Nash Equilibria in Finite-Horizon Multiagent Concurrent Games

Senthil Rajasekaran
Computer Science Department
Rice University
sr79@rice.edu

Moshe Y. Vardi
Computer Science Department
Rice University
vardi@rice.edu

ABSTRACT

The problem of finding pure strategy Nash equilibria in multiagent concurrent games with finite-horizon temporal goals has received some recent attention. Earlier work solved this problem through the use of Rabin automata. In this work, we take advantage of the finite-horizon nature of the agents' goals and show that checking for and finding pure strategy Nash equilibria can be done using a combination of safety games and lasso testing in Büchi automata. To separate strategic reasoning from temporal reasoning, we model agents' goals by deterministic finite-word automata (DFAs), since finite-horizon logics such as $\mathrm{LTL_f}$ and $\mathrm{LDL_f}$ are reasoned about through conversion to equivalent DFAs. This allow us characterize the complexity of the problem as PSPACE complete.

KEYWORDS

Boolean Games; Nash Equilibria; Automata; Temporal Logics

ACM Reference Format:
Senthil Rajasekaran and Moshe Y. Vardi. 2021. Nash Equilibria in Finite-Horizon Multiagent Concurrent Games. In *Proc. of the 20th International Conference on Autonomous agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

Game theory provides a powerful framework for modeling problems in system design and verification [4, 8, 17]. In particular, two-player games have been used in synthesis problems for temporal logics [14]. In these games, one player takes on the role of the system that tries to realize a property and the other takes on the role of the environment that tries to falsify the property. Within the scope of multiplayer games, two-player zero-sum games are the easiest to analyze, since they are purely adversarial – there is no reason for either player to do anything but maximize their own utility at the expense of the other.

When there are multiple agents with multiple goals, pure antagonism is not a reasonable assumption [19]. *Concurrent games* are a fundamental model of such multiagent systems [1, 11]. *Iterated Boolean Games* (iBG) [5] are a restriction of concurrent games introduced in part to generalize temporal synthesis problems to the multiagent setting. In an iBG, each agent has a temporal goal, usually expressed in *Linear Time Temporal Logic* (LTL) [13], and is given control over a unique set of boolean variables. At each time step, the agents collectively decide a setting to all boolean variables by individually and concurrently assigning values to their own variables. This creates an infinite sequence of boolean assignments (a *trace*) that is used to determined which goals are satisfied and which are not [5]. In this paper, we generalize the iBG formalism slightly to admit arbitrary finite alphabets rather than just truth assignments to boolean variables, as discussed below.

The concept of the *Nash Equilibrium* [12] is widely accepted as an important notion of a solution in multiagent games and represents a situation where agents cannot improve their outcomes unilaterally. In this paper we consider deterministic agents, and therefore the notion of a Nash equilibrium in this paper that of pure strategy Nash equilibrium [16]. While the problem of finding Nash equilibria in an iBG where the agents have infinite-horizon temporal goals is well studied, the analogous problem with finite-horizon temporal goals has only recently received attention [6]. In that work the automated equilibrium analysis is done through reasoning about automata on infinite words, specifically, *Rabin automata*. In this work we use simpler constructions - both safety games and *Büchi automata*.

Here we address a more abstract version of the multi-agent finite-horizon temporal-equilibrium problem by analyzing concurrent iterated games in which each agent is given their own *Deterministic Finite Word Automata* (DFA) goal. The reason for this is twofold. First, essentially all finite-horizon temporal logics are reasoned about through conversion to equivalent DFA, including the popular logics $\mathrm{LTL_f}$ and $\mathrm{LDL_f}$ [2, 3]. Thus, using DFA goals offers us a general way of dealing with a variety of temporal formalisms. Furthermore, using DFA goals enables us to separate the complexity of temporal reasoning from the complexity of strategic reasoning. Our focus on DFAs also ties in to a growing interest in DFAs as graphical models that can be reasoned about directly in a number of related fields; see [7, 10, 20] for a few examples in the context of machine learning. In this work we prove that, once a set of agents W is fixed, determining whether a pure strategy Nash equilibrium in which only the agents in W have their goals met exists is PSPACE-complete. The reason why we use the set W is that we approach the problem from the perspective of a system planner, and the set W represents the agents whose goals we would like to see met.

