From Space-Time to Space-Order: Directly Planning a Temporal Planning Graph by Redefining CBS (Extended Abstract)

Yu Wu, Rishi Veerapaneni, Jiaoyang Li, Maxim Likhachev

Carnegie Mellon University
{yuwu3, rveerapa}@andrew.cmu.edu, {jiaoyanl, maxim}@cs.cmu.edu

1 Introduction

Multi-Agent Path Finding (MAPF) tries to find collision-free trajectories that usually minimize the net travel time of all the agents. Typically, MAPF methods find "space-time" paths that require agents to be at specific locations at specific timesteps and typically do not reason about velocity or kinematic constraints. Thus, these space-time paths are impossible to directly follow on real systems.

To allow agents to flexibly follow these space-time paths, Hönig et al. (2016) convert space-time paths to a temporal plan graph (TPG). The main idea is that agents can travel at arbitrary speeds as long as they visit overlapping locations in the same "temporal precedence ordering" (i.e. relative sequence) as their space-time paths. For example, if 3 agents A, B, C have space-time paths that visit state s at timesteps 32, 14, 35 respectively, they remove the time dependency and just require them to visit s in the relative order B, A, C. Real-world agents can now travel more robust to delays or at faster speeds, only having to wait and coordinate at these overlapped locations. The current pipeline to plan for a real-world multi-agent planning system is planning space-time paths first and post-processing them into a TPG.

However, this two-stage process means that we cannot reason about the coordination between agents. Once the space-time paths are computed, the TPG post-processing is fixed. Prior work (Wagner, Veerapaneni, and Likhachev 2022) shows that the independently computed space-time paths can result in TPGs that require a large amount of coordination. Instead, if we could directly plan a TPG that agents could follow, we could directly reason about and minimize the amount of coordination that the agents require. Minimizing coordination reduces the amount of agent communication needed which is a constraint in real-world multi-agent systems (Gielis, Shankar, and Prorok 2022) and can improve robustness to agent's delays due to imperfect execution.

Our core theoretical insight is that we can view an agent's path in a TPG as a set of spaces with relative visitation orders (e.g. visiting s first vs visiting s second). A TPG then consists of a set of these *space-visitation order* paths which are free of collision and deadlock. Armed with this knowledge, we can then adapt the general Conflict-Based Search (CBS) framework to create Space-Order CBS, which produces a valid TPG directly while optimizing the amount of coordination. We experimentally show that Space-Order CBS produces TPGs with less coordination than planning MAPF plans and post-processing to TPGs, and is thus more robust against random delays during execution.

2 Space-Order Perspective of a TPG

Vertex-Perspective TPG A regular space-time path is a sequence of (s, t) vertices which requires the agent to reach space (e.g. location) s at time t. Instead of defining a TPG from an "edge" perspective as originally done in Hönig et al. (2016), we redefine a TPG from a vertex perspective. A TPG path is a sequence of (s, r) vertices, where s represents space and r represents visitation order. For example, an agent j at $(s, r = 2)$ means that j is the 3rd agent to visit s (r is zero-indexed). Figure 1(d) shows how the TPG in (c) can be reinterpreted as (s, r) vertices. A valid TPG is then a set of agent space-order paths without any conflicts or deadlocks.

Coordination in TPG Coordination is required when agents j, k visit state s at vertices (s, r_j) and (s, r_k), respectively. If $r_k < r_j$, agent k has to wait for agent j to leave s before it can enter s. We propose two metrics to quantify and minimize coordination. One natural metric is the total number of these possible wait interactions between pairs of agents. We define this value as the C_{total} of a TPG. We define another metric C_{unique} as the total number of *unique* pairs of agents who coordinate with each other, regardless of number of instances where these two agents coordinate. In the example in Figure 1, $C_{total} = 2$ and $C_{unique} = 1$.

3 Space-Order CBS

To directly plan a TPG as a set of space-order paths while minimizing the coordination objectives we defined, we adapt the Conflict-Based Search (CBS) (Sharon et al. 2015) framework and design Space-Order CBS (SO-CBS). CBS is a popular weakly complete and optimal space-time MAPF planner. It employs a high-level constraint tree (CT) that searches over conflicts in a best-first manner and a low-level A* search that searches individual paths while respecting constraints. SO-CBS redesigns a vast majority of CBS. Our low-level planner searches for a valid space-order path where vertex (s, r) can go to (s', r') where s' is a neighbor of s, and r' can be any unconstrained value (as op-

See https://arxiv.org/abs/2404.15137 for full version.

