Coupon collector model to determine the expected time to win a sequence dice game

Lasisi Olalekan Hameed

University of Passau

January 28, 2023

UNIVERSITÄT
PASSAU

Introduction: Sequential Games

- Involve players or teams of players who do not make decisions concurrently.

- One player’s decisions impact the results and decisions of subsequent players or teams.

- Examples: Go, backgammon, chess, tic-tac-toe, and sequence dice.

Introduction: Sequence Dice Game

- Developed by Douglas Reuter (born March 17, 1949) over a 4 year period [Rya18]
- Published commercially as "Sequence Five" in 1982 [Rya18]
- Comprises 60 chips, game board, two dice, and instructions
- Chips divided into 3 equal groups of 20 (red, green, and blue)

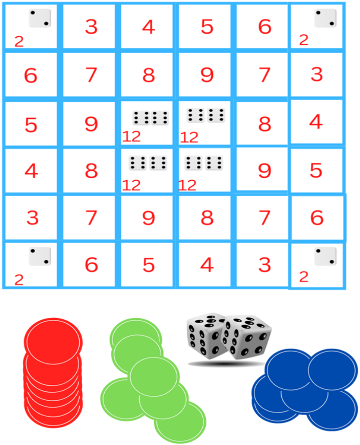


Figure 1: Sample representation of the sequence dice game board with chips and dice

Introduction: Sequence Dice Game Cont‘d

- Game is designed for 2-4 players or teams of players.
- Game board includes 36 spaces with numbers from 2 - 9, 12.
- Each number appears randomly four times apart from 10 and 11.
- Start: The game chips are divided amongst teams or players.

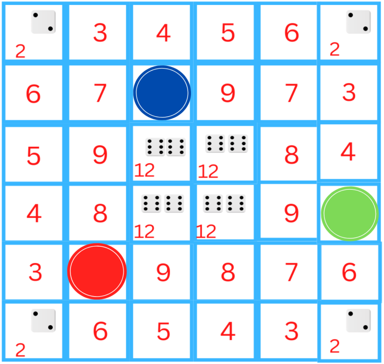


Figure 2: Sample representation of the sequence dice game board with chips

Introduction: Sequence Dice Game Cont‘d

- Players or teams take turns rolling two dice to determine who takes the first turn

- Special rolls require specific actions:

- 10 is defensive roll, 11 is wild roll, 2 or 12 allows player to take another turn

- Objective: Create a sequence of five chips in a row on the board

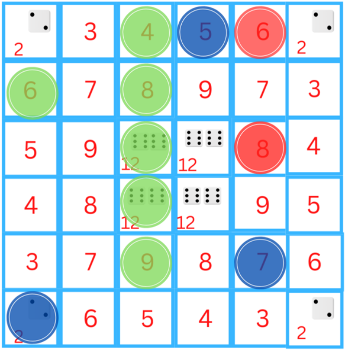


Figure 3: Sample representation of accumulated 5 chips in a row on a sequence dice board

Introduction: Motivation

- Games have been a part of human life throughout all civilizations.
- Previous studies on ludo, snake and ladders, monopoly, e.t.c.
- No research on expected time to win sequence dice game.

Introduction: Research Questions

- Expected number of die rolls to obtain a sequence of five chips.
    - One-dimensional with simplified rules.
    - One-dimensional with original rules.
    - Full board with simplified rules.
    - Full board with original rules.
- Effects of strategies on expected time to win game with full playing board.

Introduction: Research Objectives

- Theorectically
  - Use coupon collector model for unequal probabilities to simplify board and analyse the expected time to win game with restricted set of rules.
- Empirically
  - Analyse expected time to win game with one-dimensional board & full playing board.
  - Propose 3 strategies and perform comparative analysis of strategies on expected time to win game with full playing board.

Background: Related Works

Author	Title	Objectives	Methodology	Results
Althoen et al, 1993	How long is a game of snakes and ladders?	1. Average length of the game. 2. Effect of the number of snakes and ladders has on the length of the game.	1. Simple experiment with 10x10 board 2. UBASIC86 Programming language. 3. Markov Chain,Gauss-Jordan reduction	1. Average game length of 39.1 moves. 2. Addition of ladder increases game length, addition of snake reduces length.
Edwin, 2021	How long is a game of snakes and ladders?	1. To improve the work of [AKS93]. 2Examining game parameters: length & number of snakes & ladders, plus die distribution.	1. Mersenne Twister pseudo-random number generator. 2. Markov chain models 3. Monte Carlo Method	1. Dice distribution greatly influences the length of the game 2. Number of ladders and snakes significantly mpacts the expected average length
Faisal et al, 2011	Complexity analysis and playing strategies for Ludo and its variant race games	Analyse performance of aggressive, fast, defensive, and random strategies to test state-space complexity of generic Ludo game.	1. 5000-game play with all four players selecting the random strategy. 2. Evaluation of each basic strategy player against the other three players playing random strategy. 3. Evaluation of all four players using all the strategies, including the random strategy, 4. Development and evaluation of a "mixed strategy"	1. Random strategy had a 0% win rate. 2. Defensive strategy had the highest winning percentage at 40%, aggressive strategy at 32%, fast strategy at 27%, 3. Mixed strategy has a 90% win rate and an average move count of 66, with an estimated state-space complexity of 1022
Tochukwu , 2021	How long is a game of Ludo?	1. To improve on the work of [AA11] 2. Conducting a study on the length of the Ludo game	1. Three strategies (random-six, once-six-tokens-out, and complete-one-token-before-next) with n-token count ranging from 1 to 10 four pure strategies (aggressive, fast, random, and defensive) and one mixed strategy	1. Random-six and once-six-token-out strategies outperformed the complete-one-token-before-next strategy. 2. Random strategy had the fewest wins, aggressive strategy had the most wins, mixed strategy is most efficient, ludo board without safe squares had a shorter average length than with safe squares.

