Available online at www.sciencedirect.com

ScienceDirect

Games and Economic Behavior 63 (2008) 642–662

GAMES and Economic Behavior

www.elsevier.com/locate/geb

# Simple search methods for finding a Nash equilibrium

Ryan Porter, Eugene Nudelman *, Yoav Shoham

*Computer Science Department, Stanford University, Stanford, CA 94305, USA*

Received 19 October 2004

Available online 22 August 2006

**Abstract**

We present two simple search methods for computing a sample Nash equilibrium in a normal-form game: one for 2-player games and one for $n$-player games. Both algorithms bias the search towards supports that are small and balanced, and employ a backtracking procedure to efficiently explore these supports. Making use of a new comprehensive testbed, we test these algorithms on many classes of games, and show that they perform well against the state of the art—the Lemke–Howson algorithm for 2-player games, and Simplicial Subdivision and Govindan–Wilson for $n$-player games.


*JEL classification:* C63; C72

*Keywords:* Nash equilibrium; Computer science; Algorithms

## 1. Introduction

This paper addresses the problem of finding a sample Nash equilibrium of a given normal form game. This notorious problem has been described as the "most fundamental computational problem" at the interface of computer science and game theory (Papadimitriou, 2001). Despite several decades of research into this problem it remains thorny; its precise computational complexity is unknown, and new algorithms have been relatively few and far between.

* Corresponding author.

*E-mail addresses:* rwporter@cs.stanford.edu (R. Porter), eugnud@cs.stanford.edu (E. Nudelman), shoham@cs.stanford.edu (Y. Shoham).

0899-8256/$ – see front matter 
doi:10.1016/j.geb.2006.03.015

The present paper makes two related contributions to the problem, both bringing to bear fundamental lessons from computer science. In a nutshell, they are these:

- The use of heuristic search techniques in algorithms.
- The use of an extensive test suite to evaluate algorithms.

The surprising result of applying these insights is that, for relevant classes of games, even very simple heuristic methods are quite effective, and significantly outperform the relatively sophisticated algorithms developed in the past. We expand on these two points below.

In the developments of novel algorithms, one can identify two extremes. The first extreme is to gain deep insight into the structure of the problem, and craft highly specialized algorithms based on this insight. The other extreme is to identify relatively shallow heuristics, and hope that these, coupled with the ever increasing computing power, are sufficient. While the first extreme is certainly more appealing, the latter often holds the upper hand in practice.[1] For example, a deep understanding of linear programming yielded polynomial-time interior point methods (Karmarkar, 1984), though in practice one tends to use methods based on the exponential Simplex algorithm (Wood and Dantzig, 1949). In reality, neither extreme is common. It is more usual to find a baseline algorithm that contains many choice points based on some amount of insight into the problem, and then apply heuristics to making these choices. For example, for the problem of propositional satisfiability, the current state of the art solvers apply heuristics to search the space spanned by the underlying Davis–Putnam procedure (Cook and Mitchell, 1997).

The use of heuristics, while essential, raises the question of algorithm evaluation. The gold standard for algorithms is trifold: soundness, completeness, and low computational complexity. An algorithm is sound if any proposed solution that it returns is in fact a solution, and it is complete if, whenever at least one solution exists, the algorithm finds one. Low computational complexity is generally taken to mean that the worst-case running time is polynomial in the size of the input.[2]

Of course, in many cases, it is not possible (or has turned out to be extremely difficult) to achieve all three simultaneously. This is particularly true when one uses heuristic methods. In this paper we focus on approaches that sacrifice the third goal, that of low worst-case complexity. Without a worst-case guarantee (and even worse, when one knows that the running time is exponential in the worst case), an algorithm must be evaluated using empirical tests, in which case the choice of problem distributions on which it is tested becomes critical. This is another important lesson from computer science—one should spend considerable effort devising an appropriate test suite, one that faithfully mirrors the domain in which the algorithms will be applied.

With these computer science lessons in mind, let us consider the extant game theoretic literature on computing a sample Nash equilibrium. Algorithm development has clearly been of the first kind, namely exploiting insights into the structure of the problem. For 2-player games,

[1] Of course, there is no difference in kind between the two extremes. To be effective, heuristics too must embody some insight into the problem. However, this insight tends to be limited and local, yielding a rule of thumb that aids in guiding the search through the space of possible solutions or algorithms, but does not directly yield a solution.

[2] In rare cases this is not sufficient. For example, in the case of linear programming, Khachiyan's ellipsoid method (Khachiyan, 1979) was the first polynomial-time algorithm, but, in practice, it could not compete with either the existing, exponential simplex method (Wood and Dantzig, 1949), or polynomial interior point methods (Karmarkar, 1984) that would later be developed. However, this will not concern us here, since there is no known polynomial-time complexity algorithm for the problem in question—computing a sample Nash equilibrium—and there are strong suspicions that one does not exist.