

# On Range of Skill\*

Thomas Dueholm Hansen and Peter Bro Miltersen and Troels Bjerre Sørensen

Department of Computer Science  
University of Aarhus

## Abstract

At AAAI'07, Zinkevich, Bowling and Burch introduced the Range of Skill measure of a two-player game and used it as a parameter in the analysis of the running time of an algorithm for finding approximate solutions to such games. They suggested that the Range of Skill of a typical natural game is a small number, but only gave heuristic arguments for this. In this paper, we provide the first methods for rigorously estimating the Range of Skill of a given game. We provide some general, asymptotic bounds that imply that the Range of Skill of a perfectly balanced game tree is almost exponential in its size (and doubly exponential in its depth). We also provide techniques that yield concrete bounds for unbalanced game trees and apply these to estimate the Range of Skill of Tic-Tac-Toe and Heads-Up Limit Texas Hold'em Poker. In particular, we show that the Range of Skill of Tic-Tac-Toe is more than 100,000.

## Introduction

Zinkevich, Bowling and Burch (2007) recently presented a new algorithm, the *Range of Skill Algorithm*, for finding approximate minimax strategies of very large two-player zero-sum imperfect information games. Their algorithm was successfully used to compute such approximate solutions to much larger game trees than was previously possible. In particular, it was applied to certain abstractions of Limit Texas Hold'em. To gain some theoretical insight into why the algorithm works so well, Zinkevich *et al.* applied the approach of *parameterized complexity*. To every symmetric game and every real value  $\epsilon > 0$ , they associated an integer valued parameter  $\text{ROS}_\epsilon(G)$  (for Range of Skill) and showed by an elegant analysis that their algorithm finds an  $\epsilon$ -approximate solution of a game  $G$  using at most  $\text{ROS}_\epsilon(G)$  iterations of its main loop. They also presented some intuition suggesting that for most natural games, the Range of Skill is a relatively small number. The intuition is derived from relating the measure to the difficulty of playing a game from a human perspective. Imagine lining up players, such that any player in the line will be able to win against all previous players, say, 75 % of the time. This captures the intuition that one is

able to gain different levels of insight on how to play a game. The difficulty of a game may then be measured by the number of players it is possible to line up. With this in mind, the Range of Skill was formally defined for a game as the length of the longest list of arbitrary strategies, called a ranked list, such that the expected payoff to the higher ranked strategy is more than some parameter  $\epsilon$  when two strategies from the list are matched against each other.

Given the impressive practical performance of the Range of Skill algorithm, it seems important to better understand the theoretical analysis. In particular, we should understand how to rigorously estimate the Range of Skill parameter for concrete games. The present paper provides the first methods for doing this.

First, we slightly adjust the definition of ROS to get a version we call AROS that also works for asymmetric games. This definition was implicit in Zinkevich *et al.*: Even though their definition was only described for symmetric games, their algorithm was only applied to asymmetric ones. The analysis of the complexity of their algorithm goes through with this definition. Then, we prove the following general results.

- For a game tree  $G$  of size  $n$  with all payoffs having absolute value at most  $\beta$  and any real number  $\epsilon > 0$ , we have  $\text{AROS}_\epsilon(G) \leq 2(2\beta n/\epsilon)^n$ .
- For a perfectly balanced and perfectly alternating perfect information binary game tree  $G$  of depth  $d$  with every non-terminal position being open and payoffs being  $-1$  or  $1$ , we have  $\text{AROS}_{0.99}(G) \geq 2^{2^{\Omega(d)}}$ .
- The Range of Skill  $\text{AROS}_1(G)$  of any combinatorial game  $G$  is at most equal to the number of leaves of the game tree of  $G$ .
- The Range of Skill  $\text{AROS}_1(G)$  of any combinatorial game  $G$  is at least the number of positions in the game tree with two immediate terminal successors with payoffs  $-1$  and  $1$ .

