sciendo FCT Fundação para a Ciência e a Tecnologia MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR CIUHCT Ludus BOARD GAME STUDIES

# Mathematical analysis of the Royal Game of Ur

*Diego J. Raposo (0000-0002-8100-8463)*
*diegoraposo@hotmail.com*

*Padraig X. Lamont (0000-0002-1330-9790)*
*padraiglamont@gmail.com*

**Abstract:** Despite many discoveries and proposals for rules for the ancient board game known as the Royal Game of Ur (RGU), no mathematical analysis has yet been performed investigating those rules. In an attempt to fill that gap, this paper presents an initial mathematical analysis of the RGU from an introductory point of view. The paper deduces the overall complexity of the RGU using a state-space and game-tree complexity analysis, allowing the RGU to be compared to the popular games Checkers, Backgammon, Ludo, Chess, and Go. The paper builds upon the fundamental laws of combinatorics and probability to improve the understanding of the game: what patterns should you expect, what moves increase your chance to win, and what moves should you avoid. The paper also presents theorems to predict the probability of future dice rolls and piece movements within the game, allowing basic inferences to be made about strategy in the RGU. The game is further examined by analysing three different influences when determining the best move: advancement and attack (beneficial to the player), and captures (detrimental to the player). These influences are used to deduce explicit equations for the advantage gained by playing each possible move from a position, which allows the formalization of a strategic algorithm to play the RGU.

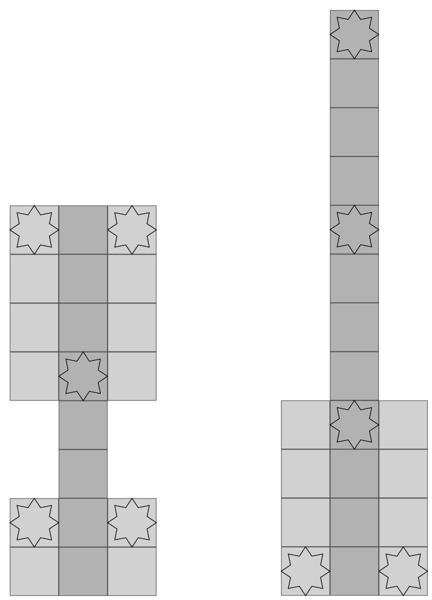
## 1 Introduction

The Royal Game of Ur (RGU) is an ancient race board game for two players. Researchers have been studying the RGU since the 1920s when Sir Leonard Woolley found five game boards in a royal tomb in Ur, Mesopotamia ("Woolley's Excavations", 2021). This discovery led to the modern naming of the game as the Royal Game of Ur. However, the game is also commonly referred to as the game of twenty squares




DOI: 10.2478/bgs-2023-0001

(de Voogt et al., 2013). The Mesopotamian variation (c. 3,000 BCE) uses the board represented in Figure 1(a), with seven pieces for each player (Becker, 2007). An Egyptian variation also exists named Aseb (c. 1,700 BCE), which has a slightly different board configuration as represented in Figure 1(b). Each player receives five pieces in Aseb instead of 7, and Aseb has fewer safe tiles than the Mesopotamian variation (de Voogt et al., 2013; Finkel, 2007). Researchers have proposed several possible rule-sets for the RGU, each formulated to match historical evidence and make the game more enjoyable (Becker, 2007; Bell, 1979; Finkel, 2007; Masters, 2021; Murray, 1952; Skiriuk, 2021). A mathematical analysis of these proposed rule-sets is desirable for both academic and educational purposes, as the strategy/chance nature of the game makes it useful as an example to study basic elements of combinatorics, probability, and computer algorithms. This analysis will also give insights into the intricacies of strategy in the RGU and how to design efficient computer programs to play it.



**(a)** Mesopotamian variation (3,000 BCE).

**(b)** Egyptian variation, Aseb (1,700 BCE).

**Figure 1:** RGU boards, with the safe and war zones indicated by light and dark grey, respectively (Finkel, 2007).

The conclusions of this paper will be built assuming only basic knowledge of combinatorics and probability. After an introduction to the mechanics of the RGU (section 2), the paper analyses the game from a combinatory perspective (section 3), a probabilistic perspective (section 4), and finally using a combination of the two (section 5). The combinatorial analysis allows the complexity of the RGU to be calculated and compared between different proposed rule-sets, and with other similar games. The


DOI: 10.2478/bgs-2023-0001

game's probabilistic analysis allows players to make predictions about future moves, allowing the most beneficial move from a position to be predicted. However, this probabilistic analysis only considers the next few moves in the game. The combination of both combinatorics and probability enables further analysis to be performed over whole games. This broad analysis is carried out by combining the long-term view gained from the combinatory analysis with the short-term view gained from the probabilistic analysis.

Whilst the complexity analysis will be performed for many rule sets, all other analyses will focus on the rules proposed by Bell (Bell, 1979), which have been further analysed by Andrea Beaker and Irving Finkel from the British Museum (BM) in the light of new evidence in 2007 (Becker, 2007; Finkel, 2007). The British Museum used these rules in the video Tom Scott vs. Irving Finkel on Youtube ("Tom Scott vs Irving Finkel: The Royal Game of Ur — PLAYTHROUGH — International Tabletop Day 2017", 2017), which increased the popularity of the RGU. However, the section analysing the complexity of the game (section 3) will also explore different board layouts and rulesets, with each adjusting the number of pieces and the path taken around the board ("Royal Game of Ur", 2021; "The Rules of Royal Game of Ur", 2021). The final section (section 6) summarizes the implications of the analysis on the optimal strategy and decision-making processes to use when playing the RGU to improve your chance of winning. Further discussion, arguments, and derivations of the results of this paper can be found in the appendices (section 8).

## 2 Basics of the game

This section describes the basic rules of the game as proposed by Bell and further refined by Andrea Beaker and Irving Finkel (Becker, 2007; Botermans, 2008; Finkel, 2007), which are:

1. Each player has 7 pieces, and the main goal of the RGU is for players to advance all their pieces along a certain path around and off the board to score them. The player who scores all their pieces first wins the game.
2. The path for one player follows the numbers in Figure 2, while their opponent's path is a mirror image of it. The initial and final tiles for pieces are given a dashed outline in Figure 2 (numbered as 0 and 15, respectively). They are not included on the physical board, but are there implicitly, and can hold any number of the player's pieces. The region between tiles 5 and 12 (dark grey in Figure 1a) can be occupied by pieces of both players, and is called the war zone. The other tiles make up the safe zones, where only one player's pieces are allowed (tiles 1, 2, 3, 4, 13 and 14).
3. The number of tiles that players can move one of their pieces in each turn is determined by the throw of four tetrahedral dice. Each dice has two marked, and two unmarked vertices. After they are rolled, the number of tiles to be moved

is determined by the number of dice with their marked vertices up. Therefore, after each throw of the four dice a piece can be moved 0, 1, 2, 3 or 4 tiles. The player can choose any of their pieces to move, as long as the move of the piece is legal.

4. Excluding the "virtual" initial and final tiles that exist off of the board, only one piece can occupy a tile at a time. It is illegal for a player to move one of their pieces onto a tile that is already occupied by another of their own pieces. However, in the war zone it is legal for the player to capture an opponent's piece by moving their piece to the same tile. After capture, the opponent's piece is moved back to the initial tile to start its advancement from the beginning. There is one exception to this rule; pieces on rosette tiles cannot be captured. Therefore, in Bell's path, pieces on tile 8 cannot be captured.

5. If a piece is moved to occupy a rosette tile (represented by stars in the tiles in Figure 1), the player is given another turn. The dice are thrown again, and unless a 0 is rolled, another one of their pieces can be moved (including the one on the rosette). This can be repeated several times, as long as there remain legal moves to play, and each piece is moved onto another rosette tile.

6. If there are legal moves available, the players cannot refuse to move on their turn. However, if there are no legal moves available, the player must pass the turn to their opponent.

7. In order to move a piece off the board, the piece must be moved exactly to the final tile. In Bell's path, the final tile is tile 15. Therefore, if the piece occupies tile 14, the player must roll a 1 to move it off the board; if it occupies tile 13, they must get a 2, and so on. If the player rolls any number higher than the exact number needed, it is illegal to move the piece off the board. Once all of a player's pieces are moved off the board to the final tile, they win the game.

The path along which the pieces must be advanced differs between rule-sets. This paper will analyse Bell's path (the same as in the British Museum proposal) (Becker, 2007; Bell, 1979; Finkel, 2007), Masters' (Masters, 2021), Skiriuk's path (Skiriuk, 2021) (for the description in English, consult Eli, 2021), and Murray's path (Murray, 1952). These paths are shown in Figure 3, alongside the path through the Aseb board (Crist et al., 2016).

## 3 Quantifying the game's complexity using combinations

The state-space and game-tree complexity metrics can be used to compare the different RGU rule-sets between one another, and to compare the RGU with the popular games Checkers, Backgammon, Ludo, Chess, and Go. These complexity metrics provide a quantitative means of comparing games that have different rules. The state-space complexity provides a measure of the number of positions that are possible in a

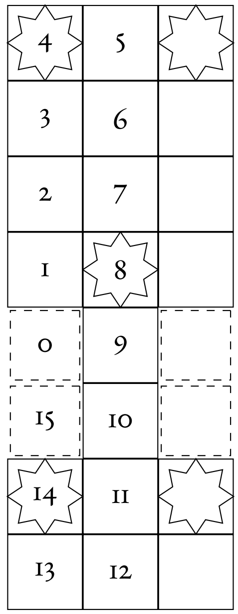
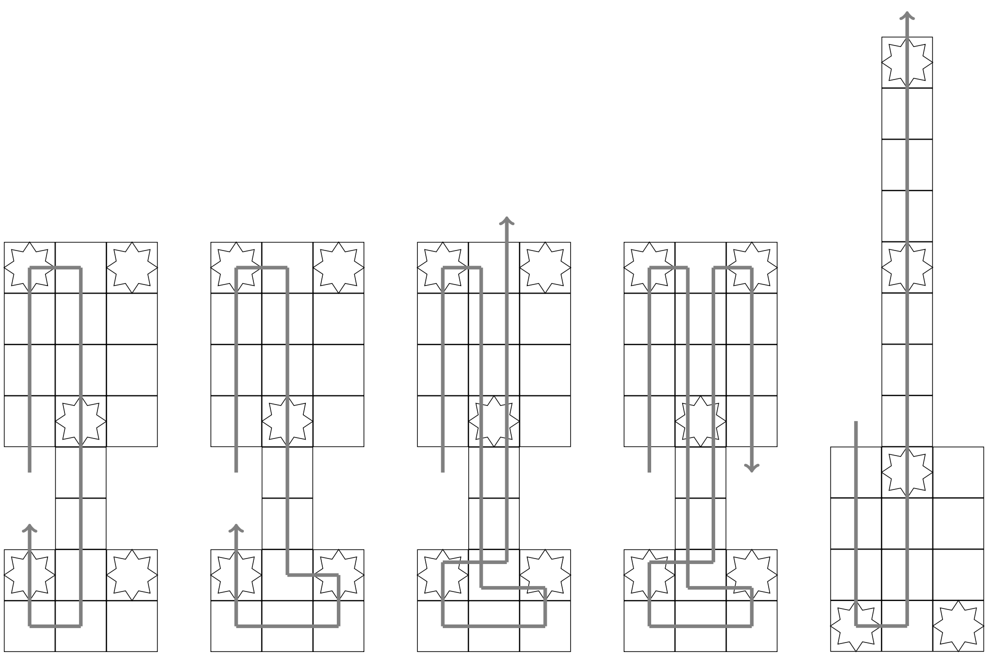



**Figure 2:** Numbering of the tiles following the Bell/BM path.



(a) Bell/BM (Bell, 1979)

(b) Masters (Masters, 2021)

(c) Skiriuk (Skiriuk, 2021)

(d) Murray (Murray, 1952)

(e) Aseb (Crist et al., 2016)

**Figure 3:** Path for the piece's of one player from different proposed rule-sets (the opponent's path is the horizontal mirror image of the paths shown).


DOI: 10.2478/bgs-2023-0001

game. We compare the state-space complexity of different rule-sets of the RGU to deduce how the piece count and the path taken across the board affect the complexity of the game. This metric also provides one perspective to compare the RGU with other games. However, the state-space complexity metric does not consider the strategic importance of game positions nor the impact of the random dice on the game. Therefore, the game-tree complexity metric is also calculated as it includes these considerations, therefore allowing more nuanced comparisons to be made between the RGU and other games.

### 3.1 State-Space Complexity

The state-space complexity (SSC) represents the total number of possible legal states that can be reached in a game. Each state counted in this measure consists of one possible arrangement of both players' pieces, on and off the board. The state-space complexity on the board of the RGU was first introduced by Geráld P. Michon (Michon, 2021). The analysis presented here will be similar, but improves the original idea to include more general cases, with comparisons and corrections.

In any position in the RGU, $p$ pieces for each player can be placed in $t'$ safe tiles (independent for each player), or in $t$ war tiles (shared by both players). The safe zone of each player includes the "virtual" initial and final tiles (tiles 0 and 15 respectively for Bell's path). The initial and final tiles may contain any number of each player's pieces off the board. Every other tile on the board can be empty or contain a single piece. The final state-space complexity value can therefore be calculated by counting the number of possible arrangements of all $2p$ pieces in the $2t'$ safe tiles and the $t$ war tiles.

If there are enough war tiles to accommodate all of both player's pieces (i.e. $t \geq 2p$), then Eq. 1 can be used to calculate all possible arrangements of pieces on the board:

$$\xi(p) = \sum_{i_g=0}^{p} \sum_{i_b=0}^{p} \left[ \sum_{i_g'=0}^{p-i_g} \alpha(i_g') \right] \left[ \sum_{i_b'=0}^{p-i_b} \alpha(i_b') \right] \beta(i_b, i_g) \tag{1}$$

If there are more pieces than war tiles (i.e. $2p > t$), then Eq. 2 must be used to calculate all the possible arrangements of pieces on the board instead:

$$\begin{aligned} \xi(p) &= \sum_{i_g=0}^{p} \sum_{i_b=0}^{t-p} \left[ \sum_{i_g'=0}^{p-i_g} \alpha(i_g') \right] \left[ \sum_{i_b'=0}^{p-i_b} \alpha(i_b') \right] \beta(i_b, i_g) \\ &+ \sum_{i_g=0}^{t-p} \sum_{i_b=t-p+1}^{p} \left[ \sum_{i_g'=0}^{p-i_g} \alpha(i_g') \right] \left[ \sum_{i_b'=0}^{p-i_b} \alpha(i_b') \right] \beta(i_b, i_g) \end{aligned} \tag{2}$$


DOI: 10.2478/bgs-2023-0001

These equations use the fundamental law of counting to count all possible states that could be reached in the RGU by splitting the board up into three sections: the grey player's safe zone, the black player's safe zone, and the war zone. The function $\alpha(i')$ counts the number of arrangements of $i'$ pieces within one player's safe zone, and $\beta(i_g, i_b)$ counts the number of possible arrangements of both player's pieces in the war zone. The equations are deduced in Appendix A, alongside a more in-depth description of their derivation.

