Explainable Multi-Agent Motion Planning

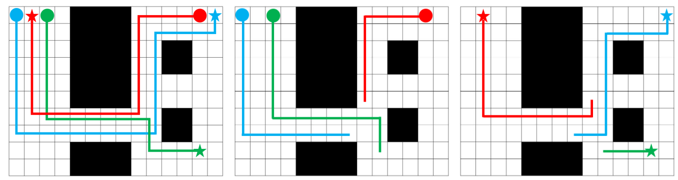
Justin Kottinger*, Shaull Almagor†, and Morteza Lahijanian*
*Aerospace Engineering Sciences, University of Colorado, Boulder, USA
Email: {justin.kottinger, morteza.lahijanian}@colorado.edu
†Computer Science, Technion, Haifa, Israel
Email: shaull@cs.technion.ac.il

I. Introduction

Multi-Agent Path Finding (MAPF) is a fundamental challenge in robotics and artificial intelligence (AI) in which the goal is to plan a path for multiple agents to reach their respective goal regions such that, when the plans are executed simultaneously, every vehicle successfully completes their path without colliding into other agents. The applications of MAPF can be found in many areas where several moving agents interact in a shared workspace. However, the use of various AI systems, including MAPF, is limited by current algorithms' inability to explain their decisions and actions to human users [9]. In many safety-critical situations, such as air-traffic control, or hazardous material warehouses, MAPF typically is not fully automated. Instead, the paths for each agent are generated using motion-planning techniques and given to a human-supervisor before execution. The supervisor's job is to verify that the automatically generated plan is safe (collision-free). Thus, these settings require the plan to be presented in a humanly-understandable manner. Specifically, the presentation should enable the supervisor to understand the path taken by individual agents and to verify that the agents do not collide. To this end, the goal of this work is to present a method of generating explainable motion plans for multi-agent systems.

Significant effort has been dedicated to providing explanations for problems in AI and machine learning. For example, the work of [6] utilized visualization to explain the result of certain machine learning algorithms that often come up with complicated classifiers. In [2], explanations are given by analyzing alternative plans with some user-defined properties. In [5], a user proposes a plan, and explanations are given as a minimal set of differences between the actual plan, and the proposed plan. A broader approach was later given in [3], where multiple types of explanations are allowed.

This work is similar to [6], where the aim is to explain the solution to a MAPF problem to a human using visualization techniques. With this goal in mind, one may suggest a sequence of images or video to explain the solution. However, watching a video takes a long time, and is nearly impossible, if there are more than 3 agents. Our approach is different in that we base the explanations on the simplicity of visual verification by human's cognitive process. Specifically, work [4], and [8] show the identification of line intersections is made very early in the cognitive process (namely in the primary visual cortex). Thus, MAPF can best be explained using a collection of *non-intersecting* path segments. An example of such an explanation is shown in Fig. 1.



(a) Full Plan **(b)** Part 1 **(c)** Part 2

Fig. 1: A plan for three agents in a grid. The circles and stars mark the initial and goal locations for the agents, respectively. (a) shows the full plan, and Figures (b), and (c), show a disjoint decomposition.

Previous work [1] on explainable MAPF focused on providing this explanation scheme over a discrete graph. That work also found that as the number of disjoint segments required to explain the path decreased, the ease of explainability increased. Furthermore, [1] proved that finding optimal explanations for pre-existing motion plans can be done in polynomial time, whereas generating plans for explainability, i.e., limiting the number of segments, is, at best, **NP-Complete**.

This work focuses on MAPF with explanations for realistic robotic systems in continuous space with kinodynamical constraints. This problem is challenging because motion planning needs to be done in the compound (continuous) state space of the agents, whose dimension is exponential in the number of agents. Moreover, kinodynamical and explanation constraints add more complexity to the planning problem. Hence, the goal of this work is to design a computationally-tractable MAPF algorithm that generates sound plans that are easily explainable. This paper proposes two approaches to this problem: abstraction-based and tree-based search. By utilizing abstraction techniques, the continuous MAPF problem can be lifted to the discrete domain. Then, the results of [1] can be employed, which showed that it is possible to compute and explain a plan in the abstraction by using an A^*-based algorithm. This method simplifies the problem, but at the cost of sacrificing completeness. To address this problem, we are currently investigating a tree-based search algorithm capable of presenting efficient explanations. We introduce preliminary results on this algorithm in this paper.