Background: Related Mathematical Models

- Nash Equilibrium
- Maximum Minimum
- Markov Chain
- Coupon Collector’s Problem
    - The Coupon Collector's Problem and Quality Control
    - Coupon-collector’s Problem With Several Parallel Collections
    - A Liability Allocation Game
    - Coupon Collecting with Quotas
    - A Coupon Collector’s Problem With Bonuses
    - Birthday Paradox and CCP

Methodology: Parameterisation of the One-dimensional Board of the Sequence Dice Game

- C: $C = \{2,3,4,5,6,7,8,9,12\} = \{c_1, c_2, c_3, \dots, c_{h-1}, c_h\}$

$$p(c) = \frac{n(h)}{n(N)} : h = \{c_1, c_2, \dots, c_h\}$$

- S: $S \subset C, U_{i=0}^{c} S_i = C$ *and* $\cap_{i=0}^{c} S_i = \emptyset$

$$S = \{S_1, S_2, S_3 S_4 S_5\}$$

$$S_i = \{w_{i+j}\}_{j=0}^{k-1}$$

$$where\ i = 1,2,\dots,m, \qquad j = 0,1,2,\dots,k-1\ , and\ k = 5.\ m = h-k+1$$

$$S_1 = \{2,3,4,5,6\}, \qquad S_2 = \{3,4,5,6,7\}, \qquad S_3 = \{4,5,67,8\},$$

$$S_4 = \{5,6,7,8,9\}, \qquad S_5 = \{6,7,8,912\}$$

- V: $V = \{10,11\}$

UNIVERSITÄT PASSAU

Methodology: Expected Time to Collect all Relevant Coupons of Equal Probability

- Let N = 9 be the number of different relevant coupons with a probability of rolling each at any time equal to $\frac{1}{N}$.

- Let T represent the number of rolls it would require to collect all N relevant coupons.

$$E(T) = E(T_1) + E(T_2) + \ldots + E(T_N)$$

$$E(T) = 1 + \frac{N}{N-1} + \frac{N}{N-2} + \cdots + \frac{N}{2} + N$$

$$E(T) = N \sum_{i=1}^{N} \frac{1}{i}$$

$$where, N = 9 \; and \; i = 1, 2, 3, 4, 5, 6, 7, 8, 9$$

$$E(T) = 9(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9})$$

$$E(T) = 25.461$$

UNIVERSITÄT PASSAU

Methodology: Expected time to collect all coupons with probabilities associated with two dice

- Calculate the conditional probability of each relevant step.

$$p(c_i) = p(c_i | V')$$

$$p(c_i | V') = \frac{p(c_i \cap V')}{p(V')}$$

$$p(V') = \frac{31}{36}$$

Relevant Steps	2	3	4	5	6	7	8	9	12
Conditional Probability	$\frac{1}{31}$	$\frac{2}{31}$	$\frac{3}{31}$	$\frac{4}{31}$	$\frac{5}{31}$	$\frac{6}{31}$	$\frac{5}{31}$	$\frac{4}{31}$	$\frac{1}{31}$
$p(c_i)$	$p(c_1)$	$p(c_2)$	$p(c_3)$	$p(c_4)$	$p(c_4)$	$p(c_5)$	$p(c_6)$	$p(c_8)$	$p(c_9)$

- $E(T) = E(T_w | V') \times E(T_R | V')$
- $E(T_w | V') = \frac{1}{p(V')}$ = 36/31 = 1.1613
- $E(X) = \sum_i^k \frac{1}{p_i} - \sum_{i<j}^k \frac{1}{p_i + p_j} + \sum_{i<j<l}^k \frac{1}{p_i + p_j + p_l} - \sum_{i<j<l<n}^k \frac{1}{p_i + p_j + p_l + + p_n} + \cdots + (-1)^{k+1}$
- $E(T_R | V')$ = 50.58586255970185
- $E(T) = 1.1613 \times 50.58586255970185$
- $E(T) = 58.7453621905818$

```python
p = [1/31, 2/31, 3/31, 4/31, 5/31, 6/31, 5/31, 4/31, 1/31]

E = 0
for i in range(len(p)):
    E += 1/p[i]
for i in range(len(p)):
    for j in range(i+1, len(p)):
        E -= 1/(p[i] + p[j])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            E += 1/(p[i] + p[j] + p[k])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            for l in range(k+1, len(p)):
                E -= 1/(p[i] + p[j] + p[k] + p[l])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            for l in range(k+1, len(p)):
                for m in range(l+1, len(p)):
                    E += 1/(p[i] + p[j] + p[k] + p[l]+ p[m])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            for l in range(k+1, len(p)):
                for m in range(l+1, len(p)):
                    for n in range(m+1, len(p)):
                        E -= 1/(p[i] + p[j] + p[k] + p[l]+ p[m]+ p[n])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            for l in range(k+1, len(p)):
                for m in range(l+1, len(p)):
                    for n in range(m+1, len(p)):
                        for o in range(n+1, len(p)):
                            E += 1/(p[i] + p[j] + p[k] + p[l]+ p[m]+ p[n
    ]+ p[o])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            for l in range(k+1, len(p)):
                for m in range(l+1, len(p)):
                    for n in range(m+1, len(p)):
                        for o in range(n+1, len(p)):
                            for q in range(o+1, len(p)):
                                E -= 1/(p[i] + p[j] + p[k] + p[l]+ p[m]+
     p[n]+ p[o]+p[q])
E += (-1)**(len(p)+1)
print('The expected number of relevant rolls to get all coupons is ', E)
```

