Online Competitive Information Gathering for Partially Observable Trajectory Games

Extended Abstract

Mel Krusniak
Vanderbilt University
Nashville, TN, United States
mel.krusniak@vanderbilt.edu

Hang Xu
Vanderbilt University
Nashville, TN, United States
hang.xu@vanderbilt.edu

Parker Palermo
Vanderbilt University
Nashville, TN, United States
parker.d.palermo@vanderbilt.edu

Forrest Laine
Vanderbilt University
Nashville, TN, United States
forrest.laine@vanderbilt.edu

ABSTRACT

When planning a trajectory through continuous space, a rational agent should consider the information limitations of itself and its counterparts, and seek out useful observations. While approximate solutions can be found to many such partially observable multi-agent problems (i.e., through reinforcement learning), doing so online and in continuous spaces is not trivial. The existing control-theoretic method of model predictive game play (MPGP) combines continuous, online control with game theoretic rational play, but does not inherently support partial observability. Our work addresses this case, presenting a method to generate information-aware plans with MPGP alongside adjustments required to deploy it on individual interacting agents. While our method is not real-time, it allows us to consider what is required to compute solutions from scratch. We evaluate the method in variants of a partially observable pursuit-evasion game, and demonstrate evidence of information gathering behavior that outperforms passive competitors.

KEYWORDS

Continuous planning; partial observability; game theory

ACM Reference Format:
Mel Krusniak, Hang Xu, Parker Palermo, and Forrest Laine. 2025. Online Competitive Information Gathering for Partially Observable Trajectory Games: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 OVERVIEW

Consider a pursuit-evasion scenario between two free-moving agents, each with a mounted "camera" used to localize the other which faces the direction of velocity. The pursuer starts randomly in one of two locations — unknown to the evader between them. In an ideal plan, the evader first views one location, then executes the remainder of the trajectory based on what it senses: a desirable behavior known as *active information gathering*. However, if future observations are not considered while the plan is calculated, the evader flees from the average pursuer location. Figure 1 visualizes the distinction. Each column shows a possible pursuer location.

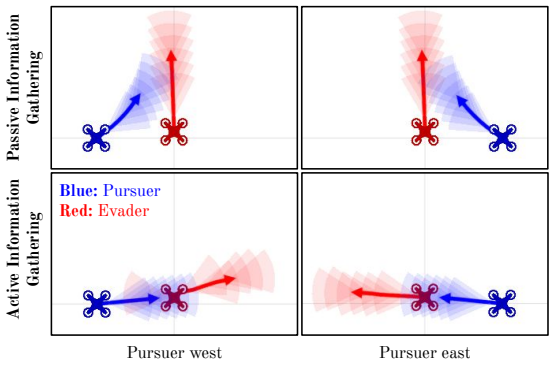

Figure 1: Illustration of information gathering modalities in a pursuit-evasion game "Tag". Cones indicate field of view at each planned timestep. The pursuer's initial position ("east" / "west") is random and unknown to the evader.

We address the online generation of such continuous, active information gathering trajectory plans using *model predictive game play* (MPGP). Standard MPGP considers only perfect information games, so we develop a variant for imperfect information games to permit active information gathering. We present a live planner for these scenarios which uses a particle representation of the full joint distribution of observation histories and states.

Our work relates to several overlapping frameworks. We primarily consider *partially observable stochastic games* (POSGs), N-player extensions of POMDPs [1], [18]. Planning in them is NP-hard [16]. Nevertheless, modern MARL advances have yielded strong strategies in (mostly discrete) POSGs like Stratego [12], Starcraft [15], and Diplomacy [2]. POSGs are related to tree-based *extensive form games*, and the distinction varies [10] [3]. Many relevant MARL methods are informed by POSGs and/or EFGs, like neural fictitious self-play [8], deep counterfactual regret minimization [5], and policy space response oracles [11], but these methods are primarily offline, and often (as in I-POMDPs [7]) limited by belief order.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Model predictive game play, on the other hand, is interpretable as a control-theoretic approach [6] [17]. In MPGP, players solve a perfect information game on a rolling horizon (essentially, multi-agent model predictive control). We merge it with *belief space planning* [13], which solves single optimization problems under *imperfect* information. Past attempts to unite the two stop at the first order of belief [14].

