



2008

# Finding Nash equilibria in two-player, zero sum games

Jeffrey Wimpee  
*Western Washington University*

Follow this and additional works at: [https://cedar.wvu.edu/computerscience\\_stupubs](https://cedar.wvu.edu/computerscience_stupubs)

 Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Wimpee, Jeffrey, "Finding Nash equilibria in two-player, zero sum games" (2008). *Computer Science Graduate and Undergraduate Student Scholarship*. 3.  
[https://cedar.wvu.edu/computerscience\\_stupubs/3](https://cedar.wvu.edu/computerscience_stupubs/3)

This Research Paper is brought to you for free and open access by the Computer Science at Western CEDAR. It has been accepted for inclusion in Computer Science Graduate and Undergraduate Student Scholarship by an authorized administrator of Western CEDAR. For more information, please contact [westerncedar@wwu.edu](mailto:westerncedar@wwu.edu).

# **Finding Nash equilibria in two-player, zero-sum games**

Jeffrey Wimpee  
Computer Science  
Western Washington University  
wimpeej@cc.wvu.edu

## **Abstract**

In many games, it is desirable to find strategies for all players that simultaneously maximize their respective worst-case payoffs. A set of strategies satisfying this criterion is called a Nash equilibrium. Because the search space of possible strategies grows rapidly as the size of the game increases, specialized algorithms are needed to efficiently find Nash equilibria. In this paper, current equilibrium-finding methods are presented and key areas for future work are identified. The first algorithm, due to Koller, Megiddo, and von Stengel, computes standard Nash equilibria in two-player, zero-sum games. The second algorithm, due to Miltersen and Sorensen, extends the method of Koller, Megiddo, and von Stengel to find proper equilibria. Both algorithms run in polynomial time in the worst case. The hardness of the equilibrium-finding problem for general-sum games highlights the need for new approximation methods.

## **Keywords**

Nash equilibrium, game theory, two-player games, zero-sum games

## **1. Introduction**

This paper is a survey of algorithms for finding Nash equilibria and proper equilibria in two-player games. A Nash equilibrium in this context is a pair of strategies, one for each player, such that each strategy is a best response to the other. In other words, no player has incentive to unilaterally deviate from his or her respective strategy. The related concept of proper equilibrium imposes additional constraints on acceptable strategies. Roughly speaking, proper equilibrium strategies take into account the possibility that a player may occasionally make suboptimal choices with small probability. In any case, the number of strategies increases exponentially with the size of the game. The objective of the research surveyed here is to present efficient algorithms for computing strategy pairs that constitute equilibria. An additional goal of this survey is to define the current frontiers of research in the field of two-player games and show possible directions for future research.

In the remainder of this paper, basic concepts and terminology of game theory are introduced. Two algorithms are presented for computing Nash equilibria in two-player, zero-sum games. Finally, several promising areas for future work are identified.

## **2. Background**

In game theory, a game is a strategic situation involving multiple agents. It specifies the choices available to the agents and the payoffs associated with each possible sequence of choices. Each agent attempts to maximize the payoff to himself, but his actions may depend on both past and future choices of the other players.

A strategy defines the action a player takes at each of his available choices. The particular strategies we are interested in finding satisfy Nash equilibrium, which may be defined by example. Suppose that each player  $P$  adopts a strategy satisfying the following condition: given that the other players' strategies remain fixed,  $P$  may not increase his own payoff by changing his strategy. A set of strategies (one for each player) that satisfies this condition is called a Nash equilibrium. Strategies may be pure or mixed: a pure strategy defines a set action for each possible decision; a mixed strategy is a probability distribution over the set of pure strategies.

A game may be represented as a set of matrices, one for each player, that specify the payoff to that player given the strategies of all players. This representation is known as the normal or strategic form. Each dimension of a matrix corresponds to the complete set of strategies available to a single player. Accordingly, in the case of a two player game, we can call one player the row player and the other the column player.

Alternatively, a game may be represented as a tree whose nodes each belong to one of the players and whose edges represent choices by that player. There may also be a pseudo-player, called Nature or Chance, representing random occurrences in the game. This tree representation is called the extensive form. In a game with imperfect information, a node may belong to an information set. Nodes in an information set represent distinct states of the game, but to the player who owns them they are indistinguishable. Associated with each leaf node is a tuple of payoffs to the players. A path from the root to a leaf describes a possible sequence of events in the game from start to finish. The probability of a player making a particular sequence of choices under a given mixed strategy is called the realization weight of that sequence. If the payoffs to each player for any given combination of strategies add up to zero, the game is called zero-sum.

