Optimal and Bounded Suboptimal Any-Angle Multi-agent Pathfinding (Extended Abstract)

Konstantin Yakovlev[1,2], **Anton Andreychuk**[2], **Roni Stern**[3]
[1]Federal Research Center for Computer Science and Control RAS
[2]AIRI
[3]Ben Gurion University of the Negev
yakovlev@airi.net, andreychuk@airi.net, sternron@bgu.ac.il

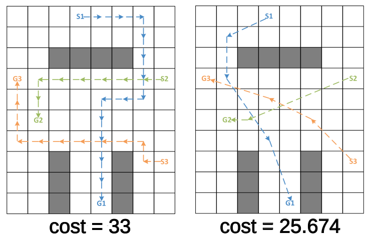

Figure 1: Two solutions of the same MAPF instance: with cardinal moves only (left) and with any-angle moves (right).

Introduction

Multi-agent pathfinding (MAPF) is the problem of finding a set of conflict-free paths for a set of agents. Most prior work on MAPF assumed that the agents move over a pre-defined graph of locations and allowed transitions between them. We focus on relaxing this assumption, allowing each agent to move between any pair of locations. This type of pathfinding is called *any-angle pathfinding* (Nash et al. 2007), We refer to its MAPF version as Any-Angle MAPF. A solution to an AA-MAPF problem is a set of n plans transferring the agents from their start vertices to their goal ones, such that each pair of plans is collision-free. We wish to solve the problem optimally w.r.t. the sum-of-cost objective, which is Figure 1 illustrates the difference between a solution to an AA-MAPF problem and a solution to a classical MAPF problem. Algorithms such as Anya (Harabor and Grastien 2013) and TO-AA-SIPP (Yakovlev and Andreychuk 2021) have been proposed for optimal single-agent any-angle path finding. In (Yakovlev and Andreychuk 2017) a suboptimal AA-MAPF solver was described. **We propose AA-CCBS, the first AA-MAPF algorithm that is guaranteed to return cost optimal solutions.**

AA-CCBS integrates the Continuous Conflict-based Search (CCBS) MAPF algorithm (Andreychuk et al. 2022) with TO-AA-SIPP. AA-CCBS is guaranteed to return optimal solutions, but scales poorly since any-angle path finding induces search trees with a very large branching factor. To mitigate this, we propose several enhancements to AA-CCBS based on techniques from classical MAPF, namely disjoint splitting (DS) (Li et al. 2019a) and multi-constraints (MC) (Walker, Sturtevant, and Felner 2020).

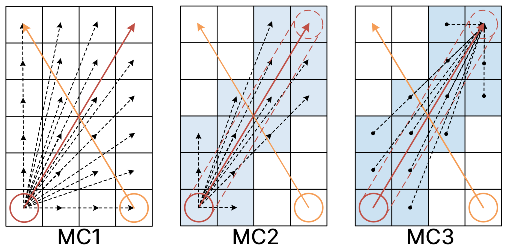

Figure 2: Actions comprising the different versions of MCs. Note that besides the shown actions, MC3 considers all the actions from MC2 as well.

Multi-Constraints in Any-Angle MAPF

CCBS resolves a conflict by adding a single constraint to one of the agents and replanning. However, adding multiple constraints (multi-constraint) when resolving a single conflict can reduce the number of high-level search iterations (Li et al. 2019b; Walker, Sturtevant, and Felner 2020). In particular, Walker et al. (2020) proposed the Time-Annotated Biclique (TAB) method adding multiple constraints in domains with non-uniform actions. For two conflicting actions, (a_i, t_i) and (a_j, t_j), TAB iterates over all actions that have the same source vertices as a_i and a_j, and identifies the subsets of them, A_i and A_j, that are *mutually conflicting*, i.e. each pair of actions from $A_i \times A_j$ lead to a conflict (if they start at t_i, t_j respectively). The multi-constraint (MC), that is added to the first agent is made of the constraints $(a_i, [t, t'])$ where $[t, t']$ is the largest time interval that is *fully included* into the unsafe intervals induced by a_i and *all* $a_j \in A_j$. The MC added to the second agent is defined similarly. TAB is applicable in AA-CCBS. We call this variant as MC1.

The problem with MC1 is that in AA-MAPF the number of mutually-conflicting actions may be large. Thus, we need to intersect numerous unsafe intervals to get the final one which is likely to get extremely trimmed, i.e. its ending point is very close to the start point. Thus the pruning power of MC diminishes. To this end we suggest two modifications.

First, we consider only a limited set of actions to form A_i

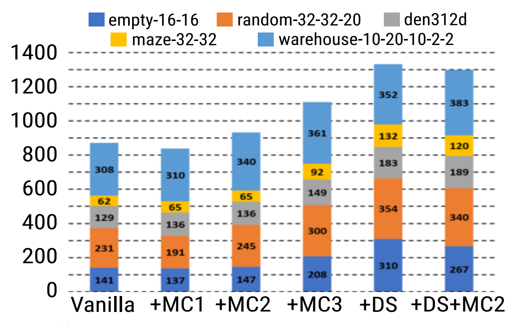

Figure 3: Number of the solved instances.

