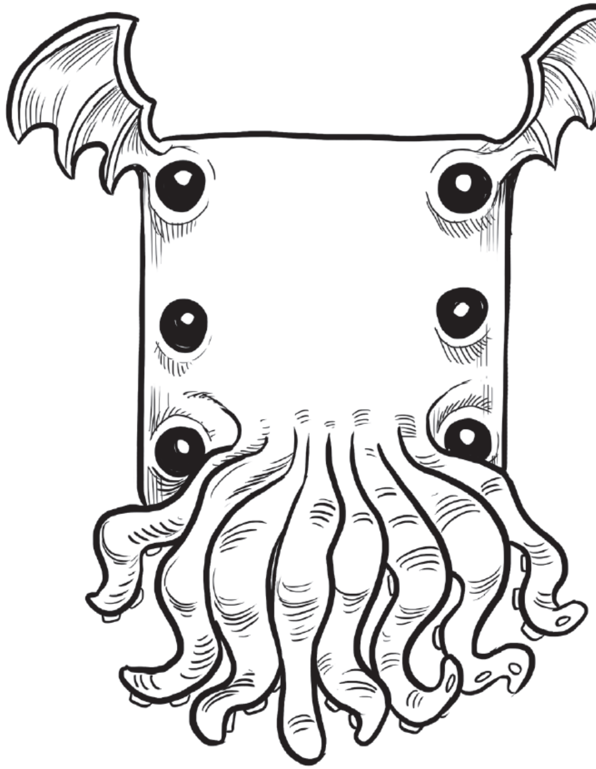
Optimal Play of the Great Rolled Ones Game

Todd W. Neller, Quan H. Nguyen, Phong T. Pham,
Linh T. Phan, and Clifton G.M. Presser

Gettysburg College
Department of Computer Science

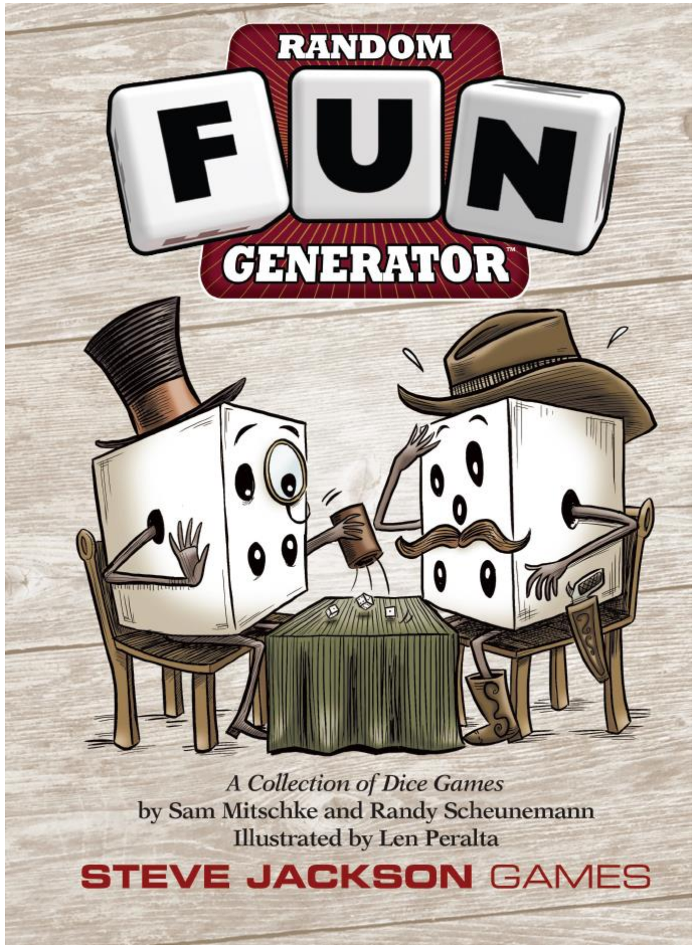
Overview



- Great Rolled Ones Rules
- Optimality Equations
- Solution Method
- Optimal Compensation Points and Visualization
- Human-Playable Policy: Simple player and ones cases
- Conclusions

