Automatica 89 (2018) 28–36

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Efficient path planning algorithms in reach-avoid problems☆

Zhengyuan Zhou [a,*], Jerry Ding [b], Haomiao Huang [c], Ryo Takei [b], Claire Tomlin [b]

[a] Department of Electrical Engineering, Stanford University, Stanford, CA, 94305, USA
[b] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, 94720, USA
[c] Kuna Systems, CA, USA

ARTICLE INFO

Article history:
Received 10 January 2014
Received in revised form 7 September 2017
Accepted 10 October 2017

Keywords:
Reach-avoid games
Fast marching methods
Path planning

ABSTRACT

We consider a multi-player differential game, referred to as a reach-avoid game, in which one set of attacking players attempts to reach a target while avoiding both obstacles and capture by a set of defending players. Unlike pursuit–evasion games, in this reach-avoid game one set of players must not only consider the other set of players, but also the target. This complexity makes finding solutions to such games computationally challenging, especially as the number of players grows. We propose an approach to solving such games in an open-loop sense, where the players commit to their control actions prior to the beginning of the game. This reduces the dimensionality of the required computations, thus enabling efficient computation of feasible solutions in real time for domains with arbitrary obstacle topologies. We describe two such formulations, each of which is conservative towards one side, and derive numerical algorithms based upon modified fast-marching methods (FMM) for computing their solutions.

1. Introduction

In a reach-avoid game, one set of players (attackers) attempts to arrive at a target set in the state-space, while avoiding a set of unsafe states, as well as interceptions by an opposing set of players (defenders). Such problems encompass a large number of robotics and control applications. For example, many safe motion-planning problems (see Karaman and Frazzoli, 2011b and the references therein) may be formulated as reach-avoid games, in which the objective is to control one or more agents into a desired target region, while avoiding a set of obstacles or possibly adversarial agents. Reach-avoid games considered here belong to the general class of differential games (Friedman, 1971; Isaacs, 1967), which include a variety of interesting problems (e.g. pursuit–evasion games Flynn, 1974, Lewin, 1986, network consensus problems under adversarial attacks Khanafer, Touri, & Basar, 2012, 2013, motion planning Chen, Zhou, & Tomlin, 2014, 2017; Karaman & Frazzoli, 2011a) and have been a subject of significant past research. For reach-avoid related games, the approaches that have been proposed often feature trade-offs between optimality of the solution (with respect to the time to achieve a player's objective), and the complexity of the computation.

Our approach in this paper to reach-avoid games is to formulate them as an open-loop game, and more specifically, as a framework of open-loop games that includes the open-loop upper value game and open-loop lower values game. The open-loop games for the reach-avoid problem formulated here are an instantiation of a general Stackelberg game (Başar & Olsder, 1999), an important class of games for modeling strategic behavior in dynamic games. In an open-loop game, the granularity of a strategy at which a player chooses is a control function that maps the entire time horizon to a trajectory. Depending on which player(s) choose(s) first, and which one(s) choose(s) in response, they are leader(s) and follower(s) respectively.

The open-loop games are conservative towards one side of players: the side that chooses first. Consequently, this level of conservatism offers performance guarantees of the solution. Namely, if a solution exists, the player is guaranteed to achieve the desired objective, irrespective of the actions of the opponent, without needing to incorporate any future state information. This is particularly well-suited for certain safety-critical applications, where state update is hard or costly to obtain, for example in GPS-denied environments. Another safety-critical application is robotic surgery, where the robotic system needs to navigate inside human body to

☆ This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N0014-08-0696) and SMARTS (N00014-09-1-1051) MURIs and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567). The material in this paper was presented at Proceedings of the IEEE International Conference on Robotics and Automation, May 14–18, 2012, St. Paul, Minnesota, USA; Proceedings of the 51st IEEE Conference on Decision and Control, December 10–13, 2012, Maui, Hawaii, USA. This paper was recommended for publication in revised form by Associate Editor Hideaki Ishii under the direction of Editor Ian R. Petersen.

* Corresponding author.
E-mail addresses: zyzhou@stanford.edu (Z. Zhou), jerryding@berkeley.edu (J. Ding), haomiao@stanford.edu (H. Huang), rrtakei@eecs.berkeley.edu (R. Takei), tomlin@eecs.berkeley.edu (C. Tomlin).

https://doi.org/10.1016/j.automatica.2017.11.035
0005-1098/

eliminate certain tissues while ensuring that no invasive damage is induced. Those safety-critical applications tend to demand the solution to satisfy strong performance guarantees (e.g. achieving the goal regardless of what the disturbance does, adversarial or non-adversarial alike), a property guaranteed by the open-loop framework.

In a static Stackelberg game (only one-round of interaction between players), an exhaust tree search can be used to find the optimal action (Başar & Olsder, 1999). However, at least in the context of reach-avoid open-loop dynamic games, no efficient algorithms have been devised to produce the optimal controls. Consequently, this paper is particularly focused upon the study of efficient and scalable computational algorithms for solving open-loop reach-avoid games. We emphasize that the proposed approach does not involve solving *any* Hamilton–Jacobi–Isaacs (HJI) equations[1] in the high dimensional space of all player states. Instead, as discussed in detail later, we reduce the problem to solving Hamilton–Jacobi–Bellman (HJB) equations in the low dimensional space of individual player states.

