International Journal of Approximate Reasoning 93 (2018) 290–326

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Approximating maxmin strategies in imperfect recall games using A-loss recall property

Jiří Čermák *, Branislav Bošanský *, Karel Horák, Viliam Lisý, Michal Pěchouček

Department of Computer Science, Czech Technical University in Prague, Czechia

ARTICLE INFO

Article history:
Received 27 June 2017
Received in revised form 14 November 2017
Accepted 17 November 2017
Available online 23 November 2017

Keywords:
Imperfect recall
Abstraction
Maxmin strategy
A-loss recall

ABSTRACT

Extensive-form games with imperfect recall are an important model of dynamic games where the players are allowed to forget previously known information. Often, imperfect recall games result from an abstraction algorithm that simplifies a large game with perfect recall. Solving imperfect recall games is known to be a hard problem, and thus it is useful to search for a subclass of imperfect recall games which offers sufficient memory savings while being efficiently solvable. The abstraction process can then be guided to result in a game from this class. We focus on a subclass of imperfect recall games called A-loss recall games. First, we provide a complete picture of the complexity of solving imperfect recall and A-loss recall games. We show that the A-loss recall property allows us to compute a best response in polynomial time (computing a best response is NP-hard in imperfect recall games). This allows us to create a practical algorithm for approximating maxmin strategies in two-player games where the maximizing player has imperfect recall and the minimizing player has A-loss recall. This algorithm is capable of solving some games with up to $5 \cdot 10^9$ states in approximately 1 hour. Finally, we demonstrate that the use of imperfect recall abstraction can reduce the size of the strategy representation to as low as 0.03% of the size of the strategy representation in the original perfect recall game without sacrificing the quality of the maxmin strategy obtained by solving this abstraction.

1. Introduction

Extensive-form games (EFGs) are a model of dynamic games with a finite number of moves and are capable of describing scenarios with stochastic events and imperfect information. EFGs can model recreational games, such as poker, as well as real-world situations in physical security [1], auctions [2], or medicine [3]. EFGs are represented as game trees where nodes correspond to states of the game and edges to actions of players. Imperfect information is modeled by grouping indistinguishable states into information sets.

There are two approaches to making decisions in EFGs. First, there are online (or game-playing) algorithms which given the observations of the game state compute the action to be played. Second, there are offline algorithms which compute (approximate) the strategy in the whole game and play according to this strategy. The latter algorithms typically provide a better approximation of equilibrium strategies in large games compared to online algorithms [4]. One exception is the recently introduced continual resolving algorithm used in DeepStack [5], which provides less exploitable strategies than existing offline algorithms in heads-up no-limit Texas Hold'em poker. The main caveat of this algorithm is that it exploits

* Corresponding authors.
E-mail addresses: cermak@agents.fel.cvut.cz (J. Čermák), bosansky@agents.fel.cvut.cz (B. Bošanský), horak@agents.fel.cvut.cz (K. Horák), lisy@agents.fel.cvut.cz (V. Lisý), pechoucek@agents.fel.cvut.cz (M. Pěchouček).

https://doi.org/10.1016/j.ijar.2017.11.010
0888-613X/

the specific structure of poker where all actions of players are observable and its generalization to other games is not straightforward. Therefore, we focus on offline algorithms.

The offline algorithms are useful in real-world applications since the strategy (a probabilistic distribution over actions in each information set) is precomputed and can be simply stored on any device. It can then be accessed by deployed units such as park rangers (see, e.g., [6]) without the need of large computational resources or good internet connection necessary when using online algorithms. The main complication of offline algorithms is, however, the size of the strategy that needs to be stored. Most of the existing offline algorithms [7–9] require that players remember all the information gained during the game – a property denoted as a *perfect recall*. The main disadvantage of perfect recall is that the size of the strategy grows exponentially with the number of moves, as the perfect memory allows the player to condition his behavior on all his actions taken in the past. Therefore, a popular approach is to use *abstractions* [10] – create an abstracted game by merging certain information sets to reduce the size of the strategy representation, solve the abstracted game, and translate the resulting strategy back to the original game. The majority of existing algorithms (e.g., see [11–13]) create perfect recall abstractions, where the requirement of perfect memory severely limits possible reductions in the size of strategies, as it still grows exponentially with the increasing number of moves in the abstracted game. To achieve additional memory savings, the assumption of perfect recall may need to be violated in the abstracted game resulting in an *imperfect recall game*.

