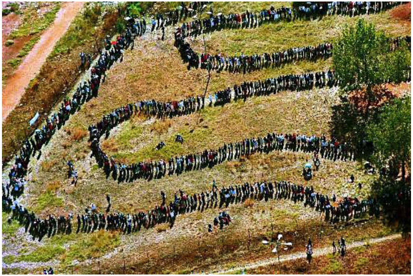
On Range of Skill

Thomas Dueholm Hansen Peter Bro Miltersen
Troels Bjerre Sørensen

Center for Algorithmic Game Theory
Department of Computer Science
University of Aarhus

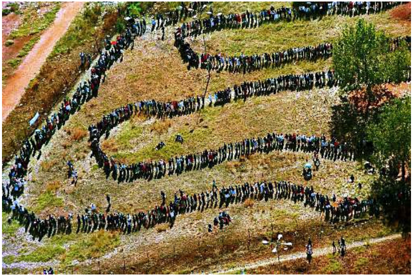
July 17, 2008

The intuition behind the Range of Skill



- Imagine lining up players, such that any player in the line will *win* against all previous players. How long can such a line get?
- A measure of difficulty of gameplaying from a human perspective.

The intuition behind the Range of Skill



- Imagine lining up players, such that any player in the line will *win* against all previous players. How long can such a line get?
- A measure of difficulty of gameplaying from a human perspective.

- Formalized by Zinkevich, Bowling and Burch (AAAI'07) as an input parameter to analyze an algorithm for computing ϵ-Nash equilibria of two-player zero-sum games, in particular imperfect information games.

The Range of Skill Algorithm

1. Let G be a two player zero-sum game with strategy space Γ_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 Γ_i.
3. Repeat
    1. Let G_1 be the game which is like G but with Player 1 restricted to strategies in Σ_1. Let v_1 be the value of G_1, and let (y_1, b_2) be an equilibrium of G_1.
    2. Let G_2 be the game which is like G but with Player 2 restricted to strategies in Σ_2. Let v_2 be the value of G_2, and let (b_1, y_2) be an equilibrium of G_2.
    3. Add b_1 to Σ_1 and b_2 to Σ_2.

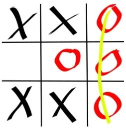
   until $v_2 - v_1 < 2\epsilon$.
4. Return (y_1, y_2).

The Range of Skill Algorithm

- Given a two-player zero-sum game G. The Range of Skill Algorithm by Zinkevich *et al.* computes a 2ϵ-Nash equilibrium in at most $\mathrm{ROS}_\epsilon(G)$ iterations.

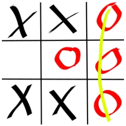
The Range of Skill Algorithm

- Given a two-player zero-sum game G. The Range of Skill Algorithm by Zinkevich *et al.* computes a 2ϵ-Nash equilibrium in at most $\mathrm{ROS}_\epsilon(G)$ iterations.
- Zinkevich *et al.* computed a 0.01-Nash equilibrium of an abstracted game of Limit Texas Hold'em Poker in 298 iterations.
- In iteration i a Nash equilibrium of a game where one player is restricted to i strategies is computed, and it is faster to solve many small games than one large.



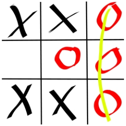
- We introduce methods for rigorously estimating the Range of Skill, showing that it is more than 100,000 for Tic-Tac-Toe and in some cases exponential in the size of the game tree.

Main results



- We introduce methods for rigorously estimating the Range of Skill, showing that it is more than 100,000 for Tic-Tac-Toe and in some cases exponential in the size of the game tree.
- We show that the Range of Skill of the abstraction of Limit Texas Hold'em Poker considered by Zinkevich *et al.* is more than 294,000.