2 CONCURRENT GAMES AND IBGS

A *concurrent game structure* (CGS) is an 8-tuple

$$(Prop, \Omega, (A_i)_{i\in\Omega}, S, \lambda, \tau, s_0 \in S, (A^i)_{i\in\Omega})$$

where $Prop$ is a finite set of *propositions*, $\Omega = \{0, \dots k-1\}$ is a finite set of *agents*, A_i is a set of *actions*, where each A_i is associated with an agent i (we also construct the set of *decisions* $D = A_0 \times A_1 \dots A_{k-1}$, S is a set of *states*, $\lambda : S \to 2^{Prop}$ is a *labeling function* that associates each state with a set of propositions that are interpreted as true in that state, $\tau : S \times D \to S$ is a deterministic *transition function* that takes a state and a decision as input and

Work supported in part by NSF grants IIS-1527668, CCF-1704883, IIS-1830549, and an award from the Maryland Procurement Office. For a longer technical report, see [15].

Proc. of the 20th International Conference on Autonomous agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

returns another state, s_0 is a state in S that serves as the *initial state*, and A^i is a DFA associated with agent i. A DFA A^i is denoted as the goal of agent i. Intuitively, agent i prefers plays in the game that satisfy A^i, that is a play such that some finite prefix of the play is accepted by A^i. It is for this reason we refer to A^i as a "goal".

We now define *iterated boolean games* (iBG), a restriction on the CGS formalism. Our formulation is slight generalization of the iBG framework introduced in [5], as we take the set of actions to be a finite alphabet rather than a set of truth assignments since we are interested in separating temporal reasoning from strategic reasoning. An iBG is defined by applying the following restrictions to the CGS formalism. Each agent i is associated with its own alphabet Σ_i. These Σ_i are disjoint and each Σ_i serves as the set of actions for agent i; an action for agent i consists of choosing a letter in Σ_i. The set of decisions is then $\Sigma = \bigtimes_{i=0}^{k-1} \Sigma_i$. The set of states corresponds to the set of decisions Σ; there is a bijection between the set of states and the set of decisions. The labeling function mirrors the element of Σ associated with each state. As in [5], we still have $\lambda(s) = s$, but with $s \in \Sigma$ now. As a slight abuse of notation, we consider the "proposition" $\sigma \in \Sigma_i$ for some i to be true at state s if σ appears in s, allowing us to generalize towards arbitrary alphabets. Finally, the transition function τ is simply right projection $\tau(s, d) = d$.

Now, we introduce a few essential definitions.

Definition 2.1 (Strategy for agent i). A strategy for agent i is a function $\pi_i : S^* \to A_i$. Intuitively, this is a function that, given the observed history of the game (represented by an element of S^*), returns an action $a_i \in A_i$.

Definition 2.2 (Strategy Profile). Let Π_i represent the set of strategies for agent i. Then, we define the set of strategy profiles $\Pi = \bigtimes_{i \in \Omega} \Pi_i$

Definition 2.3 (Primary Trace resulting from a Strategy Profile). Given a strategy profile π, the primary trace of π is the unique trace t that satisfies

(1) $t[0] = \pi(\epsilon)$
(2) $t[i] = \pi(t[0], \ldots t[i-1])$

We denote this trace as t_π.

Given a trace $t \in S^\omega$, define the *winning set* $W_t = \{i \in \Omega \ : \ t \models A^i\}$ to be the set of agents whose DFA goals are satisfied by a finite prefix of the trace t. The *losing set* is then defined as Ω / W_t.