Great Rolled Ones

- A **jeopardy dice game** for 2 or more players. (Here we consider 2 player only.)
  - Jeopardy ("Push your luck") Dice Game – Primary mechanic: Roll/hold decisions where holding *secures* turn progress, whereas rolling *risks* all turn progress for potentially greater turn progress.
- First published in 2020 by Sam Mitschke and Randy Scheunemann
  - Similar to the dice game Zombie Dice
  - Both are jeopardy dice games in the Ten Thousand dice game family



Great Rolled Ones Rules

- 2 or more players using **5 standard (d6) dice**.
- Players will have the **same number of turns**. A turn consists of **a sequence of player dice rolls where rolled 1s are set aside**.
- The **turn ends when** either the player
  - decides to **hold** (i.e. stop rolling) and score the **total number of non-1s rolled**, or
  - has **rolled three or more 1s**, ending the turn with no score change.
- A round consists of each player taking one turn in sequence.
- Any player ending their turn with a **goal score of 50** or more causes that to be the last round of the game.
- At the end of the last round, the player with the highest score wins.
- (We assume that a player is constrained to attempt to exceed the score of the current leader in the last round.)

Great Rolled Ones Example Round

Player	Roll	Result (Decision)
1	1, 1, 3, 4, 5	Two 1s set aside, turn total 3 (roll)
1	2, 2, 4	No 1s set aside, turn total 6 (roll)
1	1, 1, 6	Two 1s set aside for a total of four 1s, ≥ three 1s → **turn ends with no score gain**
2	4, 4, 4, 4, 5	No 1s set aside, turn total 5 (roll)
2	4, 4, 4, 5, 5	No 1s set aside, turn total 10 (roll)
2	1, 1, 2, 4, 5	Two 1s set aside, turn total 13 (hold) → **turn ends with a score gain of 13**

Optimality Equations: Probability of Rolling 1s

Nonterminal states are described as the 5-tuple (p, i, j, k, o), where p is the current player number (1 or 2), i is the current player score, j is the opponent score, k is the turn total, and o is the number of rolled 1s set aside.

Let $P_{\text{new1s}}(d, o_{\text{new}})$ denote the probability that o_{new} of d dice rolled are 1s $(0 \leq o_{\text{new}} \leq d \leq 5)$:

$$P_{\text{new1s}}(d, o_{\text{new}}) = \binom{d}{o_{\text{new}}} \left(\frac{1}{6}\right)^{o_{\text{new}}} \left(\frac{5}{6}\right)^{(d-o_{\text{new}})}$$

Optimality Equations: Probability of Player 2 Exceeding Player 1's Winning Score

Let $P_{\text{exceed}}(\Delta, o)$ denote the probability that player 2 will exceed player 1's score ≥ 50 where $\Delta = j - (i + k)$ (their score difference) and o is the number of rolled 1s set aside on player 2's final turn. Then,

$$P_{\text{exceed}}(\Delta, o) = \begin{cases} 0 & \text{if } o \geq 3 \\ 1 & \text{if } \Delta < 0 \\ \sum_{n=0}^{2-o} P_{\text{new1s}}(5 - o, n) P_{\text{exceed}}(\Delta - (5 - o'), o') & \text{otherwise} \\ \qquad \text{where } o' = o + n & \end{cases}$$

Optimality Equations: Probability of Winning with a Roll

The probability of winning with a roll $P_{\text{roll}}(p, i, j, k, o)$ under the assumption of optimal play thereafter is:

$$P_{\text{roll}}(p,i,j,k,o) = \begin{cases} P_{\text{exceed}}(j-i, o) & \text{if } p = 2 \text{ and } j \geq 50 \\ \sum_{n=0}^{2-o} P_{\text{new1s}}(5-o, n) P(p, i, j, k+5-o', o') + \sum_{n=3-o}^{5-o} P_{\text{new1s}}(5-o, n)(1 - P(3-p, j, i, 0, 0)) & \text{otherwise} \end{cases}$$

