

The Cost of Classical Multi-Agent Path Finding

Alvin Combrink
Chalmers University of Technology
Gothenburg, Sweden
combrink@chalmers.se

Sabino Francesco Roselli
Chalmers University of Technology
Gothenburg, Sweden
rsabino@chalmers.se

Martin Fabian
Chalmers University of Technology
Gothenburg, Sweden
fabian@chalmers.se

Abstract

Multi-Agent Path Finding (MAPF) is the problem of planning conflict-free paths for multiple agents in a shared space, each from its start to its goal. Classical MAPF has been the dominant formulation for many years, with its assumptions of discrete time and graph-based conflicts presumably easing the search for solutions. These assumptions limit the physical environments and agents for which a solution is truly collision-free, and also place an upper bound on solution quality that no algorithmic improvements can lift. This work investigates how much solution quality, and in what contexts, the classical MAPF formulation forfeits. Continuous-time MAPF (MAPF_R) relaxes these assumptions, making it a natural counter-formulation to compare against across various agent counts and sizes, and graph connectedness, topologies, and resolutions. We find that continuous time and agent shape consideration are worth relatively little on their own; their value comes from enabling an expanded range of move actions, on average improving solution quality by at least 5% on narrow and constrained maps and 17% on maps with open spaces. In some cases, the improvements exceed 20%. Doubling the map resolution with classical MAPF recovers less than 3%, meaning that little of what is forfeited can be bought back through more compute. This work therefore provides insight on when classical MAPF is a reasonable simplification, and when MAPF_R unlocks significantly higher-quality solutions.

Keywords

Classical MAPF, Continuous-time MAPF, Optimality

1 Introduction

Multi-Agent Path Finding (MAPF) is the problem of planning agent movements on a graph, to get them from where they are to where they need to be without conflicting with each other. This is often paired with an objective function defining a solution’s quality by way of a *cost*, with optimal solutions minimizing the cost. The applications are many, including warehouses and factories [6, 25, 38, 42], video games [12, 26, 35], road traffic [11, 22], and airport management [18, 27]. The overwhelming majority of MAPF research adopts the *classical MAPF* formulation [36], for which decades of sustained research has enabled algorithms to find optimal solutions for the low hundreds of agents [5, 19, 21, 34], and suboptimal [24, 46], bounded-suboptimal [4, 23], and anytime algorithms [17, 28, 29] to the thousands. Scalability is bounded in principle by NP-hardness [1, 44], yet the proliferation of increasingly scalable algorithms suggests we have not yet reached these limits. Solution quality, on the other hand, is bounded absolutely by the chosen problem formulation; no amount of algorithmic progress can produce a higher-quality solution while remaining in the formulation’s solution space. The dominance of classical MAPF is historically well-founded since the assumptions it makes supposedly ease the search, and for much of

the MAPF field’s history, more faithful formulations were simply intractable at useful scales. Simply put, a solution to a simplified model was better than no solution at all.

In the classical formulation, time is discrete, graphs are limited to unit-length edges, and agents conflict when simultaneously occupying the same vertex or traversing the same edge at the same time [36]. If agents model real robots in a shared physical space, this formulation restricts how and to what resolution the environment can be modelled, leaves agent shapes implicitly bounded in size to the graph, and incurs artificial delays that consequently inflate the solution cost (we elaborate on this in Section 3). *Continuous-time MAPF* (MAPF_R) [3] is perhaps the most similar formulation to drop these assumptions, such that agents can traverse edges at any real-valued time, these traversals or waits at vertices taking any positive real-valued duration, and conflicts considering the agents’ shapes. MAPF_R can therefore model a wider range of physical systems, including agent movements for which classical solutions cannot guarantee collision-freeness. MAPF_R does not lift all limiting assumptions, such as agents only moving along a discrete set of trajectories instead of in the entire continuous space; we do not investigate the cost of those assumptions here.

Classical MAPF nevertheless remains the wide-spread formulation. MAPF_R on the other hand has received relatively little attention. However, recent advances [2, 3, 7–9, 15] pushing MAPF_R to hundreds of agents demonstrate that MAPF_R scalability also responds to research effort. This makes the following question increasingly relevant:

What solution quality does the classical MAPF formulation forfeit relative to MAPF_R, and in what contexts is this forfeit substantial?

Existing work only partially answers this question. Using the MAPF_R algorithm CCBS [3] and variants of ICTS [39], solution quality improves by 5–25% on average when adding non-cardinal edges (such as diagonals) to the map. Any-Angle MAPF [10] maximises these non-cardinal edges, for which Any-Angle CCBS [43] provides an illustrative example of such a solution having ~22% lower cost compared to only using cardinal edges. CE-CBS [14], a discrete-time inexact method addressing the related problem of continuous *space* rather than continuous *time*, reports solution quality increasing as agent radius decreases. Finally, [45] compares the exact classical MAPF algorithm CBS [34] with several inexact MAPF_R methods, finding lower-cost MAPF_R solutions on only some problems.

These existing works do not fully address the question: ICTS [39] relies on time discretization and reports results for only a set number of agents; CCBS is a suboptimal [16, 41] (although often near-optimal [9]) algorithm and reports only aggregated results; and Any-Angle CCBS’s correctness guarantees build on CCBS’s without, in our opinion, delivering sufficient theoretical proof that it

does not inherit CCBS’s inexactness. A suboptimal MAPF_R algorithm improving on the optimal classical MAPF solution still does not reveal the true magnitude of the gap, and since a classical MAPF solution is a valid MAPF_R solution¹, not improving on the optimal classical solution says only something about the algorithm that was used. Therefore, these existing studies work only with suboptimal MAPF_R solutions, adjacent problems, or without direct and comprehensive comparisons with the classical formulation.

Recent work provides an exact MAPF_R algorithm, OC-CBS [9]. Our preliminary comparisons between OC-CBS and CBS, however, revealed OC-CBS’s relatively poor scalability to impede a comparison at sufficiently high agent counts within a reasonable runtime. AOC-CBS [8] is an anytime-optimal MAPF_R algorithm, building on OC-CBS, which early on provides an incumbent solution and progressively refines it until eventually converging on an optimal solution. Although AOC-CBS is comparable to OC-CBS for finding optimal solutions, it importantly provides bounds on the optimal cost, thereby enabling a comparison between the classical and MAPF_R formulations at higher agent counts without necessarily converging.

This work contributes the following. First, we give a direct comparison between the optimal cost of classical MAPF and MAPF_R . Using AOC-CBS’s bounds rather than requiring an optimal MAPF_R solution allows the comparison to reach higher agent counts while still keeping the comparison on the formulations rather than on the algorithms. Second, we quantify this cost difference over discrete versus continuous time, agent size, and map connectedness, on four maps ranging from open spaces to narrow corridors. We also test whether refining the map resolution allows classical MAPF to recover the loss. Third, we identify the contexts in which classical MAPF remains an appropriate simplification, and those in which the forfeited solution quality motivates further MAPF_R scalability efforts. Our repository² provides the full results and settings.

