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, Gettysburg, USA
tneller@gettysburg.edu

Abstract. In this paper, we solve and visualize optimal play for the Great Rolled Ones jeopardy dice game by Mitschke and Scheunemann [4, p. 4–5]. We share the second player advantage and compute that the first player should start with 3 compensation points (komi) for greatest fairness. We present a spectrum of human-playable strategies that trade off greater play complexity for better performance, and collectively clarify key considerations for excellent play.

1 Introduction

Great Rolled Ones is a jeopardy dice game first published in 2020 by Sam Mitschke and Randy Scheunemann [4, p. 4–5] that is similar to the game Zombie Dice [1]. Both are jeopardy dice games [3, Ch. 6] in the Ten Thousand dice game family [2]. In this paper, we analyze Great Rolled Ones, computing optimal play as well as providing additional insights to gameplay.

We begin by describing the rules of Great Rolled Ones, and then define 2-player optimality equations and our method for solving them. We calculate compensation points (komi) for a fairest game, visualize the policy, and share observations on the optimal roll/hold boundary. We then present an array of human-playable policies we have devised along with their performances against the optimal policy. The policies demonstrate different design trade-offs of greater complexity for greater win rates, and highlight key play policy considerations. Finally, we discuss future work and summarize our conclusions.

2 Rules

Great Rolled Ones (GRO) is a dice game for two or more players using 5 standard (d6) dice. In this paper, we will focus on the two-player GRO game. Players will have the same number of turns. A *turn* consists of a sequence of player dice rolls where rolled 1 s are set aside. The turn ends when either the player decides to *hold* (i.e. stop rolling) and score the total number of non-1 s rolled, or has rolled three or more 1 s, ending the turn and scoring 0 points. A *round* consists of each player taking one turn in sequence. Any player ending their turn with a goal

M. Hartisch et al. (Eds.): ACG 2023, LNCS 14528, pp. 50–59, 2024.
https://doi.org/10.1007/978-3-031-54968-7_5

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.

Given that the rules refer to a singular winner ("cultist with the most rituals", i.e. player with the most points) after the last round where "everyone else loses", this implies that no player can opt to draw, and thus a player is constrained to attempt to exceed the score of the current leader in the last round. An optimal player must attempt to win when a prior player of that round has reached 50 or more points.

Example round:

- Player 1 initially rolls {1, 1, 3, 4, 5}. Two 1 s were rolled and set aside, so 3 is added to the *turn total* of non-1 s rolled. Player 1 chooses to *roll* the remaining three non-1 dice again with a result of {2, 2, 4}. No 1 s were rolled and set aside, so 3 is again added to the turn total for a new turn total of 6. Player 1 chooses to roll the three remaining non-1 dice again with a result of {1, 1, 6}. Two 1 s are set aside, for a total of four 1 s. Three or more 1 s ends a turn scoring 0 points, so play passes to player 2 with no score change.
- Player 2 initially rolls {4, 4, 4, 4, 5}, sets no 1 s aside, has a turn total of 5, and chooses to roll again. Player 2 rolls {4, 4, 4, 5, 5}, setting no 1 s aside, has a new turn total of 10, and chooses to roll again. Player 2 rolls {1, 1, 2, 4, 5}, sets two 1 s aside, has a new turn total of 13, and chooses to *hold*, scoring 13 points and ending the round.

The game thus consists of roll/hold risk assessment in a race to achieve the top score of 50 or more points within the same number of turns as other players. Given the player scores, the turn total, and the number of 1 s set aside, should the current player roll or hold so as to maximize the probability of winning?

3 Optimality Equations and Solution Method

We here define optimality equations for the GRO two-player game where player 2 must seek to exceed player 1's score when it is at least 50.

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 1 s set aside.

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

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

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 1 s set aside on player 2's final turn. Then,

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

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, 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:

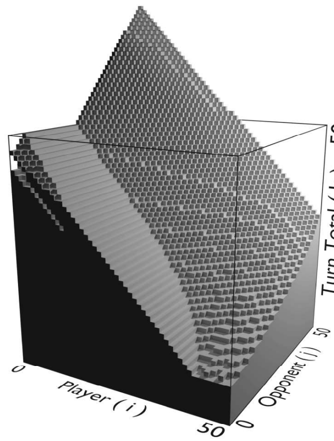
$$P(p, i, j, k, o) = \max(P_{\text{roll}}(p, i, j, k, o), P_{\text{hold}}(p, i, j, k, o))$$

What remains is to bound our nonterminal states for computation. The rules have no restriction on how high a turn total (and thus a score) can go. Our approach is to create a high enough artificial maximum score M (e.g. 100) bounding i, j, and k such that optimal policy does not expand play to further nonterminal states for any tested increase in M.