$E(T_R | V')$ = 50.58586255970185

Methodology: Expected Time to Collect Any Subsets of C

- Let T_i be the random time to collect S_i and T be the random time to collect any set, then
- $T = \min(T_1, T_2, T_3, T_4, T_5)$
- $E(T) \leq \min(E(T_1), E(T_2), E(T_3), E(T_4), E(T_5))$
- $E(T_i) = E\left(T_{W_i} \middle| V'\right) \times E\left(T_{R_i} \middle| V'\right)$
- $S_1 = \{c1, c2, c3, c4, c5\} = \{2, 3, 4, 5, 6\}$
- $E\left(T_{W_1} \middle| V'\right) = \frac{1}{\left(p_{w_{1+j}} \middle| V'\right)} = \frac{1}{\frac{1}{31}+\frac{2}{31}+\frac{3}{31}+\frac{4}{31}+\frac{5}{31}} =$
- $\frac{31}{15} = 2.0667$
- $E\left(T_{R_1} \middle| V'\right) = \sum_i^k \frac{1}{p_i} - \sum_{i<j}^k \frac{1}{p_i+p_j} +$

$\sum_{i<j<l}^k \frac{1}{p_i+p_j+p_l} - \sum_{i<j<l<n}^k \frac{1}{p_i+p_j+p_l++p_n} + \cdots + (-1)^{k+1}$

```python
p = [1/31, 2/31, 3/31, 4/31, 5/31]

E = 0
for i in range(len(p)):
    E += 1/p[i]
for i in range(len(p)):
    for j in range(i+1, len(p)):
        E -= 1/(p[i] + p[j])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            E += 1/(p[i] + p[j] + p[k])
for i in range(len(p)):
    for j in range(i+1, len(p)):
        for k in range(j+1, len(p)):
            for l in range(k+1, len(p)):
                E -= 1/(p[i] + p[j] + p[k] + p[l])
E += (-1)**(len(p)+1)
print('The expected number of relevant rolls to get all coupons of S1 is ', E)
```

UNIVERSITÄT PASSAU

Methodology: Expected Time to Collect Any Subsets of C

- $E(T_1) = E(T_{W_1}|V') \times E(T_{R_1}|V')$
- $E(T_1)$ = 2.0667 x 37.52530525030525
- $E(T_1)$ = 77.5535483608059

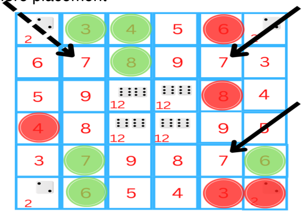
Subsets	S_1	S_2	S_3	S_4	S_5
elements w_{i+j}	$\{c_1,c_2,c_3,c_4,c_5\}$	$\{c_2,c_3,c_4,c_5,c_6\}$	$\{c_3,c_4,c_5,c_6,c_7\}$	$\{c_4,c_5,c_6,c_7,c_8\}$	$\{c_5,c_6,c_7,c_8,c_9\}$
element values	{2,3,4,5,6}	{3,4,5,6,7}	{4,5,6,7,8}	{5,6,7,8,9}	{6,7,8,9,12}
$p(w_{i+j})$	$\frac{1}{31}, \frac{2}{31}, \frac{3}{31}, \frac{4}{31}, \frac{5}{31}$	$\frac{2}{31}, \frac{3}{31}, \frac{4}{31}, \frac{5}{31}, \frac{6}{31}$	$\frac{3}{31}, \frac{4}{31}, \frac{5}{31}, \frac{6}{31}, \frac{5}{31}$	$\frac{4}{31}, \frac{5}{31}, \frac{6}{31}, \frac{5}{31}, \frac{4}{31}$	$\frac{5}{31}, \frac{6}{31}, \frac{5}{31}, \frac{4}{31}, \frac{1}{31}$
$E(T_{W_i} \mid V')$	2.0667	1.55	1.3478	1.2916	1.4762
$E(T_{R_i} \mid V')$	37.5253	21.8026	16.5018	15.0084	33.6025
$E(T_i)$	77.5535	33.7940	22.2411	19.3848	49.6040

- $E(T) \leq \min(E(T_1), E(T_2), E(T_3), E(T_4), E(T_5))$
- $E(T) \leq \min(77.5535, 33.7940, 22.2411, 19.3848, 49.6040)$
- $E(T) \leq 19.3848$

Methodology: Simulation – Strategies

- Offensive Strategy:

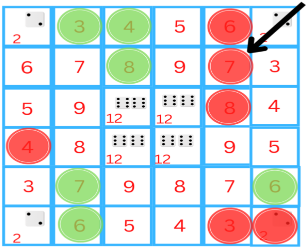
State of the board before placement



Indicates where player 2 will not place their chip

Indicates possible cells where player 2 can place their chip

State of the board after placement



Algorithm 1: Offensive Strategy

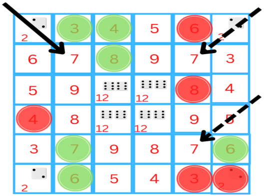
```
for row, column, diagonal in board do
    if row.count(own_chips) ≥ column.count(own_chips) and row.count(own_chips) ≥
     diagonal.count(own_chips) and dice_roll in row then
        row[dice_roll] ← own_chip ;      /* Player places chip in the row with the most
         player chips */
        break
    else if column.count(own_chips) ≥ row.count(own_chips) and column.count(own_chips) ≥
     diagonal.count(own_chips) and dice_roll in column then
        column[dice_roll] ← own_chip ;    /* Player places chip in the column with the
         most player chips */
        break
    else if diagonal.count(own_chips) ≥ row.count(own_chips) and diagonal.count(own_chips) ≥
     column.count(own_chips) and dice_roll in diagonal then
        diagonal[dice_roll] ← own_chip ;  /* Player places chip in the diagonal with the
         most player chips */
        break
    else if row.count(own_chips) ≥ 2 and dice_roll in row then
        row[dice_roll] ← own_chip ;     /* Player places chip in the row that has two or
         more player chips */
        break
    else if column.count(own_chips) ≥ 2 and dice_roll in column then
        column[dice_roll] ← own_chip ; /* Player places chip in the column that has two
         or more player chips */
        break
    else if diagonal.count(own_chips) ≥ 2 and dice_roll in diagonal then
        diagonal[dice_roll] ← own_chip ;  /* Player places chip in the diagonal that has
         two or more player chips */
        break
    else
        random_strategy();
        break
    end
end
```