Also, we describe techniques for improving the latter two bounds for concrete game trees. Armed with the general techniques, we study some concrete games. Tic-Tac-Toe was suggested by Zinkevich *et al.* as a game of very low Range of Skill. We show that  $\text{AROS}_1$  of Tic-Tac-Toe is in fact between 104,615 and 131,840. The main game of

\*Work supported by *Center for Algorithmic Game Theory*, funded by the Carlsberg Foundation.  
Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

study of Zinkevich *et al.* was an abstraction of Limit Texas Hold'em Poker, where the number of bets in each round is restricted to three. We show that  $\text{AROS}_\epsilon$  for this game is at least  $1470\epsilon^{-1}$ . The latter concrete result is particularly interesting for  $2\epsilon = 1/100$ . This was the approximation achieved by Zinkevich *et al.* when they computed an approximate solution to the poker game. The actual number of iterations of their main loop needed to achieve this approximation is reported to be 298. In contrast, the upper bound on the number of iterations given by the Range of Skill is no less than 294,000 and possibly much larger: In contrast to the case of Tic-Tac-Toe the bounds on the Range of Skill of the poker game are very far from being tight. The discrepancy between these numbers suggests that while the Range of Skill *algorithm* seems to be an extremely attractive way of approximately solving large games in practice, it is less clear that the *analysis* in terms of Range of Skill is a convincing way of providing theoretical evidence for this.

### Preliminaries

Throughout the paper, we are going to consider two-player zero-sum games with Player 1 trying to maximize payoff and Player 2 trying to minimize payoff. The (expected) payoff when Player 1 plays (mixed) strategy  $b_1$  and Player 2 plays (mixed) strategy  $b_2$  will be denoted  $u(b_1, b_2)$ .

The formal definition of Range of Skill proposed by Zinkevich *et al.* only applies to symmetric games, and their algorithm was also only described for symmetric games, even though it is exclusively applied to asymmetric games in their paper (note that *all* turn based games are asymmetric). For the discussions of this paper, it is important to appropriately fix the setup for asymmetric games. We describe in Figure 1 what we believe is the most natural variant of the Range of Skill algorithm for asymmetric games. In particular, for other variants, it does not seem obvious how to appropriately define the Range of Skill measure so that it upper bounds the complexity of the algorithm.

We now define a corresponding Asymmetric Range of Skill measure. Recall that a *strategy profile* for a two-player game is a pair of strategies, one for each player.

**Definition 1.** *Given a two-player zero-sum game  $G$  with payoff function  $u$ , define a list of strategy profiles  $(b_1^i, b_2^i), i = 1..N$  to be an  $\epsilon$ -ranked list, if for all  $i > j$ ,  $u(b_1^i, b_2^j) - u(b_1^j, b_2^i) \geq 2\epsilon$ . The Asymmetric Range of Skill or  $\text{AROS}_\epsilon(G)$  is the length of the longest  $\epsilon$ -ranked list.*

Note that in the definition, the strategies in each profile are not played against each other. Rather,  $(b_1^i, b_2^i)$  should be thought of as two strategies a single player  $i$  adopts for playing a game; one for playing as Player 1 and one for playing as Player 2. With this interpretation,  $u(b_1^i, b_2^j) - u(b_1^j, b_2^i)$  is the expected payoff for  $i$  in a tournament where  $i$  plays  $j$  twice, first as Player 1, then as Player 2.

For the case of a symmetric game, this definition agrees with the definition of ROS of Zinkevich *et al.*, except that we require a soft inequality ( $\geq 2\epsilon$ ) rather than a sharp one ( $> 2\epsilon$ ). We make this change for convenience, as it allows us to focus on the interesting case  $\epsilon = 1$  (see below), but clearly the spirit of the definition remains intact, and any

1. Let  $G$  be a two player zero-sum game with strategy space  $\Gamma_i$  for Player  $i, i = 1, 2$ .
2. For  $i = 1, 2$ , let  $\Sigma_i = \{b_i^0\}$ , where  $b_i^0$  is an arbitrary element of  $\Gamma_i$ .
3. Repeat
  - (a) Let  $G_1$  be the game which is like  $G$  but with Player 1 restricted to strategies in  $\Sigma_1$ . Let  $v_1$  be the value of  $G_1$ , and let  $(y_1, b_2)$  be an equilibrium (i.e., pair of minimax mixed strategies) of  $G_1$ .
  - (b) Let  $G_2$  be the game which is like  $G$  but with Player 2 restricted to strategies in  $\Sigma_2$ . Let  $v_2$  be the value of  $G_2$ , and let  $(b_1, y_2)$  be an equilibrium of  $G_2$ .
  - (c) Add  $b_1$  to  $\Sigma_1$  and  $b_2$  to  $\Sigma_2$ .  
until  $v_2 - v_1 < 2\epsilon$ .
4. Return  $(y_1, y_2)$ .