II. Problem Formulation and Approach

Problem. We consider $k \in \mathbb{N}$ robotic agents with dynamics

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i, \mathbf{u}_i), \quad \mathbf{x}_i \in X_i \subseteq \mathbb{R}^{n_i}, \quad \mathbf{u}_i = U_i \subseteq \mathbb{R}^{m_i} \qquad (1)$$

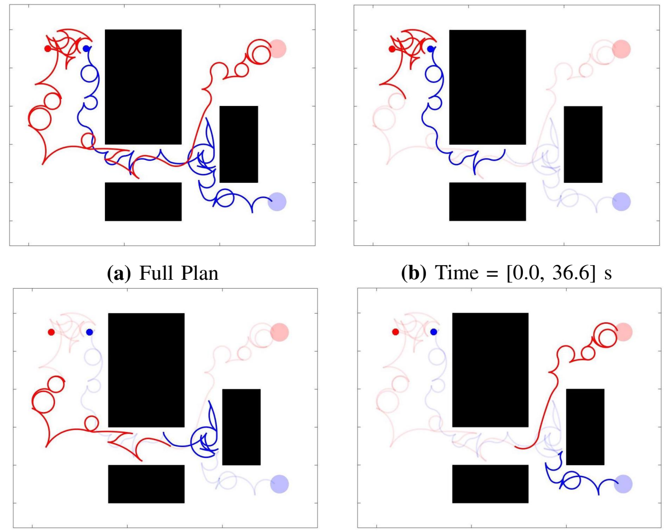
where $i \in \{1, \ldots, k\}$, X_i and U_i are the state space and input space for agent i, respectively, $f_i : X_i \times U_i \to X_i$ is an integratable and possibly nonlinear function. Once f_i is given a control input and integrated for some non-zero time duration $\Delta t = [t_1, t_2]$, a *trajectory segment* is formed for agent i denoted by $\mathbf{x}_i^{t_1:t_2}$.

We assume that agents share a 2-D workspace $W \subset \mathbb{R}^2$. We define $\text{PROJ}_W^{X_i}$ to be function that projects trajectory $\mathbf{x}_i^{t_1:t_2}$ onto workspace W. Then, we say two trajectory segments $\mathbf{x}_i^{t_1:t_2}$ and $\mathbf{x}_j^{t_1:t_2}$ are *disjoint* if $\text{PROJ}_W^{X_i}(\mathbf{x}_i^{t_1:t_2}) \cap \text{PROJ}_W^{X_j}(\mathbf{x}_j^{t_1:t_2}) = \varnothing$. Further, given $m \in \mathbb{N}$ time intervals, we define *trajectory* $T_i = \{\mathbf{x}_i^{t_0:t_1}, \mathbf{x}_i^{t_1:t_2}, \ldots, \mathbf{x}_i^{t_{m-1}:t_m}\}$ to be a set of m trajectory segments that takes agent i from an initial point to a desired goal region $X_i^G \subset X_i$.

Given a planning problem consisting of k agents in a shared workspace W, the goal of continuous MAPF is to find a trajectory T_i for every agent such that no agent collides with obstacles nor with other agents and $\mathbf{x}_i(t_m) \in X_i^G$ for all $i \in \{1, \ldots, k\}$. The goal of *explainable* MAPF adds two constraints: (i) perform a segmentation of the trajectories such that the segments are disjoint, i.e., compute time durations $[t_0, t_1], \ldots, [t_{m-1}, t_m]$ such that the corresponding segments in each time interval are disjoint, and (ii) the number of segments (explanations) cannot be larger than $r \in \mathbb{N}$, the maximum desired number of explanations, i.e., $m \leq r$.