UNIVERSITÄT PASSAU

Methodology: Simulation – Strategies cont'd

- **Defensive Strategy:**

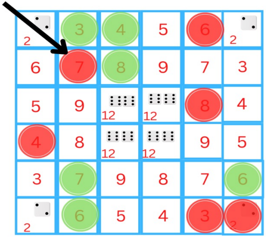
State of the board before placement



Indicates where player 2 will not place their chip

Indicates where player 2 will place thier chip

State of the board after placement



Algorithm 2: Defensive Strategy

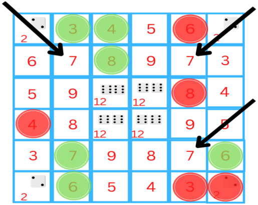
```
for row, column, diagonal in board do
    if row.count(opponent_chips) ≥ column.count(opponent˙chips) and
     row.count(opponent_chips) ≥ diagonal.count(opponent_chips) and dice_roll in
     row then
        row[dice_roll] ← own_chip ;    /* Player places chip in the row with
         the most opponent's chips */
        break
    else if column.count(opponent_chips) ≥ row.count(opponent_chips) and
     column.count(opponent_chips) ≥ diagonal.count(opponent_chips) and dice_roll
     in column then
        column[dice_roll] ← own_chip ;    /* Player places chip in the column
         with the most opponent's chips */
        break
    else if diagonal.count(opponent_chips) ≥ row.count(opponent_chips) and
     diagonal.count(opponent_chips) ≥ column.count(opponent_chips) and dice_roll
     in diagonal then
        diagonal[dice_roll] ← own_chip ;    /* Player places chip in the
         diagonal with the most opponent's chips */
        break
    else if row.count(opponent_chips) ≥ 2 and dice_roll in row then
        row[dice_roll] ← own_chip ;    /* Player places chip in the row that
         has two or more opponent's chips */
        break
    else if column.count(opponent_chips) ≥ 2 and dice_roll in column then
        column[dice_roll] ← own_chip ;    /* Player places chip in the column
         that has two or more opponent's chips */
        break
    else if diagonal.count(opponent_chips) ≥ 2 and dice_roll in diagonal then
        diagonal[dice_roll] ← own_chip ;    /* Player places chip in the
        break
    else
        random_strategy();
        break
    end
end
```

UNIVERSITÄT PASSAU

Methodology: Simulation – Strategies

- Random Strategy:

State of the board before placement



Indicates possible cells where player 2 can place their chip

Algorithm 3: Random Strategy

---

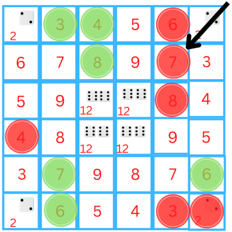
if $dice\_roll \notin \{10, 11\}$ **then**

  choose *index* randomly from *dice_roll_in_board* ;

  place *own_chip* in *board*[*index*] ;   `/* Player places their chip randomly on the board */`

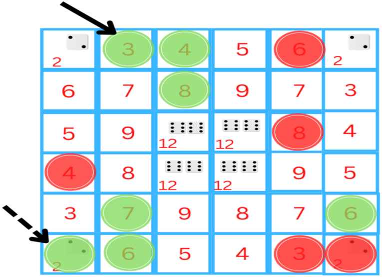
end

State of the board after placement



Methodology: Simulation – Decisions for rule 10

- Lowest Probability:



Indicates where player 2 cannot remove opponent's chip

Indicates where player 2 will remove opponent's chip

Algorithm 1: Lowest Probability Decision for Rule of 10

if *dice_roll* $= 10$ *and opponente_chips on board* **then**

  for *each opponent_chip p on board* **do**

    $p \leftarrow prob(p)$;

  end

  Except where $p = 2$ and $p = 12$;

  $index \leftarrow min(p)$;

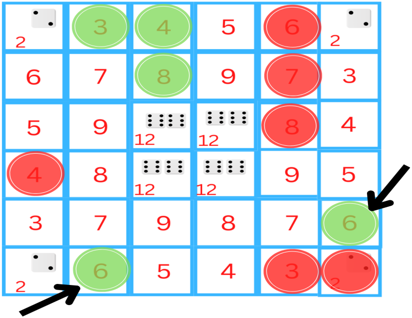
  $board[index].remove(chip)$;

  /* Remove opponent's chip from the cell with the lowest probability */

end

Methodology: Simulation – Decisions for rule 10 cont'd

- Highest Probability:

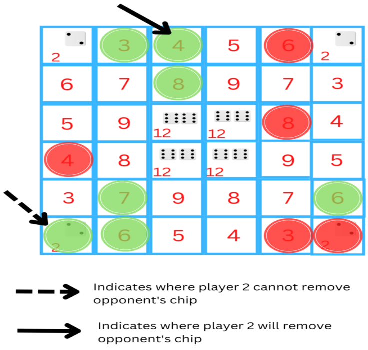


Algorithm 2: Highest Probability Decision for Rule of 10

```
if dice_roll = 10 and opponent_chip on board then
    for each opponent_chip p on board do
        p ← prob(p);
    end
    Except where p = 2 and p = 12;
    index ← max(p);
    board[index].remove(chip);
    /* Remove opponent's chip from the cell with the highest probability */
end
```

Methodology: Simulation – Decisions for rule 10 cont’d

- Random:

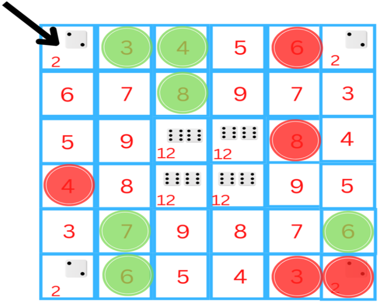


Algorithm 3: Random Decision for Rule of 10

if *dice_roll* = 10 **then**
  if *opponent_chip p on board* **then**
    Except where $p = 2$ and $p = 12$;
    $index \leftarrow random.choice(opponent\_chips)$;
    $board[index].remove(chip)$; /* Player removes opponent's chip randomly from the board */
  end
end

Methodology: Simulation – Decisions for rule 11

- Lowest Probability:



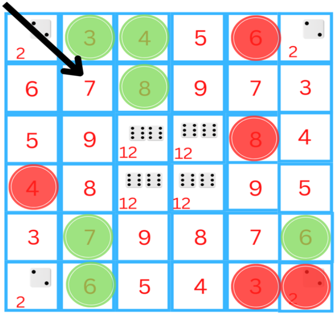
Indicates where player 2 will place thier chip

Algorithm 4: Lowest Probability Decision for Rule of 11

if *dice_roll = 11 and cell is not covered by chip on board* **then**
  for *each uncovered cell on board* **do**
    Compute probabilities of each uncovered cell on the board ;
    index ← *min(probability_of_uncovered_cell_on_board)*;
  end
  board[index] ← *own_chip* ; `/* Player places their chip on a cell with the lowest probability */`
end

Methodology: Simulation – Decisions for rule 11 cont’d

- Highest Probability:



→ Indicates where player 2 will place thier chip

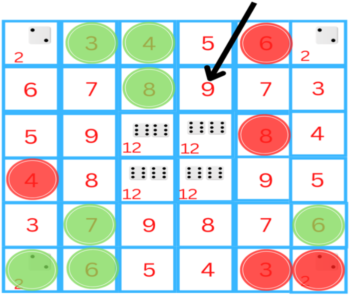
Algorithm 5: Highest Probability Decision for Rule of 11

---

if *dice_roll = 11 and cell is not covered by chip on board* **then**
    for *each uncovered cell on board* **do**
        Compute probabilities of each uncovered cell on the board ;
        index ← *max(probability_of_uncovered_cell_on_board)*;
    end
    board[index] ← *own_chip* ; `/* Player places their chip on a cell with the highest probability */`
end

Methodology: Simulation – Decisions for rule 11 cont'd

- Random:



→ Indicates where player 2 will place their chip

Algorithm 6: Random Decision for rule of 11

---

if *dice_roll = 11 and cell is not covered by chip on board* **then**
  $index \leftarrow random.choice(uncovered\_cell\_on\_board)$;
  $board[index] \leftarrow own\_chip$; `/* Player places their chip randomly on the board */`
end

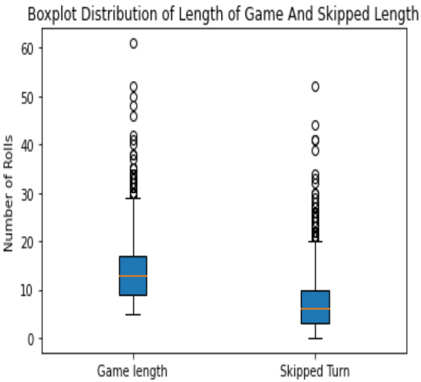
Methodology: Simulation – Variations

- One-dimensional board:
  - 1 player with simplified rules
  - 1 player with original rules – decisions for rule 10 and11
  - 2 players with simplified rules
  - 2 players with original rules – decisions for rule 10 and 11
- Full or standard board:
  - 1 player with simplified rules – strategies
  - 1 player with original rules – decisions and strategies
  - 2 players with simplified rules – strategies
  - 2 players with original ruels – decisions and strategies

Results: One-dimensional board

- One Player With Simplified Rules:

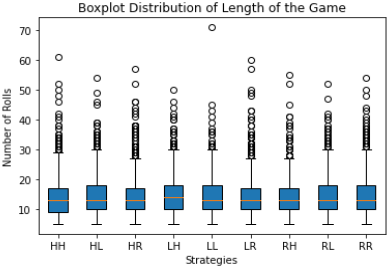
	Metrics	Value
1	Average Dice Rolls	14.29
2	Average Skipped Turns	7.87
3	Number of Wins	1000.00
4	Game Overs	0.00

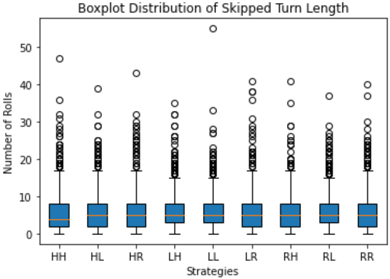


Results: One-dimensional board

- One Player With Original Rules:

	Decisions	Average Dice Rolls	Average Skipped Turns	Number of Wins	Game Overs
1	Highest-Highest	14.29	5.86	1000	0
2	Highest-Lowest	14.30	5.90	999	1
3	Highest-Random	14.30	5.85	1000	0
4	Lowest-Highest	14.43	6.23	1000	0
5	Lowest-Lowest	14.47	6.07	1000	0
6	Lowest-Random	14.43	5.78	1000	0
7	Random-Highest	14.37	5.66	1000	0
8	Random-Lowest	14.37	5.90	1000	0
9	Random-Random	14.39	6.02	999	1

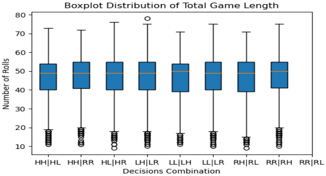


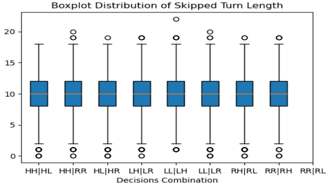


One-dimensional board : Two Players With Original Rules

- Results when both players used the same decisions for rule 10 and different decisions for rule 11:

	Decisions	Average game length	Average player one game length	Average player two game length	Player one win length	Player one % of wins	Player two win length	Player two % of wins	Player one skipped turn length	Player two skipped turn length	Average skipped turn	Draw length	% of draws
1	Highest - Highest|Highest - Lowest	46.35	23.47	22.89	204	20.4	205	20.5	6913	6753	13.67	591	59.1
2	Highest - Highest|Highest - Random	46.38	23.48	22.90	181	18.1	210	21.0	7068	6717	13.79	609	60.9
3	Highest - Lowest|Highest - Random	46.49	23.52	22.97	186	18.6	209	20.9	6883	6718	13.60	605	60.5
4	Lowest - Highest|Lowest - Random	46.42	23.48	22.94	212	21.2	203	20.3	7036	6776	13.81	585	58.5
5	Lowest - Lowest|Lowest - Highest	46.42	23.48	22.94	208	20.8	205	20.5	6969	6841	13.81	587	58.7
6	Lowest - Lowest|Lowest - Random	46.43	23.48	22.95	196	19.6	191	19.1	6887	6750	13.64	613	61.3
7	Random - Highest|Random - Lowest	46.46	23.50	22.96	179	17.9	186	18.6	7079	6916	13.99	635	63.5
8	Random - Random|Random - Highest	46.44	23.49	22.95	198	19.8	193	19.3	6951	6685	13.64	609	60.9
9	Random - Random|Random - Lowest	46.45	23.50	22.95	203	20.3	169	16.9	7218	7079	14.30	628	62.8



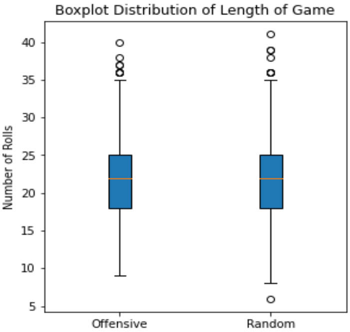


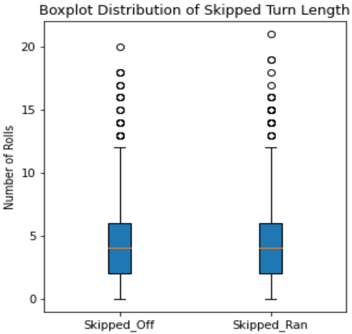
One-dimensional board: Two Players With Original Rules

- Results when both players used the same decisions for rules 10 and 11
- Results when a player used the same decisions for rules 10 and 11 while the opponent used a different but same decision for rules 10 and 11.

Full board : One Player With Simplified Rules

	Strategies	Average Dice Rolls	Number of Wins	% of Wins	Game Overs	% of Game Overs	Skipped Turns Count	Average Skipped Turns
1	Offensive	21.60	796	79.6	204	20.4	4590	4.59
2	Random	21.53	766	76.6	234	23.4	4449	4.45

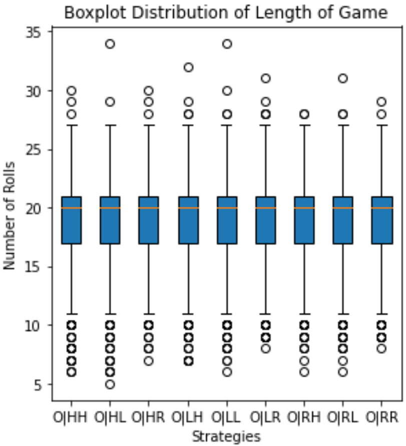


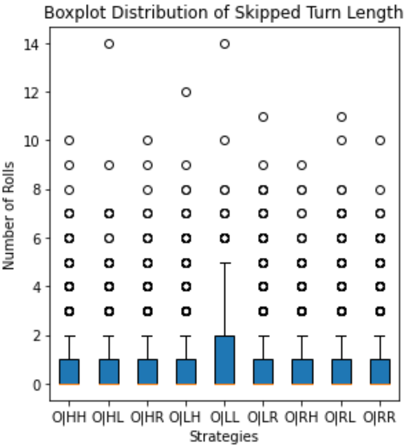


Full board : One Player With Original Rules

	Strategies	Average Game Length	Number of Wins	% of Wins	Skipped Count	Average Skipped Turn	Game Overs	% of Game Overs
1	Offensive-highest-highest	18.80	532	53.2	817	0.82	468	46.8
2	Offensive-highest-lowest	18.76	534	53.4	777	0.78	466	46.6
3	Offensive-highest-random	18.84	509	50.9	881	0.88	491	49.1
4	Offensive-lowest-highest	18.81	608	60.8	908	0.91	392	39.2
5	Offensive-lowest-lowest	18.82	577	57.7	1027	1.03	423	42.3
6	Offensive-lowest-random	18.86	556	55.6	983	0.98	444	44.4
7	Offensive-random-highest	18.86	556	55.6	889	0.89	444	44.4
8	Offensive-random-lowest	18.86	569	56.9	863	0.86	431	43.1
9	Offensive-random-random	18.87	531	53.1	902	0.90	469	46.9
10	Random-highest-highest	18.93	411	41.1	817	0.82	589	58.9
11	Random-highest-lowest	18.98	405	40.5	836	0.84	595	59.5
12	Random-highest-random	19.02	437	43.7	965	0.96	563	56.3
13	Random-lowest-highest	19.06	500	50.0	1123	1.12	500	50.0
14	Random-lowest-lowest	19.09	507	50.7	1049	1.05	493	49.3
15	Random-lowest-random	19.09	536	53.6	1066	1.07	464	46.4
16	Random-random-highest	19.11	459	45.9	921	0.92	541	54.1
17	Random-random-lowest	19.13	469	46.9	883	0.88	531	53.1
18	Random-random-random	19.14	465	46.5	848	0.85	535	53.5