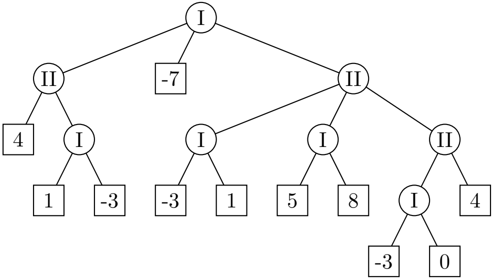
Main results



- We introduce methods for rigorously estimating the Range of Skill, showing that it is more than 100,000 for Tic-Tac-Toe and in some cases exponential in the size of the game tree.
- We show that the Range of Skill of the abstraction of Limit Texas Hold'em Poker considered by Zinkevich *et al.* is more than 294,000.
  - Note that this is 1,000 times more than the number of iterations required by the algorithm. We conclude that the Range of Skill does not accurately predict the running time.

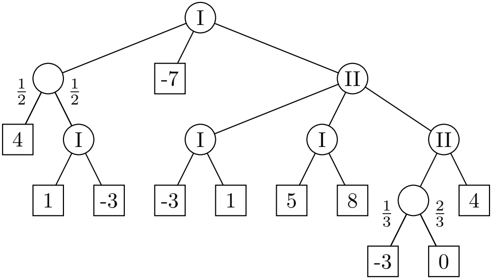
Terminology

- Two-player zero-sum extensive form game.



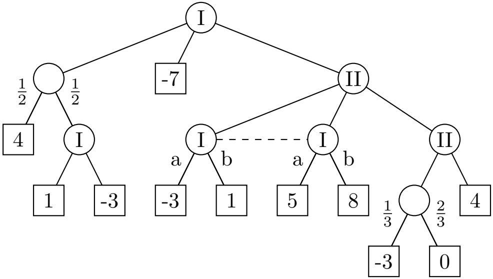
Terminology

- Two-player zero-sum extensive form game.
- Nodes of chance.



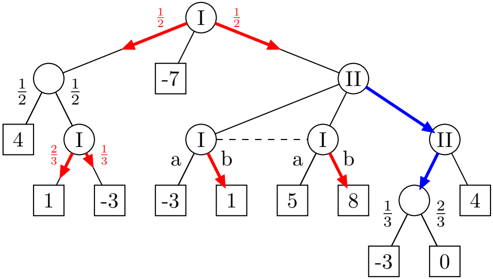
Terminology

- Two-player zero-sum extensive form game.
- Nodes of chance.
- Perfect/imperfect information.



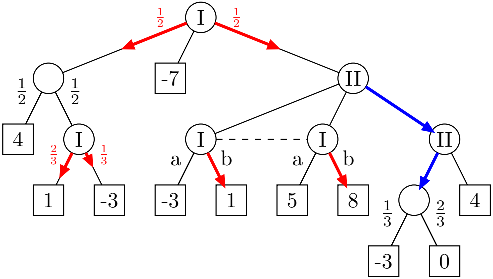
Terminology

- Two-player zero-sum extensive form game.
- Nodes of chance.
- Perfect/imperfect information.
- Strategy profile.



Terminology

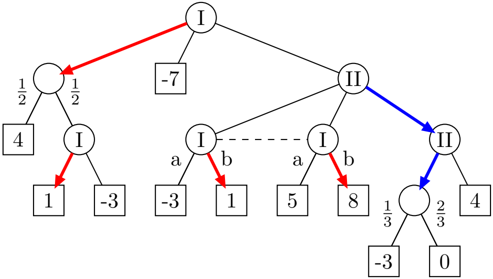
- Two-player zero-sum extensive form game.
- Nodes of chance.
- Perfect/imperfect information.
- Strategy profile.
- Payoff/utility.



$$u(s_1, s_2) = \frac{1}{2}\left(2 + \frac{1}{2}\left(\frac{2}{3} - 1\right)\right) - \frac{1}{2} = \frac{5}{12}$$

Terminology

- Two-player zero-sum extensive form game.
- Nodes of chance.
- Perfect/imperfect information.
- Strategy profile.
- Payoff/utility.
- Nash equilibrium and ϵ-Nash equilibrium.