2 Background

We formally introduce the classical MAPF and MAPF_R formulations, as well as their respective algorithms CBS and AOC-CBS used here.

2.1 Classical MAPF

The classical MAPF formulation [36] is posed on an undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with vertices $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each agent $a \in \mathcal{A}$ is initially at a unique start $s_a \in \mathcal{V}$ and must reach a unique goal $g_a \in \mathcal{V}$. Time is discrete; at every timestep, each agent either waits at its current vertex for another timestep (wait action), or moves along an adjacent edge to arrive at a neighbouring vertex at the next timestep (move action). Two agents conflict if they are located at the same vertex or traverse the same edge at the same timestep. A solution consists of one plan for each agent $a \in \mathcal{A}$, with a ’s plan specifying a sequence of actions that start at s_a , end at g_a , and do not conflict with any other plans. There exist many ways to rank solutions. Here, we use the *Sum-of-Costs* (SOC): the cost of a plan is its duration (timestep of the last move action); SOC is the sum over all plan costs, which an optimal solution minimises.

¹Assuming both problems use the same map and that the classical solution respects the requirements for conflict-freeness in the real system (see Section 3).

²<https://github.com/Adcombrink/classical-vs-continuous-mapf>

2.2 Continuous-time MAPF

The MAPF_R formulation is originally introduced in [3] and further generalised in [8]. For brevity, we present a specified version in 2D space with homogeneous and circular agents.

Compared to the classical formulation, time in MAPF_R is no longer discrete: move actions can happen at any real-valued time and take any positive time to complete, and agents can wait at vertices for any positive duration. Each vertex $v \in \mathcal{V}$ maps to a specific position in $\mathbb{R}^2$ where an agent waiting at v is located. Similarly, an edge $e \in \mathcal{E}$ defines a trajectory $\mathbb{R} \mapsto \mathbb{R}^2$ mapping from time to a position where an agent traversing e is located. Conflicts now consider the agents’ shapes, with two agents conflicting if their shapes overlap. Furthermore, since conflicts are defined by the agents’ positions and shapes instead of their location on the graph, changing the shapes also changes what an optimal solution is. For instance, multiple agents can traverse the same edge simultaneously, separated by just enough space to not conflict.

2.3 Conflict-Based Search

CBS [34] is a classical MAPF algorithm that searches on two levels to find an optimal solution. A high-level search is done in a binary constraint tree (CT): every CT node represents a candidate solution and a set of constraints; every plan in a node’s candidate solution is found using a low-level planner and has minimum duration while satisfying the node’s constraint set. The CT initially contains only the root with an empty constraint set, with every agent plan representing the shortest path without considering conflicts with other agents. At every iteration of the search, the CT leaf node with the minimum candidate solution cost is selected for expansion. If this node’s candidate solution is a solution, then it is returned as the optimal. Otherwise, the node is expanded by spawning two child nodes that inherit the current node’s constraint set, selecting a conflict, creating one constraint for each agent involved in the conflict, and adding each constraint to a respective child node. The candidate solutions at these new child nodes are replanned by the low-level planner to satisfy the expanded constraint set. This method is guaranteed to produce an optimal solution if one exists.

2.4 Anytime-Optimal Continuous-time CBS

AOC-CBS [8] is an anytime-optimal MAPF_R algorithm that early on produces an incumbent solution, and over the algorithm’s runtime improves the solution to eventually return the optimal, all while providing bounds on the optimal cost. It is based on OC-CBS which in turn is based on the continuous-time version of CBS, CCBS [3]. AOC-CBS’s search is therefore similar in structure to CBS, consisting of a high-level search in the CT and a low-level path planner. The main difference with CBS is that the constraints in AOC-CBS are posed in continuous time, and the low-level path planner uses SIPP [31] to work in continuous time. This search forms the lower bound on the optimal cost; just like CBS, it searches optimistically from the bottom up, starting without any conflict-avoiding constraints and progressively adding them until a conflict-free solution is produced. Since constraints are only ever added during expansions, the lowest-cost candidate solution over all CT leaf nodes provides a lower bound on the optimal cost. Unlike its predecessors, AOC-CBS runs a portfolio of policies that select and expand CT

nodes in parallel, thereby using multiple CPU processors. Each policy also applies a configurable set of repair functions that search for a potentially suboptimal incumbent solution. The best-found incumbent is made available even if the optimal has yet to be found, while also forming an upper bound on the optimal cost.

3 The Cost of the Classical MAPF Assumptions

This section discusses the implications of the classical MAPF assumptions and qualitatively motivates why they are believed to worsen solution quality. The discussion is set in the context of agents representing physical robots in a shared space, with conflicts representing collisions. The assumptions discussed here are that time is discrete and conflicts are defined only by the agents' locations on the graph and not by their spatial positions or shapes. From these assumptions come interconnected and limiting implications that concern both the validity (conflict-freeness) and the quality of a solution when applied to the real system.

Synchronised timesteps induce artificial delays. Consider a graph with variable-length edges. To maintain synchronicity among agents (as discrete time demands), agent 1 traversing a short edge must be delayed to match agent 2 traversing a longer edge. Whether this delay on agent 1 comes via waiting somewhere or by reducing its speed, the cost increases nonetheless. How such a delay is implemented affects the real agents' spatial positions and therefore requires care to preserve solution validity. As clarified in the next points, using only unit-length edges does not completely avoid these issues; artificial delays will still exist and solution validity depends on more than just edge lengths.

Graph-based conflicts induce artificial delays. In the common square grid map layout with unit-length edges and circular agents [37], consider two agents with radius r , where agent 1 moves to agent 2's vertex while at the same time agent 2 moves away. If agent 2 moves along an edge perpendicular to agent 1's direction, then agent 1 could start moving $1 - 2\sqrt{2}r$ earlier than agent 2 without colliding. Taking this to the limit with agent 1 leaving at the same time as agent 2 gives the maximum radius under the classical formulation: $1 - 2\sqrt{2}r = 0 \Leftrightarrow r = \sqrt{2}/4$. On the other hand, if agent 2 moves along the edge parallel to agent 1's direction, then agent 1 can begin its traversal $1 - 2r$ earlier than agent 2. Thus, in this parallel case, even with the classical formulation's limit $r = \sqrt{2}/4$, agent 1 could move $1 - 2r \approx 0.29$ time units earlier. The smaller r is, the earlier agent 1 can traverse its edge relative to agent 2. The artificial delays induced by classical MAPF not considering agent shape therefore grow as the agents get smaller.