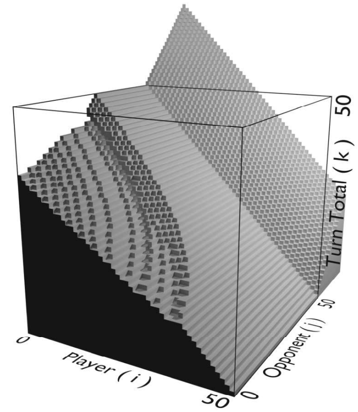
Having bounded our nonterminal state space representation such that $p \in \{1, 2\}, 0 \leq i, j, k \leq M, 0 \leq o \leq 2$, we apply value iteration as in [5] until the maximum probability change of an iteration is less than $\epsilon = 10^{-14}$.

4 Optimal Policy

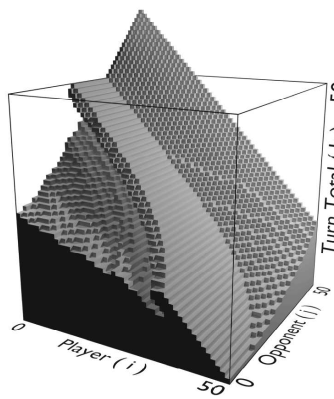
The optimal roll/hold boundaries of GRO are shown in Fig. 1. Each subfigure depicts a 3-dimensional (i, j, k) roll/hold boundary for each possible pair of player p and rolled ones o. Axes are player score i, opponent score j, and turn total k. Given a current state inside or outside of the appropriate solid, an optimal player should roll or hold, respectively.



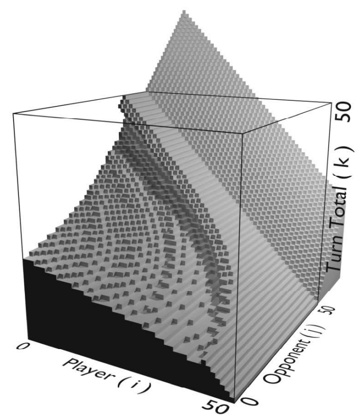
(a) Player 1 with no 1s rolled



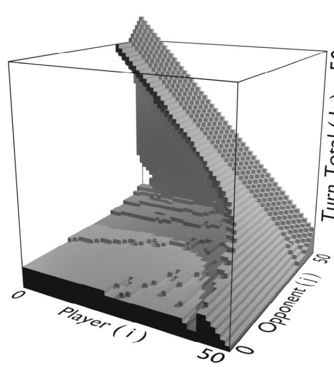
(b) Player 2 with no 1s rolled



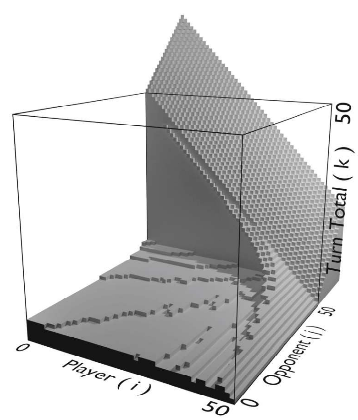
(c) Player 1 with one 1 rolled



(d) Player 2 with one 1 rolled



(e) Player 1 with two 1s rolled



(f) Player 2 with two 1s rolled

Fig. 1. GRO optimal play visualization. A player in a state inside or outside the gray solid should roll or hold, respectively. Subfigures are by p, o cases, and axes follow i, j, k state variables. (Color figure online)

We first observe a few expected similarities between these roll/hold solids. First, the $i+k=50$ diagonal plane indicating a rolling for the goal score appears in situations where player(s) are close to the end of the game or have little to risk with many dice to roll. Player 2 must exceed player 1's score when it reaches the goal score, so the plane $i+k=j+1$ is also a prominent hold plane. As a player has fewer dice to roll, play becomes more conservative.

There are some interesting differences and subtleties to observe as well. Player 1 plays more aggressively than player 2 with higher minimum hold values with all other state variables being equal. Also, there are interesting nonlinearities when player 1 seeks to not just reach 50 points, but to far enough exceed 50 so as to make it unlikely that player 2 will exceed their final score. Player 2, having the opportunity to exceed player 1's final score, has an advantage and generally plays so as to keep within striking distance of player 1's score.

Most interesting and complex are the roll/hold boundaries when a player has rolled one 1. Here we observe nonlinearities in the roll/hold surface for both players. Whereas one might approximate player with no 1 s or two 1 s as "always roll" and "hold at 5", respectively, the roll/hold surface shape is relatively complex when the current player has rolled one 1 and player scores are not close to the goal.

5 Komi

The win rate of player 1 when play is optimal is ∼0.4495, a ∼10% gap from the second player win rate. This second player advantage comes from the fact that, while both players have the same number of turns to win, player 2 has the informational advantage of knowing what score must be exceeded when player 1 scores 50 or more points first.

Komi, i.e. compensation points in the game of Go, serve to make a game more fair. For GRO, fairest optimal play komi would start player 1 with 3 compensation points bringing player 1's win rate up to 0.4955, or within 1% of perfectly fair play.

6 Human-Playable Policies

In this section, we present a range of human-playable policies mapping states to roll/hold actions that trade off greater complexity for greater win rate. By *human-playable*, we mean that all roll/hold decisions may be made through simple mental math. As we will see, these policies range from extremely simple rules to very-complex sub-cases requiring memorization of several constants in order to approximate roll/hold surfaces.