$$u(s_1, s_2) = 2 + \frac{1}{2} = \frac{5}{2}$$

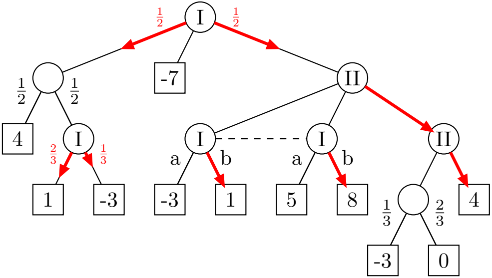
Range of Skill

Definition (Range of Skill)

Given a two-player zero-sum game G with payoff function u, define a list of strategy profiles $(s_1^i, s_2^i), i = 1..N$, to be an ϵ-ranked list, if for all $i > j$, $u(s_1^i, s_2^j) - u(s_1^j, s_2^i) \geq 2\epsilon$. The *Range of Skill* or $\mathrm{ROS}_\epsilon(G)$ is the length of the longest ϵ-ranked list.

Definition (Range of Skill)

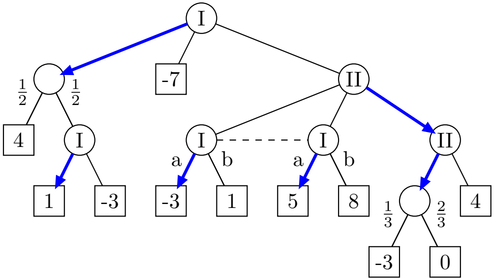
Given a two-player zero-sum game G with payoff function u, define a list of strategy profiles (s_1^i, s_2^i), $i = 1..N$, to be an ϵ-ranked list, if for all $i > j$, $u(s_1^i, s_2^j) - u(s_1^j, s_2^i) \geq 2\epsilon$. The *Range of Skill* or $\mathrm{ROS}_\epsilon(G)$ is the length of the longest ϵ-ranked list.



Strategy profile (s_1^1, s_2^1).

Definition (Range of Skill)

Given a two-player zero-sum game G with payoff function u, define a list of strategy profiles (s_1^i, s_2^i), $i = 1..N$, to be an ϵ-ranked list, if for all $i > j$, $u(s_1^i, s_2^j) - u(s_1^j, s_2^i) \geq 2\epsilon$. The *Range of Skill* or $\mathrm{ROS}_\epsilon(G)$ is the length of the longest ϵ-ranked list.

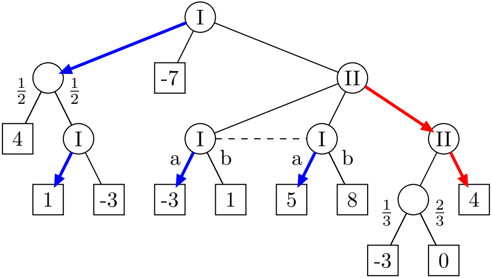


Strategy profile (s_1^2, s_2^2).

Range of Skill

Definition (Range of Skill)

Given a two-player zero-sum game G with payoff function u, define a list of strategy profiles $(s_1^i, s_2^i), i = 1..N$, to be an ϵ-ranked list, if for all $i > j$, $u(s_1^i, s_2^j) - u(s_1^j, s_2^i) \geq 2\epsilon$. The *Range of Skill* or $\mathrm{ROS}_\epsilon(G)$ is the length of the longest ϵ-ranked list.



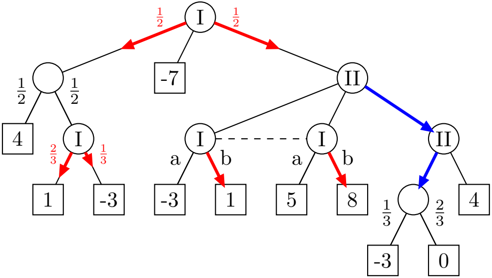
$\{(s_1^1, s_2^1), (s_1^2, s_2^2)\}$ is a 1-ranked list:

$$u(s_1^2, s_2^1) = \frac{5}{2}$$

Range of Skill

Definition (Range of Skill)

Given a two-player zero-sum game G with payoff function u, define a list of strategy profiles $(s_1^i, s_2^i), i = 1..N$, to be an ϵ-ranked list, if for all $i > j$, $u(s_1^i, s_2^j) - u(s_1^j, s_2^i) \geq 2\epsilon$. The *Range of Skill* or $\mathrm{ROS}_\epsilon(G)$ is the length of the longest ϵ-ranked list.