Graph-based conflicts limit the graph resolution. With discrete time, unit-length edges, and circular agents, agent radius must satisfy $r \leq \sin(\alpha/2)/2$ units, where α is the smallest angle between any two edges in the graph. In the square grid case, $\alpha = 90^\circ \Rightarrow r \leq \sqrt{2}/4$. If the agents are any larger than this limit then the classical solution is not guaranteed to be valid. This practically means that the resolution at which the graph represents the physical environment is upper-bounded by the size of the agents. A coarser graph permits only coarser movements, leaving less freedom in how the agents are coordinated. In some cases, the size of existing agents and the required graph resolution may outright prohibit the use of classical MAPF.

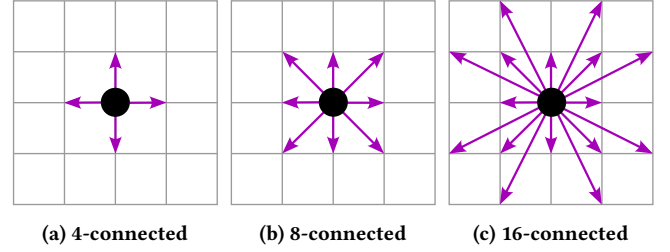


Figure 1: Three levels of map connectedness. 4-connectedness includes only cardinal edges, 8-connectedness adds diagonals, and 16-connectedness adds knight's moves.

Graph-based conflicts inhibit spatially overlapping edges.

Since conflicts are defined entirely by agent locations on the graph, no two edges can spatially overlap without risking invalidity. This prohibits, e.g., the inclusion of both diagonals of a cell: since two agents traversing each respective diagonal at the same time will be at the centre at exactly the same time, a collision occurs regardless of their shape. Classical MAPF does not consider this a conflict since the agents are on different edges. One workaround could be to only include one of the diagonals, thereby including at least some of these shorter moves. This together with time being discrete, however, leads to the same issue of synchronicity-induced delays as discussed in the first point of this section. It would also reduce the maximum allowed agent radius for validity, due to the smaller angle between edges.

These limitations of classical MAPF surely reduce its achievable solution quality. After all, a diagonal between two opposing corners of a cell is $\sim 29\%$ shorter than the Manhattan distance in 2D, and $\sim 42\%$ shorter in 3D. Fig. 1 shows three levels of connectedness (known as 2^k -neighbourhoods [33]), going from only cardinal edges (4), to adding diagonals (8), to adding knight's moves (16). The improvements to cost with increasing connectedness are however diminishing, and is bounded by the cost of each agent travelling the Euclidean distance from their respective start to goal vertices. For instance, going from 4 to 8-connectedness shortens a knight's move by 19.5%, but going from 8 to 16-connectedness shortens the same move by only an additional 7.4%. Nonetheless, the potential in cost reduction from even only adding diagonals is significant, which classical MAPF forfeits.

Classical MAPF requires unit-length edges, but nothing limits this to square grids. A triangular lattice (see Fig. 5a) is, so to speak, 6-connected since there are six different outgoing directions from a vertex, possibly allowing agents to move more directly toward their goal vertices than with 4-connectedness (Fig. 5b). This representation must still respect validity limits on agent size relative to the edge length, which is even more constrained for triangles than squares due to the smaller angle between edges. We elaborate further on this in Section 6, though we only investigate square lattices in this study.

3.1 Can Map Resolution Recover the Loss?

Considering that classical MAPF solution-validity upper-bounds the graph resolution based on agent size, a natural question that

may arise for systems with small agents is: *If agents are small, can classical MAPF achieve the same solution quality as MAPF_R by refining the map with smaller edges?* Our main experiments in Section 5 keep the map static while reducing the size of the agents, which does not test this. In practice, if the agents are small then the map can be made finer until the radius of the agents becomes $\sqrt{2}/4$ of an edge length. Using a fine map can only improve the cost, since a solution on a coarse grid is also a solution on a fine grid (assuming the coarse edges are split evenly into fine edges). We argue that the artificial delays induced by agent synchronicity and not considering agent shapes can partly be mitigated by using finer maps, up to the point when the graph cannot be made finer without risking solution validity. However, no amount of refinement can achieve the gains from using non-cardinal edges in the map. Manhattan distance is, after all, the same regardless of how fine the grid is.

4 Study Design

The purpose of this study is to compare the SOC of optimal classical MAPF and MAPF_R solutions over (i) discrete and continuous time, (ii) various agent sizes, and (iii) various levels of graph connectedness. Since the classical formulation only works in discrete time, does not exploit agent shapes, and is not able to handle non-unit-length edges, we only search for the optimal classical MAPF solution on 4-connected maps, implicitly using agents with radius $r = \sqrt{2}/4$, which is then compared with the various MAPF_R optima along (i)–(iii). Comparisons are done on different maps, with the aim of gaining a crude understanding of how graph topology affects the comparison. To this end, we use CBS for obtaining an optimal cost DT_{opt} of a classical MAPF solution, and AOC-CBS for obtaining a lower bound CT_{LB} and an upper bound CT_{UB} on the optimal cost of a MAPF_R solution. We report the bounds Δ_{LB} and Δ_{UB} on the true cost change $\Delta \in [\Delta_{\text{LB}}, \Delta_{\text{UB}}]$ that an optimal MAPF_R solution returns relative to an optimal classical MAPF solution, where

$$\Delta_{\text{LB}} = 100 \cdot \left(\frac{CT_{\text{LB}}}{DT_{\text{opt}}} - 1 \right) \quad \text{and} \quad \Delta_{\text{UB}} = 100 \cdot \min \left(0, \frac{CT_{\text{UB}}}{DT_{\text{opt}}} - 1 \right).$$

Both bounds are multiplied by 100 to be read in percent. Δ_{UB} is clipped to 0 since any classical solution is also a MAPF_R solution; the optimal MAPF_R solution can only improve on the classical solution. Therefore, we already know that the optimal MAPF_R cost is less than or equal to DT_{opt} and therefore that $CT_{\text{UB}} = DT_{\text{opt}}$ is a valid upper bound, hence $\Delta_{\text{UB}} \leq 0\%$. However, our repository does provide the unclipped results to allow practitioners to compare these two specific algorithms (CBS and AOC-CBS) in terms of solution quality and runtime performance, which is otherwise outside the scope of this work.

We report the mean Δ_{LB} and Δ_{UB} , minimum Δ_{LB} , and maximum Δ_{UB} . For a given map and agent count, these statistics are taken over the set of scenarios (see Section 4.2) for which both Δ_{LB} and Δ_{UB} have been found at every combination of agent radii and connectedness. This is to ensure a fair comparison across these values. In practice, AOC-CBS always returns CT_{LB} , and Δ_{LB} is known to be at most 0%, so this set of commonly solved scenarios depends entirely on CBS returning an optimal solution. We therefore also report CBS’s success rate at solving scenarios at each agent count, indicating the amount of data support for the corresponding bounds. The number of agents is increased until the success rate falls below 40%

(reporting only on those above) or until the spread of Δ becomes too large to provide any useful information.