Each policy is evaluated against the optimal policy with each having equal probability of playing first. Policy evaluation follows the same value-iteration-style algorithm of [6]. The performance of each is summarized in Fig. 2.

We present each policy as a method that returns whether or not to roll in the given state.

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

6.1 Roll with 4 or 5 Dice Policy

Simplest is to always roll 4 or 5 dice (unless player 2 can hold and win), and always hold with 3 or fewer dice (unless player 2 must exceed player 1's game-ending score):

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

Surprisingly, Algorithm 1 wins only ~5.4% less than the optimal policy. For all of the nuances of optimal play, this trivial baseline performance immediately hints at high human play performance possibilities.

6.2 Fixed Hold-At Policy

Next, we consider a policy where we need only remember a few turn total thresholds.

Requiring memorization of only two hold-at constants (24 and 4), Algorithm 2 reduces the optimal play gap to ~2.7%.

6.3 Simple Player and Ones Cases

The fixed hold-at policy had the same play policy for both players with the exception of player 2's game-ending constraints. This next policy breaks down cases not only by number of ones rolled o, but also by current player number p.

Algorithm 3 also requires memorization of only two constants (20 and 5), yet requires more case memorization. Even so, breaking down cases according to

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 ∧ j ≥ 50 ∧ i + k ≤ j then          // player 2 must exceed player 1
    return true
else if i + k ≥ 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
```

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
```

player p and the number of ones rolled o impressively reduces the optimal play gap to $\sim$2.0%.

In the trade-off of increased cognitive complexity for increased performance, this algorithm might represent a preferred middle ground for players. In prose, we might describe this policy as follows:

> 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 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.

6.4 Keep Pace, End Race, by Case

Algorithm 4 also breaks down cases by player p and number of ones set aside o, computing hold-at values sensitive to score difference $\delta = j - i$ combined with roll-to-the-end thresholds.

This policy requires even more case analysis, being sensitive to individual scores or score sums reaching progress thresholds. Ten constants are considerably more to remember, as well. Still, this extra work even better approximates optimal play performance, closing the optimal play gap to $\sim$1.0%.

7 Future Work

One might use supervised learning on our roll/hold or win probability tables to compress a good play policy in memory without significantly sacrificing performance. Given that relatively simple human playable policies can perform within a few percent of optimal, it would be interesting to see what memory reductions via supervised learning are possible that closely approximate optimal play.

We conjecture that memory-efficient supervised learning models that approximate the probability of winning for each (p, o) pair could be used with a one-step backup of optimality equations in order to make an excellent, compact computational approximation of optimal play policy.

Another possibility for future work is to survey expected game lengths of the most popular jeopardy dice games, and tune the GRO goal score so as to optimize its game length. In games of chance, there is a trade-off between game brevity and the rewarding of player skill. With few decisions, a player's skill is difficult to discern through the game's variance. With many decisions, a player's skill will be rewarded with a noticeable gain in win rate (e.g. backgammon). However, a game of chance with too many decisions can become tedious.

We believe there is potential to use our analytical tools or reinforcement learning approximations of optimal play in order to advance AI-assisted game design for jeopardy dice games.

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
```

8 Conclusions

In this paper, we have computed and visualized optimal play for the 2-player case of the Great Rolled Ones jeopardy dice game. We determined that the first player should start with 3 points for fairest play. In visualizing roll-hold boundaries, we showed a number of interesting nonlinear features, and gave visual insight to the play implications of player 2's advantage from always having the last turn.

In addition, we presented a variety of human-playable strategies, ranging from trivial to complex, with optimal play performance gaps ranging from ∼5.4% to ∼1.0%, respectively. For casual play, we are especially pleased to recommend Algorithm 3 with an optimal play performance gap of only ∼2.0%.

The Great Rolled Ones game has a fairly complex optimal roll-hold policy boundary, as shown in Fig. 1, and yet relatively simple human-playable policies offer decent performance against optimal play, revealing some of the key considerations for excellent play.

References

1. Zombie dice. https://boardgamegeek.com/boardgame/62871/zombie-dice Accessed 24 May 2023
2. Busche, M., Neller, T.W.: Optimal play of the farkle dice game. In: Winands, M.H.M., van den Herik, H.J., Kosters, W.A. (eds.) ACG 2017. LNCS, vol. 10664, pp. 63–72. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71649-7_6
3. Knizia, R.: Dice Games Properly Explained. Elliot Right-Way Books, Brighton Road, Lower Kingswood, Tadworth, Surrey, KT20 6TD U.K. (1999)
4. Mitschke, S., Scheunemann, R.: Random Fun Generator. Steve Jackson Games, Inc. (2000)
5. Neller, T.W., Presser, C.G.: Optimal play of the dice game pig. UMAP J. **25**(1), 25–47 (2004)
6. Neller, T.W., Presser, C.G.: Practical play of the dice game pig. UMAP J. **31**(1), 5–19 (2010)