A player can (and should) never hold at the beginning of the turn when the turn total is 0, so we express this by treating such rule-breaking as a loss. Thus, …

Optimality Equations: Probability of Winning with a Hold, Roll/Hold Decision

the probability of winning with a hold $P_{\text{hold}}(p, i, j, k, o)$ under the assumption of optimal play thereafter is:

$$P_{\text{hold}}(p,i,j,k,o) = \begin{cases} 0 & \text{if } k = 0 \text{ or } (p = 2 \text{ and } j \geq 50, i) \\ 1 & \text{if } p = 2 \text{ and } i + k \geq 50, j \\ 1 - P(3-p, j, i+k, 0, 0) & \text{otherwise} \end{cases}$$

Then the probability of winning $P(p, i, j, k, o)$ under the assumption of optimal play is:

$$P(p,i,j,k,o) = \max(P_{\text{roll}}(p,i,j,k,o), P_{\text{hold}}(p,i,j,k,o))$$

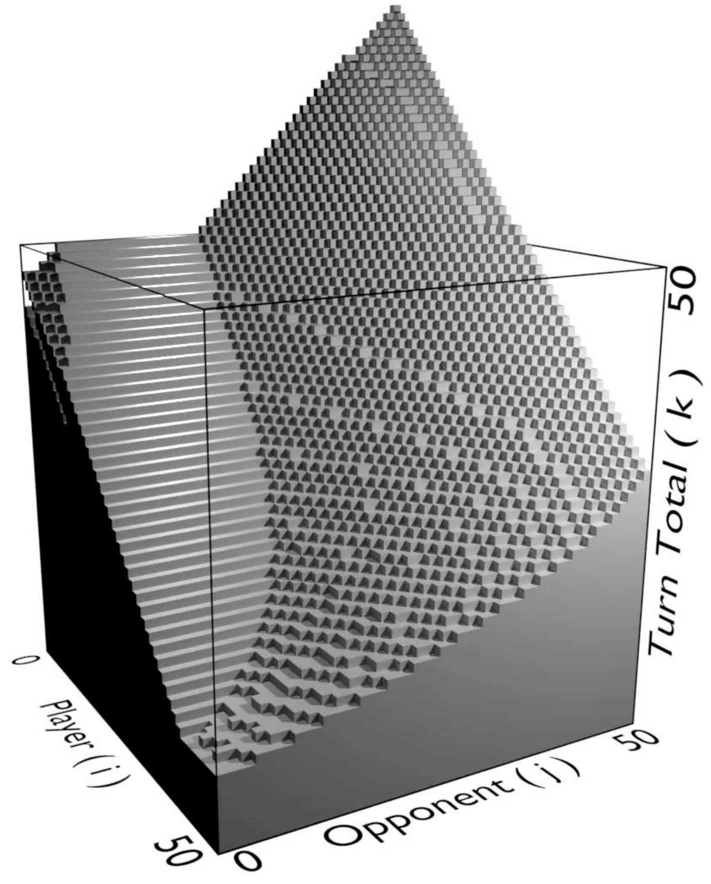
Solving Optimality Equations

- Equations (P_{new1s}, P_{exceed}) are solved through **dynamic programming** first.
- Cyclic, recursive *P* is solved through a **variation of value iteration**:
  - From initial arbitrary *P* estimates, substitute estimates in equation right-hand sides.
  - Compute the left-hand side *P* values as new, better estimates.
  - Terminate iterations of previous steps when the maximum change to a *P* estimate is $\leq 1\times10^{-14}$.