They can be applied to all RGU rule-sets where (a) the path has one entry and one exit for each player, (b) there is at least one safe tile per player ($t' \neq 0$), and (c) there are more war tiles than pieces per player ($t \geq p$). In paths where there are no safe tiles (such as in Murray's path), or there are tiles that the player must cross two times (such as in Murray's and Skiriuk's paths), the equations are different and more complicated. The exact equations for these cases will not be presented here, but a lower-bound approximation can be made by ignoring the direction of pieces on the paths. This approximation reduces Skiriuk's to Masters' path (both have $t' = 4$ and $t = 16$), and Murray's path to a simple combination problem of $i_b$ black pieces and $i_g$ grey pieces in a full war board ($t = 0$ and $t' = 20$). The approximated equation for Murray's path is:

$$\xi(p) = \sum_{i_g=0}^{p} \sum_{i_b=0}^{p} (p - i_b + 1)(p - i_g + 1)\beta(i_b, i_g) \tag{3}$$

The equations presented above are too complicated to be calculated manually. Therefore, a computer program was created using the Fortran programming language to make the calculations, with its results shown in Figure 4 and Tab. 1. The source code of the program used to make these calculations is shown in Appendix B.

We can compare the different rule-sets for playing the RGU using their state-space complexities. This will allow us to identify the effect that the game path and the number of pieces has on the complexity of the RGU (Michon, 2021). The results of our calculations are presented in Figure 4, where the vertical axis represents the base-10 logarithm of the complexity, and the horizontal axis represents the number of pieces assigned to each player.

In Figure 4 it can be seen that Murray's and Masters/Skiriuk's paths both have the highest state-space complexity of all the rule-sets presented. This is despite the simplified calculations of the complexity of Murray's and Skiriuk's paths that underestimate their true SSC. Murray's and Skiriuk's paths have a significantly higher SSC than Aseb and Bell's paths, due to their longer length and their increased number of war zone tiles. The similarity between Murray's and Masters' results also indicate that the four safe tiles introduced in Masters' path have little impact on the state-space complexity of the game.

The curve in the results in Figure 4 is a common trend when counting the possible

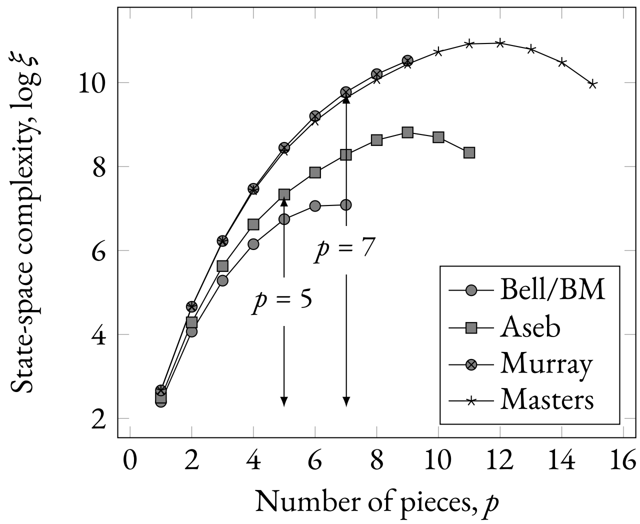



**Figure 4:** SSC for RGU rule-sets. The result for Murray's path is an approximation. The data ranges from $p = 1$ to $p = t - 1$ for each case.

arrangements (combinations) of items: the combinations rise to a maximum and then decrease. This occurs due to the symmetry of the binomial and multinomial functions that are used to calculate the combinations of pieces in the safe and war zones. However, the summations that are used to count the possible states also introduce a quadratic growth component to the complexity of the RGU as the number of pieces, $p$, is increased. These two growth characteristics act in unison to give the results above.

Additionally, among the RGU paths tested, the order of SSC is: Bell < Aseb < Masters $\lessapprox$ Murray. As a matter of comparison, the data presented by Gérald P. Michon (Michon, 2021) for the SSC of RGU in Bell's and Master's path are $\cong 1.4 \cdot 10^8$ and $\cong 5.0 \cdot 10^9$, respectively. Therefore Michon's analysis led to an order of magnitude overestimation of the SSC of Bell's path, but only a $\sim 15\%$ overestimate of the SSC of Master's path. This indicates that Michon's analysis provides practical results, despite its many approximations, with a more straightforward derivation.

However, the SSC is not an all-inclusive metric for the complexity of games. Herik et al. argued that the game-tree complexity has a higher value in measuring the complexity of games than the state-space complexity (van den Herik et al., 2002). For example, the distribution of safe tiles at the start and end of the board in the RGU significantly changes the optimal strategy, despite the SSC remaining the same. The SSC is only one factor that can indicate the complexity of a game, and including other metrics will give a better picture of a game's complexity.

### 3.2 Game-Tree Complexity

The game-tree complexity (GTC) represents all possible sequences of turns from each player until the game is complete (Eklov, 2021a). It counts all possible dice rolls and move choices of one RGU player throughout their possible games. In the RGU, most game states can be reached from several different series of dice rolls and chosen moves, and therefore GTC $\gg$ SSC.

Heyden's equation can be applied to estimate the GTC using statistics measured about the game: GTC = $(b \cdot c)^N$ (Eklov, 2021a, 2021b; Heyden, 2009). In this equation, $b$ represents the average number of choices a player must pick between each turn, called the decision branching factor. Similarly, $c$ represents the number of options selected between by random chance each turn, called the chance branching factor. In the RGU, the chance branching factor is 5, as there are five possible dice rolls: 0, 1, 2, 3, or 4. The final value, $N$, is an estimate for the average number of moves that each player is given in a typical game. The values of $b$ and $N$ used for our calculations of the GTC of the RGU were derived by simulating 2000 games between two RGU artificial intelligence agents. The agents decided their moves using the expectimax algorithm with a search depth of 7 (Lamont, 2021). An average decision branching factor of $b = 2.45$ and an average number of turns per player of $N = 106.35$ were measured using these simulations[1]. Using these results, we can make an estimate of the RGU's GTC for games between two players using the Bell/BM rules would be $(2.45 \cdot 5)^{106.35} = 5.3 \cdot 10^{115}$.

### 3.3 Comparisons to other games

The state-space complexity of the RGU is shown alongside estimates for the SSC of other popular games in Tab. 1. All rule-sets of the RGU have orders of magnitude smaller state-space complexities than Backgammon and Ludo, despite both Backgammon and Ludo being considered low-state complex games (Schaeffer et al., 2007). The games of Checkers, Chess, and Go all have much higher SSC than the RGU, despite Chess and Go being considered high-state complex games (Schaeffer et al., 2007). This demonstrates that the RGU is much less complex than other popular games in terms of the number of possible positions that can be reached.

The comparison between the GTC of the RGU with other games is also presented in Tab. 1. Interestingly, despite the RGU having a much smaller SSC than Checkers and Ludo, its GTC is much higher. This inconsistency supports the idea that any single complexity metric is insufficient to capture the complexity of board games. The GTC of the RGU is most similar to that of Chess. However, Chess does not contain elements of chance, and therefore players are given many more decisions to make in Chess than in the RGU. Due to this disparity, the GTC comparison between games with random elements (e.g., the RGU) and games without it (e.g., Chess) is contentious. However, the GTC of the RGU remains smaller than Backgammon, another game with randomness: $10^{115} \ll 10^{150}$ (Heyden, 2009; van den Herik et al., 2002)[2]. As the

[1] These numbers apply only to the BM set of rules. Other rule-sets (and paths) would lead to different averages.

[2] A more detailed calculation assumes $b = 16$ (Papahristou & Refanidis, 2012) and $c = 20$ (Tesauro, 2002) for Backgammon, so for an average number of turns of $N = 55$ (Papahristou & Refanidis, 2012),

**Table 1:** State-space complexity (SSC) and game-tree complexity (GTC) of RGU rule-sets, and other popular games.

| Game | SSC | GTC |
|---|---|---|
| RGU[a] | | |
| Bell (with $p = 7$) | $\cong 1.2 \cdot 10^7$ | $\cong 5.3 \cdot 10^{115}$ |
| Aseb (with $p = 5$) | $\cong 1.9 \cdot 10^8$ | |
| Masters (with $p = 7$) | $\cong 4.3 \cdot 10^9$ | |
| Murray (with $p = 7$) | $\cong 5.9 \cdot 10^9$ | |
| Backgammon[b] | $\cong 10^{20}$ | $\cong 10^{144}$ |
| Ludo[c] | $\cong 10^{22}$ | $\cong 10^{91}$ |
| Checkers[d] | $\cong 10^{21}$ | $\cong 10^{31}$ |
| Chess[d] | $\cong 10^{46}$ | $\cong 10^{123}$ |
| Go ($19 \times 19$)[d] | $\cong 10^{172}$ | $\cong 10^{360}$ |

[a] *This work*.
[b] Reference (Tesauro, 2002).
[c] Reference (Alvi & Ahmed, 2011). The SSC is an upper bound (the lower bound is one order of magnitude below it). The GTC is calculated assuming a mixed (aggressive/defensive/fast) strategy.
[d] Reference (van den Herik et al., 2002).

RGU and Backgammon are similar games, and the GTC of the RGU is much smaller than the GTC of Backgammon, we can infer that the RGU involves less skill to play than Backgammon. This conclusion aligns with anecdotal accounts of the RGU being a more casual and less competitive game than Backgammon. However, the calculated GTC of $10^{115}$ is still high compared to Checkers and Ludo, which are both games known to contain strategy. This comparison implies that it is likely that strategy will be important in the RGU as well. The following section will delve deeper into this strategy through the application of probability to the RGU.

## 4 Making strategy predictions using probability

### 4.1 Analysis of a single roll of the dice

In the RGU, players must start their turn by throwing their dice to see how far they can move one of their pieces. Therefore, the probabilities of each dice roll are a critical consideration for strategy in the RGU. The rolls 0, 1, 2, 3, and 4 are all possible when rolling the game's four tetrahedral dice, with their two marked and two unmarked corners. However, each of these rolls is not equally likely. Rolling a two is much more likely than rolling a zero or a four. This is due to the dice each having a 50% chance of having their marked side land upwards, which together with all four dice forms a

the GTC for Backgammon would be $(16 \cdot 20)^{55} \cong 6.1 \cdot 10^{137}$, approximately.

binomial distribution (Rozanov, 2013).

The outcome of each dice can be formalized as having the value 1 if its marked side is upwards, or 0 otherwise. A roll of four dice can then be described by a four-digit 'event', with the outcome of each dice encoded as one digit (e.g., 1010, 0111, or 0100 are all possible events). As the outcome of each individual dice has a 50% chance of being a 1, and a 50% chance of being a 0, then the total probability of each possible event is $P(\text{event}) = 0.5^4 = \frac{1}{16} = 6.25\%$. As these events are also mutually exclusive[3], the binomial distribution arises as there are more ways to roll a 2 than there are to roll a 1 or a 3, and there is only one way to roll 0 or 4. A roll of 0 will always have the event 0000. Conversely, a roll of 1 can have all the following events: 1000, 0100, 0010, 0001. Therefore, rolling a 1 is four times more likely than rolling a zero:

$$\begin{aligned} P(1000 \textbf{ or } 0100 \textbf{ or } 0010 \textbf{ or } 0001) &= P(1000) + P(0100) + P(0010) + P(0001) \\ &= 6.25\% + 6.25\% + 6.25\% + 6.25\% \\ &= 25\% \end{aligned}$$

The same approach can be used to calculate the probability, $P(\text{roll})$, of rolling a 0, 2, 3, or 4. This leads to the probabilities $P(0) = P(4) = 6.25\%$, $P(1) = P(3) = 25\%$ and $P(2) = 37.5\%$. These results are shown in Figure 5.

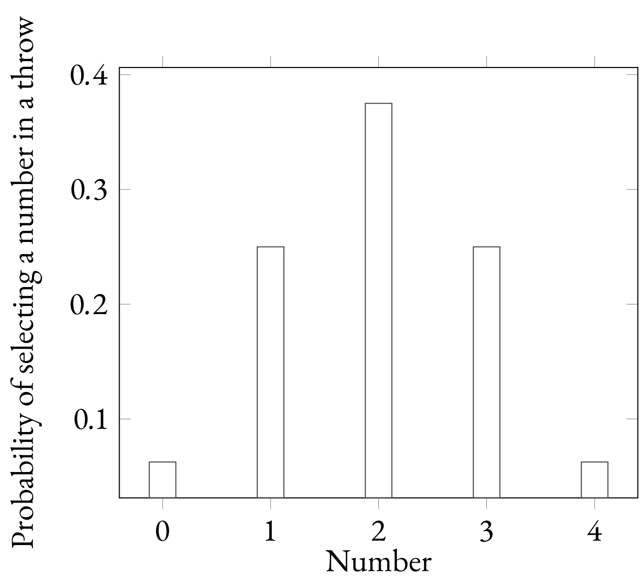



**Figure 5:** Probability mass function of one throw of four tetrahedral dice in the RGU.

[3]Only one of the events can ever occur in each throw of the dice.

We can also use this concept to calculate the chance of rolling one of a set of rolls by adding their probabilities. For example, the chance of rolling a 1 or a 3, $P(1 \textbf{ or } 3)$, is simply the sum of $P(1)$ and $P(3)$ (as they are mutually exclusive),

$$\begin{aligned} P(1 \textbf{ or } 3) &= P(1) + P(3) \\ &= 25\% + 25\% \\ &= 50\% \end{aligned}$$

Therefore, while rolling a 2 is the most common roll, it is more likely that you will roll a 1 or a 3. These probability calculations can help players to predict the most likely moves that may appear in the future. They may be used, for example, to select moves that increase the odds of performing or avoiding captures in future moves. A common situation that can arise in the RGU that requires these considerations is shown in Figure 6.