4.1 Evaluation Dimensions

Isolating time from agent size is not entirely possible since continuous time is required to exploit agent shapes, and agent shapes are required for continuous time else the agents can be moved arbitrarily close to each other. For this reason, the closest we get in this work to a pure comparison between discrete and continuous time is on 4-connected graphs with agent radius $r = \sqrt{2}/4$. These tests show to what extent MAPF_R can avoid delays induced by agent movement being synchronised and their shapes not being considered.

To evaluate the impact of agent size on solution quality, we sweep over agent radii $r \in \{\sqrt{2}/16, \sqrt{2}/8, \sqrt{2}/4\}$. The sweep is limited to at most $\sqrt{2}/4$ since any larger would compromise the validity of the classical MAPF solutions, while lower radii would not. As discussed in the previous section, the classical formulation enforcing synchronised actions and using shape-agnostic conflict definitions incurs artificial delays. With smaller agents, these delays grow relatively larger. We therefore aim to understand how large the impact of agent size is.

The impact of graph connectedness on solution quality is evaluated by using the three levels of connectedness shown in Fig. 1: 4-connected (cardinal movements), 8-connected (+ diagonals), and 16-connected (+ knight’s moves). Non-cardinal edges are only added if an agent with $r = \sqrt{2}/4$ can traverse it without intersecting a non-traversable cell; these same edges are used for all agent radii. The addition of more edges yields a richer set of move actions that allow both for shorter paths and a greater coverage of the space for more options when avoiding collisions.

The above dimensions are cross-evaluated to understand their combined effect on solution quality.

4.2 Maps and Scenarios

Our main experiments are done on four maps from the MovingAI benchmark suite [37]: *Empty-32-32*, *Room-64-64-8*, *Warehouse-10-20-10-2-2*, and *Maze-32-32-2*, shown in Fig. 2. These maps provide a crude progression from open spaces to narrow corridors and choke-points. Large open spaces allow for more non-cardinal edges and a larger number of equivalent-cost paths for agents to take. They also lead to more localised conflict avoidance since agents in general only need short delays to avoid other agents. Narrow corridors and tight spaces, on the other hand, do not allow for as many non-cardinal edges, and they also incur higher delays due to larger mutually exclusive areas. For instance, an agent must wait for another agent to clear an entire corridor before it can itself move through it.

We use the 25 *random* scenarios provided by [37] for each map: a scenario defines a list of start-goal pairs; a problem with n agents is constructed by assigning to the agents the first n start-goal pairs in the scenario. An exception is made for *Empty-32-32* which instead uses scaled *Empty-16-16* scenarios, to keep results on these two maps comparable for the experiments described in the next section.

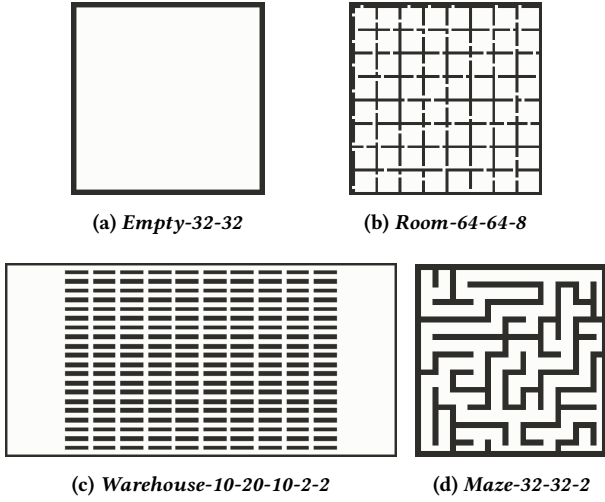


Figure 2: The four MovingAI [37] maps on which our main experiments compare the two formulations. White cells are traversable, black cells are not.

4.3 Leveraging Map Resolution

Based on Section 3.1, we aim to test how much of MAPF_R 's solution quality the classical MAPF formulation can recover by using a finer map resolution. To do so, we compare solution quality on *Empty-32-32* (a 32×32 grid) with *Empty-16-16* (a 16×16 grid); *Empty-32-32* has twice the resolution. Practically, we use agent radius $\sqrt{2}/4$ on *Empty-32-32* and $\sqrt{2}/8$ on *Empty-16-16*, and use *Empty-16-16*'s random scenarios on both maps but with start-goal positions scaled by 2 and solution costs by $1/2$ when applied to *Empty-32-32*. The problems on both maps are therefore equivalent, except that the effective graph resolution is doubled on *Empty-32-32*, allowing for agents to take steps of half the length compared to on *Empty-16-16*. We therefore say that *Empty-32-32* is the finer map, while *Empty-16-16* is the coarser.

4.4 Settings and Hardware

All experiments are run on AMD EPYC 9755 Turin 2.7 GHz 128-core processors, with each run using at most 4-cores and 16 GB of DDR5 memory. Both CBS and AOC-CBS are given a 300 second runtime limit. We use the default CBS settings as provided in the implementation by [20]. For AOC-CBS, we use a four-policy (4 CPU core) gap-cadence portfolio, which prioritises finding solutions by applying repair functions while the optimality gap upper bound is above 5%, and thereafter gradually reduces repair function use to instead focus on raising CT_{LB} . Our repository provides detailed settings for each algorithm.

5 Results

Fig. 3 shows the cost reduction of MAPF_R relative to classical MAPF on each of the maps, connectedness levels, and agent radii. For a given graph corresponding to a specific map and connectedness, each colour corresponds to a specific agent radius, for which there are four lines: the two thicker lines show the mean Δ_{LB} and Δ_{UB}

over scenarios, bounding the shaded area within which the true value of Δ is; the two thinner lines above and below (overlapping in some places) show the maximum Δ_{UB} and minimum Δ_{LB} over scenarios; and the dots show which agent counts were tested. CBS's success rate over scenarios is shown under each subplot.

To summarise the main findings before taking a closer look: on 4-connected maps, Δ_{LB} extends down to at most only a few percent, meaning that the cost reduction of MAPF_R over classical MAPF is relatively small at these tested agent counts. That is, the direct improvement from time being continuous and agent shapes being considered is limited, despite smaller agents allowing for lower-cost solutions (most clearly seen in Fig. 3 at lower agent counts). The results instead show that the value of MAPF_R comes from admitting higher levels of connectedness, which improves the cost change by at least an order of magnitude. The advantages from higher connectedness and smaller agents appear to be additive, each contributing to tighter $[\Delta_{\text{LB}}, \Delta_{\text{UB}}]$ bounds at lower costs. Across the maps, we see the improvements from higher connectedness decreasing as the open spaces become less, which is expected since narrower maps allow for fewer non-cardinal edge connections.