Figure 1: Asymmetric Range of Skill algorithm

concrete lower or upper bound can be converted between ROS and AROS by perturbing  $\epsilon$  up or down. More importantly, the proof of Zinkevich *et al.* immediately generalizes to show the following theorem (recall that an  $\epsilon$ -equilibrium is a strategy profile where no player may gain more than  $\epsilon$  by deviating):

**Theorem 2.** *The Asymmetric Range of Skill algorithm terminates after at most  $\text{AROS}_\epsilon(G)$  iterations of its main loop and computes a  $2\epsilon$ -equilibrium.*

*Proof.* That a  $2\epsilon$ -equilibrium is computed follows from the fact that when the procedure terminates, for values  $v_1$  and  $v_2$  with  $v_2 - v_1 < 2\epsilon$ , the strategy  $y_1$  for Player 1 is guaranteed to achieve a gain of at least  $v_1$  against an optimal, unrestricted counter strategy while the strategy  $y_2$  for Player 2 is guaranteed to achieve a loss of at most  $v_2$  against an optimal, unrestricted counter strategy.

Next, we estimate the number of iterations. Let the name of a variable in Figure 1 with superscript  $j$  added denote its value in the  $j$ 'th iteration of this loop after executing (b) but before executing (c). Suppose the loop has  $N$  iterations. Let  $0 \leq j < k < N$ . Since  $b_2^j \in \Sigma_2^k$  and  $b_1^k$  is a minimax strategy in  $G_2$ , we have  $u(b_1^k, b_2^j) \geq v_2^k$ . Similarly,  $b_1^j \in \Sigma_1^k$  implies  $u(b_1^j, b_2^k) \leq v_1^k$ . Also, since  $k < N$ , we have  $v_2^k - v_1^k \geq 2\epsilon$ . These inequalities together imply  $u(b_1^k, b_2^j) - u(b_1^j, b_2^k) \geq 2\epsilon$ . This means that the strategy profiles  $(b_1^j, b_2^j)$  for  $j = 0..N - 1$  form an  $\epsilon$ -ranked list and hence  $N$  is at most  $\text{AROS}_\epsilon(G)$ .  $\square$

By a *combinatorial game* we mean a perfect information game with no moves of chance and with all payoffs at leaves being 1,  $-1$  or 0 (i.e. win/lose/tie for Player 1). For combinatorial games, the case  $\epsilon = 1$  (the largest meaningful value of  $\epsilon$  for these payoffs) is particularly natural. Note that if  $(b_1^i, b_2^i), i = 1..N$  is a 1-ranked list for a combinatorial game we must have for all  $i > j$  that  $u(b_1^i, b_2^j) = 1$

and  $u(b_1^j, b_2^j) = -1$ . That is,  $i$  beats  $j$  with probability 1, no matter who starts the game. Further, when considering  $\text{AROS}_1(G)$  for combinatorial games, we can without loss of generality restrict attention to *pure strategies*. Indeed, when some strategy beats another strategy with probability 1, any random choice it makes can be frozen to an arbitrary deterministic one without changing this fact.

In some parts of the paper, it is convenient to operate with game trees satisfying certain niceness conditions. It is easy to transform any tree into one satisfying these:

**Definition 3.** A node  $x$  of a combinatorial game  $G$  is said to be open if the subtree rooted at  $x$  contains leaves of payoff both  $-1$  and  $1$ . The open tree of  $G$  is the largest embedded subtree of  $G$  for which every internal node is open. Furthermore, we denote by the reduced open tree the open tree that has been transformed by repeatedly doing the following.

- Merging nodes that are not alternating (i.e., successive nodes controlled by the same player).
- Removing internal nodes of outdegree 1 by extending the edge from the parent to the child.
- Removing leaves that has the same payoff as a sibling leaf.

## Asymptotic results

**Theorem 4.** Let any two-player zero-sum extensive-form game of perfect recall be given. Let  $n$  be the total number of actions in the game tree and let  $\beta$  be the largest absolute value of the payoff at any leaf. Then, for any  $\epsilon > 0$ , we have  $\text{AROS}_\epsilon(G) \leq 2(2\beta n/\epsilon)^n$ .