The notion of normal form proper equilibrium is a refinement of the concept of Nash equilibrium. Proper equilibrium seeks to address certain logical inconsistencies in Nash equilibria. In particular, Nash equilibrium does not necessarily require certain moves to maximize payoff to the player playing the move, if the information set containing the move is reached with zero probability. In that case, the particular move chosen is undefined. Assuming players are perfectly rational, this has no impact on the game since the suboptimal move can never be reached in the game tree. However, in practice, players may make mistakes and make non-equilibrium choices. In such cases, it is desirable to capitalize on mistakes made by the opponent in order to maximize one's payoff. Thus, reasonable behavior should be specified at all nodes of the tree, even ones that can be reached only by "unreasonable" behavior.

### **3. Methods**

In this section, two algorithms for computing Nash equilibria in two-player games are described. The first, due to Koller, Megiddo, and von Stengel, finds standard Nash equilibria. The method of Miltersen and Sorensen finds proper equilibria.

Koller et al introduced an efficient algorithm for computing Nash equilibria in

two-player games represented in extensive form [3]. Unlike previous methods, this algorithm does not require that the game first be converted into normal form. This feature represents a significant improvement, since conversion from extensive to normal form exponentially increases the size of the game representation. Their method for general-sum games is equivalent to solving a special case of quadratic programming called the linear complementarity problem. No polynomial-time algorithm is known for solving LCPs, though they can be solved, for example, by the Lemke-Howson algorithm in exponential time. However, in the special case of a two-player zero-sum game, their method corresponds to the solution of a linear program, for which polynomial-time algorithms (such as Karmarkar's interior-point method) are known.

The linear program to be solved to find the column player's equilibrium strategy  $y$  in a two-player, zero-sum game is

$$\begin{array}{ll} \min_{y,p} & e^T p \\ \text{s.t.} & -Ay + E^T p \geq 0 \\ & -Fy = f \\ & y \geq 0 \end{array}$$

The objective function  $e^T p$  represents the payoff from the column player to the row player.  $e$  and  $f$  are the scalar 1.  $A$  is the payoff matrix for Player 1.  $E$  and  $F$  are  $I \times m$  and  $I \times n$  unit matrices, respectively.

To find the corresponding equilibrium strategy  $x$  for the row player, we must solve the dual problem. By the theorem of strong duality, a solution to the primal problem determines a solution to its dual. The dual of the first program is

$$\begin{array}{ll} \max_{x,q} & -q^T f \\ \text{s.t.} & x^T(-A) - q^T f \leq 0 \\ & x^T E^T = e^T \\ & x \geq 0 \end{array}$$

where the variables are defined as in the previous program.

The solution of these linear programs suffices to determine a Nash equilibrium in a two-player, zero-sum game. If the game is in normal form, then the entries of  $A$  are known simply from the description of the game. However, the method of Koller, Megiddo, and von Stengel specifies how these variables are determined in an extensive form game.

In an extensive form game, the entries of  $E$  are computed as follows. The first entry in the first row is 1 and the rest of the entries in the row are 0. Each other row corresponds to an information set, and each column corresponds to a sequence, with the

first column being the empty sequence. The entries of each row are 1 if the last choice in the sequence is a choice in that row's information set, and -1 in the column corresponding to the choice leading into the information set. The entries of  $x$  correspond to the probability of choosing a given sequence. Then  $Ex = e$  corresponds to the constraint that the realization weight of a sequence must equal the sum of the realization weights of its child sequences. This ensures that the  $x$  corresponds to a some mixed strategy, allowing us to use the linear program previously defined. The entries of  $A$  are determined as follows: for each leaf, consider the entry in the matrix (where the rows and columns represent sequences belonging to the respective players) which leads to that leaf. The entry of the matrix is then the product of all chance probabilities on that sequence times the payoff associated with the leaf. All other entries of the matrix are zero. The values of  $y$ ,  $F$ ,  $f$  and  $B$  are defined similarly.

Miltersen and Sorensen have extended the algorithm of Koller, Megiddo and von Stengel to compute proper equilibria in two-player, zero-sum extensive games [4]. The core of the algorithm is the repeated solution of the following linear programs:

$$\begin{array}{rcl}
 \max_{x,q,t} & t & \\
 \text{s.t.} & -A^T x + F^T q + m^{(k)} t & \leq \sum_{0 < i < k} m^{(i)} v^{(i)} \\
 & Ex & = e \\
 & f^T q & = v^{(0)} \\
 & x & \geq 0 \\
 & t & \geq 0
 \end{array}$$

and

$$\begin{array}{rcl}
 \max_{x,q,u,s} & 1^T u & \\
 \text{s.t.} & -A^T x + F^T q + u + \sum_{0 < i \leq k} m^{(i)} v^{(i)} s & \leq 0 \\
 & Ex - es & = 0 \\
 & f^T q - v^{(0)} s & = 0 \\
 & 0 \leq u & \leq 1 \\
 & x & \geq 0 \\
 & s & \geq 1
 \end{array}$$

where  $A$ ,  $F$ ,  $e$ ,  $f$ , and  $x$  are defined as in the Koller, Megiddo and von Stengel method,  $m$  represents actions that are mistakes for Player 2,  $v^{(k)}$  is the value to be gained from exploiting mistakes, and  $t$  is the payoff to Player 1. The number of iterations required is bounded by the number of actions owned by Player 2 in the game tree, and the algorithm yields a pair of proper equilibrium strategies.