(i) **Discrete versus continuous time:** The 4-connected maps with $r = \sqrt{2}/4$ provide our cleanest comparison between discrete and continuous time, without the effects of smaller agents or higher map connectedness. Keep in mind, however, that agent shapes are still considered, meaning that this is not a pure discrete/continuous time comparison. The results show Δ_{LB} reaching down to at most only a few percent, meaning that the MAPF_R formulation does relatively little to improve the solution quality at these agent counts. The spread of Δ remains relatively high throughout the agent count range, which means that AOC-CBS for these specific cases is unable to narrow the bounds by much within the given runtime.

(ii) **Smaller agents:** Smaller agent radii generally improve both Δ_{LB} and Δ_{UB} across all maps and connectedness levels. The cases at low agent counts where Δ_{UB} for smaller agents is below Δ_{LB} for larger agents empirically confirms that smaller agents allow for higher-quality solutions. Δ_{UB} corresponds to actual MAPF_R solutions found by AOC-CBS; the fact that smaller agent radii decreases Δ_{UB} means that smaller agents may make it easier for AOC-CBS to find solutions; the same may be true for other MAPF_R algorithms. This makes intuitive sense since collisions become less likely when agent sweep-volumes are smaller. However, AOC-CBS finding lower-cost solutions is confounded with the fact that smaller agents permit lower-cost solutions, and so we refrain from drawing any more conclusions regarding the impact of agent size on the ease of finding solutions.

With smaller agents, the spread of Δ is narrower at the lower end of the agent range, revealing a trend of decreasing Δ with more agents. This suggests that MAPF_R pays off with more agents. However, since AOC-CBS is unable to maintain sufficiently narrow bounds as the number of agents increases, we are not able to draw any further conclusions regarding how far this trend of decreasing Δ goes. We leave these questions for future investigations using more scalable MAPF_R algorithms than AOC-CBS.

(iii) **Map connectedness:** Increasing map connectedness has by far the largest impact on solution quality out of all our tested dimensions. Going from 4-connectedness to 16-connectedness on

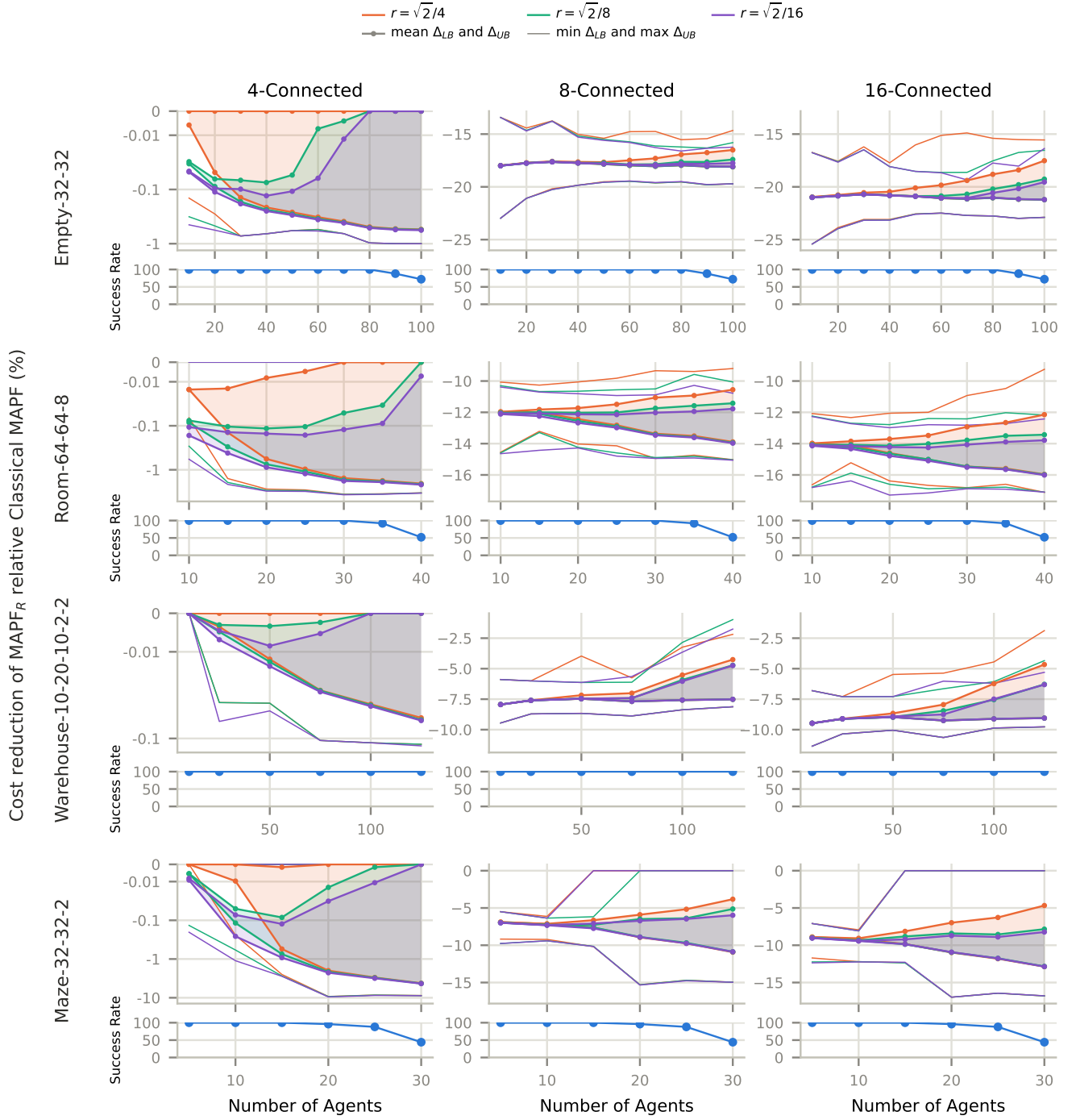


Figure 3: The solution cost reduction of MAPF_R relative to classical MAPF, on four maps (rows) with different connectedness (columns) and agent radius (colours). Each plot shows mean bounds on the cost change, $\Delta \in [\Delta_{LB}, \Delta_{UB}]$, together with the minimum Δ_{LB} and maximum Δ_{UB} , in percentage. The y-axes for the 4-connected graphs are symmetric-logarithmic while the 8- and 16-connected are linear. Each map has its own agent count on the x-axes. CBS’s success rate is shown below each corresponding plot.

Empty-32-32 improves Δ from fractions of a percent to over 17% on average. This is due to the expanded set of move actions shortening paths and offering more alternatives for conflict avoidance. The results show that the largest increase can already be achieved with 8-connectedness, while 16-connectedness provides moderate further improvements. The reduction in cost from higher connectedness tapers off over the maps *Empty-32-32* $\rightarrow$ *Room-64-64-8* $\rightarrow$ *Warehouse-10-20-10-2-2* $\rightarrow$ *Maze-32-32-2*, when qualitatively going from most to least open spaces. However, even on *Maze-32-32-2* with long and narrow corridors, the average improvement is still significant.