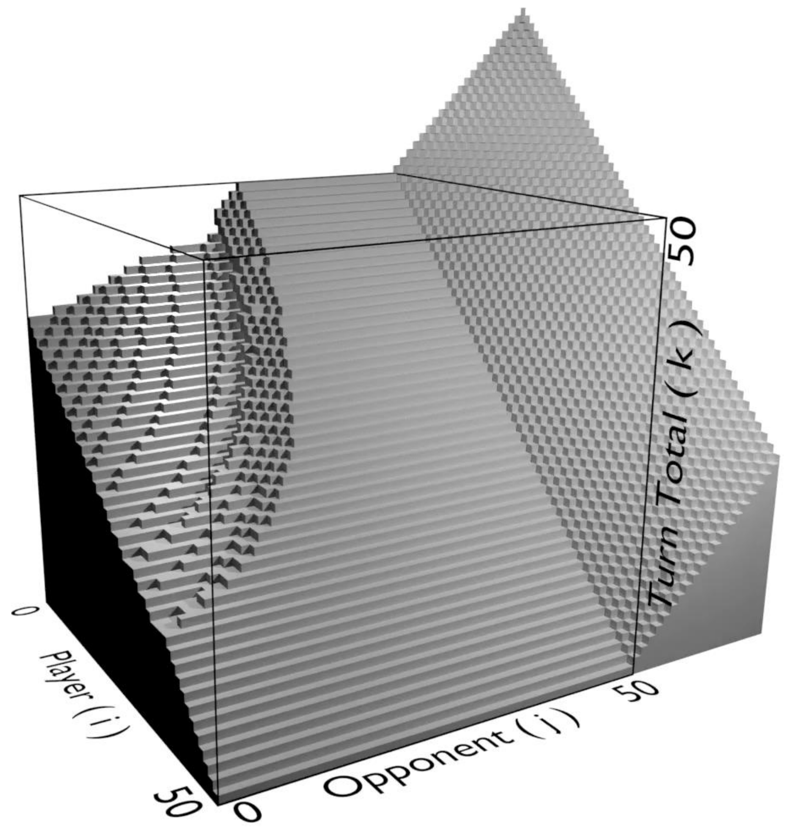
First Player Advantage and Compensation Points (Komi)

- Player 1 finishes with ≥ 50 → Player 2 must exceed Player 1's score
- Player 2 has a knowledge advantage, knowing what score is needed to win.
- With optimal play, player 1 and player 2 have win rates of 0.4495 and 0.5505, respectively (a **10% gap**!).
- In the game of Go, "komi" are compensation points designed to make games more fair.
- In the Great Rolled Ones game, **player 1 should start with 3 compensation points (komi)**, bringing player 1's win rate up to 0.4955 (a **0.9% gap**) for most fair play.

Optimal Play (zero 1s set aside)