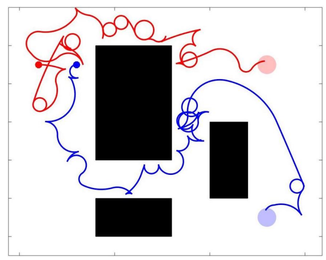
Approach. We propose two approaches to this problem. The first approach is based on an abstraction of the planning problem to a graph by discretizing the workspace W and designing control laws that guarantee the realization of edges of the graph with a fixed time duration in the continuous domain as detailed in [1]. Then, the planning problem on the graph can be solved using the A^*-based planner proposed in [1]. The second approach is based on centralized motion planning with a sampling-based tree search. Specifically, we build upon a *rapidly-exploring random tree* (RRT) [7] to plan with respect to explainability while maintaining the probabilistically completeness property of RRT. The tree is initialized with a maximum number of allowed disjoint trajectories required to explain the path. Then, as the tree grows, each node is given a cost that is equivalent to the number of decompositions required to explain the trajectory from the initial point, to the node. The tree only grows nodes that have costs within the pre-specified threshold. The result is a trajectory that is decomposed into a satisfiable number of disjoint segments.

The work [1] finds segmentation using greedy search, and utilizes this to maintain optimal decompositions during planning. This, however, becomes computationally infeasible in the continuous domain. Thus, our planner recursively tracks the number of intersections. In the worst case, the number of intersections is equal to the number of compositions required to explain the path. Our planner exploits this concept by only performing greedy search on trajectories where the number of intersections are greater than threshold r. This method limits the computational cost to allow for feasible MAPF.



(a) Full Plan **(b)** Time = [0.0, 36.6] s

(c) Time = [36.6, 87.9] s **(d)** Time = [87.9, 123.6] s



(e) Full plan with one segment.

Fig. 2: Two agents travel from their starting points (left) to their circular goal regions (right). The plan in (a) is explained with 3 segments (b)-(d), and plan in (e) explained with 1 segment.

III. Experiments

We present preliminary results of our two approaches here. First, we consider three agents with second-order unicycle dynamics in the environment in Fig. 1a. We used the abstraction method for this case, which resulted in a plan that can be explained in two segments, as shown in Figs. 1b and 1c.

The second experiment involves two agents with first-order dynamics. The environment is shown in Fig. 2a. We use our RRT-based planner, which is capable of restricting plans to a desired level of explainability. If no limitation is given, RRT returns a path that requires 8 disjoint trajectory segments to explain the entire path in 30 seconds. The plan, however, is hard to explain. Alternatively, the user can limit the number of explanations to 3, as shown in Figs. 2b-2d. This solution took about 5 minutes of computation time. Restricting the explanations to 1, the planner returns the plan and explanations in Fig. 2e, which took about 10 minutes to compute. These case studies demonstrate that the planning becomes more difficult as the desired number of explanations is reduced.

IV. Conclusion

This work exploits the humans' natural cognitive process to explain multi-agent motion plans using images of disjoint path segments within a 2-D workspace. An abstraction-based method allows the use of the discrete planner developed in [1]. Alternatively, our probabilistically complete tree-based planner takes the maximum allowable number of explanations from the user and returns a plan with a satisfiable explanation scheme.

References

[1] Shaull Almagor and Morteza Lahijanian. Explainable multi agent path finding. In *To appear in Int'l Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 2020.

[2] Rebecca Eifler, Michael Cashmore, Hoffmann Jorg, Danielle Magazzeni, and Marcel Steinmetz. Explaining the space of plans through plan-property dependencies. *Proceedings of the 2nd Workshop on Explainable Planning (XAIP 2019)*.

[3] Maria Fox, Derek Long, and Daniele Magazzeni. Explainable planning, 2017.

[4] David H. Hubel and Torsten N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3), 1959. doi: 10.1113/jphysiol.1959.sp006308.

[5] Subbarao Kambhampati. Synthesizing explainable behavior for human-ai collaboration. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 1–2, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

[6] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), Mar 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08987-4. URL http://dx.doi.org/10.1038/s41467-019-08987-4.

[7] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[8] Shiming Tang, Tai Sing Lee, Ming Li, Yimeng Zhang, Yue Xu, Fang Liu, Benjamin Teo, and Hongfei Jiang. Complex pattern selectivity in macaque primary visual cortex revealed by large-scale two-photon imaging.

[9] Matt Turek. Explainable artificial intelligence. URL https://www.darpa.mil/program/explainable-artificial-intelligence.