5.1 Impact of Higher Map Resolution

Fig. 4 visualises the cost improvement from doubling the map resolution. The MAPF_R 4-connected solutions are clipped to the solution quality of the classical MAPF solutions on the same map, since they are also a solution to the MAPF_R problem. By doubling the map resolution, with classical MAPF or MAPF_R, a cost reduction of less than 3% is achieved for these agent counts. It is not possible to increase the resolution to achieve further improvements without compromising the validity of the classical MAPF solutions; this is therefore the extent of the improvements that are possible through a higher map resolution in this context. These gains are marginal in comparison to increasing the map connectedness, which reduces the cost by more than 16% with 8-connectedness and 21% with 16-connectedness. These results mirror those in Fig. 3: here we show that using a twice as fine map improves costs slightly more than what continuous-time and shape-awareness do on their own, but still relatively little compared to using higher map connectedness. Interestingly, Δ_{LB} is almost identical across resolutions, however, Δ_{UB} on the fine map (most pronounced with 4 and 8-connectedness) is significantly closer to Δ_{LB} . This means that using a finer map does not help AOC-CBS raise CT_{LB} that much, but it seems to help AOC-CBS find lower-cost solutions and thereby lower CT_{UB} .

6 Discussion

The results show that continuous time and shape-awareness do improve solution quality on their own, but only by a few percent at the agent counts we tested. Higher connectedness improves it by at least an order of magnitude more. This is not to say that continuous-time or shape-awareness are not needed, in actuality they are necessary for enabling non-unit-length and spatially overlapping edges. A cardinal edge and a diagonal respectively have lengths 1 and $\sqrt{2}$, which no common timestep divides, so a formulation with synchronised timesteps cannot represent both faithfully. Besides, since classical MAPF defines conflicts by graph location, the two diagonals of a cell cannot both be included without risking the solution’s validity. Adding diagonals therefore requires relaxing both the discrete-time and graph-based conflict assumptions. This is also why, to our knowledge, no classical MAPF algorithm incorporates non-unit-length edges. Therefore, the true cost of the classical MAPF assumptions is mainly paid through ruling out a richer set of move actions.

The experiments over different map topologies show that classical MAPF forfeits a significant part of the achievable solution quality on maps with open spaces, where many non-cardinal edges would otherwise be possible. This loss is more modest on maps

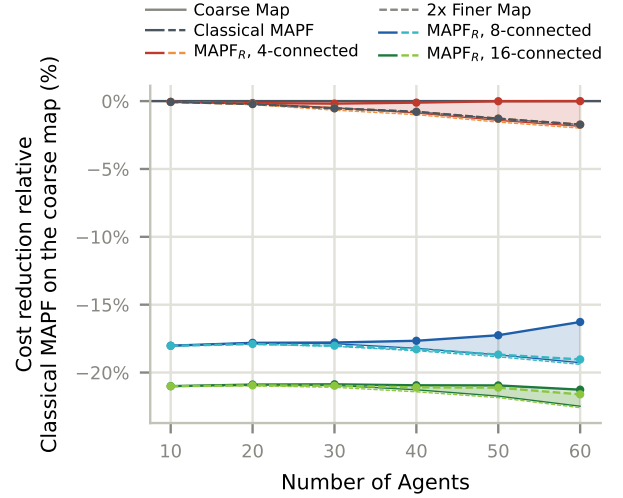


Figure 4: The solution cost reduction (in percent) from using classical MAPF on the fine map, or MAPF_R with various connectedness levels on both the coarse and fine map, compared to the baseline of classical MAPF on the coarse map. For each setting, the dotted line shows Δ_{UB} (with dots showing specifically which agent counts were tested), a corresponding thinner line showing Δ_{LB} , and the shared area showing where the true reduction $\Delta \in [\Delta_{LB}, \Delta_{UB}]$ must be.

with narrow corridors and tight spaces. The classical formulation therefore remains a reasonable simplification on these more constrained maps, especially given how scalable and robust its algorithms have become after decades of research effort. If MAPF_R algorithms achieve the level of scalability that real-world applications require, then the additional solution quality that they unlock becomes practically attainable. Our results also show that much of the solution quality is attained already at 8-connectedness; for this reason research effort may be best spent on these lower connectedness levels.

Furthermore, although MAPF research often uses *square* lattice graphs, classical MAPF does not require that. Higher-quality solutions may be obtained through other lattices, such as the triangular or hexagonal lattices in Fig. 5. We acknowledge the practical difficulty of this, since many real-world settings are built around right angles. Nonetheless, with agent radius limited to $r \leq \sin(\alpha/2)/2$ where α is the smallest angle between adjacent edges in the graph, this gives for triangular lattices $r \leq 0.25$ ($\alpha = 60^\circ$), for square lattices $r \leq 0.35$ ($\alpha = 90^\circ$), and for hexagonal lattices $r \leq 0.43$ ($\alpha = 120^\circ$). Larger permitted agent radii mean that a finer map resolution can be used; although the actual density of vertices is also decided by the lattice’s shape. The shape also determines the connectedness: six for triangles, four for squares, three for hexagons. This presents a trade-off, where more connected lattices force longer edges. However, as our results show, connectedness has a far larger impact on solution quality than map resolution. We leave investigations into the trade-off between different lattices to future work.

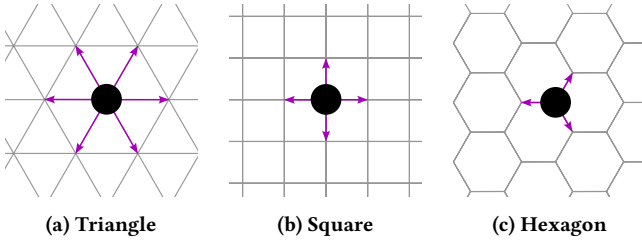


Figure 5: Three types of lattices, with the agent size fixed and the edge lengths set to the minimum that classical MAPF validity permits.

Another interesting direction, based on connectedness being the strongest improver of solution quality, is MAPF with Asynchronous Actions [32, 41]. There, classical MAPF’s discrete-time assumption is relaxed to allow for non-unit-length edges, while graph-based conflicts are still kept. The latter means that agent sizes and edge lengths are still bounded to each other for solution validity, and overlapping edges are still prohibited. However, since it is more similar to the classical MAPF formulation than what MAPF_R is, it is possible that many of the algorithmic developments extending classical MAPF scalability could be transferred to this formulation. Although, for the time being we have yet to see it applied to agent counts of the same magnitude as what classical MAPF has been applied to [32, 40, 41].