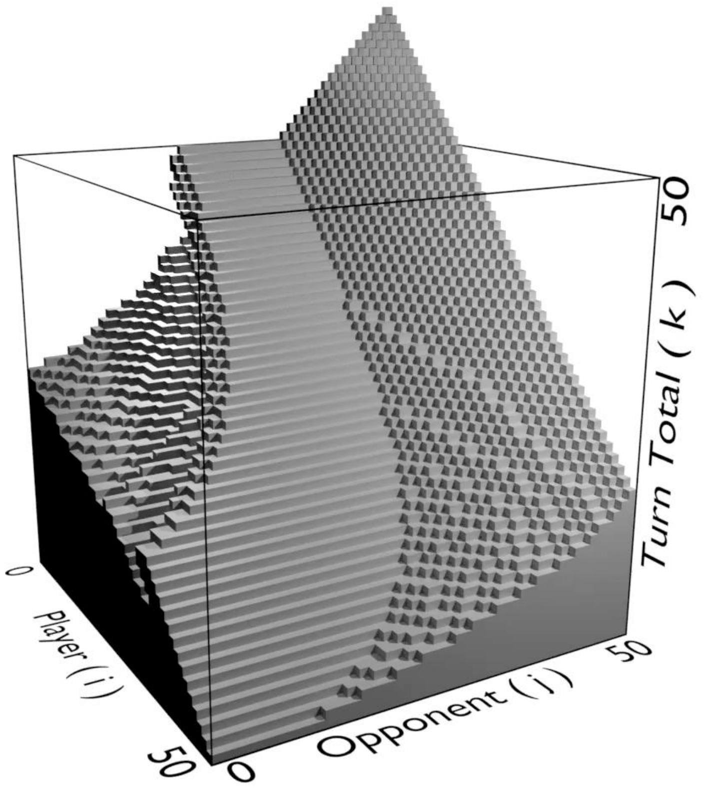


Player 1

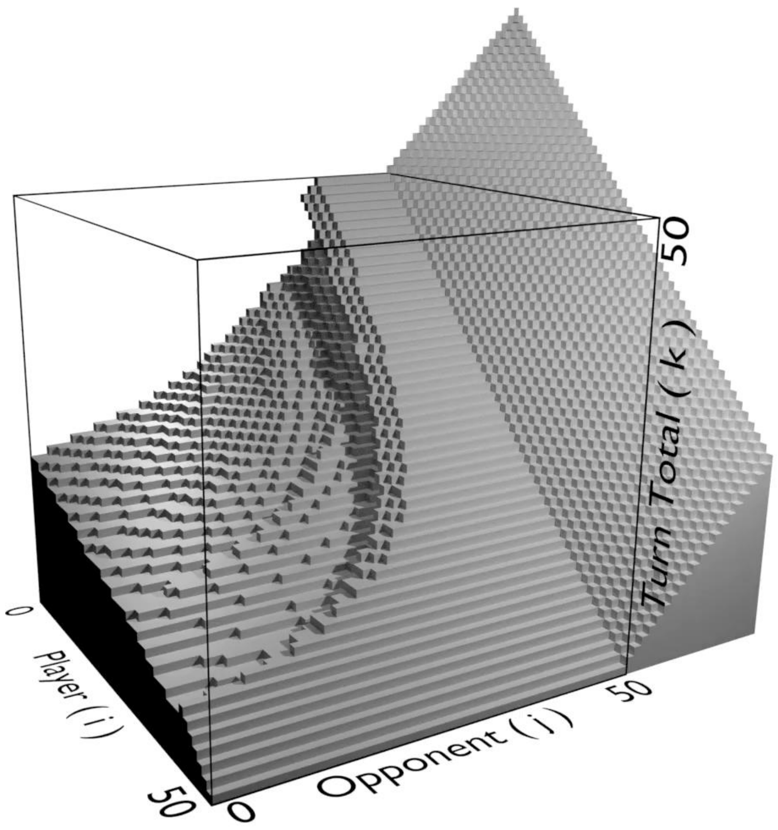


Player 2

Optimal Play (one 1 set aside)

