A Combined Learning- and Search-based Approach to Complete Multi-Agent Path Finding

Guillaume Sartoretti*, Sven Koenig†, Howie Choset*
*Carnegie Mellon University - The Robotics Institute
†University of Southern California - Department of Computer Science

Multi-agent path finding (MAPF) considers the problem of planning collision-free paths for numerous agents from their current position to their assigned goal, in a potentially obstacle-ridden environment. MAPF is a central problem in robotics, with applications in automated distribution centers, airplane taxiing, as well as multi-agent search and rescue [1, 2, 3]. However, as the number of agents in the system grows, so does the combinatorial complexity of coordinating them, and MAPF remains a NP-hard problem even when approximating optimal solutions [4, 5]. To mitigate the curse of dimensionality associated with coupled approaches (e.g., simple A^*), where a single planner plans paths for all agents in their joint configuration space, the community recently focused on dynamically-coupled approaches that only extend the configuration space of the agents when necessary (usually around collisions between individual paths) [6, 7, 8, 9]. Since these dynamically-coupled planners remain centralized, they often come with provable performance guarantees, such as *path optimality* or *completeness*, i.e., the planner will always find a solution to a problem if a solution exists. When producing bounded-suboptimal paths, our prior works in dynamically-coupled planners can handle up to 200 agents [6] in uniform-randomly generated grid worlds. In this context, recent works have also focused on fully decoupled approaches, where each agent plans its own path [10, 11, 12] to handle larger teams. However, since joint actions cannot be considered, the resulting paths are usually suboptimal. Moreover, deadlocks can also appear in fully decoupled MAPF, making most of these approaches incomplete [13].

In this context, our recent work, PRIMAL [14], focused on fully decoupled MAPF by combining distributed reinforcement learning (RL) and imitation learning (IL) from a dynamically-coupled planner. By relying on a good collaborative reward structure, careful training, and demonstrations of (joint) collaboration from the centralized planner, we obtain an essentially single-agent policy allowing agents to plan individual collision-free paths to their goal, while exhibiting implicit collaboration. This policy, once trained, can be copied onto an arbitrary number of agents, and we show successful MAPF for up to 1024-agent teams in grid worlds with relatively low obstacle density. However, as the grid size becomes smaller, and/or the obstacle density larger, PRIMAL starts to struggle since it cannot exhibit the complex coordinated movements needed to complete such problems, and is usually outperformed by dynamically-coupled planners. However, we note that even when PRIMAL does not allow all agents to reach their goal, it usually brings most agents to their goal and gets a handful of them close to it (see Fig.1 for an example). Building upon this observation, and as a natural way to bridge the gap between decoupled and dynamically-coupled planners, this work investigates the combined use of PRIMAL and a centralized complete planner (here, ODrM* [8]), to finally obtain fast, scalable, and complete MAPF.

Specifically, we propose to use PRIMAL for a fixed number of time steps, during which agents plan their path individually based on their current state (in their local observation range, here 10×10). After this first phase, and only if all agents are not already on goal, we rely on a short instance of centralized, dynamically-coupled planning by running the bounded-suboptimal version of ODrM* ($\epsilon = 3$) for a maximum of 5 seconds, to try and get all remaining agents to on goal. ODrM* is ran from the current state of the system at the end of the first phase, and plans for all agents, since those resting on goal might need to move away to let other agents reach their goal.

Figure 2 shows the result of this combined learning- and search-based MAPF approach to the same experiments we previously considered in [14]. For each team size, we measure the success rate over the same (randomly generated) 100 experiments across all planners. We consider three key cases: a case with a medium-size world with high obstacle density, one with a small world with medium obstacle density, and finally one with a large world with high obstacle density. These example have been handpicked to show the advantage and limits of our approach. We also note that results for cases where PRIMAL particularly shines (low obstacle densities) are not affected by the additional use of a dynamically-coupled planner. We believe that this effect can be explained by the fact that empty environments are particularly difficult for a dynamically-coupled planner, since most agents end up colliding, often resulting in fully coupled planning.