AOC-CBS returns valid bounds on the optimal MAPF_R cost, so the direction and validity of our results do not depend on which specific algorithm was used. What does depend on it is the width of these bounds and the agent count that can be reached while keeping these bounds narrow enough to offer insights. This is clearest on the map *Maze-32-32-2* with $r = \sqrt{2}/4$ and 4-connectedness, where at relatively small agent counts the bounds extend from 0% down to -10%, effectively offering very little insight. We suspect the cause of this to be long corridors, which are known to be particularly difficult for CBS-style algorithms. The CBS implementation used here [20] uses specific techniques to address this, however, corresponding techniques have to our knowledge not yet been developed for continuous-time and are therefore not implemented in AOC-CBS. Narrower bands would tell us whether or not Δ keeps decreasing as the agent count grows, thereby answering the question regarding whether the cost reduction from continuous-time and shape-awareness alone eventually become significant or not. Since MAPF_R generalises classical MAPF, which is NP-hard [44], as far as we know there will always be an agent count beyond which the bounds widen; a more scalable algorithm than AOC-CBS only moves that point further out.

This study uses only 2D grid maps. A 3D diagonal is even shorter than a 2D diagonal when compared to the corresponding Manhattan distance. Both classical MAPF and MAPF_R can also be applied to 3D; therefore, comparing these two formulations in 3D space would make for interesting future work. Furthermore, 2D grid maps do not exercise the full representational power of MAPF_R. *Roadmaps*, which need not follow any regular structure, can be adaptively tuned to have a higher resolution in parts of the environment where it is needed, such as at junctions and high-traffic areas in the

environment. This freedom is not present in classical MAPF since map resolution is limited to agent size. Our results should therefore be read as a conservative estimate of what the classical formulation forfeits. Finally, classical MAPF plans for synchronous systems and generally needs post-processing before non-holonomic robots can execute them [13, 30]. MAPF_R supports kinodynamically feasible trajectories directly [8], so the cost gap we measure is likely a lower bound on the gap in execution.

7 Conclusion

Classical MAPF has been the dominant MAPF formulation for many years, with the scalability of modern algorithms enabling the planning of thousands of agents. However, classical MAPF fundamentally assumes discrete time and graph-based conflicts, thereby forfeiting solution quality. This work investigated the extent of this forfeit by comparing optimal classical MAPF solutions to those of Continuous-time MAPF (MAPF_R), a formulation that relaxes several of the classical MAPF assumptions. A comparison was done over discrete and continuous time, various agent sizes, and graph connectedness, resolution, and topology. At our tested agent counts, continuous-time and shape-awareness alone account for marginal improvements; however, they enable an expanded set of move actions that lead to significant improvements in solution quality. Average improvements on narrow and constrained maps are at least 5%, on open maps at least 17%, and in some cases more than 20%. We also find that naively refining the map, which cannot be done unboundedly in classical MAPF without compromising the solution’s validity, only slightly raises solution quality. Classical MAPF therefore cannot recover what its assumptions forgo through more compute. We do however pinpoint contexts where the forfeit is only moderate: maps with narrow corridors and tight spaces, where agents are relatively large compared to the graph edges, and where only cardinal movements are possible. This work therefore quantifies what classical MAPF loses through its assumptions, which had previously only been partially observed or hinted at.

Acknowledgments

Computational resources on the Arrhenius supercomputer were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), funded by the Swedish Research Council.

References

- [1] Artem Agafonov and Konstantin Yakovlev. 2025. *Multi-Agent Path Finding for Large Agents is Intractable*. SAGE Publications, 4856–4863. doi:10.3233/FAIA251395
- [2] Anton Andreychuk, Konstantin Yakovlev, Eli Boyarski, and Roni Stern. 2021. Improving continuous-time conflict based search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11220–11227.
- [3] Anton Andreychuk, Konstantin Yakovlev, Pavel Surynek, Dor Atzmon, and Roni Stern. 2022. Multi-agent pathfinding with continuous time. *Artificial Intelligence* 305 (2022), 103662. doi:10.1016/j.artint.2022.103662
- [4] Max Barer, Guni Sharon, Roni Stern, and Ariel Felner. 2014. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Proceedings of the international symposium on combinatorial Search*, Vol. 5. 19–27.
- [5] Eli Boyarski, Ariel Felner, Roni Stern, Guni Sharon, Oded Betzalel, David Tolpin, and Eyal Shimony. 2015. ICBS: The improved conflict-based search algorithm for multi-agent pathfinding. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 6. 223–225.
- [6] Kyle Brown, Oriana Peltzer, Martin A Sehr, Mac Schwager, and Mykel J Kochenderfer. 2020. Optimal sequential task assignment and path finding for multi-agent

