Exclusion Method for Finding Nash Equilibrium in Multi-Player Games

(Extended Abstract)

Kimmo Berg
Aalto University School of Science
Dept. of Mathematics and Systems Analysis
kimmo.berg@aalto.fi

Tuomas Sandholm
Carnegie Mellon University
Computer Science Department
sandholm@cs.cmu.edu

ABSTRACT

We present a complete algorithm for finding an ϵ-Nash equilibrium, for arbitrarily small ϵ, in games with more than two players. The best prior complete algorithm has significantly worse complexity and has, to our knowledge, never been implemented. The main components of our tree-search-based method are a node-selection strategy, an exclusion oracle, and a subdivision scheme. The node-selection strategy determines the next region to be explored—based on the region's size and an estimate of whether the region contains an equilibrium. The exclusion oracle provides a provably correct sufficient condition for there not to exist an equilibrium in the region. The subdivision scheme determines how the region is split if it cannot be excluded. Unlike well-known incomplete methods, our method does not need to proceed locally, which avoids it getting stuck in a local minimum that may be far from any actual equilibrium. The run time grows rapidly with the game size, and this suggests a hybrid scheme where one of the relatively fast prior incomplete algorithms is run, and if it fails to find an equilibrium, then our method is used.

Keywords

Nash equilibrium; multi-player games; noncooperative games

1. INTRODUCTION

Finding a Nash equilibrium is an important and well-studied problem. Finding (even an approximate) Nash equilibrium in a multi-player game is PPAD-complete [2, 3]. Several techniques have been proposed: simplicial subdivision [10], homotopy [4, 5], mixed-integer programming [9], support enumeration [8], and differential evolution methods. The only prior complete method for finding an approximate Nash equilibrium is an exhaustive grid search in the space of probabilities [1], and it had not been implemented.

We present a complete method that improves the best-known upper bound for computing an approximate Nash equilibrium. It is guaranteed to find an ϵ-Nash equilibrium, for arbitrarily small ϵ, in finite time that depends on ϵ.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. NORMAL-FORM GAMES

Let $N = \{1, .., n\}$ be the set of players and $A_i = \{a_1, .., a_m\}$ is the set of pure actions for player i. The function $u_i : A \mapsto \mathbf{R}$ gives player i's payoff. Each player i assigns a probability $p_i(a_j) \geq 0$ for each pure action $a_j \in A_i$ such that $\sum_{k=1}^{m} p_i(a_k) = 1$. A strategy p^* is a Nash equilibrium if $u_i(a_j, p^*_{-i}) \leq u_i(p^*_i, p^*_{-i})$ for all $a_j \in A_i$ and $i \in N$. At least one Nash equilibrium always exists in these finite games [6].

We can define player i's regret of action a_j as $r_i(a_j, p) = u_i(a_j, p_{-i}) - u_i(p)$. The regret of player i is $r_i(p) = \max_{a_j \in A_i} r_i(a_j, p)$ and $r(p) = \max_{i \in N} r_i(p)$. A strategy p^* is an ϵ-Nash equilibrium if the regret is small enough: $r(p^*) = \epsilon$.

3. TREE-SEARCH-BASED METHOD

We propose a tree-search-based method for finding a Nash equilibrium in multi-player games. It splits the space of strategy probabilities into smaller regions until a solution is found, that is, when the regrets are small enough.

3.1 Region selection (node-selection strategy)

An important choice in the method is the order in which the regions are examined. We propose a ranking function $g(R, p^0)$ that uses the size of the region $d(R)$, the regret, and its gradient evaluated at one point $p^0 \in R$:

$$g(R, p^0) = \max_{i \in N} r_i(p^0)/(d(R) \cdot M_i(p^0)), \tag{1}$$

$$M_i(p^0) = \max_{a_j \in A_i(p^0)} \|\nabla r_i(a_j, p^0)\|_\infty \tag{2}$$

is the maximum derivative at p^0 of player i's regret over all the variables in p' and $d(R) = \max_{q \in R} \|q - p^0\|$.