2 APPROACH

Consider a POSG $\mathcal{G}$, defined as a tuple with the following elements:

- $\mathcal{N} := 1..N$, the set of players;
- $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{Z}$, the sets of states x, actions a, and observations z (joint across players and time);
- $T(x_t|x_{t-1}, a_{t-1})$ and $O(z_t|x_t)$, the public transition and observation models;
- $r(x_t)$, the (simplified) reward functions; and
- $X_0(x_0)$, the joint prior distribution over players' states. (This distributes over initial configurations of the game for all players. These initial states may be correlated between players.)

We index over both players and time: i.e., $x^{(i)}$ denotes player i's states for all timesteps, x_t denotes *all* players' states at timestep t, and undecorated x denotes the joint state at all timesteps. We define player i's *cost* $c^{(i)}(x) := -\sum_{t=1}^{\infty} r^{(i)}(x_t)$. We consider only complete, differentiable POSGs with deterministic strategies.

We encounter two necessities when adapting POSGs for MPGP. First, rather than planning single trajectories, players map future observation histories to trajectories. To do this, we consider finite past and future of lengths T_{past} and T_{future} respectively, and represent mappings from observation histories $z^{(i)}_{[t]} := [z^{(i)}_{t-T_{\text{past}}+1}, ..., z^{(i)}_t]$ to actions $a^{(i)}_t$ as policies π with parameters θ. Second, players must account for the past and future observations of *other* players. Therefore, players track the joint distribution of observation histories and states $q_{\bar{t}}(x_{\bar{t}}, z_{[\bar{t}]}))$ from X_0 to the planning time $\bar{t}$, across all players (including themselves). This distribution is unconditioned on past observations (with one exception discussed later).

In total, at each step, each player $k \in \mathcal{N}$ solves the game

$$\left\{ \underset{\theta^{(i)}}{\text{argmin}} \int_{\substack{x \in \mathcal{X}^{1+T_{\text{future}}} \\ z \in \mathcal{Z}^{T_{\text{past}}+1+T_{\text{future}}}}} c^{(i)}(x) p(x, z|x_{\bar{t}}, z_{[\bar{t}]}) q_{\bar{t}}(x_{\bar{t}}, z_{[\bar{t}]}) \right\}_i \quad (1)$$

at each horizon (seeking Nash equilibrium across all $i \in \mathcal{N}$), where

$$p(x, z|x_{\bar{t}}, z_{[\bar{t}]}) = \prod_{t=\bar{t}}^{\bar{t}+T_{\text{future}}} T(x_{t+1}|x_t, \pi_\theta(z_{[t]}))\, O(z_t|x_t) \quad (2)$$

and executes $\pi^{(k)}_\theta(z^{(i)}_{[t]})$, using real observation history $\bar{z}^{(i)}_{[t]}$.

Accordingly, our approach is as follows for each player k:

(1) At $\bar{t} = 0$, initialize $q_0(x, z)$ as a particle representation of X_0.
(2) Solve the equilibrium problem in Eq. 1: estimate the integral using Monte Carlo sampling of $q_{\bar{t}}$ and perform gradient play.
(3) Execute $\pi^{(k)}_\theta(z^{(i)}_{[t]})$. Record $z^{(k)}_{\bar{t}+1}$.
(4) Generate $q_{\bar{t}+1}$: Step particles in $q_{\bar{t}}$ forward with $\mathcal{T}$ and O.
(5) Repeat from (2) with $\bar{t} + 1$.

We include two particle-updating refinements for (4). First, agents perform step (2) N_{eq} times and update each particle with a random solution to account for potential equilibrium disagreement among players. Second, with very low probability γ, a particle is updated with fixed observation $\bar{z}^{(k)}_{\bar{t}}$ in (4) and reweighted (like a Bayes filter), ensuring the planner's true history is represented.

3 RESULTS

We implemented this approach in the Julia language [4], representing policies as feedforward neural networks with Flux.jl [9]. We applied it on a two-dimensional, continuous pursuit-evasion game Tag, in which agents control their velocity, and their position x_{pos} evolves through simple integrator dynamics for $T_{\text{future}} = T_{\text{past}} = 7$ timesteps. The pursuer minimizes $||x^{(1)}_{\text{pos}} - x^{(2)}_{\text{pos}}||_2$ while the evader maximizes it at every timestep. Players know their own locations, and observe the opponent with Gaussian noise proportional to the opponent's angular distance outside a fixed field of view in the direction of motion. Initial positions are normally distributed around the origin. There is no process noise.