- robotic assembly planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 441–447.
- [7] Alvin Combrink, Sabino Francesco Roselli, and Martin Fabian. 2025. Prioritized Planning for Continuous-time Lifelong Multi-agent Pathfinding. In *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, Vol. 1. 1454–1459. doi:10.1109/CoDIT66093.2025.11321711
 - [8] Alvin Combrink, Sabino Francesco Roselli, and Martin Fabian. 2026. AOC-CBS: Anytime-Optimal Continuous-time Conflict-Based Search for Generalised Multi-Agent Path Finding. arXiv:2608.08175 [cs.MA] <https://arxiv.org/abs/2608.08175>
 - [9] Alvin Combrink, Sabino Francesco Roselli, and Martin Fabian. 2026. Optimal Multi-Agent Path Finding in Continuous Time. arXiv:2508.16410 [cs.MA] <https://arxiv.org/abs/2508.16410>
 - [10] Kenny Daniel, Alex Nash, Sven Koenig, and Ariel Felner. 2010. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research* 39 (2010), 533–579.
 - [11] Christofel Rio Goenawan. 2025. ASTM : Autonomous Smart Traffic Management System Using Artificial Intelligence CNN and LSTM. arXiv:2410.10929 [cs.LG]
 - [12] Daniel Harabor, Ryan Hechenberger, and Thomas Jahn. 2022. Benchmarks for pathfinding search: Iron harvest. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 15. 218–222.
 - [13] Wolfgang Hönl, TK Kumar, Liron Cohen, Hang Ma, Hong Xu, Nora Ayanian, and Sven Koenig. 2016. Multi-agent path finding with kinematic constraints. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 26. 477–485.
 - [14] Kristýna Janovská and Pavel Surynek. 2024. Multi-agent Path Finding in Continuous Environment. arXiv:2409.10680 [cs.MA] <https://arxiv.org/abs/2409.10680>
 - [15] Kazumi Kasaura, Mai Nishimura, and Ryo Yonetani. 2022. Prioritized Safe Interval Path Planning for Multi-Agent Pathfinding With Continuous Time on 2D Roadmaps. *IEEE Robotics and Automation Letters* 7, 4 (2022), 10494–10501. doi:10.1109/LRA.2022.3187265
 - [16] Andy Li, Zhe Chen, Mor Vered, and Daniel Harabor. 2025. Revisiting Conflict Based Search with Continuous-Time. arXiv:2501.07744 [cs.MA]
 - [17] Jiaoyang Li, Zhe Chen, Daniel Harabor, Peter J Stuckey, and Sven Koenig. 2022. MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 10256–10265.
 - [18] Jiaoyang Li, Mimi Gong, Zi Liang, Weizi Liu, Zhongyi Tong, Liangchen Yi, Robert Morris, Corina Pasareanu, and Sven Koenig. 2019. *Departure Scheduling and Taxiway Path Planning under Uncertainty*. 2930. arXiv:<https://arc.aiaa.org/doi/pdf/10.2514/6.2019-2930> doi:10.2514/6.2019-2930
 - [19] Jiaoyang Li, Daniel Harabor, Peter J Stuckey, Ariel Felner, Hang Ma, and Sven Koenig. 2019. Disjoint splitting for multi-agent path finding with conflict-based search. In *Proceedings of the international conference on automated planning and scheduling*, Vol. 29. 279–283.
 - [20] Jiaoyang Li, Daniel Harabor, Peter J. Stuckey, Hang Ma, Graeme Gange, and Sven Koenig. 2021. Pairwise symmetry reasoning for multi-agent path finding search. *Artificial Intelligence* 301 (2021), 103574. doi:10.1016/j.artint.2021.103574
 - [21] Jiaoyang Li, Daniel Harabor, Peter J Stuckey, Hang Ma, and Sven Koenig. 2019. Symmetry-breaking constraints for grid-based multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 6087–6095.
 - [22] Jiaoyang Li, The Anh Hoang, Eugene Lin, Hai L. Vu, and Sven Koenig. 2023. Intersection Coordination with Priority-Based Search for Autonomous Vehicles. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 10 (6 2023), 11578–11585. doi:10.1609/aaai.v37i10.26368
 - [23] Jiaoyang Li, Wheeler Ruml, and Sven Koenig. 2021. EECBS: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 12353–12362.
 - [24] Hang Ma, Daniel Harabor, Peter J Stuckey, Jiaoyang Li, and Sven Koenig. 2019. Searching with consistent prioritization for multi-agent path finding. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7643–7650.
 - [25] Hang Ma and Sven Koenig. 2017. AI buzzwords explained: multi-agent path finding (MAPF). *AI Matters* 3, 3 (Oct. 2017), 15–19. doi:10.1145/3137574.3137579
 - [26] Hang Ma, Jingxing Yang, Liron Cohen, T. K. Kumar, and Sven Koenig. 2017. Feasibility Study: Moving Non-Homogeneous Teams in Congested Video Game Environments. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 13, 1 (Oct. 2017), 270–272. doi:10.1609/aiide.v13i1.12919
 - [27] Robert Morris, Corina S Pasareanu, Kasper Søe Luckow, Waqar Malik, Hang Ma, TK Satish Kumar, and Sven Koenig. 2016. Planning, Scheduling and Monitoring for Airport Surface Operations.. In *AAAI Workshop: Planning for Hybrid Systems*. 608–614.
 - [28] Keisuke Okumura. 2023. Improving LaCAM for scalable eventually optimal multi-agent pathfinding. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (Macao, P.R.China) (IJCAI '23)*. Article 28, 9 pages. doi:10.24963/ijcai.2023/28
 - [29] Keisuke Okumura. 2024. Engineering LaCAM*: Towards Real-Time, Large-Scale, and Near-Optimal Multi-Agent Pathfinding. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
 - [30] Keisuke Okumura, Guang Yang, Zhan Gao, Heedo Woo, and Amanda Prorok. 2026. Concrete multi-agent path planning enabling kinodynamically aggressive maneuvers. *npj Robotics* 4, 1 (14 Mar 2026), 20. doi:10.1038/s44182-026-00083-2
 - [31] Mike Phillips and Maxim Likhachev. 2011. SIPP: Safe interval path planning for dynamic environments. In *2011 IEEE International Conference on Robotics and Automation*. 5628–5635. doi:10.1109/ICRA.2011.5980306
 - [32] Zhongqiang Ren, Sivakumar Rathinam, and Howie Choset. 2021. Loosely synchronized search for multi-agent path finding with asynchronous actions. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 9714–9719.
 - [33] Nicolás Rivera, Carlos Hernández, and Jorge Baier. 2017. Grid pathfinding on the 2k neighborhoods. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.
 - [34] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219 (2015), 40–66.
 - [35] Jamie Snape, Stephen J. Guy, Ming C. Lin, Dinesh Manocha, and Jur Van den Berg. 2012. Reciprocal Collision Avoidance and Multi-Agent Navigation for Video Games. In *MAPF@AAAI*. <https://api.semanticscholar.org/CorpusID:10768014>
 - [36] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, and Roman Bartak. 2019. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the Twelfth International Symposium on Combinatorial Search (SoCS 2019)*, Vol. 10. 151–158.
 - [37] Nathan R. Sturtevant. 2012. Benchmarks for Grid-Based Pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games* 4, 2 (2012), 144–148. doi:10.1109/TCIAIG.2012.2197681
 - [38] Sumanth Varambally, Jiaoyang Li, and Sven Koenig. 2022. Which MAPF model works best for automated warehousing?. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 15. 190–198.
 - [39] Thayne T. Walker, Nathan R. Sturtevant, and Ariel Felner. 2018. Extended Increasing Cost Tree Search for Non-Unit Cost Domains. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 534–540. doi:10.24963/ijcai.2018/74
 - [40] Xuemian Wu, Shizhe Zhao, and Zhongqiang Ren. 2026. BCBS-AA: Bounded Sub-Optimal Conflict-Based Search for Multi-Agent Path Finding with Asynchronous Actions. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 19. 193–201.
 - [41] Xuemian Wu, Shizhe Zhao, and Zhongqiang Ren. 2026. Conflict-Based Search for Multi Agent Path Finding with Asynchronous Actions. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (Paphos, Cyprus) (AAMAS '26)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1174–1182. doi:10.65109/FSCJ9273
 - [42] Peter R Wurman, Raffaello D’Andrea, and Mick Mountz. 2008. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine* 29, 1 (2008), 9–9.
 - [43] Konstantin Yakovlev, Anton Andreychuk, and Roni Stern. 2024. Optimal and bounded suboptimal any-angle multi-agent pathfinding. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7996–8001.
 - [44] Jingjin Yu and Steven LaValle. 2013. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 27. 1443–1449.
 - [45] Chuanlong Zang, Anna Mannucci, Isabelle Barz, Philipp Schillinger, Florian Lier, and Wolfgang Hönl. 2026. GRACE: A Unified 2D Multi-Robot Path Planning Simulator & Benchmark for Grid, Roadmap, And Continuous Environments. arXiv:2603.10858 [cs.RO]
 - [46] Shuyang Zhang, Jiaoyang Li, Taoan Huang, Sven Koenig, and Bistra Dilkina. 2022. Learning a priority ordering for prioritized planning in multi-agent path finding. In *Proceedings of the International Symposium on Combinatorial Search*, Vol. 15. 208–216.