Solving Extensive-form Games with Double-oracle Methods (Doctoral Consortium)

Branislav Bošanský
Agent Technology Center, Dept. of Computer Science,
Faculty of Electrical Engineering, Czech Technical University in Prague
bosansky@agents.fel.cvut.cz

ABSTRACT

We investigate iterative algorithms for computing exact Nash equilibria in two-player zero-sum extensive-form games. The algorithms use an algorithmic framework of double-oracle methods. The main idea is to restrict the game by allowing the players to play only some of the strategies, and then iteratively solve this restricted game and exploit fast best-response algorithms to add additional strategies to the restricted game for the next iteration. The experimental evaluation on different games shows that the double-oracle methods often provide significant improvement in running-time, and can find exact solution of much larger games compared to the existing approaches.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: [Multiagent systems]

Keywords

game theory, extensive-form games, algorithms, iterative approaches

1. INTRODUCTION

Non-cooperative game theory provides a mathematical framework for describing the behavior of self-interested agents. Recent results in the field of computational game theory led to a number of applications of game-theoretic models, primarily in security domains [10]. However, the existing game-theoretic security models do not capture more complex strategies and tactics that arise in real-world situations, such as using scouts, decoys, or intelligence to retrieve information. Moreover, most of the existing security models focus on one-time interactions, when players do not gain significant information about the actions of the other players during the course of the game.

If we consider the opposite and allow the players to gain new information, the situations can be modeled as extensive-form games (EFGs) with imperfect information. In general, the uncertainty about the current state of the game and unknown moves of the opponent makes the games very hard to solve. Even the easiest subclass of zero-sum EFGs contains instances of pursuit-evasion games as well as classical board games, such as Poker or Kriegspiel (blind chess), solving which is a challenging and ongoing task. Therefore, it is of a great interest to study and design novel algorithms that significantly reduce the computational requirements and enable solving larger instances of EFGs. We focus on zero-sum games and we exploit the iterative approach known in game theory as the *oracle methods* [8] that can typically find a solution of the game without considering all possible strategies.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

2. TECHNICAL BACKGROUND AND RELATED WORK

Extensive-form games (EFGs) are visually represented as game trees. We study two-player games with *Nature* player capturing the stochastic events. Each node in the game tree is assigned to a player that acts in the situation represented by this node and a utility function is defined for the terminal nodes (leafs of the game tree). Player's uncertainty is represented using the *information sets* that form a partition over the nodes assigned to the player: every node in the game tree belongs to exactly one information set. When playing the game, the player knows the current information set, but has no information about the exact node. We assume *perfect recall*, which means that all nodes in any information set have the same history of actions for player acting in this information set (i.e., players cannot misremember their own actions).

2.1 Solving Extensive-form Games

A *pure strategy* of a player in an EFG is an assignment of an action to play in each information set assigned to the player and *mixed strategies* are probability distributions over pure strategies. Solving a game involves computing a profile of strategies (a single mixed strategy for each player) satisfying certain conditions defined by a *solution concept*. The most common is the Nash equilibrium solution concept, where each player plays the best response to the mixed strategies of the other players.

The standard way of computing an exact Nash equilibrium in EFGs is to use the compact *sequence-form* representation [6], where the sequences represent ordered lists of actions of a single player. With the sequence form, a Nash equilibrium can be found as a solution of a linear program (LP). However, solving a full LP is a very difficult computational task for problems of realistic size; hence, approximation methods are commonly used to solve these problems in practice. These methods include: regret minimization techniques (e.g., CFR) [11], later improved with sampling methods [7]; Nesterov's Excessive Gap Technique (EGT) [5]; or variants of Monte-Carlo tree search algorithms applied on imperfect information games (e.g., in [9]). The first two classes of algorithms guarantee convergence to approximate ϵ-Nash equilibrium, while the third method has no theoretical guarantees for imperfect-information games, although it can produce good strategies in practice [9]. Iterative double-oracle algorithms thus present an alternative that reduces the computational requirements of solving the full LP.

3. DOUBLE-ORACLE ALGORITHMS

Double-oracle algorithms are based on the research on decomposition methods for solving large-scale optimization problems and they have been successfully used to solve large normal-form games [8, 10]. Our goal is to adapt these methods for extensive-form games (EFGs) as well.

The generic structure of the double-oracle algorithm consists of iterating through three main steps until convergence: (1) create a restricted game by limiting the set of strategies that each player is allowed to play; (2) compute a pair of Nash equilibrium strategies in this restricted game; (3) for each player, compute a best response against the equilibrium strategy of the opponent (the best response may be *any* strategy in the complete game). The best-response strategies found in step 3 are added to the restricted game and allowed in the next iteration. The algorithm terminates if the value of the best response against the equilibrium strategies does not improve the game value of the restricted game. In the worst case, this approach may need to enumerate all possible strategies, but in typical cases a solution can be found by exploring a small fraction of the strategy space.