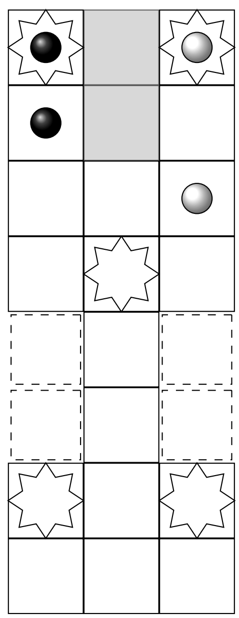


**Figure 6:** An example position in the RGU where the correct move to make is contentious. The player with the black pieces has rolled a 2 and must decide which black piece to move.

In the position in Figure 6, the player with the black pieces has rolled a 2 and must decide which piece to move. They must decide whether to move their piece in tile 3 to tile 5, or move their piece from tile 4 to tile 6. If they chose to move their piece from tile 3, their opponent has a 50% chance of rolling a 1 or a 3 to capture their piece. Conversely, if they move their piece from tile 4, their opponent has a 44% chance of


DOI: 10.2478/bgs-2023-0001

rolling a 2 or a 4 to capture their piece. Therefore, the second option has a 6% lower likelihood of their opponent capturing their piece in the next turn. Additionally, the second option frees the rosette tile for their other piece and improves the chance of getting to the center rosette. Therefore, the second move from tile 4 is likely better, despite the two moves appearing similar at first glance.

## 4.2 Analysis of a series of rolls of the dice

We can analyse more complex positions in the RGU by considering the probabilities of throwing multiple specific dice rolls in a row. Each throw of the dice is independent from previous rolls, and therefore the throws of the dice are considered statistically independent[4]. Therefore, the probability of rolling a specific sequence of numbers can be calculated as the product of all their individual probabilities,

$$P\left(\text{roll}_1 \textbf{ then } \text{roll}_2 \textbf{ then} \ldots \textbf{ then } \text{roll}_N\right) = \prod_{i=1}^{N} P\left(\text{roll}_i\right) \tag{4}$$

This method can be applied to calculate the chance of any sequence of rolls. Similar to our notation for the rolls of individual dice, we can represent one sequence of rolls as an 'event' consisting of one digit per outcome (e.g., 23 would represent a roll of 2 followed by a roll of 3). The probability of rolling a 1 followed by 4 would then be represented by $P(14)$, which would equal $P(1) \cdot P(4)$. Since multiplication is commutative, the order of the sequence does not affect the probability (e.g., $P(14) = P(41)$). Additionally, since $P(0) = P(4)$ and $P(1) = P(3)$, they can be interchanged without affecting the probability of the sequence (e.g., $P(14) = P(30)$). The probabilities of many common sequences are listed in Tab. 2.

The equation to calculate the probability of longer sequences may be generalized to consider only the number of each roll in the sequence (e.g. the sequence 14114 contains three 1's and two 4's). This simplification of the equation can be made as the order of a sequence does not affect its likelihood. The general equation for the probability of a sequence is shown in Eq. 5. The variable $N_i$ refers to the number of times the roll $i$ appears in the sequence of $N$ rolls overall.

$$P\left(\text{Sequence}\right) = P(0)^{N_0} P(1)^{N_1} P(2)^{N_2} P(3)^{N_3} P(4)^{N_4} = \left(\frac{6^{N_2} 4^{N_1+N_3}}{16^N}\right) \tag{5}$$

For example, we can use this equation to calculate the probability of rushing a piece through the entire board in a single play. This is not possible with Bell's path, but

[4] The term "statistically independent" is used with the meaning of "mutually independent", not just "pairwise independent". This requires that there is no intersection between any possible sets of events.

**Table 2:** Probabilities for sequences of dice throws in the RGU, as percentage values.

| $P$(Sequence) | Probability (%) |
|---|---|
| $P(22)$ | 14 |
| $P(12) = P(23)$ | 9.4 |
| $P(11) = P(13) = P(33)$ | 6.2 |
| $P(222)$ | 5.3 |
| $P(122) = P(223)$ | 3.5 |
| $P(112) = P(123) = P(233)$ | 2.4 |
| $P(02) = P(24)$ | 2.3 |
| $P(01) = P(03) = P(14) = P(34)$ | 1.6 |
| $P(111) = P(113) = P(133) = P(333)$ | 1.6 |
| $P(022) = P(224)$ | 0.89 |
| $P(012) = P(124) = P(023) = P(234)$ | 0.59 |
| $P(00) = P(04) = P(44)$ | 0.39 |
| $P(011) = P(013) = P(033) = P(114) = P(134) = P(334)$ | 0.39 |
| $P(002) = P(024) = P(244)$ | 0.15 |
| $P(001) = P(003) = P(014) = P(144) = P(034) = P(344)$ | 0.10 |
| $P(000) = P(004) = P(044) = P(444)$ | 0.024 |

theoretically can happen in Masters' path (sequence of four 4's and a 1), Skiriuk's path (sequence of six 4's), Murray's path (sequence of seven 4's), and in Aseb (equivalent to Masters'). The probabilities are 0.00038%, 0.0015% and 0.00000037%, respectively. Therefore, in Murray's path (the most likely path where this could occur), according to the reasoning fully developed in section 5, it would be expected that a piece could be rushed through the entire board once in every 800 games.

We can also use the probabilities of sequences of rolls to inform strategical decision making in the RGU. Consider the board presented in Figure 7.

Here the question is: what are the estimated odds of the black piece reaching tile 6 and taking out the grey piece before it can escape? This would require the player with the grey pieces to roll either a 0 or a 1. The player with the black pieces would then need to roll a 3 followed by a 2 if the grey player rolled a 0, or two 3's if the grey player rolled a 1. Therefore, the two most likely sequences of rolls that would lead to the capture of the grey piece are 032 and 133. These two sequences have the probabilities 0.59% and 1.6%, respectively (see Tab. 2). Therefore, the chance that grey's piece will be captured in the next few moves is low, despite its placement in a vulnerable position in the war zone.

The exact chance that the player with the black piece can take out the grey piece is difficult to calculate due to the exponential branching of possibilities in the game


DOI: 10.2478/bgs-2023-0001

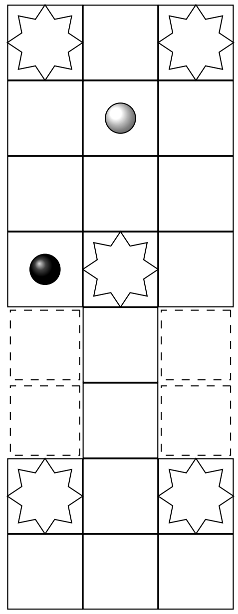


**Figure 7:** Example RGU board where the probability of a sequence of rolls can be applied to estimate odds for decision making. It is the grey player's turn, and they have no pieces left to play.

tree of the RGU. For example, an arbitrary number of 0 rolls could precede the two sequences above. However, if players limit their calculation to the most common shorter sequences, they can effectively estimate the odds of different outcomes to inform their move choices.

### 4.3 Assigning value to moves

While the probabilities of rolls underpin strategy in the RGU, they are insufficient on their own to determine the best move from a position. To determine the best move, one must consider which move gives them the most advantage. We will quantify this advantage by considering the value of moves, which measures the loss or gain to your chance of winning after playing a move. This expected value can give a much better indication of the best move to make from a position than just probabilities by considering the degree of gain from each move. For example, capturing an opponent's piece at the end of the war zone is more beneficial than capturing it at the start of the war zone.

The true value of any move is the amount that it increases your chance to win the game. However, this value is computationally intractable to calculate directly. Therefore, this paper proposes a method to calculate an approximation of the true value, called the expected value. This expected value is calculated using piece movement as a substitute for winning chance. This substitution assumes that you are more likely to win if you can move your pieces toward the end of the board faster than your opponent. The use of piece movement as a substitute for winning chance simplifies the calculation of the value of moves significantly by decoupling the estimated value from the game's outcome.

Therefore, by this expected value metric, advancing your pieces is beneficial as it brings you closer to victory. Similarly, capturing an opponent's piece is advantageous as it reverses all of that piece's advancement. Conversely, if one of your own pieces is captured, it is detrimental to your progress. This expected value metric can be calculated for a move as the following:

$$\begin{aligned} E_{\text{advance}}(\mathbf{r}) &= r \\ E_{\text{capture}}(\mathbf{r}, \mathbf{c}) &= r + c \end{aligned} \tag{6}$$

In the above equation, $r$ represents the amount that a player's own piece was advanced (i.e. the player's roll), and $c$ represents the position of the captured piece on the board (i.e. the number of tiles it was advanced before it was captured). The expected value of this move for the opposing player is the negation of the expected value for the player who made the move.

### 4.4 Refining move values using decision trees

The expected value metric described in the previous section is too simplistic to facilitate high levels of play when used on its own. However, by expanding a decision tree based upon possible future sequences of moves, accumulating the expected values of each move, and aggregating the results, more accurate estimates can be made. These 'refined' estimates are termed the gain, $G$, of a move. Similar methods of decision tree expansion and aggregation to determine value have been widely used for games with mixed chance and skill aspects in the past, such as Backgammon (Packel, 2006).

An efficient algorithm to aggregate the expected values of outcomes in decision trees with the presence of chance was proposed by Bruce W. Ballard in 1983 (Ballard, 1983). Ballard proposed calculating a weighted average, whereby the expected value of sequences are weighted by their probability, and players are expected to choose the move that maximizes their value. This method is very effective for artificial intelligence agents. However, it is very demanding for human players to perform during games. Therefore, this paper explores a different aggregation method.

Instead, this paper aggregates the decision trees by independently analysing three different strategy considerations in the RGU: advancement and attack (beneficial to the player), and capture (detrimental to the player). This approach of separating the strategy of the RGU into distinct components helps to simplify the calculations of the gain. This allows us to understand and analyse strategy more easily from positions in the RGU.

Additionally, this paper does not assume that the roll that was made is known. Ignoring the roll that the player made allows this paper to gain insights into positional strategy in the RGU. However, this assumption diminishes the effectiveness of this method while playing games. When playing games, the roll that was made should be considered. The roll that was made can be considered by ignoring the weighted sum over all possible rolls when calculating $W$, $A$, and $C$ below.

### 4.4.1 Calculation of Gain

The gain is calculated based upon a single turn of a player, and a possible counter-attack from the opponent. Multiple rolls in a single turn are also considered, as when a piece lands on rosette tiles, another roll of the dice is granted. The gain is then adjusted based upon the sequence of moves of the chosen piece after it has landed on rosette tiles. The piece to be moved is represented by the parameter $\mathbf{A}$. The gain of the sequence of moves can then be calculated by evaluating the following expression for the gain, $G$:

$$G(\mathbf{A}) = W + A + C \tag{7}$$

The three contributions to $G(\mathbf{A})$ represent the gain for walking the pieces forward on the board ($W$), the gain for capturing a piece of the opponent ($A$), and the loss due to the player's piece being captured in the next turn ($C$). While $W$ and $A$ increase the gain of a move, $C$ decreases it. These contributions are further divided into separate terms for each move in the sequence (e.g. $W_0$ for the first move, $W_1$ for the second move, $W_2$ for the third move, etc...). The full expressions for $W$, $A$, and $C$ are deduced in the Appendix C, and are presented and interpreted as:

- Walking ($W$): The walking term, $W$, measures the expected value of advancing a piece in a single turn. It is the sum of the contributions of all legal moves of the chosen piece in a single turn. The contribution from each move is the expected value of advancement for the move, from Eq. 6, multiplied by the probability of that move, $P(r)$. Without rosettes, the total walking term is given by the following sum over all rolls that lead to legal moves:

$$W_0 = \sum_{r_0} r_0 \cdot P(r_0) \tag{8}$$

However, for every rosette tile the piece can legally reach in the turn, a new term is added to include the probability of reaching that rosette:

$$\begin{aligned} W_n &= W_{n-1} \cdot P(r_n) \\ &= W_0 \cdot \prod_{i=1}^{n} P(r_i) \end{aligned} \tag{9}$$

The equation above is repeated for every new rosette that can be reached by moving the piece, resulting in the following expression for $W$ where $n$ represents the number of rosette tiles that can be reached by the piece,

$$W = W_0 + \sum_n W_n \tag{10}$$

- Attack ($A$): A player can only ever capture one of the opponent's pieces in a single turn. This arises as only a single piece can occupy a tile at a time, and player's cannot capture pieces on rosette tiles. Therefore, player's cannot gain an additional move and capture a piece in a single turn. The attack contribution depends on the number of the war zone tile reached by the piece, $n_a$, and the probability it can be reached by the piece, $P(\mathbf{AB}\ldots)$. The probability of a piece reaching a given tile is the product of the probabilities of the numbers that must be selected for that last move to be possible, $P(\mathbf{AB}\ldots)$ for a sequence $\mathbf{AB}\ldots$ (including cases where $\mathbf{A} = \mathbf{B} = \ldots$). If, within this sequence, the pieces can attack the opponents pieces in multiple war zone tiles, the resulting value of $A$ encloses all pieces that can reach all these tiles, and all sequences $s$ inside. For instance, a sequence **ABC** includes the value of attacks on the first (sequence **A**), second (**AB**), and third (**ABC**) moves. Therefore, this calculation requires a triple summation over the pieces on the board ($p$), the tiles in the war zone ($a$), and possible sequences to capture those pieces ($s$):

$$A = \sum_p \sum_a \sum_s n_a P_a = \sum_p \sum_a n_a \sum_s P_a \tag{11}$$

If there are more possible paths, more pieces that can be captured, or more pieces that are within range to attack a piece, then the value of $A$ will be higher. Also, since $n_a$ is the number of the tile containing the captured piece, captures at the end of the war zone have a higher gain than captures at the start of the war zone (more than twice as much, in some cases). The probability $P_a$ may include the chance of many rolls in the same turn. Therefore, $P_a$ can be calculated as the product of the probability of all the rolls in the turn.