Solving imperfect recall games is known to be a difficult problem (see, e.g., [14–16]). Since we are interested mainly in solving imperfect recall games created by an abstraction algorithm, we focus on finding an efficiently solvable subclass of imperfect recall games. The abstraction algorithm can then be guided to result in a game from this class. Existing approaches create very specific abstracted games, so that perfect recall algorithms are still applicable: e.g., in *chance relaxed skew well-formed games* [17,18] or in *normal-form games with sequential strategies* [19,1]. The restrictions posed by these classes are unnecessarily strict, which can prevent us from fully exploiting the possibilities of abstractions and compact representation of dynamic games. We focus on a different subclass of imperfect recall games called *A-loss recall games* [20,21] where each loss of a memory of a player can be traced back to forgetting his own actions.

We provide the following contributions. We present a complete picture of the problem of solving imperfect recall games and show which computational tasks become easier when restricting to A-loss recall. Next, we use the properties of the A-loss recall to provide the first family of algorithms capable of approximating the strategies with the best worst-case expected outcome (maxmin strategies). In order to achieve this result, we require only the minimizing player to have A-loss recall, while the maximizing player is allowed to have imperfect recall. Finally, we experimentally demonstrate the effectiveness of the use of imperfect recall abstractions to reduce the size of strategies to be stored.

One of the most important theoretical properties discussed in this paper is the complexity of computing a best response since it is a subproblem of many algorithms solving EFGs. In general, it is NP-hard to find a best response in imperfect recall game. In games where the best responding player has A-loss recall, however, finding a best response can be done using the same algorithm as in the perfect recall case. Hence the problem is polynomially solvable [20,21]. We use this property to design the first family of algorithms for approximating maxmin strategies in imperfect recall games where the maximizing player has imperfect recall and the minimizing player has A-loss recall. Additionally, we provide novel necessary and sufficient (i.e., if and only if) condition for the existence of a Nash equilibrium (NE) in behavioral strategies in A-loss recall games, making A-loss recall games the only subclass of imperfect recall games for which such condition is known. Thus we show that A-loss recall forms an interesting subclass of imperfect recall. On the other hand, we extend the known hardness results of computing solution concepts in imperfect recall games due to Koller and Meggido [15] and Hansen [16] to A-loss recall games.[1]

From the computational perspective, we provide a novel approximate algorithm, denoted IRABnB (Imperfect Recall Abstraction Branch-and-Bound algorithm), for computing maxmin strategies in imperfect recall games where the maximizing player has imperfect recall and the minimizing player has A-loss recall.[2] We base the algorithm on the sequence-form linear program for computing maxmin strategies in perfect recall games [7,24] extended by bilinear constraints necessary for the correct representation of strategies of the maximizing player in imperfect recall games. We approximate the bilinear terms using recent Multiparametric Disaggregation Technique (MDT) [25] and provide a mixed-integer linear program (MILP) for approximating maxmin strategies. We propose a novel branch-and-bound algorithm that repeatedly solves the linear relaxation of the MILP. It simultaneously tightens the constraints that approximate bilinear terms and searches for the optimal assignment to the relaxed binary variables from the MILP. We prove that the algorithm has guaranteed convergence to maxmin strategy and we provide a bound on the number of steps needed.

We further extend the IRABnB algorithm by incremental strategy generation technique. The resulting algorithm is denoted DOIRABnB[3] (Double Oracle Imperfect Recall Abstraction Branch-and-Bound algorithm). While such techniques exist

[1] A part of this work appeared in [22]. Here we significantly improve the proofs and overall presentation. We further strengthen the relationship between A-loss recall games and chance relaxed skew well-formed games and provide more examples.

[2] A part of this work appeared in [23]. Here we mainly improve the description of the algorithm and proofs. We also provide a discussion of the heuristics used (section 5.2.1). Notice that in this paper we refer to the BnB algorithm from [23] as IRABnB to remove the naming clash with general branch-and-bound algorithm.

[3] A part of this work appeared in [26]. Here, we provide an improved version of the DOIRABnB algorithm which is more efficient. Furthermore, we significantly improve the description of the algorithm and the proof of correctness. Finally, we extend the experimental evaluation of the algorithm. Notice that in this paper we refer to the DOBnB algorithm from [26] as DOIRABnB to improve the clarity of presentation.