*Proof.* We shall in fact only look at the case where the largest absolute value of any payoff is 1. The general case follows easily by scaling. Assume  $\text{AROS}_\epsilon(G) = N$ , and let  $\{(b_1^j, b_2^j)\}_{j=0}^{N-1}$  be an  $\epsilon$ -ranked list. We are going to use sequence form representation of mixed strategies. For a game of perfect recall, the sequence form representation  $x$  ( $y$ ) of a mixed strategy  $b_1$  ( $b_2$ ) belonging to Player 1 (Player 2) has the following properties (see Koller, Megiddo and von Stengel (1994) for details):

- $x$  ( $y$ ) is a real vector containing at most the total number of actions of Player 1 (Player 2) in the game. Every entry in  $x$  and  $y$  is between 0 and 1.
- The expected payoff for Player 1 when Player 1 plays  $b_1$  and Player 2 plays  $b_2$  is given by  $x^\top Ay$ , where  $A$  is a matrix depending on the game.
- The absolute values of entries of the vector  $Ay$  as well as the vector  $x^\top A$  are all bounded by the largest absolute value of the payoff at any leaf of the game.

We let  $x^j$  be the sequence form representation of  $b_1^j$  and  $y^j$  be the sequence form representation of  $b_2^j$ . Also, let  $\tilde{x}^j$  ( $\tilde{y}^j$ ) be  $x^j$  ( $y^j$ ) rounded to  $r$  bits of precision, with  $r = \lceil \log(1/\epsilon) + \log n \rceil$ . Let  $s^j$  be a string containing the binary representation of all values  $\tilde{x}^j, \tilde{y}^j$ . Note that  $s$  has length  $rn$ . We claim:

- For all  $k > j$ , we have that  $\tilde{x}^{k^\top} Ay^j - x^{j^\top} A\tilde{y}^k > 0$  and for all  $k < j$  we have that  $\tilde{x}^{k^\top} Ay^j - x^{j^\top} A\tilde{y}^k < 0$ .

We only prove the first half of the claim; the proof of the second half is similar. From the definition of an  $\epsilon$ -ranked list, we have  $x^{k^\top} Ay^j - x^{j^\top} Ay^k \geq 2\epsilon$ . Since each entry of  $\tilde{x}^k$  differs from the corresponding entry of  $x^k$  and each entry of  $\tilde{y}^k$  differs from the corresponding entry of  $y^k$  by strictly less than  $2^{-r} \leq \epsilon/n$  and the entries in  $Ay^k$  and  $x^{k^\top} A$  are bounded in absolute value by 1, the claim follows.

The claim implies that each string  $s^k$  can be shared by at most two different values of  $k$ . Indeed,  $\tilde{x}^k$  and  $\tilde{y}^k$  may be reconstructed from  $s^k$  and the claim implies that we can almost reconstruct  $k$  from  $\tilde{x}^k$  and  $\tilde{y}^k$ : It is either the largest value  $j$  for which  $\tilde{x}^{k^\top} Ay^j - x^{j^\top} A\tilde{y}^k > 0$  or the smallest value  $j$  for which  $\tilde{x}^{k^\top} Ay^j - x^{j^\top} A\tilde{y}^k < 0$ . Thus, we have that  $2^{rn} \geq N/2$ . That is,  $N \leq 2^{\lceil \log(1/\epsilon) + \log n \rceil n + 1} \leq 2^{(1 + \log(1/\epsilon) + \log n)n + 1} = 2(2n/\epsilon)^n$ .  $\square$

Note that combining Theorem 4 with Theorem 2 provides an upper bound on the running time of the Asymmetric Range of Skill algorithm as a function of the size of the game tree. The bound is exponential, but a priori, it was not obvious that even an exponential bound could be given.

Next, we turn to lower bounds on Range of Skill, showing that Theorem 2 does not imply that the Asymmetric Range of Skill algorithm has a running time which is polynomially bounded in the size of the game tree. Zinkevich *et al.* mentions that the game where both players choose a number between 1 and  $n$ , and the largest number wins, has Range of Skill linear in  $n$ . Our general approach for lower bounding the Range of Skill is to find embeddings of this game within any given game  $G$ . The following lower bound is the first example of this method.

**Theorem 5.** For any  $\epsilon > 0$  there is a constant  $k_\epsilon$  so that the following is true. Let  $G$  be a game that contains as an embedded subtree a perfectly balanced, perfectly alternating, perfect information open tree of depth  $k_\epsilon d$  with no nodes of chance and with payoffs  $-1$  and  $1$  at the leaves. Then,  $\text{AROS}_{1-\epsilon}(G) \geq 2^{2^d}$ .