- Capture ($C$): the same way the player can reach the opponent's pieces in the war zone, it can be attacked by the opponent on the next turn. The gain of the opponent is $n_c P_c$, if the player's piece is on tile $n_c$ and can be reached by the adversary with the probability $P_c$. From the player's perspective, this is a potential loss of $n_c P_c$ for each sequence that could lead to a capture of their piece in the next turn. The resulting loss for the player is:

$$C = -\sum_p \sum_c \sum_s n_c P_c = -\sum_p \sum_c n_c \sum_s P_c \tag{12}$$

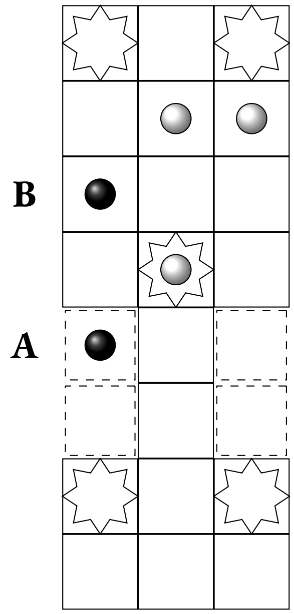



**Figure 8:** Board example for the determination of which move to make in the RGU based on expected values determinations. The black pieces are labeled by letters A and B.

The same considerations from the attack contribution ($A$) are valid for captures ($C$), which is simply an attack from the opponent's point of view. Therefore, the loss to the player upon capture is the negation of the opponent's potential gain from an attack.

### 4.4.2 Application of Gain

Let's apply the decision tree approach just described in order to make a choice in a RGU puzzle. Consider the example presented in Figure 8. The possible sequences of moves in this case are **AA**, **AB**, **BA** and **BB**. This arises as A and B are the only two pieces available to move for the black player, and both can reach the rosette on tile 4. If either piece is moved onto the rosette tile, then the black player could choose to use their next roll to move the same piece again, or move their other piece. Therefore, the rosette leads to a branching in the decision tree of possible moves in the player's turn, as shown in Figure 9. The single-piece moves here do not need to be considered explicitly, as the gain from subsets of moves are included implicitly (e.g. **AA** and **AB** together represent the gain of the sequence **A**). This arises as if **AA** and **AB** have high gain values, then **A** is implicitly more likely to be chosen as the first move.

The analysis of this decision tree to calculate the gain of each sequence of moves can be used to evaluate the best move from the position. The gain of each of the possible sequences are compared in Tab. 3. The details of the calculations of these values can be found in Appendix C.

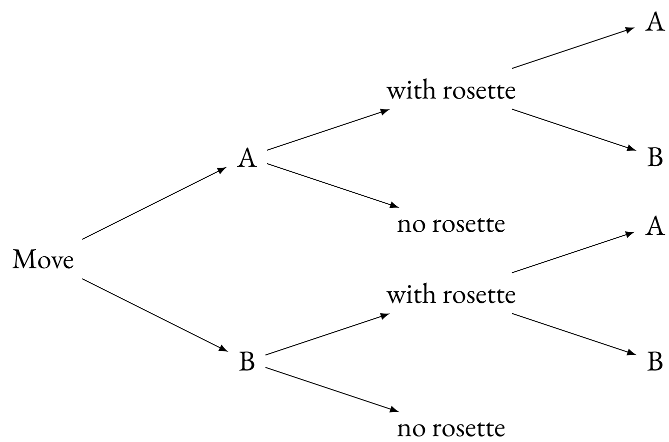



**Figure 9:** Ramification of possible paths the player of black pieces may choose, moving the pieces A or B and, if the rosette is reached, having to decide to move A or B with the new selected throw.

**Table 3:** Contribution from different sources to the decision tree evaluation of which sequence of pieces to move on the board of Figure 8.

| Sequence | Walking ($W$) | Attack ($A$) | Capture ($C$) | Total ($G$) |
|---|---|---|---|---|
| **AA** | 1.35 | 0.14 | −5.13 | −3.64 |
| **AB** | 1.24 | 0.02 | −4.25 | −2.99 |
| **BA** | 2.61 | 0.38 | −4.25 | −1.27 |
| **BB** | 2.61 | 0.84 | −9.38 | −5.92 |

The inspection of this table allows us to quickly discover that the gain, $G$, of all possible sequences are negative. This indicates that all possible move sequences will likely end in loss to the player instead of gain. However, the least negative value is given by the **BA** sequence, indicating that it is considered the best sequence to move according to our model. The small value of $A$ in all sequences arises as there is only one piece of the opponents that can be reached to attack, and the sequences of moves required to attack it are unlikely (either the sequence 4 or 22 are required). Conversely, the $W$ contribution is significant, as in sequences where **B** is moved first the walking contribution is over 1 tile greater than in sequences where **A** is moved first. However, the most significant contribution from this position is $C$, as the chance of capture of any pieces that are moved into the war zone is high. Therefore, the reduced chance of capture and increased chance of piece movement from the sequence **BA** make it the best sequence to play according to our model.

## 5 Combining combinations and probability

The original combinatorics analysis of the RGU gave us insight into the game's complexity, and the probability analysis gave us insight into decision-making strategies for specific moves. However, we can also gain more broad insight into strategy in the RGU by combining both combinatorics and probability.

### 5.1 Expected number of dice rolls in each game

In any match of the RGU, there are a limited number of turns for each player. In the combinatorics section we used the estimate of $N = 106$ turns per player, which we will use again in this section. Using this estimation, we can calculate the expected number of times that players will roll each value of the dice using the equation $\overline{N_r} = N \cdot P(r)$, where $r$ is the value of the dice (i.e. 0, 1, 2, 3, or 4)[5]. Consequently, in a typical match, players should expect seven 0's, twenty-seven 1's, forty 2's, twenty-seven 3's, and seven 4's (the same distribution as the probabilities shown in Figure 5). However, there are several limiting factors that should be considered when using these predictions to make any inferences about strategy:

- Players cannot use these numbers to predict future rolls of the dice. If a player has rolled five 1's in a row, it is not less likely that they will roll a 1 in the next throw of the dice. This is the gambler's fallacy (Rabin, 2002; Rabin & Vayanos, 2010);
- The number of rolls of each number that players will receive is not guaranteed. Although unlikely, you could roll twenty 0's in a game of the RGU (the chance of rolling twenty 0's in a game is 0.00062%, as will be derived soon);

[5]This result arises from basic probability theory.

- The value of $N$ is an approximation. Games of the RGU can be quick or long, depending on chance and the strategy of each player.

A possible approach to gain more insight towards the numbers of each roll for several matches is to observe the confidence interval around $N_r$. In order to do so, it is useful to calculate the order-independent probability of getting $N_0$ zeros, $N_1$ ones, $N_2$ twos, and so on. This probability follows a multinomial distribution, and can therefore be calculated as follows:

$$\begin{aligned} P(\text{Frequency}) &= P(\text{Sequence})\binom{N}{N_0, N_1, N_2, N_3, N_4} \\ &= \frac{P(0)^{N_0}P(1)^{N_1}P(2)^{N_2}P(3)^{N_3}P(4)^{N_4}N!}{N_0!N_1!N_2!N_3!N_4!} \end{aligned} \tag{13}$$

This equation is equivalent to the product of the probability of any specific sequence with those numbers of rolls, and the number of possible sequences. For example, the probability of rolling one 1 and one 2 is 9.375%, and there are two sequences that contain one 1 and one 2 (12 and 21). Therefore, $P(\textit{one } 1, \textit{one } 2)$ is $2 \cdot 9.375\% = 18.75\%$.

However, confidence intervals in multidimensional functions are harder to evaluate. Luckily, we can make a simplification to treat this problem as a binomial problem. Instead, let's redefine the 'event' whose probability we are interested in. We will consider an 'event' only in relation to the occurrence of a single number $r$ in a sequence of rolls. This changes the problem into a binomial problem, as we are only interested in whether a roll was $r$, or was not $r$. In this case, the probability of rolling $r$ exactly $N_r$ times in a sequence of $N$ rolls follows the following binomial distribution:

$$P(N_r) = P(r)^{N_r}\,[1 - P(r)]^{N-N_r}\binom{N}{N_r} \tag{14}$$

For instance, the chances of getting $N_0 = 20$ zeros in $N = 106$ turns is:

$$P(N_0 = 20) = P(0)^{20}\,[1 - P(0)]^{106-20}\binom{106}{20} = 6.25 \cdot 10^{-6} = 0.000625\%$$

When $N$ and $N_r/N$ are large, the confidence intervals of binomial functions approximately follow a normal distribution (Lawrence D. Brown & DasGrupta, 2001). Using this approximation, we can estimate how probable it is that $N_r$ lies between $\overline{N_r} - \Delta N_r$ and $\overline{N_r} + \Delta N_r$, with a certain confidence level. The value of $\Delta N_r$ is given by:

$$\Delta N_r = z \cdot N \sqrt{\frac{P(r)\,[1 - P(r)]}{N}} \tag{15}$$

In this equation, $z$ depends on the confidence level, and represents the number of standard deviations away from the mean of the normal distribution. For example, for a 90% confidence level, $z = 1.645$. This represents that for normal distributions, 90% of samples fall within 1.645 standard deviations of the mean. This gives the results that in 90% of games, the number of 0's and 4's will fall between 3 and 11 (each), the number of 1's and 3's will fall between 19 and 34 (each), and the number of 2's will fall between 32 and 48. This demonstrates that there is a large variation in the expected number of rolls observed in games of the RGU, which reinforces that the number of rolls within games of the RGU is unpredictable.

## 5.2 Expected number of turns to move pieces

In games of the RGU, the expected number of dice rolls to advance a piece by $S$ tiles is $S/2$. This arises due to the expected mean dice roll of 2, leading to an average of 2 tiles moved per dice roll. Therefore, the most probable sequences of dice rolls to move a piece by $S$ tiles will be $S/2$ moves in length. Additionally, when players encounter a rosette tile, they are granted an extra move, leading the average number of turns to move a piece $S$ tiles to be less than $S/2$. However, when moves of a piece are blocked by other pieces, the number of turns required will increase.

For example, taking control of the centre rosette is commonly considered strategically advantageous by players. To move a piece from the start of the board to the centre rosette, players must advance a piece by $S = 8$ tiles. Therefore, it should most often take $S/2 = 4$ moves to get to the centre rosette. Additionally, due to the rosette on tile 4, players will likely be granted an extra move and take only 3 turns to get to the centre rosette. This result shows that the player who goes first will have a significant edge in getting to the central rosette first. However, this advantage is not without risk, as 3 of the tiles that must be advanced past to get to the central rosette are in the war zone. Therefore, often players will opt for the safer option of moving more of their pieces onto the board instead of risking their opponent capturing their piece early, without retribution.

Another example is to consider all of the possible sequences to advance a piece by $S = 4$ tiles. We would expect that most often this would take players $S/2 = 2$ moves. To verify this, we calculate the probabilities of all length-1, length-2, and length-3 sequences of moves to advance a piece by 4 tiles in Tab. 4. We must also consider all the sequence combinations, which is accomplished by the multiplications in the probability columns (e.g. the sequences 004, 040, and 400 all have the same probability, so we only list 004 once). This is a direct application of Eq. 13.

In these results, the length-2 sequences are the most likely to allow the movement of

**Table 4:** Probabilities for sequences of dice rolls with sum four from $M = 1$ to $M = 3$ dice rolls. The multiplication within the probability columns is a factor to consider all combinations of each sequence.

| Length-1 | | Length-2 | | Length-3 | |
|---|---|---|---|---|---|
| Sequence | Probability (%) | Sequence | Probability (%) | Sequence | Probability (%) |
| (4) | 6.25 | (04) | $0.39 \cdot 2 = 0.78$ | (004) | $0.024 \cdot 3 = 0.073$ |
| – | – | (13) | $6.25 \cdot 2 = 12.50$ | (013) | $0.39 \cdot 6 = 2.34$ |
| – | – | (22) | 14.06 | (022) | $0.88 \cdot 3 = 2.64$ |
| – | – | – | – | (112) | $2.34 \cdot 3 = 7.03$ |

a piece by 4 tiles. However, the overall probability of moving your piece exactly 4 tiles in 1, 2, or 3 moves is still only 45.7%. This low probability is due to these calculations ignoring the possibility of selecting the pieces you wish to move. Sequences such as 232 still allow you to move one of your pieces by exactly 4 tiles, as long as you can use the dice roll of 3 on another one of your pieces. Therefore, despite moving a piece by 4 tiles taking approximately 2 moves, those moves may not be in order. Thus, in real games, it may take more turns to move your pieces to precise locations, as you will likely have to distribute your dice rolls between your pieces to do so.

## 6 Conclusion

The mathematical proofs and arguments presented in this paper can be used for teaching, as a basis for further research, and to improve the tactics of players.

In section 3.1 we found that the state-space complexity of the Royal Game of Ur (RGU) is small when compared with similar games such as Backgammon or Ludo, and much smaller than Chess or Checkers. Conversely, the game-tree complexity of the RGU calculated in section 3.2 was much higher due to the inclusion of chance in the calculation. The game-tree complexity of the RGU was estimated to be much larger than the game-tree complexity of Checkers and Ludo. This suggests that even though the RGU has a small number of possible game states compared to other games, its high game-tree complexity suggests that strategy is present. This matches the anecdotal accounts of the presence of strategy in the RGU.

In section 3.1 the state-space complexity (SSC) of the RGU was also compared between the rule-sets proposed for the game. It was found that the SSC was much larger for the rule-sets with the longest paths. Therefore, both Murray's and Masters' paths lead to the most possible outcomes. Conversely, one of the most popular paths played today, Bell's path, is the least complex in terms of state-space complexity of all the rule-sets analysed here. Perhaps then, the simplicity and approachability of the rule-set popularised by Irving Finkel has played into its widespread adoption.

The influence of the number of starting pieces on the SSC of the RGU was also calculated for many different RGU rule-sets. It was observed that the SSC was highest when the number of starting pieces was slightly smaller than the number of war-zone tiles in the rule-set's path. For Bell's path, it was found that 7 starting pieces led to the highest SSC, which is the same number of starting pieces as used by the rule-set popularised by Irving Finkel.