Definition 2.4 (Nash Equilibrium). [5] Let G be an iBG and $\pi = \langle \pi_0, \pi_1 \ldots \pi_{k-1} \rangle$ be a strategy profile. We denote $W_\pi = W_{t_\pi}$. The profile π is a *Nash equilibrium* if for every $i \in \Omega / W_t$ we have that given all strategy profiles of the form $\pi' = \langle \pi_0, \pi_1 \ldots \pi_i' \ldots \pi_{k-1} \rangle$, for every $\pi_i' \in \Pi_i$, it is the case that $i \in \Omega / W_{\pi'}$.

Definition 2.5 (W-NE Strategy Profile). Let G be an iBG, $W \subseteq \Omega$ a set of agents and π a strategy profile in G. We say π is a *W-NE strategy profile* if it is a Nash equilibria in which $W_\pi = W$.

3 COMPLEXITY

The main result of our work is the characterization of the following problem:

Given an iBG G and a set of agents W, does a W-NE strategy profile exist?

as PSPACE-complete. In order to do so, we establish novel upper and lower bounds for the problem. We only present a high level overview here; a full write up can be found at [15].

3.1 Upper Bound

We characterize our notion of a Nash equilibrium as two separate conditions. Thus, a strategy profile π is a W-NE strategy profile in an iBG G iff it satisfies both the

(1) Primary-Trace Condition: The primary infinite trace t_π defined by π satisfies the goals A^j precisely for $j \in W$. The trace $t_\pi = x_0, x_1, \ldots$ for π is once again defined as follows
  (a) $x_0 = \varepsilon$
  (b) $x_{i+1} = x_0, \ldots, x_i, \pi(x_0, \ldots, x_i)$
(2) j-Deviant-Trace Condition: Each j-*deviant* trace $t = y_0, y_1, \ldots$, for $j \notin W$, does not satisfy the goal A^j.
  For $\alpha \in \Sigma$, we introduce the notation $\alpha[-j]$ to refer to $\alpha|_{\Sigma \setminus \Sigma_j}$ (that is, α with Σ_j projected out). A trace $t = y_0, y_1, \ldots$ is j-deviant if
  (a) $y_0 = \varepsilon$
  (b) $y_{i+1} = y_0, \ldots, y_i, \alpha$, where $\alpha \in \Sigma$ and $\alpha[-j] = \pi(y_i)[-j]$
  (c) t is not the primary trace

In order to capture this property we create a deterministic top-down Büchi tree automata T_W, which recognizes all W-NE strategies in G, and then test it for nonemptiness to see if a W-NE strategy exists. Our approach allows us to separate reasoning about the j-Deviant-Trace Condition from reasoning about the Primary-Trace Condition. Specifically, we consider the j-Deviant-Trace Condition through solving a series of safety games, and then test for both the Primary-Trace Condition and the j-Deviant-Trace Condition by testing a Büchi word automaton for nonemptiness, a problem which can generally be done in NLOGSPACE [18]. Since our automaton is exponential in size, we get

Theorem 3.1. *The problem of deciding whether there exists a W-NE strategy profile for an iBG G and a set $W \subseteq \Omega$ of agents is in PSPACE.*

3.2 Lower Bound

The lower bound comes from a reduction from the PSPACE-complete problem of DFA Intersection Emptiness (DFAIE). The DFAIE problem is as follows: Given k DFAs $A^0 \ldots A^{k-1}$ with a common alphabet Σ, decide whether $\bigcap_{0 \leq i \leq k-1} A^i \neq \emptyset$ [9]. We are able to incorporate this in our formalism by applying a synchronization modification to the goal DFAs in an iBG which takes away the temporal aspect of our problem. Therefore,

Theorem 3.2. *The problem of deciding whether there exists a W-NE strategy profile for an iBG G and a set $W \subseteq \Omega$ of agents is PSPACE-hard.*

Theorem 3.3. *The problem of deciding whether there exists a W-NE strategy profile for an iBG G and a set $W \subseteq \Omega$ of agents is PSPACE-complete.*

REFERENCES

[1] Rajeev Alur, Thomas A Henzinger, and Orna Kupferman. 2002. Alternating-time temporal logic. *J. ACM* 49, 5 (2002), 672–713.