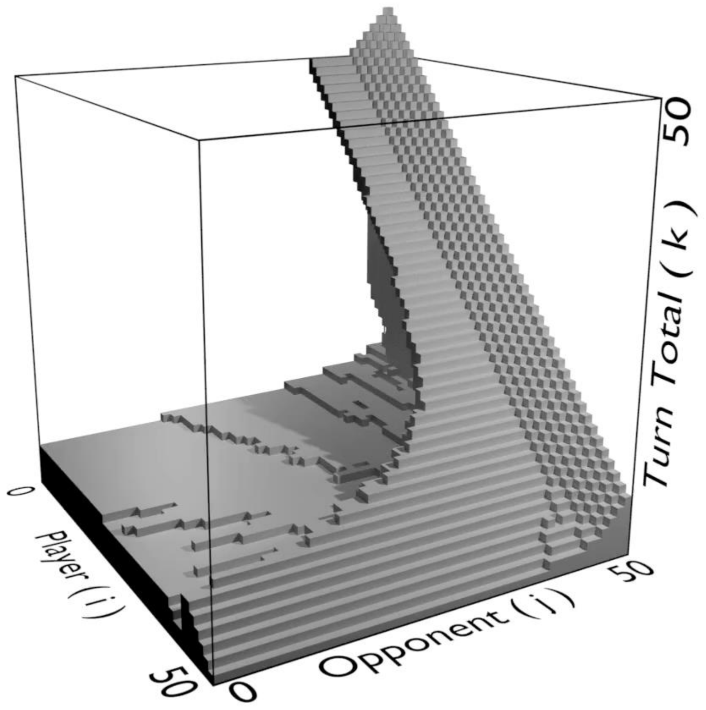


Player 1

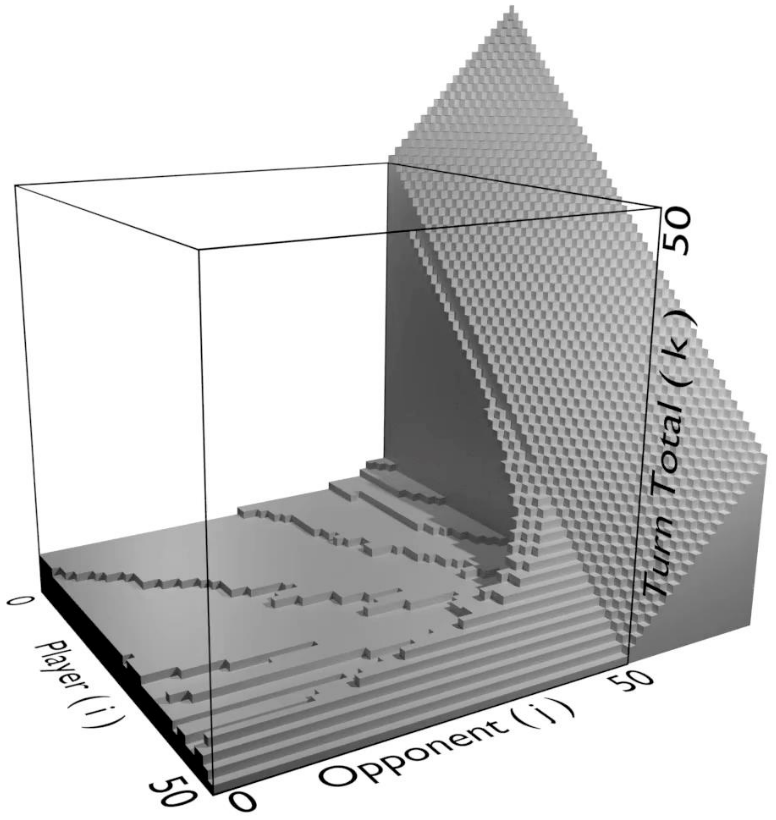


Player 2

Optimal Play (two 1s set aside)