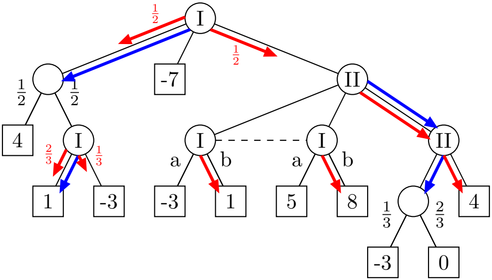


$\{(s_1^1, s_2^1), (s_1^2, s_2^2)\}$ is a 1-ranked list:

$$u(s_1^1, s_2^2) = \frac{5}{12}$$

Definition (Range of Skill)

Given a two-player zero-sum game G with payoff function u, define a list of strategy profiles $(s_1^i, s_2^i), i = 1..N$, to be an ϵ-ranked list, if for all $i > j$, $u(s_1^i, s_2^j) - u(s_1^j, s_2^i) \geq 2\epsilon$. The *Range of Skill* or $\mathrm{ROS}_\epsilon(G)$ is the length of the longest ϵ-ranked list.



$\{(s_1^1, s_2^1), (s_1^2, s_2^2)\}$ is a 1-ranked list:

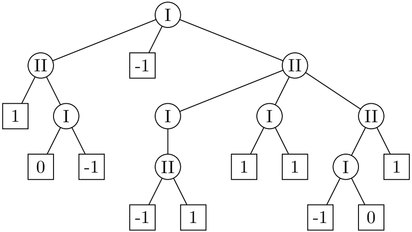
$$u(s_1^2, s_2^1) - u(s_1^1, s_2^2) =$$

$$\frac{5}{2} - \frac{5}{12} = \frac{25}{12} \geq 2$$

ROS_1 for combinatorial games

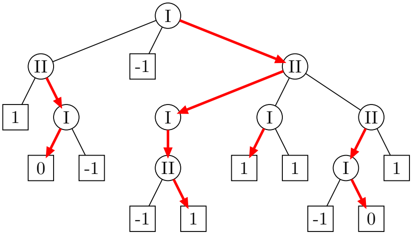
In order to get bounds for Tic-Tac-Toe and Limit Hold'em Poker, we will consider ROS_1 for *combinatorial games*, by which we mean two-player zero-sum extensive form games with:

- Perfect information.
- No moves of chance.
- Payoffs being 1, -1 and 0 (win/lose/tie for Player 1).



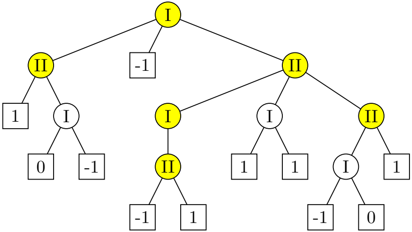
For ROS_1 the higher ranked strategy profile wins no matter who starts the game.

ROS_1 for combinatorial games



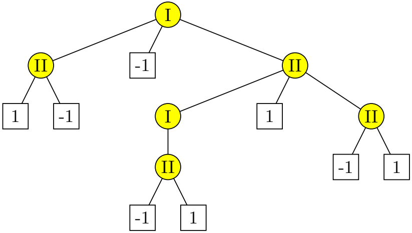
- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.

ROS_1 for combinatorial games



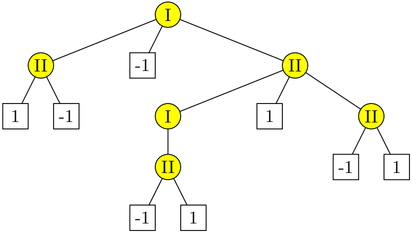
- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.
- We can restrict our attention to the *open* part of the game tree (both 1 and -1 can be reached).