1.1. Related work

For certain games, it is possible to construct strategies (sometimes even optimal) for the players analytically or geometrically. Defending a line segment on a plane without obstacles has been considered in Kawecki, Kraska, Majcherek, and Zola (2009) and Rzymowski (2009), where the motion has been restricted to fixed maximum speed and moving along line segments, respectively. In both cases, optimal strategies have been found for defending the target set (a line segment). A class of methods has been proposed for safe motion-planning in the presence of moving obstacles by computing the future set of states an obstacle may occupy, given the dynamics of the controlled agent and the obstacles. This set of states are then treated as obstacles in the joint state-time space, and paths are planned which avoid these states (Fiorini & Shiller, 1998; Fraichard & Asama, 2004; Van Den Berg & Overmars, 2008). They are well suited for scenarios in which the obstacle motion can be unpredictable or even adversarial, so that in the presence of hard constraints on safety, one needs to account for the worst-case possibility that the disturbances may actively attempt to collide with the agent. However, these approaches tend to be limited to simple game configurations without complex static obstacle configurations or inhomogeneous speed constraints from varying terrain, issues that our formulation (Section 2.1) captures and that our computational framework (Section 4) addresses.

For general cases, the classical approach is to formulate the game as a minimax problem, in which a value function is defined representing the time-to-reach of the attackers at the target, subject to the constraint that it is not captured by the defender. This value is then computed via a related Hamilton–Jacobi–Isaacs (HJI) partial differential equation (PDE), with appropriate boundary conditions (Başar & Olsder, 1999; Evans & Souganidis, 1984; Isaacs, 1967). Solutions are typically found either using the method of characteristics (Başar & Olsder, 1999; Isaacs, 1967), where optimal trajectories are computed by integrating backward from a known terminal condition, or via numerical approximation of the HJI equation on grids (Falcone & Ferretti, 2002). For the HJI method, the number of grid points needed to approximate the value function typically grow exponentially in the number of continuous states. As such, finding solution strategies for such games can be computationally expensive, even for games involving a single attacker and a single defender. For differential games with multiple players, the computational burden also scale exponentially in the number of players. On a related note, in particular differential game scenarios, approximate dynamic programming (ADP) techniques have been developed to efficiently compute game solutions. For example, clever policy iterations have been designed to solve certain two-player zero-sum games (Johnson, Bhasin, & Dixon, 2011; Vamvoudakis & Lewis, 2012). However, the differential game scenario considered in this paper has a non-smooth value function. The application of ADP methods to such scenarios has been found to be challenging from a numerical convergence standpoint (Munos, Baird, & Moore, 1999). In closing, we mention that another related research thread concerns reach-avoid games under imperfect or incomplete information (Doyen & Raskin, 2010; Sebbane, 2014), where the players do not necessarily know the locations of the other players at all times.

As a final note, closely related to reach-avoid problems is the class of pursuit–evasion problems (Flynn, 1974; Lewin, 1986; Liu, Zhou, Tomlin, & Hedrick, 2013; Zhou et al., 2016). A reach-avoid game shares some similar aspects of a pursuit–evasion game: the attacker (similar to an evader) needs to avoid the capture of the defender (similar to a pursuer) because capture would result in the attackers losing the game instantly. However, there is a key distinction between the two: the utility of an agent in a reach-avoid game is fundamentally different from that of a pursuit–evasion game. In the latter, an evader's utility is solely based on whether it will ever be captured, whereas in the former, the attacker's utility depends on how long it takes to reach the target. Consequently, a reach-avoid problem is much more challenging because the attacker not only needs to avoid capture, but also needs to reach a per-determined target set. The recent work (Zhou et al., 2016) studies a multi-pursuer–single-evader pursuit–evasion problem, where an analytical cooperative pursuit strategy for the pursuers is derived using Voronoi partitions. This geometric approach (and the analytical results therein) are feasible because all agents in the pursuit–evasion problem are assumed to have constant speed. Conversely, in our current reach-avoid setting, in addition to the complexity just mentioned, we also allow for arbitrary speed profiles for all agents. As a result, these two levels of generality dictate that a completely different and non-analytical approach be taken.

1.2. Our contributions

In this paper, we provide a computationally tractable framework for solving open-loop reach-avoid games. Our major contribution is the algorithmic framework for computing open-loop values. Specifically, we develop (Section 4) efficient and novel numerical algorithms (a set of modified fast-marching methods) that allow us to quickly compute solutions to these open-loop games, in the form of a set of open-loop player trajectories with provable properties. To the best of our knowledge, this is the first set of efficient algorithms for computing open-loop values. We note that the two open-loop values, in addition to being interesting for study in their own right, also provide bounds on the closed-loop value (in general the solution to an HJI equation, which is typically intractable to solve) of the reach-avoid game. In this sense, our open-loop framework can be interpreted as a computationally efficient approximation of the closed-loop value for reach-avoid games, through the trade-off of a certain degree of optimality for a reduction in computational complexity. In particular, there exist non-trivial cases where solving for open-loop values also yield closed-loop values. We also emphasize the independent modeling value of the open-loop formulations: since they are conservative towards one set of players, the resulting solutions provide worst-case guarantees that will be useful in safety-critical applications. Some of the theoretical results were presented in two previous publications (Takei, Huang, Ding, & Tomlin, 2012; Zhou, Takei, Huang, & Tomlin, 2012). This work unifies the presentation of the

[1] This corresponds to a closed-loop formulation. See Section 1.1 for a discussion and related work.