[2] Giuseppe De Giacomo and Moshe Y. Vardi. 2013. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, Francesca Rossi (Ed.). IJCAI/AAAI, 854–860. http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6997

[3] Giuseppe De Giacomo and Moshe Y. Vardi. 2015. Synthesis for LTL and LDL on Finite Traces. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Qiang Yang and Michael J. Wooldridge (Eds.). AAAI Press, 1558–1564. http://ijcai.org/Abstract/15/223

[4] E. Grädel, W. Thomas, and T. Wilke. 2002. *Automata, Logics, and Infinite Games: A Guide to Current Research*. Springer.

[5] Julian Gutierrez, Paul Harrenstein, and Michael J. Wooldridge. 2015. Iterated Boolean games. *Inf. Comput.* 242 (2015), 53–79. https://doi.org/10.1016/j.ic.2015.03.011

[6] Julian Gutierrez, Giuseppe Perelli, and Michael J. Wooldridge. 2017. Iterated Games with LDL Goals over Finite Traces. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee (Eds.). ACM, 696–704. http://dl.acm.org/citation.cfm?id=3091225

[7] Mohammadhosein Hasanbeig, Natasha Yogananda Jeppu, Alessandro Abate, Tom Melham, and Daniel Kroening. 2019. DeepSynth: Program Synthesis for Automatic Task Segmentation in Deep Reinforcement Learning. *CoRR* abs/1911.10244 (2019). arXiv:1911.10244 http://arxiv.org/abs/1911.10244

[8] Thomas A. Henzinger. 2005. Games in system design and verification. In *Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2005), Singapore, June 10-12, 2005*, Ron van der Meyden (Ed.). National University of Singapore, 1–4. https://dl.acm.org/citation.cfm?id=1089935

[9] Dexter Kozen. 1977. Lower Bounds for Natural Proof Systems. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*. IEEE Computer Society, 254–266. https://doi.org/10.1109/SFCS.1977.16

[10] Joshua J. Michalenko, Ameesh Shah, Abhinav Verma, Richard G. Baraniuk, Swarat Chaudhuri, and Ankit B. Patel. 2019. Representing Formal Languages: A Comparison Between Finite Automata and Recurrent Neural Networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=H1zeHnA9KX

[11] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y Vardi. 2014. Reasoning about strategies: On the model-checking problem. *ACM Trans. on Computational Logic* 15, 4 (2014), 1–47.

[12] John F. Nash. 1950. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences* 36, 1 (1950), 48–49. https://doi.org/10.1073/pnas.36.1.48 arXiv:https://www.pnas.org/content/36/1/48.full.pdf

[13] Amir Pnueli. 1977. The Temporal Logic of Programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*. IEEE Computer Society, 46–57. https://doi.org/10.1109/SFCS.1977.32

[14] Amir Pnueli and Roni Rosner. 1989. On the Synthesis of a Reactive Module. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*. ACM Press, 179–190. https://doi.org/10.1145/75277.75293

[15] Senthil Rajasekaran and Moshe Y. Vardi. 2021. Nash Equilibria in Finite-Horizon Multiagent Concurrent Games. arXiv:2101.00716 [cs.GT]

[16] Yoav Shoham and Kevin Leyton-Brown. 2009. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

[17] Johan van Benthem. 2011. Logic Games: From Tools to Models of Interaction. In *Proof, Computation and Agency - Logic at the Crossroads*, Johan van Benthem, Amitabha Gupta, and Rohit Parikh (Eds.). Synthese library, Vol. 352. Springer, 183–216. https://doi.org/10.1007/978-94-007-0080-2_11

[18] M.Y. Vardi and P. Wolper. 1994. Reasoning about Infinite Computations. *Information and Computation* 115, 1 (1994), 1–37.

[19] Michael J. Wooldridge. 2009. *An Introduction to MultiAgent Systems, Second Edition*. Wiley.

[20] Eran Yahav. 2018. From Programs to Interpretable Deep Models and Back. In *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10981)*, Hana Chockler and Georg Weissenbacher (Eds.). Springer, 27–37. https://doi.org/10.1007/978-3-319-96145-3_2