ROS_1 for combinatorial games



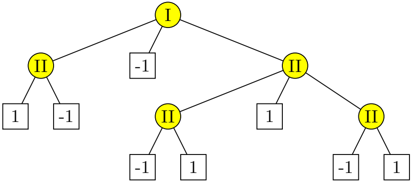
- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.
- We can restrict our attention to the *open* part of the game tree (both 1 and -1 can be reached).

ROS_1 for combinatorial games



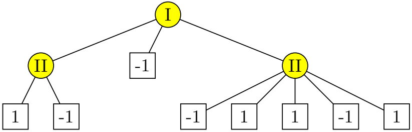
- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.
- We can restrict our attention to the *open* part of the game tree (both 1 and -1 can be reached).
- We can reduce this open tree.

ROS_1 for combinatorial games



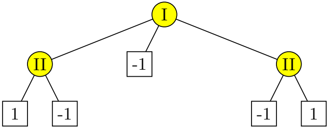
- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.
- We can restrict our attention to the *open* part of the game tree (both 1 and -1 can be reached).
- We can reduce this open tree.

ROS_1 for combinatorial games



- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.
- We can restrict our attention to the *open* part of the game tree (both 1 and -1 can be reached).
- We can reduce this open tree.

ROS_1 for combinatorial games



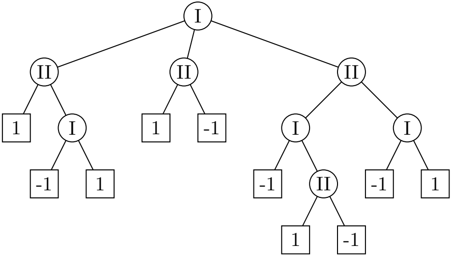
- For ROS_1 we can restrict our attention to strategies where all probability mass is assigned to one action.
- We can restrict our attention to the *open* part of the game tree (both 1 and -1 can be reached).
- We can reduce this open tree.

Constructing strategy profiles for a 1-ranked list

The main idea: The players each choose a number in some range, and during the game **communicate** to find out who chose the higher number in order to let this player win.

Constructing strategy profiles for a 1-ranked list

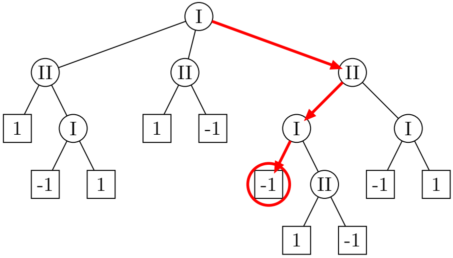
The main idea: The players each choose a number in some range, and during the game **communicate** to find out who chose the higher number in order to let this player win.



- Fix some representation and impose an ordering. Losing leaves to the left and winning leaves to the right.

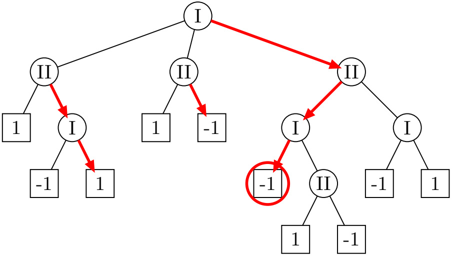
Constructing strategy profiles for a 1-ranked list

The main idea: The players each choose a number in some range, and during the game **communicate** to find out who chose the higher number in order to let this player win.



- Fix some representation and impose an ordering. Losing leaves to the left and winning leaves to the right.
- Pick a leaf that the players are *going for*.

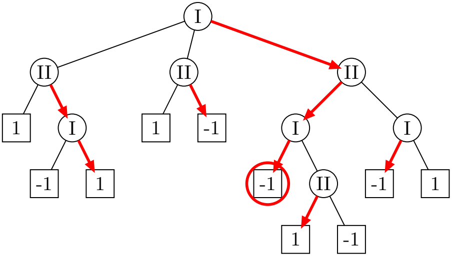
The main idea: The players each choose a number in some range, and during the game **communicate** to find out who chose the higher number in order to let this player win.