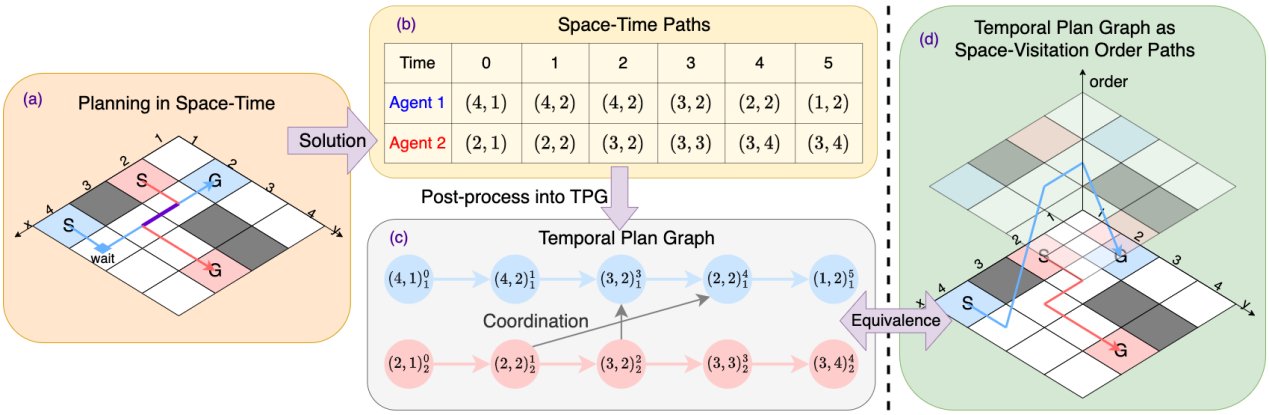

Figure 1: Existing MAPF methods compute space-time paths that require agents to be at a specific location at a specific time (b). (a) shows how space-time paths may require agents to wait (e.g. the blue agent waiting at (4,2)), and have spatial overlaps in their paths (denoted in purple). To allow agents to move at arbitrary times, the output space-time paths can be post-processed into a Temporal Plan Graph which defines a precedence order (c). Cross edges (gray) in the TPG define coordination instances. These occur at overlapping regions (e.g. purple in (a)) which require coordination to avoid collisions if agents move at arbitrary time. In (c), $(3,2)^2_2 \rightarrow (3,2)^3_1$ represents agent 1 needing to wait for agent 2 to leave $(3,2)$ before it can enter. We show that by redefining vertices to space-visitation order (s,r) vertices representing the r^{th} agent to visit state s, a TPG is equivalent to agents having a set of non-intersecting space-visitation order paths (d). This equivalence allows us to directly plan a TPG.

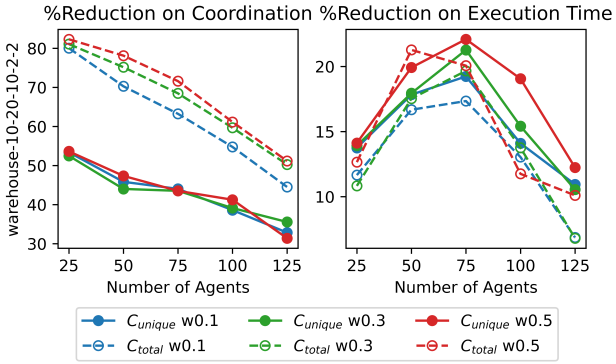

Figure 2: The left plot shows our percentage reduction on coordination (measured by C_{total}). The right plot shows our reduction on execution time when simulated with random delays. Each line represents a different setting for SO-CBS.

posed to $r+1$ in space-time search). We describe criteria necessary for a valid TPG and invent novel constraints to resolve conflicts while maintaining weak-completeness. Instead of optimizing the path length L, SO-CBS optimizes $w*C+(1-w)*L$, where C is the coordination metric (e.g. C_{total} or C_{unique}) and w is a hyper-parameter. This allows SO-CBS to jointly reason on path length and coordination.

4 Experimental Evaluation

We experimentally compare SO-CBS against the baseline of running ECBS and then post-processing into a TPG (ECBS-POST) and show that SO-CBS produces valid TPGs that have less coordination between agents and are robust to random delays during execution. We show results on a warehouse map in Figure 2 and the trends are generally applicable to the other 5 maps we tested on. The SO-CBS results are shown with two C_{total} and C_{unique} objectives and three different cost weights. We compare the coordination (measured by C_{total}) of TPGs planned by both methods in the left subplot of Figure 2. SO-CBS achieves improvements of 30-90% on C_{total}, providing evidence that SO-CBS utilizes free space to avoid coordinating between agents.

To test robustness with delays, we simulated execution with 5% of the agents having 20% random delays, with each delay having a duration of 100 timesteps. Agents are required to follow their TPG execution policy and we evaluate the wait time due to coordination and the total execution time summed over all agents, which are shown in the right subplot of Figure 2. The best-performing setup is using C_{unique} objective with a cost weight of 0.5, achieving 15%-35% improvements in wait time (not shown in Figure) and 10%-20% improvements in total execution time.

Acknowledgements

This material is partially supported by NSF Grant IIS-2328671.

References

Gielis, J.; Shankar, A.; and Prorok, A. 2022. A Critical Review of Communications in Multi-Robot Systems. *CoRR*, abs/2206.09484.

Hönig, W.; Kumar, T. S.; Cohen, L.; Ma, H.; Xu, H.; Ayanian, N.; and Koenig, S. 2016. Multi-agent path finding with kinematic constraints. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. R. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219: 40–66.

Wagner, A.; Veerapaneni, R.; and Likhachev, M. 2022. Minimizing Coordination in Multi-Agent Path Finding with Dynamic Execution. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 18(1): 61–69.