2021, June). A decentralized token-based

negotiation approach for multi-agent path finding. In European Conference on Multi-Agent Systems (pp. 264-280). Cham: Springer International Publishing.
41. Liu, Y., Han, D., Wang, L., & Xu, C. Z. (2021). HGHA: task allocation and path planning for warehouse agents. Assembly Automation, 41(2), 165-173.
42. Yan, Z., & Wu, C. (2023, October). Neural Neighborhood Search for Multi-agent Path Finding. In The Twelfth International Conference on Learning Representations.
43. Mai, S., Benecke, T., & Mostaghim, S. (2023, March). MACO: A Real-World Inspired Benchmark for Multi-objective Evolutionary Algorithms. In International Conference on Evolutionary Multi-Criterion Optimization (pp. 305-318). Cham: Springer Nature Switzerland.
44. Ma, J., & Lian, D. (2022, April). Attention-cooperated reinforcement learning for multi-agent path planning. In International Conference on Database Systems for Advanced Applications (pp. 272-290). Cham: Springer International Publishing.
45. Jia, Y., Song, Y., Xiong, B., Cheng, J., Zhang, W., Yang, S. X., & Kwong, S. (2024). Hierarchical Perception-Improving for Decentralized Multi-Robot Motion Planning in Complex Scenarios. IEEE Transactions on Intelligent Transportation Systems.
46. Yu, C., Yang, X., Gao, J., Yang, H., Wang, Y., & Wu, Y. (2022, October). Learning efficient multi-agent cooperative visual exploration. In European Conference on Computer Vision (pp. 497-515). Cham: Springer Nature Switzerland.
47. Kiadi, M., Villar, J. R., & Tan, Q. (2021). Synthesized a* multi-robot path planning in an indoor smart lab using distributed cloud computing. In 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020) 15 (pp. 580-589). Springer International Publishing.
48. Liu, Q., Gao, J., Zhu, D., Chen, P., Guo, J., & Li, Y. Multi-Agent Target Assignment and Path Finding for Intelligent Warehouse: A Cooperative Multi-Agent Deep Reinforcement Learning Perspective.
49. Brandao, M., Mansouri, M., Mohammed, A., Luff, P., & Coles, A. (2022). Explainability in multi-agent path/motion planning.
50. Surynek, P. (2020, September). Multi-agent path finding modulo theory with continuous movements and the sum of costs objective. In German Conference on Artificial Intelligence (Künstliche Intelligenz) (pp. 219-232). Cham: Springer International Publishing.
51. Garg, K., Zhang, S., So, O., Dawson, C., & Fan, C. (2024). Learning safe control for multi-robot systems: Methods, verification, and open challenges. Annual Reviews in Control, 57, 100948.
52. Pianpak, P., Li, J., & Son, T. C. (2023, May). Load Balancing in Distributed Multi-Agent Path Finder (DMAPF). In International Workshop on Engineering Multi-Agent Systems (pp. 130-147). Cham: Springer Nature Switzerland.
53. Surynek, P. (2020, September). Logic-based multi-agent path finding with continuous movements and the sum of costs objective. In Russian Conference on Artificial Intelligence (pp. 85-99). Cham: Springer International Publishing.
54. Ye, X., Deng, Z., Shi, Y., & Shen, W. (2023). Toward energy-efficient routing of multiple AGVs with multi-agent reinforcement learning. Sensors, 23(12), 5615.