*Proof.* The Greater Than problem on  $S = \{1, \dots, N\}$  is the following communication problem (for formal definitions of two-party communication protocols and complexity, see Kushilevitz and Nisan (1996)): Alice and Bob each get a number in  $S$  and must communicate by transmitting bits to determine which number is the larger (they are promised that they are distinct). Combining Nisan (1993) with Newman (1991), we have that for any  $\epsilon > 0$ , there is a  $c$  and a private coin (meaning that each player has a separate source of randomness, not accessible to the other player) randomized communication protocol for the Greater Than problem on  $\{1, \dots, 2^{2^d}\}$  with error probability  $\epsilon$  and at most  $cd$  bits communicated.

Two players can simulate a communication protocol by making moves in a perfectly balanced, perfectly alternating, perfect information open game tree, arbitrarily associating in each position of the tree the communication bit 0 to one action and the communication bit 1 to another. In this way, a tree of depth  $2M + 1$  enables them to simulate any communication protocol of communication complexity  $M$ . The loss of a factor of two is due to the fact that the protocol will

specify in any situation one of the players to communicate next - if this is not the player to move, the player to move will move arbitrarily. Since the position arrived at after simulating the protocol is non-terminal, it is still possible for each player to win. Thus, the players may let the output bit of the protocol determine who actually wins the game.

With a tree of depth larger than  $2cd$ , we can associate to any number  $j$  in  $\{1, \dots, 2^{2^d}\}$  the mixed strategy profile  $(b_1^j, b_2^j)$  where both strategies consists of simulating in this way the Nisan-Newman communication protocol for the Greater Than problem on input  $j$  (with  $b_1^j$  simulating Alice and  $b_2^j$  simulating Bob) followed by selecting an appropriate leaf.

Then, by construction,  $\{(b_1^j, b_2^j)\}_j$  is an  $\epsilon$ -ranked list.  $\square$

It is clear from the definition of AROS that for a fixed game  $G$ ,  $\text{AROS}_\epsilon(G)$  is a non-increasing function in  $\epsilon$ . We conclude this section with a theorem giving more precise information. This theorem will be useful for lower bounding the Range of Skill of Texas Hold'em for relevant values of  $\epsilon$ .

**Theorem 6.** *For any game  $G$ , any  $\epsilon > 0$ , and any integer  $k$ , we have  $\text{AROS}_{\epsilon/k}(G) \geq k(\text{AROS}_\epsilon(G) - 1) + 1$ .*

*Proof.* We show how to construct a longer  $\epsilon/k$ -ranked list  $p$  from an  $\epsilon$ -ranked list  $b$ , where the  $j$ 'th element of  $b$  is the mixed strategy profile  $b^j = (b_1^j, b_2^j)$ ,  $j = 0..N - 1$ . The idea is to take all pairs of adjacent elements of the list and insert  $k - 1$  convex combinations between each of these pairs.

More precisely, we define

$$p^{kj+i} = ((k-i)/k)b^j + (i/k)b^{j+1}$$

for  $j = 0..N - 2$  and  $i = 0..k - 1$  and we let  $p^{k(N-1)} = b^{N-1}$ . It is easy to see that the resulting list is  $\epsilon/k$ -ranked.  $\square$

## Range of Skill for combinatorial games

The asymptotic lower bound of Theorem 5 suggests that the Range of Skill of many natural games are huge numbers. However, the theorem has two drawbacks.

- To apply the theorem successfully we need a perfectly balanced, perfect information game tree of a certain depth embedded in the game of interest. Many game trees are quite unbalanced. Also, the value of  $k_\epsilon$  is not explicitly stated. We might estimate it by going through the arguments of Nisan and Newman, but would find that it is rather large. So, despite of being a superpolynomial bound, the theorem would provide poor estimates of the Range of Skill for many concrete games.
- The use of mixed strategies is essential for the argument. Thus, the theorem provides *no* lower bound on  $\text{AROS}_1$ .

In this section, we address both issues. First, it is easy to see that going from  $\text{AROS}_{1-\epsilon}$  to  $\text{AROS}_1$ , we encounter a "phase transition": The Range of Skill is now bounded by the size of the game tree.