3.2 Exclusion oracle

An exclusion oracle can detect that a Nash equilibrium cannot be in the region. If this is the case, the region can be excluded, and otherwise the region is subdivided. We determine a global upper bound for the gradient of the regret:

$$M_i^* = \max_{p \in P'} \max_{a_j \in A_i(p)} \|\nabla r_i(a_j, p)\|_\infty.$$

THEOREM 3.1. *A p^0-centered ball with radius s cannot contain 0-Nash if*

$$r_i(p^0) > s \cdot M_i^*, \quad \textit{for some } i \in N. \tag{3}$$

Table 1: Results for the random general-sum games.

n	m	Dim	Avg Time	Median	Timeout %	Avg ϵ at timeout
3	2	3	0.04	0.02	0	-
3	3	6	18.2	0.8	1	0.0028
3	5	12	900	900	100	0.07
4	2	4	21.5	0.47	1	0.0026
4	3	8	343	96	27	0.003
5	2	5	110	27	7	0.0038
5	3	10	576	652	49	0.003

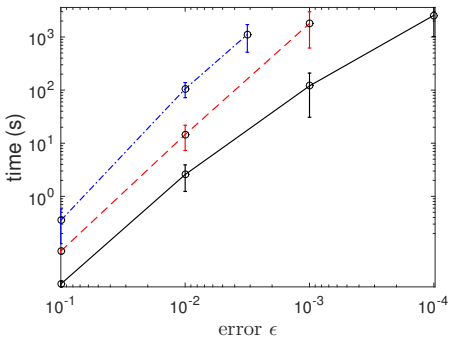

Figure 1: 3 to 5-player games with 3 actions.

3.3 Subdivision of the region

There are many ways how the search space can be subdivided. In this paper, we use hyperrectangles which makes the memory requirement linear in dimension. It is also easy to compute the middle point p^0 in the hyperrectangle and the maximum distance d to the corner points:

$$p^0_x = (u_x + l_x)/2,$$

$$d = 1/2 \cdot \sqrt{\sum_{x \in p'} (u_x - l_x)^2},$$

where l_x and u_x are the lower and upper bounds of the hyperrectangle in dimension x. We use a bisection method and split the hyperrectangle along the longest edge.

Algorithm 1
Repeat until the stopping condition is met

1. Select the region with minimal value of g in Eq. (1).
2. Compute $r(p^0)$. If Eq. (3) holds, exclude and goto 1.
3. Bisect the region.

THEOREM 3.2. *Any bisection algorithm excludes all points with $r(p) \geq \epsilon$ within $2^{(m-1)n\lceil \log\left(\frac{2M^*}{\epsilon}\right)/\log(2)\rceil} - 1$ iterations, where $M^* = \max_{i \in N} M^*_i$.*

The only prior complete algorithm is an exhaustive grid search by Babichenko et al. [1]. It is slower: it runs in $(k+1)^{nm}$ iterations, where $k > 8(\log m + \log n - \log \epsilon + \log 8)/\epsilon^2$.

THEOREM 3.3. *The number of iterations required by any bisection algorithm (with small enough ϵ and large enough m and n) is smaller than $(k+1)^{mn}$.*

4. EXPERIMENTS

We tested the algorithms on randomly generated normal-form games (time limit 900s) and on GAMUT games [7]; we used target accuracy $\epsilon = 10^{-3}$. One can see from the results that the running times grow rapidly as the search space dimension increases.

Figure 1 shows the computation times for different ϵ in three-player (black solid line), four-player (red dashed line) and five-player (blue dash-dotted line) games, with three actions per player. Moreover, we have tested different node-selection strategies: depth-first, breadth-first and random search order, and they were slower than the presented one.

5. CONCLUSION

Our aim was not to develop the fastest incomplete method but a method that is guaranteed to find an ϵ-Nash equilibrium on all instances. In contrast, the homotopy methods, e.g., are often much faster in practice as they improve the current solution locally, but they may not find a solution. That suggests a hybrid scheme where one runs some relatively fast incomplete method first, and if it fails to find an equilibrium, then one runs our algorithm. That achieves the best of both worlds: the speed of the prior algorithms and the completeness of ours. It would also be extremely interesting to develop improvements to each of the three component techniques used in our tree search.

Acknowledgments

This material is based on work supported by the NSF under grants 1320620 and 1546752.

REFERENCES

[1] Y. Babichenko, S. Barman, and R. Peretz. Simple approximate equilibria in large games. EC '14, p. 753–770, 2014.

[2] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3), 2009.

[3] C. Daskalakis. On the complexity of approximating a Nash equilibrium. *ACM Trans. on Alg.*, 9(3), 2013.

[4] S. Govindan and R. Wilson. A global Newton method to compute Nash equilibria. *JET*, 110:65–86, 2003.

[5] P. J.-J. Herings and A. van den Elzen. Computation of the Nash equilibrium selected by the tracing procedure in n-person games. *GEB*, 38:89–117, 2002.

[6] J. Nash. Equilibrium points in n-person games. *PNAS*, 36:48–49, 1950.

[7] E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms. AAMAS '04, p. 880–887, 2004.

[8] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *GEB*, 63:642–662, 2008.

[9] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. *AAAI*, p. 495–501, 2005.

[10] G. van der Laan, A. Talman, and L. van der Heyden. Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. *Math of OR*, 12(3):377–397, 1987.