- Fix some representation and impose an ordering. Losing leaves to the left and winning leaves to the right.
- Pick a leaf that the players are *going for*.
- To the left: "I'm higher ranked than you".

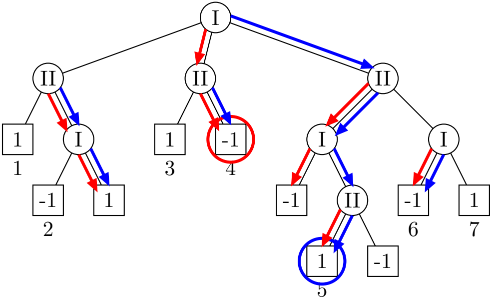
Constructing strategy profiles for a 1-ranked list

The main idea: The players each choose a number in some range, and during the game **communicate** to find out who chose the higher number in order to let this player win.

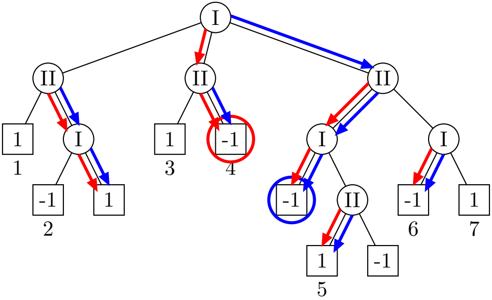


- Fix some representation and impose an ordering. Losing leaves to the left and winning leaves to the right.
- Pick a leaf that the players are *going for*.
- To the left: “I’m higher ranked than you”.
- To the right: “I’m lower ranked than you”.

Is it possible to construct strategy profiles for a 1-ranked list from all the leaves?

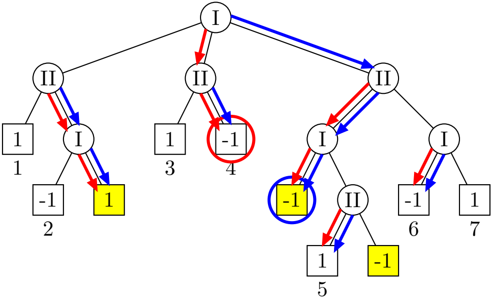


Is it possible to construct strategy profiles for a 1-ranked list from all the leaves? No!



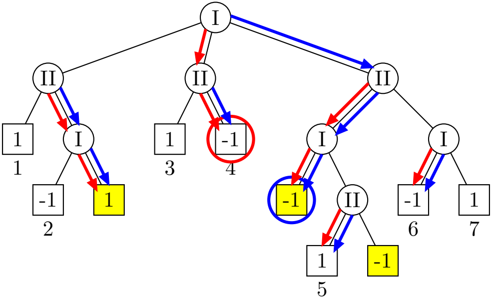
Lower bound

Is it possible to construct strategy profiles for a 1-ranked list from all the leaves? No! We call these leaves *problematic*.



Lower bound

Is it possible to construct strategy profiles for a 1-ranked list from all the leaves? No! We call these leaves *problematic*.



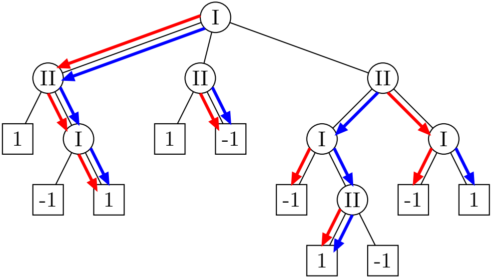
Theorem (Lower bound on ROS_1)

$\mathrm{ROS}_1(G)$ *of a combinatorial game* G *is lower bounded by the number of leaves of the reduced open tree of* G *that are non-problematic.*

Upper bound

Theorem (Upper bound on ROS_1)

$\mathrm{ROS}_1(G)$ *of a combinatorial game* G *is upper bounded by the number of leaves of the reduced open tree of* G*.*



Proof: The pigeonhole principle.