**Theorem 7.** *The number of leaves in the reduced open tree of a combinatorial game  $G$  is an upper bound on  $\text{AROS}_1(G)$ .*

*Proof.* Let  $(b_1^j, b_2^j)$ ,  $j = 0..N - 1$  be the longest 1-ranked list. As mentioned in the Preliminaries section, we can without loss of generality assume that all strategies in the list are pure.

Let  $m$  be the number of leaves in the reduced open tree of  $G$ . If  $N > m$  we would, by the pigeonhole principle, have  $i$  and  $j$ , with  $i > j$ , in the longest 1-ranked list, so that when the strategies in the profile  $(b_1^i, b_2^j)$  are played against each other, the same leaf is reached as when the strategies in the strategy profile  $(b_1^i, b_2^i)$  are played against each other. Clearly, this is also the leaf reached when  $b_1^i$  is played against  $b_2^i$  and when  $b_1^i$  is played against  $b_2^j$ . But this contradicts the fact that the list is 1-ranked as this implies that Player 2 wins in the first case and that Player 1 wins in the second.  $\square$

We next present a way to lower bound  $\text{AROS}_1$  which yields good bounds for concrete games and in many natural cases beats the figures for  $\text{AROS}_{1-\epsilon}$  that could be obtained by working out the constant  $k_\epsilon$  in Theorem 5. We first describe a way of constructing strategy profiles that will be useful for constructing 1-ranked lists.

Given a combinatorial game  $G$ , we impose an ordering on the reduced open tree  $T$  of  $G$ , such that for some fixed representation of  $T$ , we let the children of any node be ordered from left to right in increasing order. We require that a leaf that makes the player in turn lose (win) the game is the leftmost (rightmost) child of its parent.

For a given strategy profile  $(b_1, b_2)$  and a given node  $x$ , we will say that the players are *going for* the leaf that will be reached if  $b_1$  and  $b_2$  are matched against each other starting at  $x$ . Note that specifying what the players are going for at every node will describe the entire strategy profile. Furthermore, we will say that the players are *going for a loss* (*going for a win*) if they are going for the leaf of lowest (highest) order of the subtree rooted at  $x$ .

We can then construct a strategy profile from a leaf  $x$  in the following way.

- If possible, the players are going for  $x$ .
- At nodes of lower order than  $x$ , the players are going for a win.
- At nodes of higher order than  $x$ , the players are going for a loss.
- At nodes that are not internal of  $T$ , actions are chosen that make sure the previously decided winner wins the game.

Figure 2 shows two applications of this construction of strategy profiles. We can now show the following lower bound on  $\text{AROS}_1(G)$ .

**Theorem 8.** *For a combinatorial game  $G$ , let  $m$  be the number of nodes of the reduced open tree  $T$  of  $G$  that has two leaves as children. Then  $\text{AROS}_1(G) \geq m$ .*

*Proof.* For every node  $x$  of  $T$  that has two leaves, we can construct a strategy profile for a 1-ranked list from either of these leaves. To see this we need to consider what happens when two such strategy profiles match up.

It is clear that at most one of the players will be going for a win at any given time and at most one will be going for a loss. Also, if the player to choose an action leading to a leaf is either going for a win or a loss, the player whose strategy is constructed from the leaf of highest order is certain to get a payoff of 1. We therefore only need to consider what happens at the node  $x$  when the player in turn is going for a leaf of  $x$ .

If the opposing player is still going for his leaf, the higher ranked player is sure to get a payoff of 1 because of the ordering of the nodes. If not, the opposing player is either going for a win or a loss, meaning that the previous choice either led to the subtree of highest or lowest order. Since every internal node of  $T$  has outdegree at least 2, both cannot be the case, and we are free to construct a strategy profile for the 1-ranked list from one of the leaves.  $\square$

The case analysis of the proof of Theorem 8 holds in general, and we can use this to construct even more strategy profiles for the 1-ranked list using the same scheme. We observe the following.

- (i) If a player chooses the action leading to the node of highest order, and his opponent is not already going for a win, then his opponent will not be going for a win in the next move either, and the other way around for the subtree of lowest order.
- (ii) The reduced open tree  $T$  is perfectly alternating, meaning that if a player  $i$  chose the action leading to the root  $r$  of some subtree of  $T$ , and player  $i$  controls the node from which an action leads to a leaf  $x$  of  $T$ , then the path from  $r$  to  $x$  must be of even length.

