Tracking Progress in Multi-Agent Path Finding

Bojie Shen, Zhe Chen, Muhammad Aamir Cheema
Daniel D. Harabor and Peter J. Stuckey

Faculty of Information Technology, Monash University, Melbourne, Australia
{bojie.shen, zhe.chen, aamir.cheema, daniel.harabor, peter.stuckey}@monash.edu

Abstract

Multi-Agent Path Finding (MAPF) is an important core problem for many new and emerging industrial applications. Many works appear on this topic each year, and a large number of substantial advancements and performance improvements have been reported. Yet measuring overall progress in MAPF is difficult: there are many potential competitors, and the computational burden for comprehensive experimentation is prohibitively large. Moreover, detailed data from past experimentation is usually unavailable. In this work, we introduce a set of methodological and visualisation tools that can help the community establish clear indicators for state-of-the-art MAPF performance and facilitate large-scale comparisons between MAPF solvers. Our objectives are to lower the barriers of entry for new researchers and to further promote the study of MAPF.

Introduction

In recent years, the number of publications on the topic of MAPF has exploded, as industrial interest continues to grow. Many works now appear, across many different venues, and there have been substantial performance improvements. To track progress in the area, the community has developed a set of standardised MAPF benchmarks (Stern et al. 2019), which cover a variety of popular application domains and synthetic/pathological test cases. In total, there are more than 1.5 million standard instances with up to thousands of moving agents per instance. Unfortunately, the computational burden associated with running this benchmark is large, which means that most researchers attempt to solve only a limited subset of instances and then only compare against a limited subset of potential competitors. Consequently, it is not entirely clear where a given MAPF solver sits on the pareto-frontier that currently defines the state-of-the-art. Another related problem is the availability of results data. Although published works include headline results, such as success rates and total problems solved, they typically do not mention which specific problems were solved, which were closed, and where the remaining gaps are. Supplementary data, such as concrete plans and best-known bounds, which can allow other researchers to verify claims and build on established results, are seldom available. Thus, despite notable advancements, and despite the availability of benchmark problem sets, we currently do not have a clear picture of overall progress in MAPF.

In this work, we introduce a new web-based system[1] for the MAPF community to track and validate the results on standardised benchmarks. We then undertake a large set of experiments, with several currently leading optimal and suboptimal solvers (Shen et al. 2023; Li et al. 2021; Lam et al. 2022; Gange, Harabor, and Stuckey 2019; Li et al. 2022; Okumura 2023), in an attempt to map the current pareto-frontier. Finally, We believe that our system can help identify the main strengths of existing research and the remaining challenges in the area. They can also be used to track progress on those challenges over time and help to lower the barrier of entry for new research on the topic of MAPF.

System Demonstration

We introduce our system for tracking progress of different methods on MAPF benchmarks. In general, there are three types of algorithms studied by the research community:

(i) **Optimal Algorithms** focus on finding exact optimal solutions. Such algorithms start from a lower-bound of the optimal solution, and progressively push the lower-bound until they find a feasible solution that is provably optimal.

(ii) **Bounded Suboptimal Algorithms** find the suboptimal solution within theoretical guarantees. These algorithms explore lower-bounds and feasible solutions simultaneously, returning a solution meets certain suboptimality.

(iii) **Unbounded Suboptimal Algorithms** focus on finding feasible solutions. These algorithms find the feasible solution fast, and keep improving it given sufficient time.

Our goal is to design a system that tracks different types of algorithms and their progress together. The critically important feature for us is the ability to handle all types of algorithms. Therefore, we focus on two important results reported by different MAPF algorithms: (a) best (i.e., largest) lower-bound value: we track this value to cover the algorithms in (i) and (ii); and (b) best (i.e., smallest SIC) solution: we record this result to cover the algorithms in (ii)

[1]Our website is accessible at: http://tracker.pathfinding.ai. A demo video giving an overview of the system is also available at: https://youtu.be/qtG6-h4FZxU. A full length manuscript is available at https://arxiv.org/abs/2305.08446.

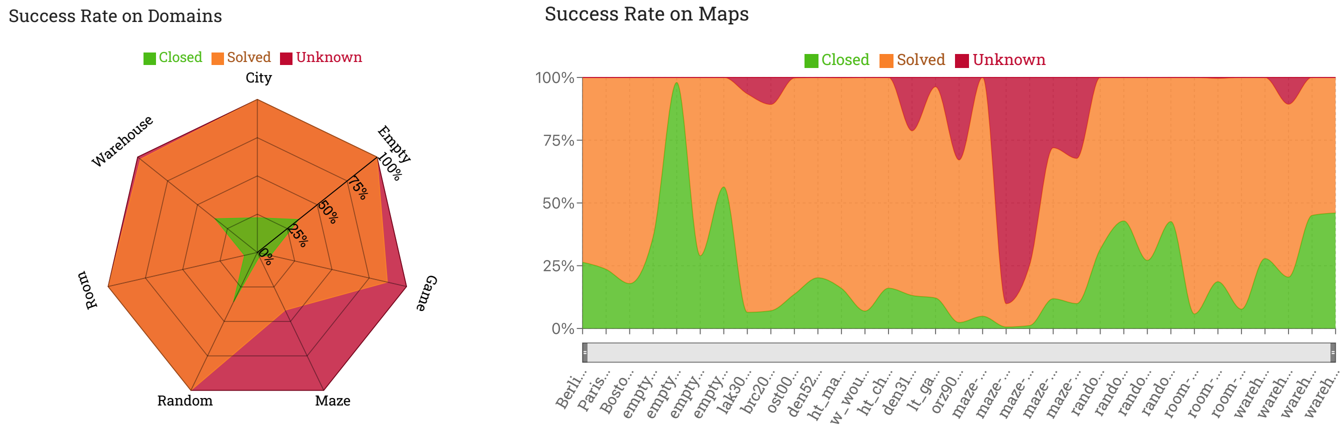

Figure 1: Screenshots taken from our website. We show the percentage of instances that were closed (green), solved (yellow), and unknown (red) for 7 different benchmark domains on the left, and for 33 different benchmark maps on the right.

and (iii). In the remainder of this section, we explain our strategies for generating insight and systemic analysis from results data, as well as a list of important things that we are tracking on different levels of the benchmark.