The probabilities of individual dice rolls and sequences of dice rolls were derived in section 4. It was shown that rolling a 2 was most likely with a 37.5% chance, followed by 1 and 3 with a 25% chance each, and then 0 and 4 with only a 6.25% chance each. These chances were used to analyse an example position that showed that while rolling a 2 is the most likely, rolling 1 or 3 is more likely. Therefore, while a piece that is 2 tiles ahead of an opponent's piece is very vulnerable, a piece that is 1 tile ahead of one opponent's piece, and 3 tiles ahead of another, is more vulnerable. This basic probabilistic analysis is too simplistic to describe more complex positions however, especially in the presence of rosette tiles.

When rosette tiles are available, sequences of dice rolls must be considered to calculate the chance of capturing a piece in a single turn. In section 4.2, the probabilities of sequences of moves were derived. It was observed that all sequences of two dice rolls that included at least one roll of 2 were more likely to occur than rolling a single 0 or 4. For example, if a piece is 4 tiles behind an opponent's piece and 2 tiles behind an empty rosette tile, then they could roll either one 4 or two 2's to capture that piece. The chance of rolling one 4 is 6.25%, while the chance of rolling two 2's is 14%, higher than the chance of rolling one 4. Therefore, considering the probability of sequences of moves is key to analysing more complex positions in the RGU. However, these probabilistic analyses overlook one key aspect of selecting the best move in the RGU: some moves are more beneficial than others.

In section 4.3 we argued that an effective metric for estimating move value was piece movement. If one move advanced your pieces further than another, then it is more beneficial. Similarly, if you capture an opponent's piece it loses its advancement, which is detrimental to your opponent and thus beneficial to you. This metric improves the analysis of positions even further, although it lacks consideration of counter-attacks from your opponent.

In section 4.4 we introduced the concept of expanding decision trees of the possible future moves and aggregating the results to calculate more accurate estimates of move value. We proposed a simplified method for this process in 4.4.2 that analysed movements based upon three considerations: walking and attack (beneficial to the player), and capture (detrimental to the player). The sum of these three considerations gives our proposed gain for a move. Moves with higher gain are considered to be more advantageous to play. The walking consideration is calculated based upon the number of tiles that pieces may be advanced when a piece is moved, and it is increased through the presence of rosette tiles. The attack consideration is calculated based upon the value of capturing an opponent's piece, and it is increased when there are more pieces that may be captured, or when they are further along in the war-zone. The capture consideration is the inverse of attack, and it represents the opportunity of any possible counter-attack from the opponent after you have moved your piece. These considerations are easier to calculate on-the-fly than traditional approaches of refining the values of moves using decision trees, and therefore we believe this is a good approach for human players to emulate to improve their game.

In section 5 we combined the use of probability theory and combinatory theory to calculate the frequencies of dice rolls expected within typical games of the RGU, and to estimate the number of turns required to move pieces to specific tiles. We found that the expected frequency of each value of dice rolls in any given match can fluctuate widely by calculating the 90% confidence interval for the frequency of each dice roll value in a typical game. This suggests that in any given game, luck could play a significant factor in determining if one player gets more beneficial rolls than their adversary. We also showed that it can be expected that $N/2$ turns will be required to move a single piece $N$ tiles, due to the mean expected dice roll of 2. This implies that sequences to move a piece $N$ tiles that are not $N/2$ moves long will be less probable to occur. Analyses such as these have a great potential to provide new insights into the RGU, and it is the current subject of further research.

## 7 Future Work

While this study addresses several key elements of the RGU, many remain unexplored, including game balance, game length, and the impact of luck on the outcome of games. Future studies could consider establishing formal methods to measure these attributes, facilitating comparisons to be made between rule sets, and potentially between games. However, these features may be complex to calculate directly. Therefore, a promising future avenue of research could be the simulation of games with AI agents. This could allow statistical measurements to be made of these features of the RGU. Additionally, further computer analysis could allow the calculation of the game-tree complexity metric for each RGU rule set, extending beyond the current focus on Bell's path.

## Acknowledgments

The authors thank James Masters, Thiago A. Nascimento, George Pollard, and other colleagues from the Royal Game of Ur community on Discord for their support in the preparation of this manuscript. You have all helped greatly through corrections, suggestions, relevant and interesting discussions, and encouragement.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## References


Alvi, F., & Ahmed, M. (2011). Complexity analysis and playing strategies for ludo and its variant race games. *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, 134–141. https://doi.org/10.1109/CIG.2011.6031999

Ballard, B. W. (1983). The *-minimax search procedure for trees containing chance nodes. *Artificial Intelligence*, *21*(3). https://doi.org/https://doi.org/10.1016/S0004-3702(83)80015-0

Becker, A. (2007). The royal game of ur. In I. L. Finkel (Ed.), *Ancient board games in perspective: Papers from the 1990 british museum colloquium* (pp. 11–15). British Museum Press.

Bell, R. C. (1979). *Board and Table Games From Many Civilizations* (Revised, Vol. 1 and 2). Dover Publications, Inc.

Botermans, J. (2008). *The book of games: Strategy, tatics & history*. Sterling.

Crist, W., Dunn-Vaturi, A.-E., & de Voogt, A. (2016). *Ancient Egyptians at Play: Board Games Across Borders*. Bloomsbury Academic.

de Voogt, A., Dunn-Vaturi, A.-E., & Eerkens, J. W. (2013). Cultural transmission in the ancient Near East: Twenty squares and fifty-eight holes. *Journal of Archaeological Science*, *40*, 1715–1730.

Eklov, E. (2021a). Game Complexity I: State-Space & Game-Tree Complexities. https://www.pipmodern.com/post/complexity-state-space-game-tree

Eklov, E. (2021b). Game Complexity II: Modern Board Game Analysis. https://www.pipmodern.com/post/complexity-modern-board-game-analysis

Eli. (2021). Royal game of ur - game of 20 squares. https://www.ancientgames.org/royal-game-ur-game-20-squares/

Finkel, I. L. (2007). On the rules for the royal game of ur. In I. L. Finkel (Ed.), *Ancient board games in perspective: Papers from the 1990 british museum colloquium* (pp. 16–32). British Museum Press.

Heyden, C. (2009). *Implementing a Computer Player for Carcassonne* [Master thesis]. Maastricht University. https://project.dke.maastrichtuniversity.nl/games/files/msc/MasterThesisCarcassonne.pdf

Lamont, P. (2021). The Royal Game of Ur Analysis. https://github.com/Sothatsit/RoyalUrAnalysis/blob/main/docs/Agents.md

Lawrence D. Brown, T. T. C., & DasGrupta, A. (2001). Interval Estimation for a Binomial Proportion. *Statistical Science*, *16*(2), 101–133.

Masters, J. (2021). The Royal Game of Ur & The Game of 20 Squares. https://www.tradgames.org.uk/games/Royal-Game-Ur.htm

Michon, G. P. (2021). Royal game of ur (game of twenty squares). http://www.numericana.com/answer/ur.htm

Murray, H. J. R. (1952). *A History of Board-games Other Than Chess*. Oxford University Press.

Packel, E. W. (2006). *The mathematics of games and gambling*. The Mathematical Association of America.

Papahristou, N., & Refanidis, I. (2012). Improving temporal difference learning performance in backgammon variants. In H. J. van den Herik & A. Plaat (Eds.), *Advances in computer games (2011)* (pp. 134–145). Springer Science + Business Media. https://www.academia.edu/2569666/Improving_Temporal_Difference_Learning_Performance_in_Backgammon_Variants

Rabin, M. (2002). Inference by Believers in the Law of Small Numbers. *The Quarterly Journal of Economics*, *117*(3), 775–816.

Rabin, M., & Vayanos, D. (2010). The Gambler's and Hot-Hand Fallacies: Theory and Applications. *The Review of Economic Studies*, *77*, 730–778.

Royal game of ur. (2021). https://gameofuronline.com/

Rozanov, Y. A. (2013). *Probability theory: A concise course*. Dover Publications.
The rules of royal game of ur. (2021). https://www.mastersofgames.com/rules/royal-ur-rules.htm
Schaeffer, J., Burch, N., Björnsson, Y., & Kishimoto, A. (2007). Checkers Is Solved. *Science*, *317*(5844), 1518–1522.
Skiriuk, D. (2021). The rules of royal game of ur. https://skyruk.livejournal.com/231444.html
Tesauro, G. (2002). Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, *134*, 181–199.
Tom Scott vs Irving Finkel: The Royal Game of Ur — PLAYTHROUGH — International Tabletop Day 2017. (2017). https://www.youtube.com/watch?v=WZskjLqo4oI&t=6s
van den Herik, H. J., Uiterwijk, J. W., & van Rijswijck, J. (2002). Games solved: Now and in the future. *Artificial Intelligence*, *134*, 277–311.
Woolley's excavations. (2021). http://www.ur-online.org/about/woolleys-excavations/

## 8 Appendices

## A Deducing the state-space complexity equations for the RGU

The number of possible arrangements of each player's pieces within the war zone can be calculated using the multinomial equation. This can be done by considering the tiles in the war zone as having three possible states: (1) they can be empty, (2) they can contain a grey piece, or (3) they can contain a black piece. The number of arrangements (i.e. combinations) of each player's pieces can therefore be calculated using the multinomial equation by counting the possible arrangements of states of each tile.

The multinomial equation calculates the combinations of a sequence of $N_1$ elements of type 1, $N_2$ elements of type 2, $N_3$ elements of type 3, and so on, in a sequence of $N = N_1 + N_2 + N_3 + \ldots$ numbers:

$$Z = \frac{N!}{\prod_{i=1}^{k} N_i!} = \binom{N}{N_1, N_2, \ldots, N_k} \tag{16}$$

For example, the number of sequences of the 3 letters $a$, $b$, and $c$, is 6 ($N = 3$, $N_1 = N_2 = N_3 = 1$):

| | | |
|---|---|---|
| *abc* | *bac* | *cab* |
| *acb* | *bca* | *cba* |

If $a$ is repeated instead of adding the third letter $c$, then the combinations reduces to $Z = 3$ ($N = 3$, $N_1 = 2$, and $N_2 = 1$):

| | | |
|---|---|---|
| *aab* | *aba* | *baa* |

If there are only two types, as in the two letter case, then the multinomial equation simplifies to the binomial equation. The binomial equation is often expressed as the combinations of $N = N_1 + N_2$ elements, with $i = N_1$ elements of type 1:

$$Z(i) = \frac{N!}{i!(N-i)!} = \binom{N}{i} \tag{17}$$

In the RGU context, we can use Eq. 16 for the calculation of the state-space complexity (SSC) of the game. The SSC is just the number of possible legal arrangements of the pieces on the board, for all possible games. It can be directly obtained for some simple games, but for most games only the order of magnitude is available through mathematical derivation or brute force (using computer algorithms). Our exact calculations will depend upon the multinomial equation (Eq. 16), its binomial special case (Eq. 17), and on the fundamental principle of counting. This principle states that if we want to determine the number of possible outcomes of sets of independent events (each with $\xi_i$ number of events for each case $i$), one must multiply them all:

$$\xi = \prod_{i=1}^{k} \xi_i \tag{18}$$

This allows us to separate the problem of counting the number of combinations of black and grey pieces into two cases: (1) counting pieces in safe tiles that can be occupied by only one player's pieces, and (2) counting pieces in war tiles that can be occupied by both grey and black pieces.

For the first case, one must consider the tiles where only one kind of piece can occupy, $t'$, and the departure and arrival tiles, where any number of pieces can stay. In the $t'$ houses ($t_\mathrm{b}$ and $t_\mathrm{g}$ for black and grey pieces, respectively), only two states are allowed: the house is occupied or not. Therefore, the number of combinations of $i'$ pieces in the $t'$ tiles is just:

$$\binom{t'}{i'} = \frac{t'!}{i'!(t'-i')!} = \binom{t'}{t'-i'} \tag{19}$$

Related to that number is the amount of pieces that are left in the departure and arrival tiles ($d, a$). If one is on the board, for instance, the $t'-1$ pieces can be separated between $t'-1$ on one side and $0$ on the other: $(t'-1, 0)$. But other arrangements are equally possible (as long as the sum of pieces is $t'-1$): $(t'-2, 1)$, $(t'-3, 2)$, $\ldots$, $(0, t'-1)$. There are, therefore, $t'-1+1$ of such cases, if 1 piece is on the board; if $i'$ pieces are on the board, the number of possible arrangements outside the board is $t'-i'+1$. The product of this number with the combinations inside the board leads to:

$$\alpha(i' \leq t') = (t'-i'+1)\binom{t'}{i'} \tag{20}$$

Eq. 20 is valid, as indicated, when $i' \leq t'$. When it is not the case, however, all tiles are occupied (number of combinations is 1) and there are only $i'-t'+1$ possibilities outside the board.

$$\alpha(i') = \begin{cases} (t' - i' + 1)\binom{t}{i'} & \text{for } i' \leq t' \\ i' - t' + 1 & \text{for } i' > t' \end{cases} \tag{21}$$

A way to put it in a single equation is:

$$\alpha(i') = (|\delta| + 1)\binom{t'}{(\delta - |\delta|)/2 + t'} \tag{22}$$

where:

$$\delta = i' - t' \tag{23}$$

Therefore, when $i' < t'$, $i' - t' < 0$, $|\delta| = -(i' - t') = t' - i'$ and $(\delta - |\delta|)/2 + t' = 2(i' - t')/2 + t' = i'$. Otherwise, if $i' \geq t'$, then $i' - t' \geq 0$, $|\delta| = i' - t'$ and $(\delta - |\delta|)/2 + t' = 2(i' - t' - i' + t')/2 + t' = t'$, and the combinatorial becomes 1.