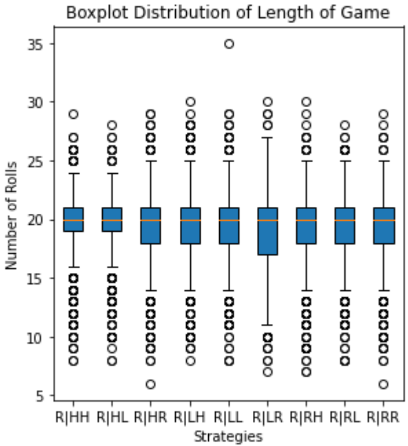
Full board : One Player With Original Rules Cont’d

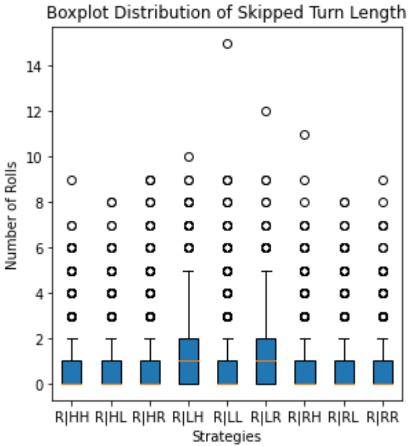




UNIVERSITÄT PASSAU

Full board : One Player With Original Rules Cont'd

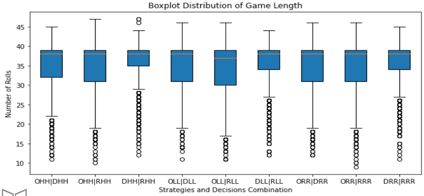


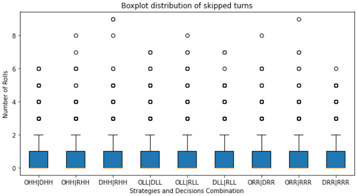


Full board : Two Player With Original Rules

- Results when both players used different placement strategies but the same decisions for rules 10 and 11

	Strategies and decisions	Average game length	Player one win count	Player one % of wins	Player two win count	Player two % of wins	Number of draws	% of draws	Average skipped turns	Player one skipped turn count	Player two skipped turn count
1	offensive - highest - highest|defensive - highest - highest	35	235	23.5	185	18.5	580	58.0	0.73	431	304
2	offensive - highest - highest|random - highest - highest	34	290	29.0	155	15.5	555	55.5	0.70	441	260
3	defensive - highest - highest|random - highest - highest	36	216	21.6	140	14.0	644	64.4	0.74	434	303
4	offensive - lowest - lowest|defensive - lowest - lowest	35	247	24.7	181	18.1	572	57.2	0.77	445	324
5	offensive - lowest - lowest|random - lowest - lowest	34	344	34.4	143	14.3	513	51.3	0.66	408	253
6	defensive - lowest - lowest|random - lowest - lowest	35	270	27.0	135	13.5	595	59.5	0.74	440	301
7	offensive - random - random|defensive - random - random	35	226	22.6	208	20.8	566	56.6	0.60	353	247
8	offensive - random - random|random - random - random	34	299	29.9	148	14.8	553	55.3	0.63	386	241
9	defensive - random - random|random - random - random	36	235	23.5	145	14.5	620	62.0	0.60	356	246



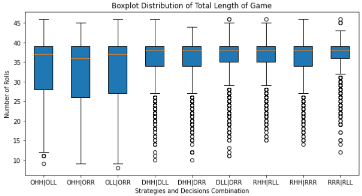


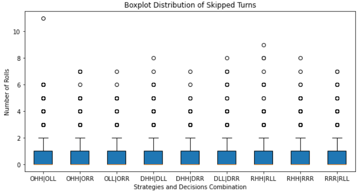
UNIVERSITÄT PASSAU

Full board : Two Player With Original Rules

- Results when both players used the same placement strategy but different fixed decisions for rules 10 and 11:

	Strategies and decisions	Average game length	Player one win count	Player one % of wins	Player two win count	Player two % of wins	Number of draws	% of draws	Average skipped turns	Player one skipped turn count	Player two skipped turn count
1	offensive - highest - highest|offensive - lowest - lowest	32	335	33.5	242	24.2	423	42.3	0.65	467	183
2	offensive - highest - highest|offensive - random - random	32	369	36.9	198	19.8	433	43.3	0.62	389	230
3	offensive - lowest - lowest|offensive - random - random	33	291	29.1	227	22.7	482	48.2	0.54	312	228
4	defensive - highest - highest|defensive - lowest - lowest	36	232	23.2	137	13.7	631	63.1	0.77	513	257
5	defensive - highest - highest|defensive - random - random	36	242	24.2	136	13.6	622	62.2	0.66	417	239
6	defensive - lowest - lowest|defensive - random - random	36	185	18.5	171	17.1	644	64.4	0.73	388	342
7	random - highest - highest|random - lowest - lowest	36	246	24.6	109	10.9	645	64.5	0.78	529	254
8	random - highest - highest|random - random - random	36	219	21.9	138	13.8	643	64.3	0.68	434	243
9	random - lowest - lowest|random - random - random	36	161	16.1	165	16.5	674	67.4	0.75	404	345



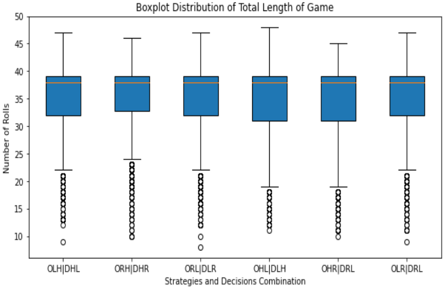


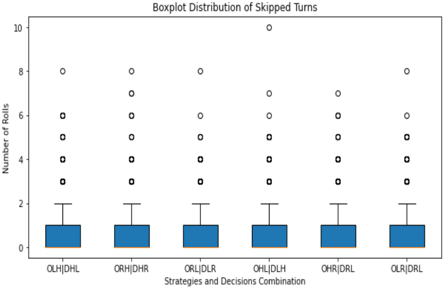
Full board : Two Player With Original Rules

