Available online at www.sciencedirect.com
SCIENCE @ DIRECT•
An International Journal
computers & mathematics with applications
ELSEVIER Computers and Mathematics with Applications 50 (2005) 823–854
www.elsevier.com/locate/camwa

Computation of Equilibria in Noncooperative Games

S. AZHAR*
Department of Computer Science and Software Engineering
Rose-Hulman Institute of Technology, 5500 Wabash Avenue, Terre Haute, IN 47803, U.S.A.

A. MCLENNAN†
Department of Economics, University of Minnesota
Minneapolis, MN 55455, U.S.A.

J. H. REIF‡
Department of Computer Science, Duke University
Durham, NC 27708, U.S.A.

(Received and accepted February 2005)

Abstract—This paper presents algorithms for finding equilibria of mixed strategy in multistage noncooperative games of incomplete information (like probabilistic blindfold chess, where at every opportunity a player can perform different moves with some probability). These algorithms accept input games in extensive form. Our main result is an algorithm for computing *sequential equilibrium*, which is the most widely accepted notion of equilibrium (for mixed strategies of noncooperative probabilistic games) in mainstream economic game theory. Previously, there were no known algorithms for computing sequential equilibria strategies (except for the special case of single stage games).

The computational aspects of passage from a recursive presentation of a game to its extensive form are also discussed. For nontrivial inputs the concatenation of this procedure with the equilibrium computation is time intensive, but has low spatial requirements. Given a recursively represented game, with a position space bound $S(n)$ and a log space computable next move relation, we can compute an example mixed strategy satisfying the sequential equilibria condition, all in space bound $O(S(n)^2)$, Furthermore, in space $O(S(n)^3)$, we can compute the connected components of mixed strategies satisfying sequential equilibria.[1]

1. INTRODUCTION

1.1. The Equilibrium Problem in Classical Game Theory

In recent years, there has been a proliferation of applications of noncooperative game theory to economics, political science, evolutionary biology, and other disciplines. A typical research

*Supported by DARPA/ARO Contract DAAL03-88-K-0195, DARPA/ISTO Contract N00014-88-K-0458, Office of Naval Research under contract N00014-87-K-0310, and Air Force contract AFSOR-87-0386. Also Supported by Duke University's James B. Duke Fellowship for Advanced Studies.
†Supported by NSF Grants MCS-8120790 and SES-8420114.
‡Supported by DARPA/ARO Contract DAAL03-88-K-0195, DARPA/ISTO Contract N00014-88-K-0458, Office of Naval Research under contract N00014-87-K-0310, and Air Force contract AFSOR-87-0386. Also, subsequently supported by National Science Foundation Grants CCF-0523555, CCF-0432038, CCF-0432047, ITR-0326157, EIA-0218376, EIA-0218359, and EIA-0086015.

0898-1221/05/$ - see front matter Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX
doi:10.1016/j.camwa.2005.02.015

project consists of modeling an environment or strategic interaction as a formally laid out game, after which conclusions are derived from the application of a solution concept that (hopefully) embodies the relevant consequences of (simultaneously) rational or optimizing behavior by all agents. Even for quite simple games, the computation of a solution concept by hand can be an arduous endeavor. In this paper, we present algorithms for the computation of some of the most important solution concepts of noncooperative game theory, and we derive complexity measures for these algorithms.

The algorithms are based on the observation that, for the type of finite game studied here, the solution concepts of interest are described by finite systems of (real) polynomial equations and inequalities. In general, a solution set of a finite system of polynomial equations and inequalities is called a *semialgebraic set*, and in recent years algorithms have been developed that pass from a given system of equations and inequalities to more direct and useful descriptions of the set determined by the system. For computer scientists, therefore, the main novelty in this paper is the elaboration of a new domain of application for these tools.

For pure and applied game theorists it is of interest to know that such algorithms are possible, of course, but it is also important to have a more detailed understanding of how they are constructed, and finally one should have some sense of their limitations. For the latter issue, the best available tools are the concepts of computational complexity studied in computer science. Our brief introduction to this subject is hopefully sufficient to enable the reader to understand the specific conclusions stated later.

Computer science has historically studied games from two perspectives. First, particular games present decision problems whose formal complexity can be studied. For example, one can consider the computational complexity of determining an optimal move in the game of *Go* as a function of the size of the board. Second, games can be used as models of particular computational problems or environments, particularly those involving parallel computation or networks of independent processors. Our work contributes to these traditions, so we will briefly review closely related literature.

As can be seen from the above description, this project is interdisciplinary, and the bulk of our work is expositional. Our aim is to provide each subaudience with enough information to appreciate the concepts and results of the unfamiliar discipline. The remainder of this introduction begins with a brief description of the game theoretic concepts studied here, followed by a quite cursory survey of the fundamental concepts of the theory of computation, intended for quite naive readers. For a more extensive introduction we recommend [2,3]. Then, we briefly survey preexisting work on decision algorithms for games, first for pure strategies, then for probabilistic solution concepts. Finally we outline the work of subsequent sections.

1.2. An Outline of Our Results

Three methods of presenting a game are considered here,

1. recursive form,
2. extensive form, and
3. normal form.

Perhaps most familiar in recreational games is the recursive presentation. There is a space of possible positions, with a designated initial position. There are rules (i.e., computational procedures) for passing from a nonterminal position to the player whose turn it is to move, the informational constraints faced by that player at the position, the set of allowed moves, and the new positions resulting from each allowed move. There are also rules determining payoffs at each terminal position. Both theoretically and practically, it is natural to require that these rules be computationally simple, as is the case, for instance, in chess.

The extensive form of a game is the most popular in modeling applications. Here, the set of all legal paths through the space of positions is laid out in a tree. The informational constraints