Given that the number of pieces can range from $i' = 0$ to $i' = p'$, the number of combinations given $p'$ is:

$$\gamma(p') = \sum_{i'=0}^{p'} \alpha(i') \tag{24}$$

Eq. 24 is not absolute, however. It implicitly assumes there are both pieces in the board and outside it, unless there are too much pieces to be on the board, and only exchanges between the pieces outside (between departure and arrival tiles) counts. However, when there are no pieces on the board ($i' = 0$), the number of combinations is just $p' + 1$. In order to consider it, one can count this case outside the summation:

$$\gamma(p') = (p' + 1) + \sum_{i'=1}^{p'} \alpha(i') \tag{25}$$

Another thing is, if $i' < t'$ and all the pieces are on the board ($i' = p'$), no other outside (in cases where $p' < t'$), the equation has to be changed, and the number of combinations is just $t'!/(p'!(t' - p')!)$. Therefore, the equations so far apply only to $p' > t'$, which is usually the case in RGU rules. Also, the equation is based on the transformation $\sum_{i'=0}^{b} f(i') = f(0) + \sum_{i'=1}^{b} f(i')$ for a function $f(i')$, which implies that if $b = 0$, there is no summation at all, and $f(i') = f(0)$. In our case:

$$\gamma(p' = 0) = (p' + 1) = 1 \tag{26}$$

These considerations and equations holds for a player only, and the possible outcomes when both players are considered is the product of the number of events of each (Eq. 18), and for a specific $p_g{}'$ of grey pieces and a $p_b{}'$ of black pieces:

$$\gamma(p_g{}')\gamma(p_b{}') = \left[\sum_{i_g{}'=0}^{p_g{}'} \alpha(i_g{}')\right] \left[\sum_{i_b{}'=0}^{p_b{}'} \alpha(i_b{}')\right] \tag{27}$$

What about the pieces on the shared places? Well, in this case there are three types of state for each tile: unoccupied, occupied by grey pieces and occupied by black pieces. According the multinomial equation (Eq. 16), the number of possibilities for $i_g$ grey pieces and $i_b$ black pieces (the remaining will be in the exclusive tiles or in departure/arrival tiles) in $t$ tiles is:

$$\beta(i_g, i_b) = \frac{t!}{i_b! i_g! (t - i_b - i_g)!} = \binom{t}{i_b, i_g} \tag{28}$$

Eq. 28 remains valid if $i_b + i_g \leq t$, otherwise the limits on the variations of $i_g$ and $i_b$ must be reconsidered to account the fact. This will be addressed soon.

Now, considering the three independent events (black and grey in secure places, and black and grey at war), we must divide the pieces which are on the shared places ($i_b$ black pieces plus $i_g$ grey pieces) and the ones that must go to the secure tiles or outside the board. The number of these is determined by setting $p' = p - i$, where $p$ is the total number of pieces (independent of the place they are). For black or grey pieces, then:

$$\gamma(p') = \gamma(p - i) = \sum_{i'=0}^{p-i} \alpha(i') \tag{29}$$

Therefore, when we consider, for a fixed number of $p_b$ black pieces and $p_g$ grey pieces on the board ($i_b$ black pieces and $i_g$ grey pieces on the war zone) we have $\gamma(p_g - i_g)\gamma(p_b - i_b)\beta(i_g, i_b)$ combinations. Since the number of pieces on the war zone can range from $i = 0$ to $i = p$, the total number when we consider them all is:

$$\xi(p_g, p_b) = \sum_{i_g=0}^{p_g} \sum_{i_b=0}^{p_b} \gamma(p_g - i_g)\gamma(p_b - i_b)\beta(i_g, i_b) \tag{30}$$

$$\xi(p_g, p_b) = \sum_{i_g=0}^{p_g} \sum_{i_b=0}^{p_b} \left[\sum_{i_g{}'=0}^{p_g - i_g} \alpha(i_g{}')\right] \left[\sum_{i_b{}'=0}^{p_b - i_b} \alpha(i_b{}')\right] \beta(i_b, i_g) \tag{31}$$

where $i_\mathrm{b}$ and $i_\mathrm{g}$ pieces are on the shared tiles, and $i_\mathrm{b}{}' = p_\mathrm{b} - i_\mathrm{b}$ and $i_\mathrm{g}{}' = p_\mathrm{g} - i_\mathrm{g}$ are on the rest of the board (or outside).

Notice that this equation is quite general: it is valid for a board with two kinds of pieces, two kinds of places (the individual and shared ones), any number of pieces for each and any number places of each case. In the RGU historical possibilities, and the equality among players (both must have the same opportunities and limitations!), it should be assumed that $t_\mathrm{g} = t_\mathrm{b} = t$ and $p_\mathrm{g} = p_\mathrm{b} = p$. That implies:

$$\xi(p) = \sum_{i_\mathrm{g}=0}^{p} \sum_{i_\mathrm{b}=0}^{p} \left[ \sum_{i_\mathrm{g}{}'=0}^{p-i_\mathrm{g}} \alpha(i_\mathrm{g}{}') \right] \left[ \sum_{i_\mathrm{b}{}'=0}^{p-i_\mathrm{b}} \alpha(i_\mathrm{b}{}') \right] \beta(i_\mathrm{b}, i_\mathrm{g}) \tag{32}$$

The equation is valid for $2p < t$. If is not the case, the pieces can fill the war zone completely, and the multinomial term can no longer be used (since it will lead to negative factorials). A way to bypass that limitation is, while assuming $p < t$, to choose one kind of piece to fill as much as it can the war zone (say, summing up $i_\mathrm{g}$ from 0 to $p$), and gradually filling the gaps with the other kind of pieces, until $t - p$ (in the example, $i_\mathrm{b}$ would range from 0 to $t - p$). Then, another summation must be made, with the opposite case, the kind of piece that previously filled the war zone ranging from 0 to $t - p$ and the other kind (previously limited), ranging from $t - p + 1$ to $p$. Therefore, for cases where $t/2 < p < t$ (cases were $p < t/2$ are covered by Eq. 32):

$$\begin{aligned} \xi(p) \quad = \quad & \sum_{i_\mathrm{g}=0}^{p} \sum_{i_\mathrm{b}=0}^{t-p} \left[ \sum_{i_\mathrm{g}{}'=0}^{p-i_\mathrm{g}} \alpha(i_\mathrm{g}{}') \right] \left[ \sum_{i_\mathrm{b}{}'=0}^{p-i_\mathrm{b}} \alpha(i_\mathrm{b}{}') \right] \beta(i_\mathrm{b}, i_\mathrm{g}) \\ + \quad & \sum_{i_\mathrm{g}=0}^{t-p} \sum_{i_\mathrm{b}=t-p+1}^{p} \left[ \sum_{i_\mathrm{g}{}'=0}^{p-i_\mathrm{g}} \alpha(i_\mathrm{g}{}') \right] \left[ \sum_{i_\mathrm{b}{}'=0}^{p-i_\mathrm{b}} \alpha(i_\mathrm{b}{}') \right] \beta(i_\mathrm{b}, i_\mathrm{g}) \end{aligned} \tag{33}$$

We emphasize that Eq. 33 do not hold when $2p < t$, since it was deduced assuming we can fill the war zone with all pieces of one type and complete the other tiles with the pieces of the adversary. Therefore, in these cases the Eq. 32 should be used instead. In contraposition, the use of Eq. 32 to cases where $2p > t$ leads to the problems discussed before. Each equation has its domain of validity. And both assumes that $p < t$ (which are the cases that concerns us).

Among the different paths suggested for the RGU, the simplest one concerns a board with no separation between safe and war zones among pieces ($t' = 0$), which is Murray's path. If we neglect the pieces direction when crossing the same tiles twice, the number of combinations can be approximated to an exact, though underestimated, value. The number of possible scenarios in this case, for $p$ pieces for each player, is:

$$\xi(p) = \sum_{i_g=0}^{p} \sum_{i_b=0}^{p} (p - i_b + 1)(p - i_g + 1)\beta(i_b, i_g) \tag{34}$$

We finish this appendix with the detailing of a small model, a mini-RGU example, which assumes $p = t' = 1$ and $t = 2$. Since $2p \leq t$, we should use Eq. 32:

$$\begin{aligned}
\xi(1) \quad = \quad & \sum_{i_g=0}^{1} \sum_{i_b=0}^{1} \left\{ \left[ 1 - i_g + 1 + \sum_{i_g{}'=1}^{1-i_g} (1 - i_g{}' + 1) \binom{1}{i_g{}'} \right] \cdot \right. \\
& \left. \left[ 1 - i_b + 1 + \sum_{i_b{}'=1}^{1-i_b} (1 - i_b{}' + 1) \binom{1}{i_b{}'} \right] \right\} \binom{2}{i_b, i_g}
\end{aligned}$$

This implies, when each term of the summations is properly accounted for:

$$\begin{aligned}
\xi(1) \quad = \quad & \left[ 1 - 0 + 1 + (1 - 1 + 1) \binom{1}{1} \right] \cdot \left[ 1 - 0 + 1 + (1 - 1 + 1) \binom{1}{1} \right] \cdot \frac{2!}{0!0!2!} \\
+ \quad & [1 - 1 + 1 + 0] \cdot \left[ 1 - 0 + 1 + (1 - 1 + 1) \binom{1}{1} \right] \cdot \frac{2!}{1!0!1!} \\
+ \quad & \left[ 1 - 0 + 1 + (1 - 1 + 1) \binom{1}{1} \right] \cdot [1 - 1 + 1 + 0] \cdot \frac{2!}{0!1!1!} \\
+ \quad & \left[ 1 - 1 + 1 + (1 - 1 + 1) \binom{1}{1} \right] \cdot \left[ 1 - 1 + 1 + (1 - 1 + 1) \binom{1}{1} \right] \cdot \frac{2!}{1!1!0!}
\end{aligned}$$

$$\xi(1) = 3 \cdot 3 \cdot 1 + 1 \cdot 3 \cdot 2 + 3 \cdot 1 \cdot 2 + 1 \cdot 1 \cdot 2 = 9 + 6 + 6 + 2 = 23$$

The possible 23 board dispositions are listed on Figure 10.

## B Algorithm to calculate the combinations in the different versions of the RGU

The algorithm is written in Fortran 95, but the logic is quite simple and expressed in the documentation reproduced along the code presented in Figure 11. The code is valid for $p < t$, and assumes $p$ and $t'$ are the same for both players.