## 4. Results

Both the methods of Koller, Megiddo, and von Stengel and Miltersen and Sorensen rely fundamentally on formulating and solving linear programs. Although the latter requires the solution of a series of linear programs, the number of such programs is polynomially bounded. Therefore, since LPs can be solved in polynomial time (using Karmarkar’s algorithm, for example), both methods are in the complexity class P.

Algorithm	Output	Computation Required	Complexity Class
Koller, Megiddo, von Stengel [3]	Standard Nash equilibrium	Single solution of linear program	P
Miltersen-Sorensen [4]	Normal form proper Nash equilibrium	Iterated solution of linear program	P

**Table 1:** A comparison of algorithms for computing standard and proper Nash equilibria in terms of the general method and time complexity.

## 5. Discussion

The algorithms presented here fulfill complementary roles. Koller, Megiddo, and von Stengel describe a method for computing standard Nash equilibria that is applicable to extensive form two-player games, both zero- and general-sum. In the latter case, it may take exponential time. The algorithm of Miltersen and Sorensen computes *proper* equilibria in two-player extensive form games, provided the game is zero-sum.

Algorithm	Advantages	Disadvantages
Koller, Megiddo, von Stengel [3]	Operates on extensive form representation, applies to general-sum games	Computed strategies may not take advantage of opponent’s mistakes, in general-sum case takes worst-case exponential time
Miltersen-Sorensen [4]	Operates on extensive form representation, eliminates certain insensible behaviors (e.g., failing to take advantage of opponent’s mistakes)	Applies to zero-sum games only, proper equilibrium concept may be also counterintuitive

**Table 2:** Advantages and disadvantages of the Koller-Megiddo-von Stengel and Miltersen-Sorensen algorithms.

## 6. Future work

The main difficulty in this area is that computing equilibria in the general-sum case is intractable. It has been shown by Chen and Deng that even in the two-player case, computing exact Nash equilibria is complete in a complexity class known as PPAD [1].

Evidence suggests that no efficient algorithms exist for such problems [5]. Therefore, finding approximate solutions is an important area of current research.  $\epsilon$ -approximate Nash equilibrium is a promising idea related to this issue. In an  $\epsilon$ -approximate Nash equilibrium, by changing strategy no player may achieve a payoff more than  $\epsilon$  plus the optimal payoff in the exact equilibrium. Currently, the best polynomial time approximation algorithm for two-player, general-sum games achieves  $\epsilon$  equal to 0.3393, where the payoffs have been linearly mapped into the  $[0, 1]$  interval [6]. However, note that in this result,  $\epsilon$  is constant. It has been shown that general problem of approximating Nash equilibrium to arbitrary  $\epsilon$  is also PPAD-complete. Thus, it appears that the best that may be hoped for in this endeavor is to find a smaller, but still constant, value for  $\epsilon$ . Ideally, a theoretical bound on the value of  $\epsilon$  could be determined.

An interesting approach to approximation is to cluster strategically-similar nodes of the game tree to reduce the problem size [2]. However, the techniques of Gilpin and Sandholm seem to be somewhat specific to poker. It would be interesting to discover more general techniques for abstracting games via clustering.

Another contribution that could be made is further refinement of the idea of proper equilibrium. Miltersen and Sorensen give an example of a game in which the first player has two moves [4]. The first move leads to a state in which the second player can make a mistake that has a payoff of 1 to the first player. Player 1's other move gives Player 2 the opportunity to make a mistake that pays 2 to Player 1. According to the proper equilibrium concept, Player 1 should make the first move with probability  $2/3$  and the second with probability  $1/3$ , instead of always giving Player 2 the opportunity to make a bigger mistake. The justification given is that if Player 1 always made the second move, Player 2 could train to avoid making the more costly mistake. However, this model of mistakes may not always be logical. For example, if a game is being played at a computer, mistakes may result from an accidental input error rather than an erroneous decision. In this situation, mistakes could be better modeled as all having equal probability.

## References

- [1] X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *Proc. 47th FOCS*, pages 261-272, 2006.
- [2] A. Gilpin and T. Sandholm. Lossless abstraction of imperfect information games. *Journal of the ACM*, 25:1-30, 2007.
- [3] D. Koller, N. Megiddo, and B. von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proc. 26th STOC*, pages 750-759, 1994.
- [4] P. Miltersen and T. Sørensen. Fast algorithms for finding proper strategies in game trees. In *Proc. 19th SODA*, pages 874-883, 2008.



- [5] C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, pages 498-532, 1994.
- [6] H. Tsaknakis and P. Spirakis. An optimization approach for approximate Nash equilibria. In *Proc. 3<sup>rd</sup> WINE*, pages 42-56, 2007.