(A_j similarly). The source of such an action is the same as the source of the initially conflicting action a_i. The destination, however, must be a vertex that is swept by the agent when executing a_i, i.e. the agent's body intersects this vertex (grid cell). Thus the resulting actions form a "stripe" along a_i – see Fig. 2 (the cells that form a stripe are highlighted). The second modification is filtering out the actions from the candidate set of MC that lead to trimming of the original unsafe interval. In particular, if $unsafe(a_i, a'_j) \not\subset unsafe(a_i, a_j)$ then a'_j is excluded from A_j. Here a_i and a_j are the original conflicting actions, and $\{a'_j\}$ is the set of actions that are in conflict with all $a \in A_i$. The combination of those two enhancements is dubbed MC2.

Finally, we also suggest a way of composing the action set of MC that was mentioned by Walker (2020) but never implemented. I.e. we enlarge the action sets of MC2 with the actions that have the same target vertex (as the original conflicting actions) but *different source vertices*. We chose those source vertices to lie on the same strip as before – see Fig. 2 (right). The rationale behind this is that these actions are likely to lead to the collisions that will happen between the same agents and nearly in the same place. Thus it is natural to include them into the same (multi) constraint.

Disjoint Split with Multi-Constraint

Disjoint splitting (DS) (Li et al. 2019a) is a powerful technique for reducing search effort for CBS-based algorithm that has been used also for CCBS (Andreychuk et al. 2021). When adopting CCBS to AA-MAPF one might consider two options: using DS as-is or use DS with MC. The first option only requires adapting TO-AA-SIPP to be able to plan with the landmarks. Integrating DS with MC is more involved, since some constraints may involve actions starting at different vertices (like MC3). Thus, we focused on combining DS with the MCs that contain actions starting from the same vertex (MC1 and MC2).

When planning with the landmarks that are made of the multiple actions starting in the same vertex and ending in the different ones, a special care should be paid to the search states that result from applying the landmark actions as these states become the start search nodes of the next search. This is because better paths to these nodes may be found, and thus additional bookkeeping is needed to ensure completeness and optimality.

Experimental Results and Future Work

We evaluate six versions of AA-CCBS, AA-CCBS (vanilla), AA-CCBS+MC1, AA-CCBS+MC2, AA-CCBS+MC3, AA-CCBS+DS, AA-CCBS+DS+MC2, on five different maps from the MovingAI benchmark (Stern et al. 2019). For each map the benchmark provides 25 scenarios files, and we measured the number of problems solved within a time-limit of 300 seconds. Fig. 3 shows the results. The results show that MC1 and MC2 do not provide significant improvement over vanilla AA-CCBS. MC3, DS, AND DS+MC2, however, notably outperform it, especially the ones that use DS.

We have presented AA-CCBS, the first optimal any-angle MAPF algorithm and showed how to incorporate existing CCBS enhancements, namely disjoint splitting and multi-constraints, to make it scale better. Future work can explore more sophisticated procedures of forming multi-constraints, identifying which sets of actions should be considered in each step, as well as adapting incremental search techniques in the any-angle low-level search.

References

Andreychuk, A.; Yakovlev, K.; Boyarski, E.; and Stern, R. 2021. Improving continuous-time conflict based search. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, 11220–11227.

Andreychuk, A.; Yakovlev, K.; Surynek, P.; Atzmon, D.; and Stern, R. 2022. Multi-agent pathfinding with continuous time. *Artificial Intelligence*.

Harabor, D.; and Grastien, A. 2013. An optimal any-angle pathfinding algorithm. In *International Conference on Automated Planning and Scheduling*, 308–311.

Li, J.; Harabor, D.; Stuckey, P. J.; Felner, A.; Ma, H.; and Koenig, S. 2019a. Disjoint splitting for multi-agent path finding with conflict-based search. In *International Conference on Automated Planning and Scheduling*, 279–283.

Li, J.; Surynek, P.; Felner, A.; Ma, H.; and Koenig, S. 2019b. Multi-agent path finding for large agents. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, 7627–7634.

Nash, A.; Daniel, K.; Koenig, S.; and Felner, A. 2007. Theta*: Any-Angle Path Planning on Grids. In *Proceedings of The 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*, 1177–1183.

Stern, R.; Sturtevant, N. R.; Felner, A.; Koenig, S.; Ma, H.; Walker, T. T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T. S.; et al. 2019. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the 12th Annual Symposium on Combinatorial Search (SoCS 2019)*, 151–158.

Walker, T. T.; Sturtevant, N. R.; and Felner, A. 2020. Generalized and sub-optimal bipartite constraints for conflict-based search. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, 7277–7284.

Yakovlev, K.; and Andreychuk, A. 2017. Any-angle pathfinding for multiple agents based on SIPP algorithm. In *International Conference on Automated Planning and Scheduling*, 586–594.

Yakovlev, K.; and Andreychuk, A. 2021. Towards Time-Optimal Any-Angle Path Planning With Dynamic Obstacles. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS 2021)*, 405–414.