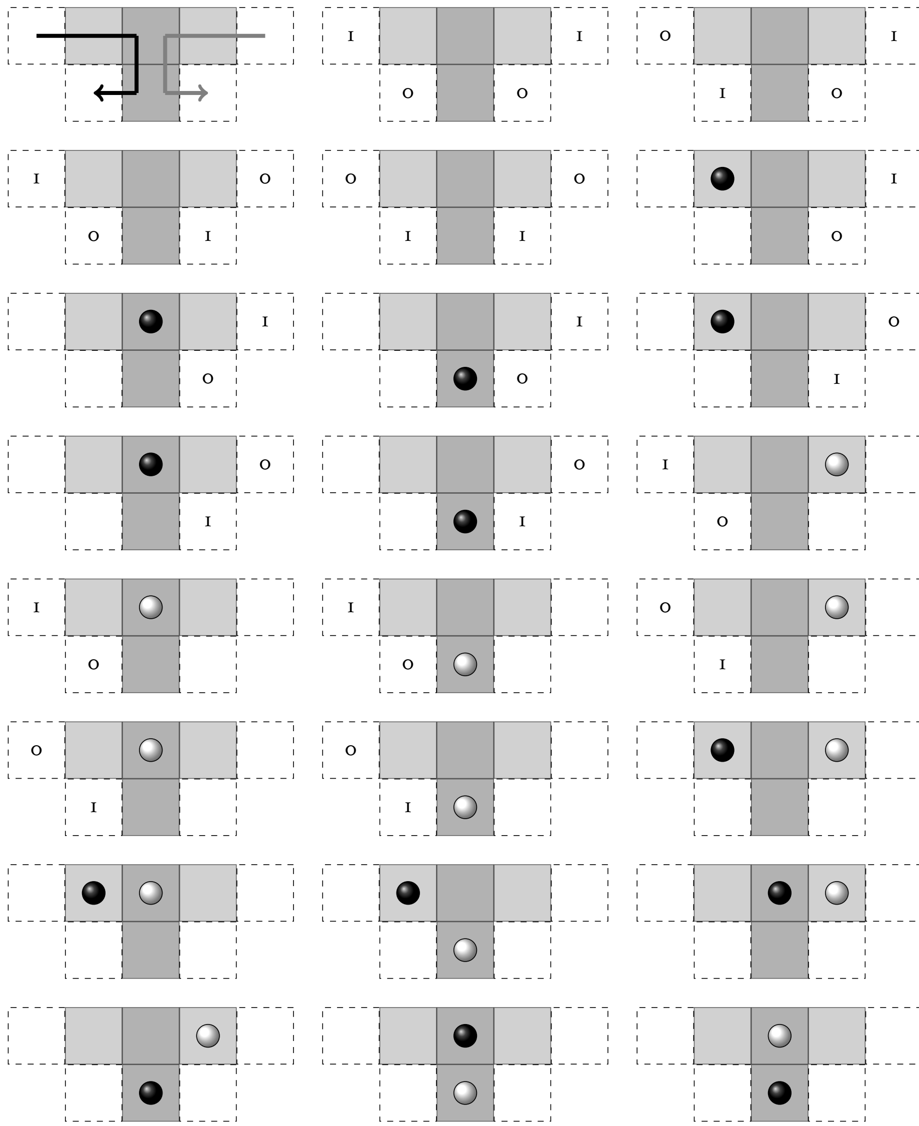



**Figure 10:** Configurations of a mini-RGU with $p = t' = 1$ and $t = 2$.


DOI: 10.2478/bgs-2023-0001

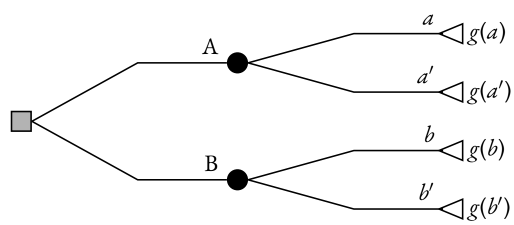
```fortran
program rgu_program
implicit none
!xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx!
!------------- This program calculate the space-state complexity of the Royal Game of Ur ---------------!
!xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx!
integer, parameter :: ikind=selected_real_kind(p=15)
real(kind=ikind) :: p,t_safe,t_war,ig_safe,ib_safe,ig_war,ib_war,fact
real(kind=ikind) :: delta_gray,delta_black,a_gray,a_black,sum1,sum2,sum3
real(kind=ikind) :: cont,top1,top2,top3,total
print *,'----------------------------------------------------------------------------'
print *,'   The program rgu calculates the number of combinations of p stones        '
print *,'       in t_safe houses and t_war houses in the Royal Game of Ur            '
print *,'----------------------------------------------------------------------------'
print *,'             Any issue, contact the developer Diego J. Raposo               '
print *,'                    (email: diegoraposo@hotmail.com)                        '
print *,'----------------------------------------------------------------------------'
print *,'Reading input file:'
open(10,file='rgu_input.txt')                                                  !Opening the input file!
read(10,*) p,t_safe,t_war                                    !Reading variables in the input file!
print *,'----------------------------------------------------------------------------'
print *, 'The value of p is',p
print *,'----------------------------------------------------------------------------'
print *, 'The value of t_safe is',t_safe
print *,'----------------------------------------------------------------------------'
print *, 'The value of t_war is',t_war
print *,'----------------------------------------------------------------------------'
open(12,file='rgu_output.txt')                              !Open a file where the output will be stored!
!do p=1.0_ikind,t_war-1                                                     !Setting a cycle, changing p!
if (2*p <= t_war) then                                    !There will be one external cycle if 2p <= t_war!
  top1 = p                                                              !Establishing the limits of the sums!
  top2 = 0.0_ikind
  top3 = p
  cont = 2.0_ikind                                                          !This allows one external cycle!
  total=0.0_ikind                                         !'total' will be added up at each internal cycle!
else                                                      !There will by two external cycles if 2p > t_war!
  top1 = p                                                              !Establishing the limits of the sums!
  top2 = 0.0_ikind
  top3 = t_war - p
  cont = 1.0_ikind                                                         !This allows two external cycles!
  total=0.0_ikind                                         !'total' will be added up at each internal cycle!
end if
  do while (cont < 3)                 !'cont' will limit to one or two cycles, depending on previous value!
    do ig_war=0,top1                                        !sumations on gray stones inside the war zone!
      do ib_war=top2,top3     !sumations on black stones inside the war zone!
        sum1=0.0_ikind                                                              !'sum1' will be added up!
        if (p-ig_war == 0.0_ikind) then        !if there is no stone left to safe zone or outside de board!
          sum1 = 0.0_ikind
        else
          if (p-ig_war <= t_safe) then                  !there is not enough stones to fill the safe place!
            do ig_safe=1.0_ikind,p-ig_war
              sum1=sum1+fact(t_safe)/(fact(ig_safe)*fact(t_safe-ig_safe)) !
            end do
          else
            do ig_safe=1.0_ikind,p-ig_war            !enough to fill the safe place, and deltas if pass it!
              delta_gray=ig_safe-t_safe
              a_gray=(delta_gray-abs(delta_gray))/2.0_ikind+t_safe
              sum1=sum1+fact(t_safe)*(abs(delta_gray)+1)/(fact(a_gray)*fact(t_safe-a_gray))
            end do
          end if
        end if
        sum1=sum1+p-ig_war+1                        !adding the case when all pieces are outside the board!
        sum2=0.0_ikind                                                              !'sum2' will be added up!
        if (p-ib_war == 0.0_ikind) then                                       !the same for the black pieces!
          sum2 = 0.0_ikind
        else
          if (p-ib_war <= t_safe) then
            do ib_safe=1.0_ikind,p-ib_war
              sum2=sum2+fact(t_safe)/(fact(ib_safe)*fact(t_safe-ib_safe))
            end do
          else
            do ib_safe=1.0_ikind,p-ib_war
              delta_black=ib_safe-t_safe
              a_black=(delta_black-abs(delta_black))/2.0_ikind+t_safe
              sum2=sum2+fact(t_safe)*(abs(delta_black)+1)/(fact(a_black)*fact(t_safe-a_black))
            end do
          end if
        end if
        sum2=sum2+p-ib_war+1
        sum3 = fact(t_war)/(fact(ib_war)*fact(ig_war)*fact(t_war-ib_war-ig_war))!combinations inside war zone!
        total=total+sum1*sum2*sum3                          !total sum, given one choice of ib_war and ig_war!
      end do
    end do                                                             !first or only double sum finishes here!
    cont = cont + 1.0_ikind                               !one or two double cycles, depending on earlier value!
    top1 = t_war - p                                           !if there are two external cycles, new limits!
    top2 = t_war - p + 1
    top3 = p
  end do
print *,'----------------------------------------------------------------------------'
print *, 'The state-space complexity number is',p, total
!write (12,*) p,total                                          !to write the results in a output textfile!
!end do                                                                       !cycle with variable p values!
print *,'----------------------------------------------------------------------------'
print *, 'Thank you for using this program!'
end program rgu_program
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! Defining factorial function
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
function fact(number)
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind) :: fact, number, z
z = 1
do i = 1, number
z = z * i
end do
fact = z
end function fact
```

**Figure 11:** Fortran code used to calculate SSC of different paths in RGU.

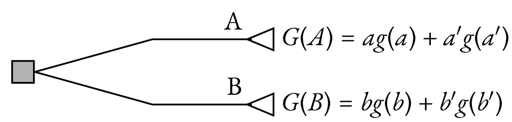

DOI: 10.2478/bgs-2023-0001

## C Simple decision tree algorithm for RGU

A decision tree is a scheme/algorithm/model that provides quantitative tools to make decisions in situations where the statistical element is present. It is composed by decision nodes (where the choices are splitted into branches, whose consequences need to be analysed), chance nodes (whose branches amounts the possible outcomes and the respective probabilities) and end nodes (associated with some numerical value, which one aim to maximize or minimize). Usually the decision, chance and end nodes are represented by tiles, circles and triangles, respectively. A typical and simple decision tree is presented in Figure 12(a).




**(a)** Decision tree scheme for deciding between A and B




**(b)** Reduction of the decision tree by simplifying the chance nodes.

**Figure 12:** Simple decision tree with two options, A and B, both of which have random events attached to them. The calculation of the gain function for each is also presented.

This three represents a choice between A or B, depending on the events that follow them. Choosing A, for instance, leads to two possible events, with probabilities $a$ and $a'$, respectively. For each possible path there is an outcome, a profit, for instance. The event with probability $a$ implies a profit of $g(a)$, while the alternative event, with probability $a'$, gives a profit of $g(a')$. The same reasoning applies to the paths along B. In order to decide between the two choices considering the probabilities and profits, the expected value of $g$ for each node is calculated and compared. Such expected value, which also may be referred as a gain function, $G$, is calculated for a chance node, considering all possible branches in this path, according to Eq. 35:

$$G(\text{chance node}) = \sum_{\text{branches}} P(\text{branch})g(\text{branch}) \tag{35}$$

Since $G$ is an expected value, the branches in the chance node must obey a normal-


DOI: 10.2478/bgs-2023-0001

ization condition. In our example, $a + a' = 1$ and $b + b' = 1$.

With Eq. 35 the statistical elements of the decision can be wrapped up in numeric values that can be compared. For instance, the path A has a expected value of $G(A) = ag(a) + a'g(a')$, that can be compared with the same function of the alternative path, $G(B) = bg(b) + b'g(b')$. If the objective is to maximize $G$, then $G(A) > G(B)$ implies A is the path that must be chosen; and B is the best one if $G(A) < G(B)$. Graphically, this reduces the tree to decision elements only, such as displayed in Figure 12(b).

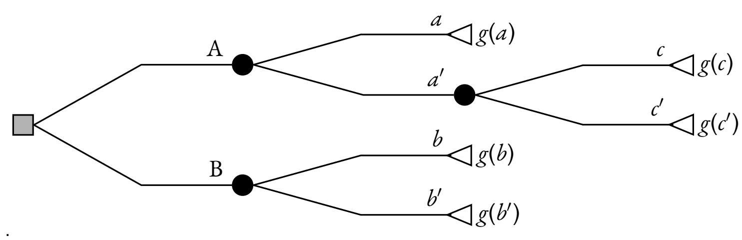
Decision trees can have nested chance nodes, one inside (or branching from) the other. One of such cases is exemplified in Figure 13(a), where the path A has a splitting on the event with probability $a'$, which is divided into two new events, one with probability $c$ and other with probability $c'$ (and their respective gains). The simplification of this case is performed by considering the two sequences, event with $a'$ followed by the event with $c$, and the event with $a'$ followed by the event with $c'$. Giving the statistical independence, the first sequence has the probability $a'c$; the second, $a'c'$. The $g$ function of these branches are the sum of each contribution, and the resulting tree is shown in Figure 13(b). Notice that this new tree branch departing from A must obey the normalization conditions just the same. The original tree imposed that $a + a' = 1$ and $c + c' = 1$. The new one must have $a + a'c + a'c' = 1$, which can be proved: $a + a'c + a'c' = a + a'(c + c') = a + a' = 1$.

The final result of the "adding gains and multiplying probabilities" is presented on Figure 13(c), after the calculation of $G$ in the respective chance nodes. Once again $G(A)$ and $G(B)$ are compared to make an informed decision.

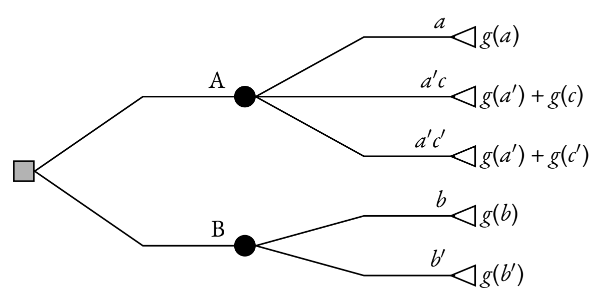
The simplest decision tree one can see while playing a match of RGU is presented in Figure 14(a). While the values of $P(i)$ for a given $i$ are known, we have not defined $g(i)$ yet. It encloses several contributions, but it is helpful consider them explicitly afterwards (namely, attacks and captures). The walking contribution ($W$) is just how many tiles the piece walk after $i$ is selected. Therefore, $g(i) = i$ in such case. The tree in 14(a) has, by that definition, a walking contribution $W$ to the gain $G$ of 2:

$$G = W = \sum_{i=1}^{4} iP(i) = 0 \cdot \frac{1}{16} + 1 \cdot \frac{4}{16} + 2 \cdot \frac{6}{16} + 3 \cdot \frac{4}{16} + 4 \cdot \frac{1}{16} = 2$$

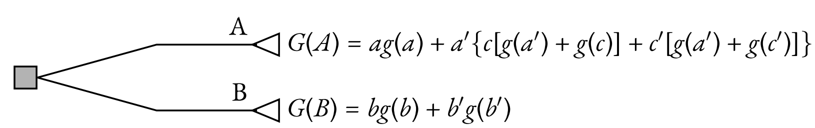
This case considers all moves legal or possible. If one or more forbidden moves are present, we can either do not count them while doing the summation for $W$ or include a subtractive term due to forbidden moves, $F$, that compensates the over counting on $W$ calculation. The gain is, then, $G = W - F$. For instance, in Figure 14(b) the selection of 1 leads to a forbidden move, so we can calculate the gain by subtracting $F = 1 \cdot (4/16)$ from $W = 2$, leading to $G = 7/4$. For simplicity, the former approach will be adopted, and only legal moves will be included on the walking summation.




**(a)** Decision tree with entangled nodes.

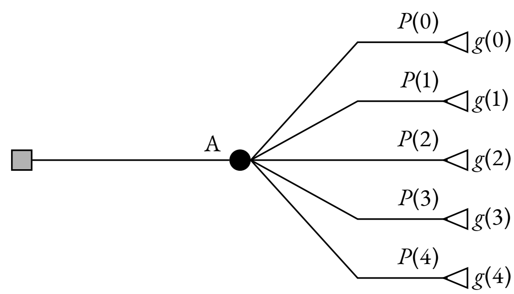



**(b)** Simplification of the entangled node.

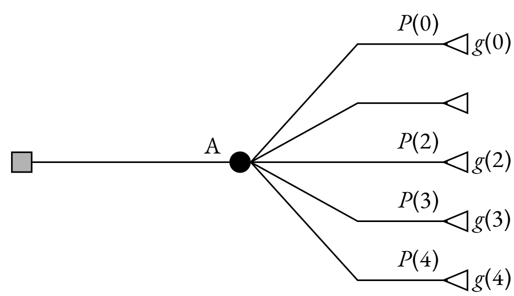



**(c)** Simplification of all nodes of the tree.

**Figure 13:** Simple decision tree with entangled chance nodes, and further simplifications.




**(a)** Decision tree that describes the possible moves A can make in RGU board.



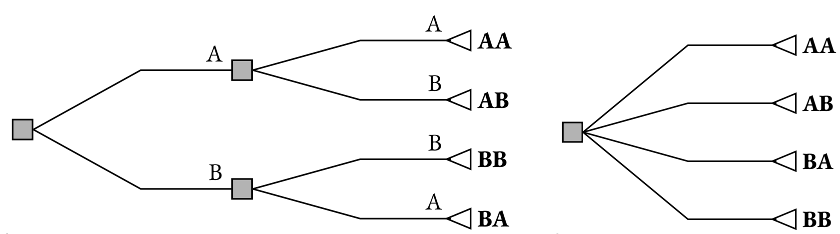

**(b)** Decision tree of RGU board, with an illegal move of the piece A if 1 is the selected number in the roll of the dice.

**Figure 14:** Decision trees for the first move on the RGU, considering only a single piece A.

Hence, the walking is expressed as a sum of the product of the legal numbers $r_0$ and the respective probabilities, $P(r_0)$:

$$W_0 = \sum_{r_0} r_0 \cdot P(r_0) \tag{36}$$

This analysis is based on a single move (zero rosettes). One particular characteristics of RGU is the possibility that one of the pieces reaches a rosette, giving the player another throw on his turn. In such a case the next bonus throw can be used to move that very piece again, or any of the other pieces (this choice is the essential strategy element of the game, hence the need for algorithms to do this kind of decision). This situation is illustrated in Figure 15(a), with the first move being centered on the decision of moving A or B and, if the rosette can be reached by either A or B, the second move decision involves the same piece or the alternative. This leads to four possible sequences that can be separated and compared in terms of gain, such as represented in Figure 15(b)




**(a)** Chain of events while choosing the movements of the pieces A and B.

**(b)** Sequence of movements of pieces A and B, individually isolated.

**Figure 15:** Sequence of two moves in a single turn, by using two pieces, A and B.

Inside each of these sequences of pieces there are chance nodes, including nested chance nodes for more than one allowed movement. Now we will generalize the treatment using some examples to facilitate the comprehension. Let's say, for instance, that the selection of $r_1 = 2$ leads the first piece to a rosette. The next move will lead to the five possible numbers (some of them possibly tied to illegal moves) inside the chance node of the first move. This is illustrated in Figure 16:

Using the simplification method for reducing the entangled chance nodes into one, the simplified tree looks like the one presented in Figure 17.