As in the end of the proof of Theorem 8 consider the problematic situation where one player  $j$  is going for a win, and the other player  $i$  chooses an action leading to a leaf  $x$  that ensures that player  $j$  loses. Since  $x$  lets the player in turn get a payoff of 1,  $x$  must be the leaf of highest order of some subtree of  $T$  rooted at some node  $r$ . Furthermore, it follows from (i) that for the largest such subtree player  $i$  chose the action leading to  $r$ , and from (ii) that the length of the path from  $r$  to  $x$  is at least two and even. We can make a similar observation for the opposite scenario. To make use of these observations we introduce the following definition.

**Definition 9.** A leaf  $x$  of the reduced open tree  $T$  of a combinatorial game is said to be problematic if

- $x$  is neither of highest nor lowest order of  $T$ .
- The length of the path from  $x$  to the root of the largest subtree of  $T$  for which  $x$  is of either highest or lowest order, is even and at least two.

The problematic leaves are exactly the ones giving rise to strategy profiles that, when matched against other strategy profiles, might produce the problematic situation. Hence, the list of all strategy profiles constructed from distinct leaves of  $T$  that are not problematic is a 1-ranked list. The

length of this 1-ranked list depends on the ordering of the leaves, which in turn depends on the representation of  $T$ . Different permutations of  $T$  therefore produce different 1-ranked lists. The length of the constructed 1-ranked list is, however, always at least as long as the number of nodes of  $T$  with two leaves.

Figure 2 illustrates the construction of strategy profiles for a 1-ranked list. The numbers below the leaves correspond to the indices of the constructed strategy profiles in the 1-ranked list, and the shaded leaves are problematic.

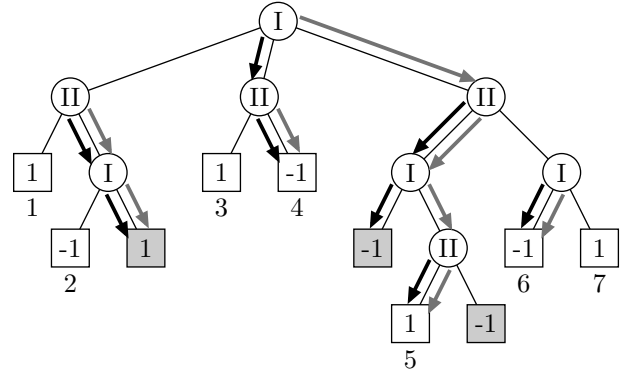


Figure 2: Strategy profiles constructed from leaf number 4 (black arrows) and leaf number 5 (gray arrows).

### Range of Skill of Tic-Tac-Toe and Limit Hold'em Poker

Using a computer program, we have counted the number of non-problematic leaves in the game of Tic-Tac-Toe. As mentioned above, this number depends on the actual representation of the game tree. We only created strategies for a single representation (i.e., permutation of actions) of the reduced open tree of Tic-Tac-Toe. It might well be possible to get tighter results by choosing different representations. The results are listed in Table 1. The source code for the program used can be found at

<http://daimi.au.dk/%7Etdh/RangeOfSkill.zip>.

Tree	Number of leaves
Game tree	255168
Open tree	153872
Reduced open tree	131840
Nodes of the reduced open tree with two leaves: 50544	
Number of non-problematic leaves: 104615	

Table 1: Tic-Tac-Toe.

The numbers in the table imply:

$$104615 \leq \text{AROS}_1(\text{Tic-Tac-Toe}) \leq 131840.$$

Our approach to finding strategies for a 1-ranked list does not apply directly to games of chance and imperfect information. For games such as poker, we can, however, ignore the random cards, and play the game as a game of no

chance and perfect information, using only the betting tree. The possibility of folding without a showdown ensures that we still have leaves of positive as well as negative payoff. For Limit Hold'em Poker this leaves us with a betting tree which is considerably smaller than the original game tree, but which we may still use to obtain lower bounds on the Range of Skill using the technique for combinatorial games described in the previous section. Note that we do not actually have a combinatorial game, as the payoffs are small integers in a wider range than  $-1, 0, 1$ . However, it is not hard to see that the lower bounds for  $\text{AROS}_1$  of the previous section are still valid for trees with arbitrary positive integers in place of 1 and arbitrary negative integers in place of  $-1$ .