We also consider two variants: in TagChain N players alternately pursue or evade the next indexed player, and in Hide&Seek a number of visual and physical obstacles are introduced.

		Passive pursuer(s)	Active pursuer(s)
Tag	Passive evader	12.58 ± 1.44 -12.58 ± 1.44	9.97 ± 1.19 -9.97 ± 1.19
	Active evader	13.97 ± 1.92 -13.97 ± 1.92	12.98 ± 1.18 -12.98 ± 1.18
TagChain	Passive evaders	17.68 ± 1.39 -25.91 ± 1.54	16.79 ± 1.43 -25.12 ± 1.57
	Active evaders	15.79 ± 1.23 -27.48 ± 1.39	15.71 ± 1.26 -27.29 ± 1.41
Hide&Seek	Passive evader	19.62 ± 1.33 -19.62 ± 1.33	17.93 ± 1.34 -17.93 ± 1.34
	Active evader	18.91 ± 1.25 -18.91 ± 1.25	16.75 ± 1.34 -16.75 ± 1.34

Table 1: Costs in passive/active configurations, per scenario

Table 1 summarizes the average costs and standard errors for each possible configuration in each game over 20 trials and 20 timesteps. ("Active," as in "active information gathering," is our method.) Indeed, in almost all cases, both pursuer cost (blue) and evader cost (red) decrease when the corresponding player gathers information actively. (The sole exception is Hide&Seek's evader, which intuitively has little incentive to gather information.)

4 CONCLUSION

This work demonstrates the potential for online planning in partially-observable continuous games via model predictive game play. Improvements in handling mixed strategies and finite recall are important avenues for further development. In our experiments, the method converged in tens of seconds before any code or hardware optimizations — placing real-time, imperfect-information planning, as required for multiple robotics applications, within reach.

REFERENCES

[1] Stefano V Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press.

[2] Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. 2022. Mastering the game of no-press Diplomacy via human-regularized reinforcement learning and planning. *arXiv preprint arXiv:2210.05492* (2022).

[3] Tyler Becker and Zachary Sunberg. 2024. Bridging the Gap between Partially Observable Stochastic Games and Sparse POMDP Methods. http://arxiv.org/abs/2405.18703 arXiv:2405.18703 [cs].

[4] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM review* 59, 1 (2017), 65–98.

[5] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. Deep counterfactual regret minimization. In *International conference on machine learning*. PMLR, 793–802.

[6] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. 2019. Algames: A fast solver for constrained dynamic games. *arXiv preprint arXiv:1910.09713* (2019).

[7] Piotr J Gmytrasiewicz and Prashant Doshi. 2004. Interactive pomdps: Properties and preliminary results. In *International Conference on Autonomous Agents: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-*, Vol. 3. 1374–1375.

[8] Johannes Heinrich and David Silver. 2016. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121* (2016).

[9] Mike Innes. 2018. Flux: Elegant machine learning with Julia. *Journal of Open Source Software* 3, 25 (2018), 602.

[10] Vojtěch Kovařík, Martin Schmid, Neil Burch, Michael Bowling, and Viliam Lisỳ. 2022. Rethinking formal models of partially observable multiagent decision making. *Artificial Intelligence* 303 (2022), 103645.

[11] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems* 30 (2017).

[12] Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. 2022. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science* 378, 6623 (2022), 990–996.

[13] Robert Platt Jr, Russ Tedrake, Leslie Pack Kaelbling, and Tomas Lozano-Perez. 2010. Belief space planning assuming maximum likelihood observations.. In *Robotics: Science and systems*, Vol. 2.

[14] Wilko Schwarting, Alyssa Pierson, Sertac Karaman, and Daniela Rus. 2021. Stochastic dynamic games in belief space. *IEEE Transactions on Robotics* 37, 6 (2021), 2157–2172.

[15] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *nature* 575, 7782 (2019), 350–354.

[16] Nikos Vlassis, Michael L Littman, and David Barber. 2012. On the computational complexity of stochastic controller optimization in POMDPs. *ACM Transactions on Computation Theory (TOCT)* 4, 4 (2012), 1–8.

[17] Zijian Wang, Riccardo Spica, and Mac Schwager. 2019. Game theoretic motion planning for multi-robot racing. In *Distributed Autonomous Robotic Systems: The 14th International Symposium*. Springer, 225–238.

[18] Yaodong Yang and Jun Wang. 2020. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583* (2020).