Calculating the expected value in the chance node we can estimate the gain associated with the sequence **AB**:


DOI: 10.2478/bgs-2023-0001

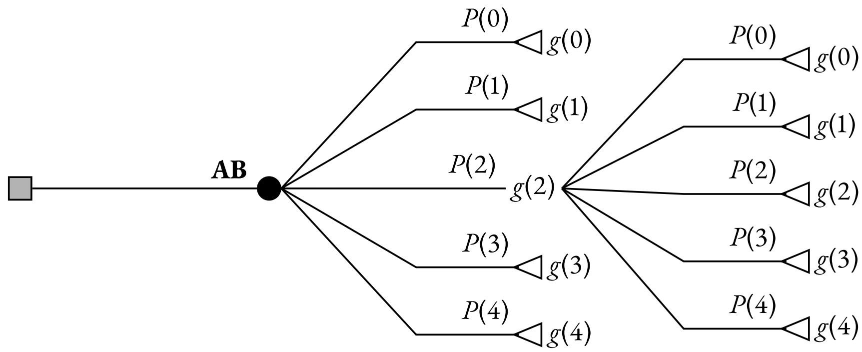



**Figure 16:** Example of a decision tree with two moves in a RGU turn.

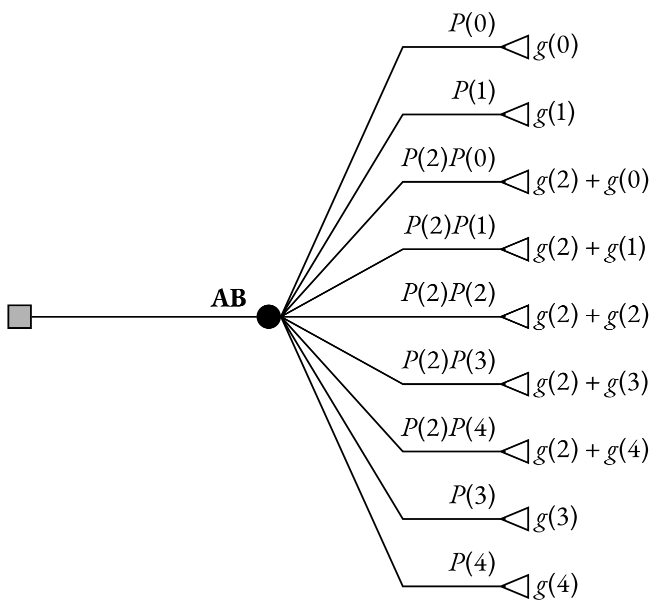



**Figure 17:** Simplification of the decision tree of Figure 16.

$$
\begin{aligned}
G &= \sum_{i\neq 2} g(i)P(i) + \sum_{i=0}^{4} \left[g(2) + g(i)\right] P(2)P(i) \\
&= \sum_{i\neq 2} g(i)P(i) + \sum_{i=0}^{4} g(2)P(2)P(i) + \sum_{i=0}^{4} g(i)P(2)P(i) \\
&= \sum_{i\neq 2} g(i)P(i) + g(2)P(2) \sum_{i=0}^{4} P(i) + P(2) \sum_{i=0}^{4} g(i)P(i) \\
&= \sum_{i\neq 2} g(i)P(i) + g(2)P(2) + P(2) \sum_{i=0}^{4} g(i)P(i) \qquad (37) \\
&= \sum_{i=0}^{4} g(i)P(i) + P(2) \sum_{i=0}^{4} g(i)P(i) \\
&= [1 + P(2)] \sum_{i=0}^{4} g(i)P(i) \\
&= [1 + P(2)]\ W_0
\end{aligned}
\qquad (38)
$$

Where the normalization condition ($\sum_{i=0}^{4} P(i) = 1$) and the previous calculation for $W_0$ were used. Notice that the first term is just $W_0$, while $W_0P(2)$ is a term due to the rosette the first piece can reach if 2 is the result of the roll of the dice. More generally, the walking contribution when two movements are allowed is the sum of $W_0$ and $W_1 = W_0P(r_1)$, considering a rosette reachable by the selection of the number $r_1$:

$$W = W_0 + W_1 = W_0 + P(r_1)W_0 = W_0\,[1 + P(r_1)] \qquad (39)$$

Following the same steps, it is easy to prove that the possibility of reaching a new rosette, hence allowing a second move if the number $r_2$ with probability $P(r_2)$ is selected, increase the equation for $W$ by a factor $W_2 = W_0P(r_1)P(r_2) = W_1P(r_2)$:

$$W = W_0 + W_1 + W_2 = W_0\,\{1 + P(r_1)\,[1 + P(r_2)]\} \qquad (40)$$

The $W_3$ can be calculated in the same way, but for simplicity we will reduce the analysis to three moves only. The general formula is given by:

$$W = W_0\,[1 + P(r_1) + P(r_1)P(r_2) + \ldots] = W_0 \left[1 + \sum_{n>0} \prod_{k=1}^{n} P(r_k)\right] \qquad (41)$$

which can be applied in a recursive manner:

$$W = W_0 + \sum_{n>0} W_n \tag{42}$$

where:

$$W_n = W_{n-1}P(r_n) = W_0 \prod_{k=1}^{n} P(r_k) \tag{43}$$

Notice that two or even three different rosettes can be reached in a single row of the dice only if all possible pieces for each throw are considered. Once the sequence is defined, there is only one rosette selected by move, for a single piece can reach a specific rosette in a single way. The different sequences, including the ones that reach the same rosette through different pieces, are compared in terms of the gain function.

The attacks contribution to the gain, either from the player or the future possible attacks of the opponent in the next turn, expressed as a capture degree, are calculated separately (for convenience) and must consider that:

- Different pieces can attack the same opponent's piece in the same tile;
- The same piece can attack different pieces in different tiles;
- The same piece can attack another in a fixed tile through distinct sequence of steps.

These variables, tile number ($n$), sequence of selected numbers in the throws in a single turn ($s$) and which piece moves ($p$), can change independently, which implies a mathematical representation that express the summation over the three variables. In other words, for a specific piece one must consider which tiles it can attack, and according which sequences of numbers each. Then the sum is performed for all pieces. The gain in this case is still the same: number of tiles the piece walk. In a attack, the player takes a piece of the opponent back to the beginning, then the opponent loses $n$ tiles in the run, which puts the player $n$ tiles ahead (gain). If the player is attacked, the opponent gain $n$ tiles of advantage at the cost of the equivalent player's loss. Given the probability of the attack of the player, $P_a$, or a attack of the opponent, $P_c$, the player gains $n_a P_a$ or loses $n_c P_c$ accordingly. Considering the variables just mentioned, the total attack value, $A$, or capture value (tendency to the adversary's attack in the next turn), $C$, for the player can be written as:

$$A = \sum_p \sum_a \sum_s n_a P_a = \sum_p \sum_a n_a \sum_s P_a \tag{44}$$

$$C = -\sum_p \sum_c \sum_s n_c P_c = -\sum_p \sum_c n_c \sum_s P_c \tag{45}$$

These concepts, in particular, are more understandable through an example, which will be, in our case, the board described in Figure 8. The detailed calculations are represented in two tables: Tab. 5 displays the values of $W$ for the possible sequences of moves (forbidden moves properly subtracted), and Tab. 6 lists the values used to calculate $A$ and $C$. The comparison among sequences and the main results are presented in section 4.3.

**Table 5:** The contribution $W$ for the decision making algorithm proposed, applied to the specific example of Figure 8.

| Sequence | Walking |
|---|---|
| **AA** | $2-2\cdot\left(\frac{6}{16}\right)+2\cdot\left(\frac{1}{16}\right)-(4+4)\cdot\left(\frac{1}{16}\right)\cdot\left(\frac{1}{16}\right)$ |
| **AB** | $2-2\cdot\left(\frac{6}{16}\right)+2\cdot\left(\frac{1}{16}\right)-(4+2)\cdot\left(\frac{1}{16}\right)\cdot\left(\frac{6}{16}\right)$ |
| **BA** | $2-0+2\cdot\left(\frac{6}{16}\right)-(2+4)\cdot\left(\frac{6}{16}\right)\cdot\left(\frac{1}{16}\right)$ |
| **BB** | $2-0+2\cdot\left(\frac{6}{16}\right)-(2+4)\cdot\left(\frac{6}{16}\right)\cdot\left(\frac{1}{16}\right)$ |

Notice that the contributions of $A$ and $C$ can be tracked down to each term, piece, sequence, tile number, in a organized way, and calculated accordingly. The attacks of the player, or possible attacks of the opponent in the next turn, are determined by the number of the tile where the attack occurs (always in the war zone). For instance, consider that B can, in a first throw, to occupy the tile 6 where a grey piece is, by rolling 4 on the dice. That leads to a gain for the player of 6 (number of tiles that the opponent has to retrocede) times 1/16 (the probability of taking 4). The opponent can reach the player in the next turn, however: it must select either 3 or the sequence 12 (going through his own rosette). These are possible losses for the player, in the form of $-6\cdot\frac{4}{16}$ (if 3) plus $-6\cdot\frac{4}{16}\cdot\frac{6}{16}$ (if 12). Including all attacks of the player and of the opponent ($A+C$), for all pieces in all sequences in the route **AA**, for instance:

$$\underbrace{+6\cdot\frac{1}{16}\cdot\frac{6}{16}}_{\substack{\text{black}\\ \text{tile 6}\\ \text{sequence 42}}}\underbrace{-5\left(\frac{6}{16}+\frac{4}{16}\cdot\frac{4}{16}\right)}_{\substack{\text{grey}\\ \text{tile 5}\\ \text{sequence 2}\\ \text{sequence 11}}}\underbrace{-6\left(\frac{4}{16}+\frac{4}{16}\cdot\frac{6}{16}\right)}_{\substack{\text{grey}\\ \text{tile 6}\\ \text{sequence 3}\\ \text{sequence 12}}}\underbrace{-7\left(\frac{1}{16}+\frac{4}{16}\cdot\frac{4}{16}\right)}_{\substack{\text{grey}\\ \text{tile 7}\\ \text{sequence 4}\\ \text{sequence 13}}}=-4.98$$

**Table 6:** The contributions $A$ and $C$ for the decision making algorithm proposed, applied to the specific example of Figure 8.

| Sequence | Attack | Capture |
|---|---|---|
| **AA** | • **A**: 0<br>• **A**:<br>– Tile 6:<br>* Seq. 42: $6 \cdot \left(\frac{1}{16}\right) \cdot \left(\frac{6}{16}\right)$ | • **A**: 0<br>• **A**:<br>– Tile 5:<br>* Seq. 2: $-5 \cdot \left(\frac{6}{16}\right)$<br>* Seq. 11: $-5 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$<br>– Tile 6:<br>* Seq. 3: $-6 \cdot \left(\frac{4}{16}\right)$<br>* Seq. 12: $-6 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{6}{16}\right)$<br>– Tile 7:<br>* Seq. 4: $-7 \cdot \left(\frac{1}{16}\right)$<br>* Seq. 13: $-7 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$ |
| **AB** | • **A**: 0<br>• **B**:<br>– Tile 6:<br>* Seq. 44: $6 \cdot \left(\frac{1}{16}\right) \cdot \left(\frac{1}{16}\right)$ | • **A**: 0<br>• **B**:<br>– Tile 5:<br>* Seq. 2: $-5 \cdot \left(\frac{6}{16}\right)$<br>* Seq. 11: $-5 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$<br>– Tile 6:<br>* Seq. 3: $-6 \cdot \left(\frac{4}{16}\right)$<br>* Seq. 12: $-6 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{6}{16}\right)$ |
| **BA** | • **B**:<br>– Tile 6:<br>* Seq. 4: $6 \cdot \left(\frac{1}{16}\right)$<br>• **A**: 0 | • **B**:<br>– Tile 5:<br>* Seq. 2: $-5 \cdot \left(\frac{6}{16}\right)$<br>* Seq. 11: $-5 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$<br>– Tile 6:<br>* Seq. 3: $-6 \cdot \left(\frac{4}{16}\right)$<br>* Seq. 12: $-6 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{6}{16}\right)$<br>• **A**: 0 |
| **BB** | • **B**:<br>– Tile 6:<br>* Seq. 4: $6 \cdot \left(\frac{1}{16}\right)$<br>• **B**:<br>– Tile 6:<br>* Seq. 22: $6 \cdot \left(\frac{6}{16}\right) \cdot \left(\frac{6}{16}\right)$ | • **B**:<br>– Tile 5:<br>* Seq. 2: $-5 \cdot \left(\frac{6}{16}\right)$<br>* Seq. 11: $-5 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$<br>– Tile 6:<br>* Seq. 3: $-6 \cdot \left(\frac{4}{16}\right)$<br>* Seq. 12: $-6 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{6}{16}\right)$<br>• **B**:<br>– Tile 5:<br>* Seq. 2: $-5 \cdot \left(\frac{6}{16}\right)$<br>* Seq. 11: $-5 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$<br>– Tile 6:<br>* Seq. 3: $-6 \cdot \left(\frac{4}{16}\right)$<br>* Seq. 12: $-6 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{6}{16}\right)$<br>– Tile 7:<br>* Seq. 4: $-7 \cdot \left(\frac{1}{16}\right)$<br>* Seq. 13: $-7 \cdot \left(\frac{4}{16}\right) \cdot \left(\frac{4}{16}\right)$ |


DOI: 10.2478/bgs-2023-0001

While it is possible to design an algorithm that implement such equations and, ultimately, can play against a silicon or human opponent, that is not necessary, for much better algorithms are already available, and some of them are very good. Paddy Lamont's Expectimax algorithm, for instance, is based on the search of possible scenarios on the board, the distance between the player and the opponent in the run of RGU, the probability of reaching that situation, and the choice of the move that leads to the maximum of the product of those quantities (Lamont, 2021). While the approach presented in this paper has a depth of one turn plus a part of the next turn (the part related to the gain of the opponent), those algorithms can evaluate until seven turns or moves ahead, without the need of the separation in different contributions. The presented results, however, have an alternative intent: to give insight of how different contributions come at play in a RGU match, and how the mathematics behind it allows the improvement of the player's decision making while playing, all while presenting the content in a step by step manner.