3.1 Double-oracle Algorithms for EFGs

First of all, we applied the double-oracle method to solve generic zero-sum EFGs with imperfect information. The main idea of works [1, 2] is to combine the sequence-form representation with the double-oracle method, and to iteratively expand the restricted game by allowing the players to play only some of the *sequences* of actions (instead of operating on the full strategy space). However, the naïve method of simply adding best-response sequences to the restricted game does not work, because the game may get malformed due to incompatibilities among the sequences resulting in incorrect solutions (a combination of sequences may be incompatible if the sequences of actions cannot be executed in full). Therefore, the algorithm creates temporary leafs in the nodes of the restricted game corresponding to the places of inconsistencies, and assigns temporary utility value corresponding to the lower bound on the utility values for the player that acts in this node to solve this issue.

The experimental results on search games and simplified variants of Poker show a great potential – the double-oracle algorithm requires significantly less memory compared to the full LP and in games with small support (i.e., the number of sequences actually used in the Nash equilibrium is low), the double-oracle algorithm is faster than the full LP in order of magnitudes.

Secondly, we applied the double-oracle method in EFGs with perfect information, but where the agents act in each state of the game simultaneously [3] – i.e., each node in the game tree corresponds to a normal-form game and a combination of actions of both players leads to a different normal-form game. The main idea of our method is to (1) use the double-oracle algorithm in each state of the game to iteratively construct and solve the corresponding normal-form game and to (2) improve the performance of the double-oracle algorithm using lower and upper bounds on the game values of the successors in each state of the game. These bound are effectively calculated by the classical alpha-beta search on serialized variants of the game. Again, the experimental results on the card game Goofspiel and pursuit-evasion games show great computational savings in orders of magnitude compared to the full search.

4. CONCLUSIONS AND FUTURE WORK

The benefits of the double-oracle algorithm are twofold. First of all, the algorithm can compute exact Nash equilibria for extensive-form games (EFGs) before constructing the complete game by identifying the promising actions that the players should play without any domain-specific knowledge. Secondly, this approach decomposes the problem of computing a Nash equilibrium into separate sub-problems. Therefore, it can be seen as a framework, in which the domain-independent methods can be replaced with the domain-specific ones to further improve the performance.

Experimental results show a great potential when using the double-oracle methods for solving EFGs. In future work, an adaptation of the algorithm for computing different solution concepts (refinements of Nash equilibria, Stackelberg equilibria) in EFGs is needed, since the Nash equilibrium has known weaknesses in EFGs, such as non-credible threats or prescription of irrational behavior in off-equilibrium paths. Secondly, this iterative approach can be further generalized to the games with even more complex space of strategies and imperfect recall, such as patrolling games [4], where Markov policies are typically used.

Acknowledgements

This research was supported by the Czech Science Foundation (grant no. P202/12/2054).

5. REFERENCES

[1] B. Bosansky, C. Kiekintveld, V. Lisy, and M. Pechoucek. Iterative Algorithm for Solving Two-player Zero-sum Extensive-form Games with Imperfect Information. In *Proc. of ECAI*, 2012.

[2] B. Bosansky, C. Kiekintveld, V. Lisy, J. Cermak, and M. Pechoucek. Double-oracle Algorithm for Computing an Exact Nash Equilibrium in Zero-sum Extensive-form Games. In *Proc. of AAMAS*, 2013.

[3] B. Bosansky, V. Lisy, J. Cermak, R. Vitek, and M. Pechoucek. Using Double-oracle Method and Serialized Alpha-Beta Search for Pruning in Simultaneous Moves Games. Under review process. Unpublished., 2013.

[4] B. Bosansky, V. Lisy, M. Jakob, and M. Pechoucek. Computing Time-Dependent Policies for Patrolling Games with Mobile Targets. In *Proc. of AAMAS*, 2011.

[5] S. Hoda, A. Gilpin, J. Peña, and T. Sandholm. Smoothing techniques for computing nash equilibria of sequential games. *Math. Oper. Res.*, 35(2):494–512, May 2010.

[6] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2), 1996.

[7] M. Lanctot, K. Waugh, M. Zinkevich, and M. Bowling. Monte carlo sampling for regret minimization in extensive games. In *Proc. of NIPS*, pages 1078–1086, 2009.

[8] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In *Proc. of ICML*, pages 536–543, 2003.

[9] M. J. V. Ponsen, S. de Jong, and M. Lanctot. Computing approximate nash equilibria and robust best-responses using sampling. *J. Artif. Intell. Res. (JAIR)*, 42:575–605, 2011.

[10] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[11] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. *Proc. of NIPS*, 20:1729–1736, 2008.