Tic-Tac-Toe

Using a computer program we constructed a 1-ranked list for Tic-Tac-Toe.

- The game tree of Tic-Tac-Toe has 255,168 leaves.
- The open tree has 153,872 leaves.
- The reduced open tree has 131,840 leaves.
- There are 104,615 non-problematic leaves.

Hence, we know that:

$$104,615 \leq \mathrm{ROS}_1(\text{Tic-Tac-Toe}) \leq 131,840$$

Limit Hold'em Poker

- Ignore cards and remove the possibility of a showdown.
- 1471 non-problematic leaves for this simplified version of the abstracted poker game considered by Zinkevich *et al.*

Theorem (Increasing the Range of Skill while lowering ϵ)

For any two-player zero-sum game G, any $\epsilon > 0$, and any integer k, we have $\mathrm{ROS}_{\epsilon/k}(G) \geq k(\mathrm{ROS}_{\epsilon}(G) - 1) + 1$.

- Using the theorem we get that ROS_{ϵ} of the abstracted poker game is at least 294,000 when $2\epsilon = 1/100$.

Asymptotic results

Theorem (Lower bound)

For any $\epsilon > 0$ there is a constant k_ϵ 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, $\mathrm{ROS}_{1-\epsilon}(G) \geq 2^{2^d}$.

From communication complexity:

- The communication complexity of a protocol is the number of bits exchanged in the worst case.
- For any $\epsilon > 0$, there exists a private coin randomized communication protocol for the Greater Than problem on $\{1, \ldots, 2^{2^d}\}$ with error probability ϵ and communication complexity at most cd, for some constant c (Combining Nisan '93 with Newman '91).

Asymptotic results

Theorem (Upper bound)

Let any two-player zero-sum extensive-form game of perfect recall be given (i.e., the players don't forget their own moves). Let n be the total number of actions in the game tree and let β be the largest absolute value of the payoff at any leaf. Then, for any $\epsilon > 0$, we have $\mathrm{ROS}_\epsilon(G) \leq 2(2\beta n/\epsilon)^n$.

Open problems

- Predict more precisely the running time of the Range of Skill Algorithm.

Open problems

- Predict more precisely the running time of the Range of Skill Algorithm. We make the following observations:
  - For any combinatorial game G, the worst-case running time of the Range of Skill Algorithm for computing a 1-equilibrium is $\mathrm{ROS}_1(G)$.

Open problems

- Predict more precisely the running time of the Range of Skill Algorithm. We make the following observations:
  - For any combinatorial game G, the worst-case running time of the Range of Skill Algorithm for computing a 1-equilibrium is $\mathrm{ROS}_1(G)$.
  - For any combinatorial game the Range of Skill Algorithm terminates in one iteration if it is initialized with a strategy profile for which every action is played with positive probability.

Open problems

- Predict more precisely the running time of the Range of Skill Algorithm. We make the following observations:
  - For any combinatorial game G, the worst-case running time of the Range of Skill Algorithm for computing a 1-equilibrium is $\mathrm{ROS}_1(G)$.
  - For any combinatorial game the Range of Skill Algorithm terminates in one iteration if it is initialized with a strategy profile for which every action is played with positive probability.
- Improve the results for games of chance and imperfect information.

Open problems

- Predict more precisely the running time of the Range of Skill Algorithm. We make the following observations:
  - For any combinatorial game G, the worst-case running time of the Range of Skill Algorithm for computing a 1-equilibrium is $\mathrm{ROS}_1(G)$.
  - For any combinatorial game the Range of Skill Algorithm terminates in one iteration if it is initialized with a strategy profile for which every action is played with positive probability.
- Improve the results for games of chance and imperfect information.
- Find a characterization that allows us to exactly compute $\mathrm{ROS}_1(G)$ of combinatorial games. Is this **NP**-hard?

The end

Thank you!