We will focus on the variant of Limit Texas Hold'em Poker to which Zinkevich *et al.* apply their algorithm. In this game there are four rounds of betting, each with up to three raises. The blinds at the beginning of the first round also count as a raise, meaning there are actually only two raises allowed in the first round. In order to avoid a random outcome of the game we trim the betting tree by removing all leaves that do not correspond to folding. We then produce a 1-ranked list the same way as for Tic-Tac-Toe. The results are listed in Table 2. In particular, the Range of Skill for  $\epsilon = 1$  is at least 1471. Combining this with Theorem 6, we get the figure for  $\text{AROS}_\epsilon$  with  $2\epsilon = 1/100$  that was mentioned in the introduction.

Tree	Number of leaves
Trimmed betting tree	1715
Open tree	1715
Reduced open tree	1610
Nodes of the reduced open tree with two leaves: 490	
Number of non-problematic leaves: 1471	

Table 2: Limit Texas Hold'em Poker.

### Open problems

We have seen that for the case of combinatorial games and with  $\epsilon = 1$ , the Range of Skill measure has attractive combinatorial properties. Indeed, it seems natural to ask if there is a simple natural characterization that would allow us to *exactly* compute  $\text{AROS}_1(G)$  for a given combinatorial game, say in time linear in the size of the tree, or at least in polynomial time. We do not have such a characterization at the moment, and one might, in fact, also speculate that this problem could be **NP-hard**. We have already seen that our approach seems to introduce a lot of variation through the choice of representation of the reduced open tree that seems hard to formalize and show an exact bound for. Also, Figure 3 shows an example where we can do even better than what our current approach accomplishes. The strategy profile indicated by the arrows, will win against any of the constructed strategies and could therefore be added to the 1-ranked list as well. This goes to show that a new extended approach would be needed to find the exact Range of Skill.

For the case of imperfect information games and small values of  $\epsilon$ , our understanding is much worse. For instance, for the Texas Hold'em abstraction, our lower bound for the

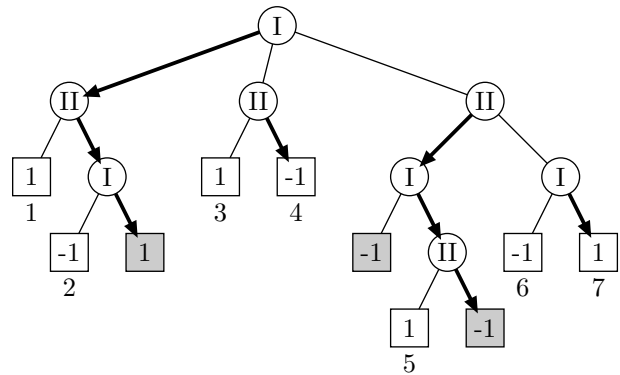


Figure 3: Example showing how to add more strategy profiles than our approach can supply.

Range of Skill is  $1470\epsilon^{-1}$ , while the best upper bound is the one given by Theorem 4. Here, the upper and lower bounds differ by several orders of magnitude and new ideas seem needed to bridge this gap.

A main conclusion of this work is that Theorem 2 does not provide a very good upper bound on the actual number of iterations of the Range of Skill algorithm. We can make the following simple observations: For a combinatorial game  $G$ , any strategy profile in a 1-ranked list is a best response to any mix of strategies of lower ranked strategy profiles. If the algorithm is initialized with the first strategy profile of the longest 1-ranked list, the number of iterations could therefore be exactly  $\text{AROS}_1(G)$ . If, on the other hand, the algorithm is initialized with a perfectly mixed strategy profile, it would terminate in only one iteration. In the more general setting it is not clear how the algorithm behaves, and it would be desirable to gain more insight into this.

### References

Koller, D.; Megiddo, N.; and von Stengel, B. 1994. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, 750–759.

Kushilevitz, E., and Nisan, N. 1996. *Communication Complexity*. Cambridge University Press, New York, USA.

Newman, I. 1991. Private vs. common random bits in communication complexity. *Inf. Proc. Lett.* 39(2):67–71.

Nisan, N. 1993. The communication complexity of threshold gates. In Miklós, V., and Szonyi, T., eds., *Combinatorics, Paul Erdős is Eighty, Volume 1*. János Bolyai Mathematical Society, Budapest. 301–315.

Zinkevich, M.; Bowling, M.; and Burch, N. 2007. A new algorithm for generating equilibria in massive zero-sum games. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 788–793.