The first observation from these results is that adding 5s of ODrM* planning cannot really help with cases involving 512 agents and more, since these cases most likely require longer planning times, even just for the few agents not already resting on their goal. Similarly, in the third example involving large worlds (80×80) with high obstacle density (30%), we note that, although adding a step of centralized planning after running PRIMAL gets the success rates of small teams (up to 32 agents) to 100%, the performance of ODrM* drops drastically for teams larger than 64 agents, and centralized

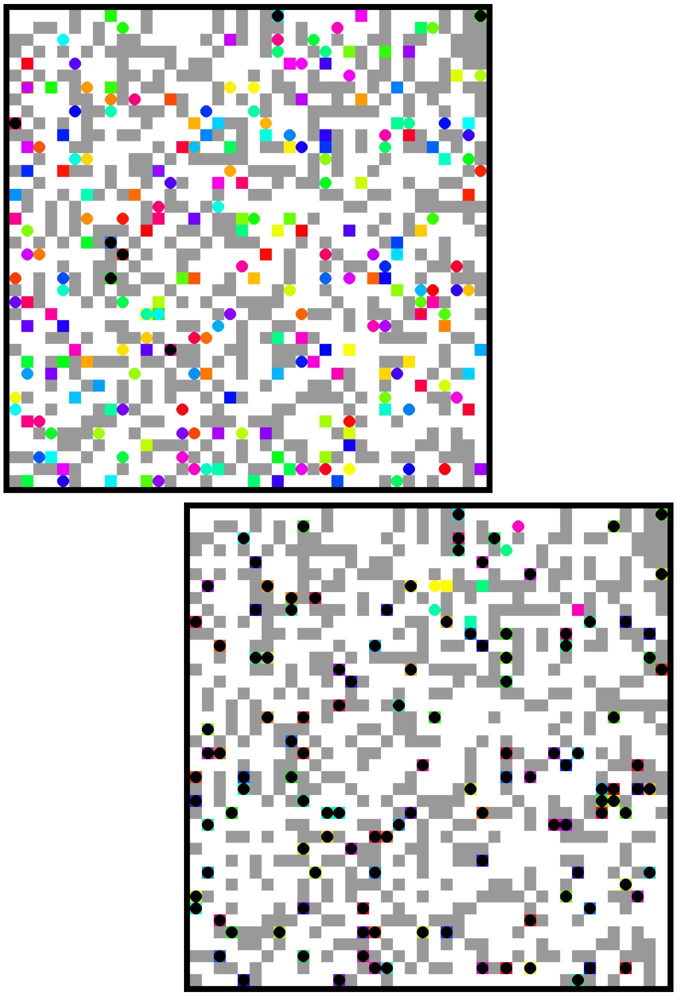

Figure 1. Key frames along an example problem, where 128 agents (color-coded squares) are planning paths to their goal (same-colored circles) in a 40×40 discrete grid world with 30% uniform-randomly placed obstacles (grey squares). Top: Initial state. Bottom: State after 256 time steps of decentralized path planning using PRIMAL. This second problem is finally solved in less than 3s using ODrM*, adding 28 movements (i.e., time steps).