Player 1

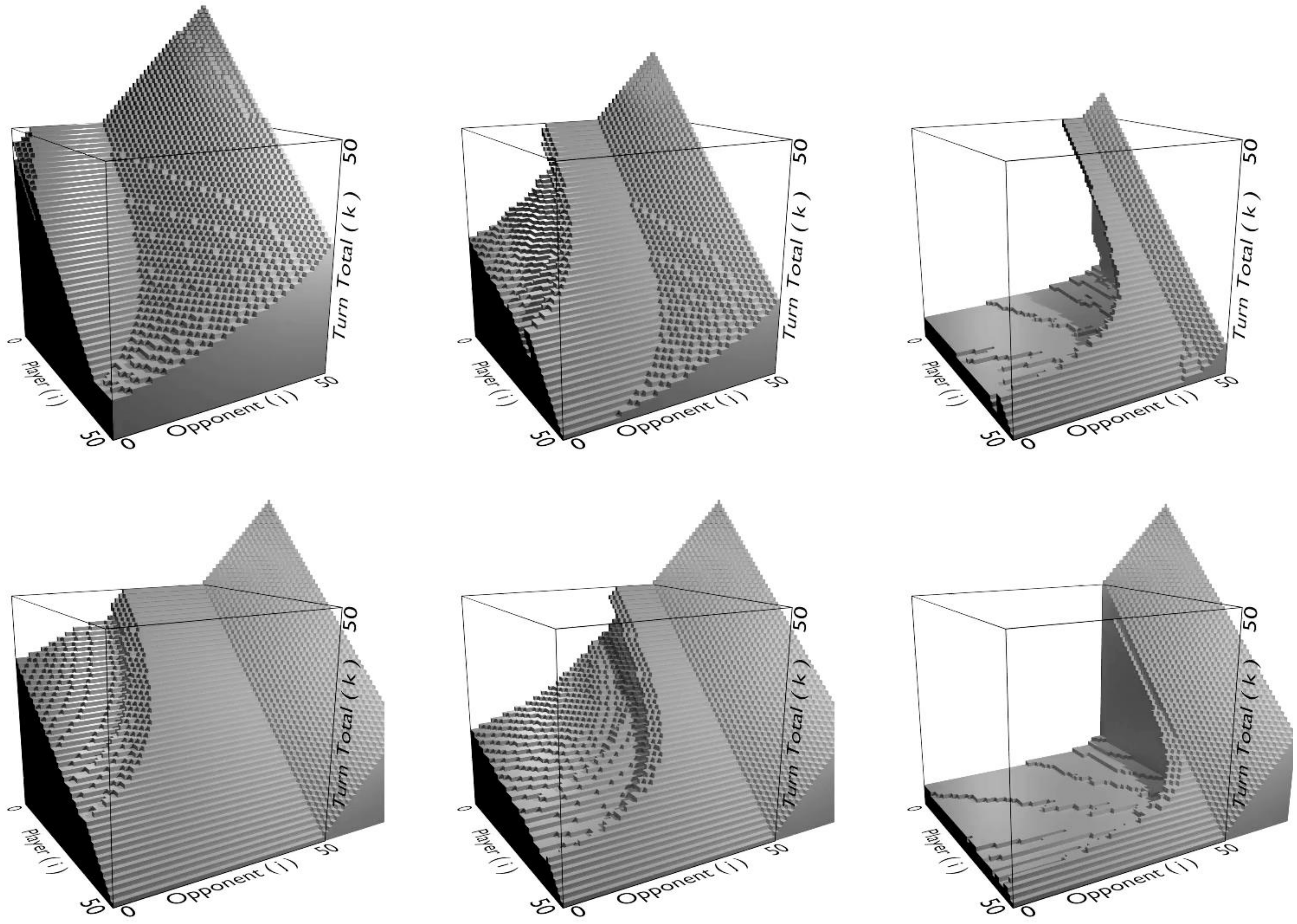


Player 2

Zero 1s | One 1 | Two 1s

Player 1

Player 2



Human-Playable Policies

- By human-playable, we mean involving simple mental arithmetic and limited recall of cases and constants.
- We observe a continuum of tradeoffs from simple play policies with modest performance, to complex play policies with excellent performance.

Policy	Difference
Roll with 4 or 5 Dice	-0.0536
Fixed Hold-At	-0.0268
Simple Player and Ones Cases	-0.0201
Keep Pace, End Race, by Case	-0.0100

Fig. 2: Differences between human-playable and optimal policy win rates

Simple Player and Ones Cases (cont.)

- For player 1,
  - roll with 5 dice.
  - With 4 dice, hold at or beyond the goal with a lead of at least 20.
- For player 2,
  - if player 1 has reached the goal score, exceed it.
  - Otherwise, if player 2 can hold and win, do so.
  - Otherwise, player 2 always keeps rolling to win with 4 or 5 dice.
- With 3 dice, both players should hold if it reaches the goal score or if the turn total is at least 5.
- Such play wins only ~2.0% less than optimal play!

Conclusions

- Optimal play has been computed for the Great Rolled Ones game.
- 3 compensation points should be given initially to Player 1 for greatest fairness
- Among the variety of human playable strategies analyzed, we shared the “Simple Player and Ones Cases” strategy that has a 2% gap from the optimal win rate:
  - (Player 2 must exceed a winning Player 1 score.)