Instance-level Tracking At the instance level, our system records the best lower-bound and solution cost as explained above. For each reported lower-bound or valid plan we also keep track of additional metadata, such as the algorithm which produced the result, names of authors, publication references and links to implementations. We then use the data to provide additional insights:

Tracking the concrete plan: each instance contains a different number of agents, however, it is not clear how these agents are distributed w.r.t. the obstacles of map, and how their solution paths interact on the map. Our system records a concrete plan for each best known solution cost and provides a visualiser to better understand those solutions.

Tracking the gap: For each instance, we may have different algorithms which contribute lower-bounds and solutions (upper bounds) separately. Together, we need to analyse how close these algorithms are in terms of finding and proving optimal solutions. Therefore, we automatically track and visualise the *suboptimality ratio* of each instance defined as $(\mathcal{S} - \mathcal{L}) / \mathcal{L}$ where $\mathcal{L}$ and $\mathcal{S}$ are the best known lower-bound and solution of the instance, respectively.

Scenario-level Tracking All instances in a scenario are categorised into three types: (i) *closed instance*: the instance has the same best lower-bound and solution cost (indicating that the solution cannot be further improved); (ii) *solved instance*: the instance has a feasible solution reported, but the current best lower-bound is less than the solution cost (i.e., improvement may be possible); and (iii) *unknown instance*: the instance has no solution reported. For each scenario, our system tracks the percentage of *closed* and *solved* instances to indicate the progress of all contributed algorithms. For scenarios of the same map, we also track the following:

Tracking progress on scenarios: For a given map, our system automatically generates plots which shows the percentage of *closed*, *solved* and *unknown* instances for every scenario. The objective here is to identify the scenarios that are hard to solve with existing MAPF algorithms, so that more attention can be paid to these.

Tracking progress on different number of agents: Each scenario contains instances with different numbers of agents. It is important to understand the scalability of MAPF algorithms across all scenarios (i.e., at what number of agents we stop making progress). Therefore, our system includes the percentages of *closed*, *solved* and *unknown* instances for different number of agents on the same map.

Domain and Map-level Tracking Finally, at the map-level of the benchmark, our system records the percentages of *closed* and *solved* instances for each map. Similar to the scenario-level, our system also generates plots to track the percentages of *closed*, *solved*, and *unknown* instances across all maps, and summarises the related maps for each domain to provide domain-level plots. Figure 1 shows an overview of the progress made on domains and maps based on the results collected in our system thus far. Researchers can use this information to focus their efforts on solving those parts of the benchmark that have shown limited progress, such as the *maze* and *game* domains, as well as specific maps within these domains, including *orz900d*, *den312d* and *maze* maps.

Participation and Comparison Another critical feature of our system is allowing other researchers to participate by submitting their algorithms/results and establish the state-of-the-art together. For all the results we collect, we make them publicly available and allow other researchers to download the results at each level. In order to make it easy for researchers to evaluate their own progress against other attempts, we also provide tools to automatically compare algorithms, across every level of the system. Our principal evaluation criteria are: # of instances a given algorithm closed; # of instances that the algorithm solved; # of instances for which it achieved the best lower-bound; # of instances for which it reported the best solution. We apply these criteria to summarise the state-of-the-art for each type of algorithm. (optimal, bounded- and unbounded-suboptimal).

Acknowledgements

This work is supported by the Australian Research Council under grant DP200100025, DP230100081, FT180100140, and by a gift from Amazon.

References

Gange, G.; Harabor, D.; and Stuckey, P. J. 2019. Lazy CBS: Implicit Conflict-Based Search using Lazy Clause Generation. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*.

Lam, E.; Le Bodic, P.; Harabor, D.; and Stuckey, P. J. 2022. Branch-and-Cut-and-Price for Multi-Agent Path Finding. *Computers & Operations Research*.

Li, J.; Chen, Z.; Harabor, D.; Stuckey, P. J.; and Koenig, S. 2022. MAPF-LNS2: Fast Repairing for Multi-Agent Path Finding via Large Neighborhood Search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Li, J.; Harabor, D.; Stuckey, P. J.; Ma, H.; Gange, G.; and Koenig, S. 2021. Pairwise Symmetry Reasoning for Multi-Agent Path Finding Search. *Artificial Intelligence*.

Okumura, K. 2023. LaCAM: Search-Based Algorithm for Quick Multi-Agent Pathfinding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Shen, B.; Chen, Z.; Li, J.; Cheema, M. A.; Harabor, D. D.; and Stuckey, P. J. 2023. Beyond pairwise reasoning in multi-agent path finding. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*.

Stern, R.; Sturtevant, N. R.; Felner, A.; Koenig, S.; Ma, H.; Walker, T. T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T. K. S.; Barták, R.; and Boyarski, E. 2019. Multi-Agent Path Finding: Definitions, Variants, and Benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*.