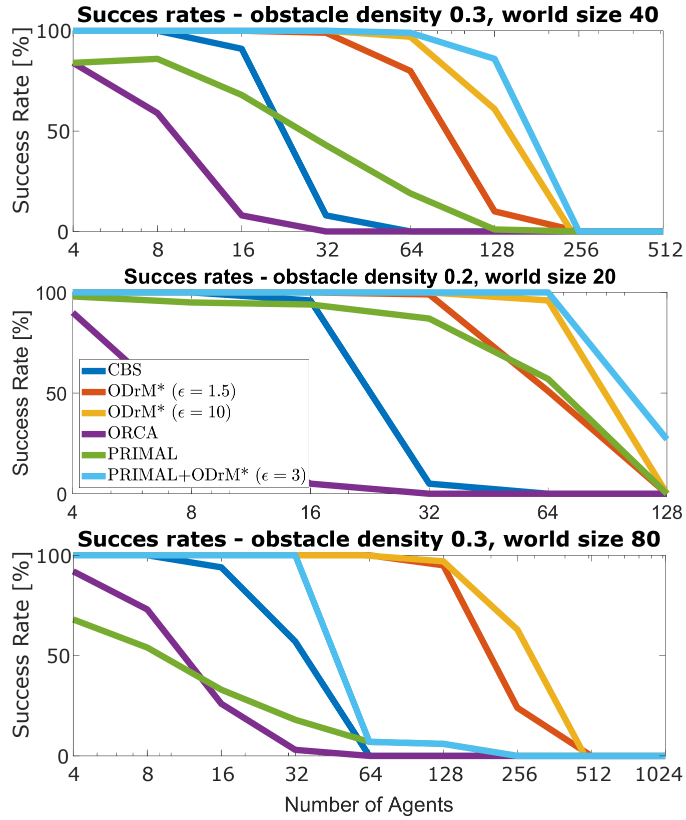

Figure 2. Success rates of the different planners in the three selected key scenarios. Our combined approach (light blue) outperforms all other planners in the two top plots (medium to high obstacle density in small to medium worlds), even though in these cases PRIMAL particularly struggles by itself. In the bottom plot, involving a large world with high obstacle density, our combined approach helps significantly for smaller teams (up to 32 agents) but cannot help beyond that and is outperformed by the other centralized planners.

planners outperform our combined approach for these larger teams. However, for small to medium worlds (40 and 20×20) and relatively high obstacle densities (20-30%), our combined approach is both able to significantly improve over the success rate of PRIMAL, and even outperform all other planners (top plots in Fig. 2). These results seem to indicate that our combined approach allows for highly scalable, as well as complete MAPF, where we supplement decoupled (and potentially online) path planning with a form of online deadlock detection and centralized planning. Although our current deadlock detection method is relatively simple, only relying on a maximal number of steps PRIMAL is allowed to run before switching to ODrM*, we believe that such a combined approach could be valuable for actual deployments in automated warehouses, where numerous agents may not be driving fast enough to prevent the seldom execution of 5s of centralized planning.

Future works of this combined approach will first focus on a more advanced dead/livelock detection approach, to further reduce the time during which PRIMAL is used without really bringing more agents to their goal. In doing so, our goal will be to try ans answer the question of "*how should agents planning paths in a decentralized manner realize that they are stuck and need centralized help to reach their goal?*" As a first step, we envision that, by observing the fact that the state of the system (or even the individual agents' states) is similar to a previously observed state, we will be able to detect oscillations in the system characteristic of a deadlock, and start the use of centralized planning earlier. Second, we would like to investigate another approach to combine decoupled and dynamically-coupled planning. Specifically, we will be looking at using PRIMAL as the single-agent policies in M*, and then let the M* planner detect and backtrack when a collision is detected. Combining both approaches in this manner will involve several new challenges, especially since PRIMAL is essentially an online path planning algorithm (making backtracking nontrivial), but this combination should provide us with very fast, yet provably complete (and maybe bounded-suboptimal) MAPF planners.

References

[1] Joseph L Baxter, EK Burke, Jonathan M Garibaldi, and Mark Norman. Multi-robot search and rescue: A potential field based approach. In *Autonomous Robots and Agents*, pages 9–16. Springer, 2007.

[2] Hamsa Balakrishnan and Yoon Jung. A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport. In *AIAA GNC*, volume 3, pages 2382–2400, 2007.

[3] Martin Bonert, LH Shu, and B Benhabib. Motion planning for multi-robot assembly systems. *International Journal of Computer Integrated Manufacturing*, 13(4):301–310, 2000.

[4] Hang Ma, Daniel Harabor, J. Li, and Sven Koenig. Searching with Consistent Prioritization for Multi-Agent Path Finding. In *AAAI*, 2019.

[5] S LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[6] G. Wagner and H. Choset. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219:1–24, 2015.

[7] Max Barer, Guni Sharon, Roni Stern, and Ariel Felner. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Proceedings of SoCS*, 2014.

[8] C Ferner, G Wagner, and H Choset. ODrM*: optimal multirobot path planning in low dimensional search spaces. In *ICRA*, pages 3854–3859, 2013.

[9] G Sharon, R Stern, A Felner, and N Sturtevant. Conflict-based search for optimal multi-agent path finding. In *Proc. of AAAI*, 2012.

[10] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research*, pages 3–19. 2011.

[11] Prasanna Velagapudi, Pradeep Varakantham, Katia Sycara, and Paul Scerri. Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 955–962. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[12] S Leroy, J-P Laumond, and T Siméon. Multiple Path Coordination for Mobile Robots: A Geometric Algorithm. In *IJCAI*, pages 1118–1123, 1999.

[13] G Sanchez and J Latombe. Using a PRM planner to compare centralized and decoupled planning for multi-robot systems. In *Proceedings of ICRA*, volume 2, pages 2112–2119, 2002.

[14] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. *IEEE Robotics and Automation Letters*, 4(3):2378–2385, July 2019. arXiv preprint.