1s Rolled	Player 1	Player2
0	Roll and **do not hold**	Roll to win
1	Hold at ≥ 50 points with a lead of ≥ 20	Roll to win
2	Hold at ≥ 5 turn total or ≥ 50 points	Hold at ≥ 5 turn total or ≥ 50 points

Roll with 4 or 5 Dice

Algorithm 1: Roll with 4 or 5 dice

Input : player p, player score i, opponent score j, turn total k, ones rolled o
Output: whether or not to roll

if $p = 2 \land j \geq 50 \land i + k \leq j$ **then** // player 2 must exceed player 1
    return *true*
else if $p = 2 \land i + k \geq 50$ **then** // player 2 must hold at goal score
    return *false*
else // roll with 4 or 5 dice
    return $o < 2$
end if

5.4% gap

Fixed Hold-At

Algorithm 2: Fixed hold-at

Input : player p, player score i, opponent score j, turn total k, ones rolled o
Output: whether or not to roll

if $p = 2 \wedge j \geq 50 \wedge i + k \leq j$ **then** `// player 2 must exceed player 1`
  return *true*
else if $i + k \geq 50$ **then** `// player 2 holds and wins`
  return *false*
else if $o = 0$ **then** `// keep rolling with 5 dice`
  return *true*
else if $o = 1$ **then** `// hold at 24 with 4 dice`
  return $k < 24$
else `// hold at 4 with 3 dice`
  return $k < 4$
end if

2.7% gap

Simple Player and Ones Cases

Algorithm 3: Simple player and ones cases

Input : player p, player score i, opponent score j, turn total k, ones rolled o
Output: whether or not to roll

```
if p = 1 then                                          // player 1 cases
    if o = 0 then                                      // keep rolling with 5 dice
        return true
    else if o = 1 then                                 // hold at goal with ≥ 20 lead with 4 dice
        return k < max(50 − i, 20 + j − i)
    else                                               // hold at 5 or goal with 3 dice
        return k < min(50 − i, 5)
    end if
else                                                   // player 2 cases
    if j ≥ 50 then                                     // player 2 must exceed player 1
        return i + k ≤ j
    else if i + k ≥ 50 then                            // hold at goal score
        return false
    else if o < 2 then                                 // roll with 4 or 5 dice
        return true
    else                                               // hold at 5 or goal with 3 dice
        return k < min(50 − i, 5)
    end if
end if
```

2.0% gap

Keep Pace, End Race, by Case

Algorithm 4: Keep pace, end race, by case

Input : player p, player score i, opponent score j, turn total k, ones rolled o

Output: whether or not to roll

```
δ ← j − i
if p = 1 then                                          // player 1 cases
    if o = 0 then                  // hold at goal with ≥ 38 lead with 5 dice
        return k < max(50 − i, 38 + δ)
    else if o = 1 then
        h ← 22 + δ                          // hold with a ≥ 22 lead with 4 dice
        if i ≥ 10 ∨ j ≥ 23 then
            // if player 1 / 2 has scored 10 / 23, resp.
            h ← max(50 − i, h)                // then at least roll for the goal
        end if
        return k < h
    else if i + j ≥ 71 then
        // reach the goal when the player score sum reaches 71
        return k < 50 − i
    else                                       // hold at 5 or goal with 3 dice
        return k < min(50 − i, 5)
    end if
else                                                   // player 2 cases
    if j ≥ 50 then                         // player 2 must exceed player 1
        return k ≤ δ
    else if o = 0 then                           // keep rolling with 5 dice
        return true
    else if o = 1 then                                     // with 4 dice
        if i ≥ 20 ∨ j ≥ 32 then
            // if player 1 / 2 has scored 20 / 32, resp.
            return k < 50 − i                        // then roll for the goal
        else                                     // else hold with ≥ 28 lead
            return k < 18 + δ
        end if
    else if i + j ≥ 84 then
        // reach the goal when the player score sum reaches 84
        return k < 50 − i
    else                                       // hold at 5 or goal with 3 dice
        return k < min(50 − i, 5)
    end if
end if
```

1.0% gap

Future Work

- Supervised learning of win probabilities for nonterminal states could compress our precise tabular computation.
- One-step backup of approximate win probabilities would likely yield excellent roll/hold decisions.
- Question: How well would different models/techniques perform for trading off performance for reduced memory requirements?