- Results when player one used an offensive strategy with different decisions for rules 10 and 11 and when player two used a defensive strategy with different decisions for rules 10 and 11:

	Strategies and decisions	Average game length	Player one win count	Player one % of wins	Player two win count	Player two % of wins	Number of draws	% of draws	Average skipped turns	Player one skipped turn count	Player two skipped turn count
1	offensive - lowest - highest|defensive - highest - lowest	35	225	22.5	185	18.5	590	59.0	0.71	400	306
2	offensive - random - highest|defensive - highest - random	35	213	21.3	192	19.2	595	59.5	0.63	375	257
3	offensive - random - lowest|defensive - lowest - random	34	248	24.8	188	18.8	564	56.4	0.66	403	259
4	offensive - highest - lowest|defensive - lowest - highest	35	225	22.5	192	19.2	583	58.3	0.63	373	258
5	offensive - highest - random|defensive - random - highest	35	204	20.4	162	16.2	634	63.4	0.65	373	276
6	offensive - lowest - random|defensive - random - lowest	35	239	23.9	172	17.2	589	58.9	0.67	385	287

Full board : Two Player With Original Rules Cont’d





Full board : Two Player With Original Rules

- Results when player one used either an offensive or defensive strategy to play against player two who used a random strategy, with both players using the same different combination of decisions for rules 10 and 11.

- Results when player one used either an offensive or defensive strategy to play against player two who used a random strategy, with both players using different combinations of decisions for rules 10 and 11

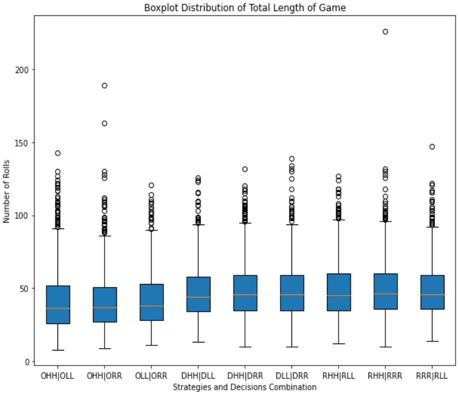
Full board : Two Players With Original Rules - Unlimited Number of Chips

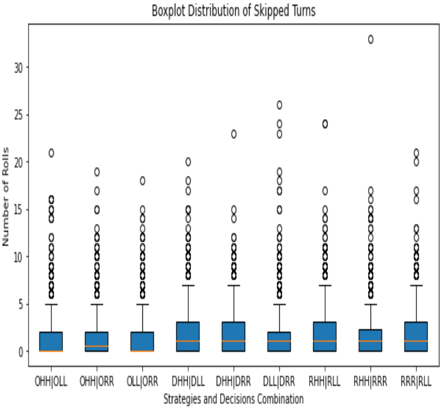
- Results when both players used the same placement strategy but different fixed decisions for rules 10 and 11 - unlimited chip:

	Strategies and decisions	Average game length	Average player one game length	Average player two game length	Player one win count	Player one % of wins	Player two win count	Player two % of wins	Average skipped turns	Player one skipped turn count	Player two skipped turn count
1	offensive - highest - highest|offensive - lowest - lowest	41.91	21.23	20.68	575	57.5	425	42.5	1.47	894	488
2	offensive - highest - highest|offensive - random - random	41.41	21.01	20.40	603	60.3	397	39.7	1.46	844	663
3	offensive - lowest - lowest|offensive - random - random	44.30	22.44	21.86	537	53.7	463	46.3	1.62	625	706
4	defensive - highest - highest|defensive - lowest - lowest	44.43	22.50	21.93	592	59.2	408	40.8	1.62	1226	695
5	defensive - highest - highest|defensive - random - random	44.50	22.54	21.96	577	57.7	423	42.3	1.62	1089	779
6	defensive - lowest - lowest|defensive - random - random	45.50	23.04	22.46	548	54.8	452	45.2	1.69	891	933
7	random - highest - highest|random - lowest - lowest	46.13	23.35	22.78	605	60.5	395	39.5	1.73	1295	783
8	random - highest - highest|random - random - random	46.16	23.37	22.80	550	55.0	450	45.0	1.73	1084	811
9	random - lowest - lowest|random - random - random	46.55	23.56	23.00	543	54.3	457	45.7	1.76	924	926

Full board : Two Players With Original Rules - Unlimited Number of Chips

- Results when both players used the same placement strategy but different fixed decisions for rules 10 and 11 - unlimited chip:





Discussion: Comparison between the simulation and theoretical results

- The shortest and easiest achievable sequence of 5 is the sequence of subset S4 = 5, 6, 7, 8, 9, requiring an expected number of 19 rolls.

- Simulation results showed an average game length of 14 rolls, implying that the expected time of the simulation is less than the minimum expected time of the theoretical result.

- Similar findings were reported in Monte Carlo simulation of Tic-Tac-Toe and in a game of Backgammon.

Effects of strategies and decisions on the game length

- 1-player, 1-Dimension board with original rules: Decision rules had no significant effect on the average game length.
- 2-player, 1-dimensional board with original rules: Average game length is 46-47 rolls - minimal impact of decision rules.
- 1-player, full board with simplified rules: Average game length was 22 rolls: No difference in the offensive and random- No strategy impact.
- 1-player, full board with original rules: offensive-lowest-highest had best result (18.81 game length, 60.8% win rate) while random-random-random had worst (19.14 game length, 46.5% win rate).
- 2-player settings, 1-dimension and full board with simplified rules ended in deadlock.
- 2-player, full board with original rules: offensive strategy had highest win rate and average game length of 32-34, defensive strategy had 34-36, and random strategy had 35-37.

Discussion: Effects of number of chips on the game length

- Two-player full board sequence dice game with rules variation, game ended in a draw 423-720 times with a limit of 20 chips.

- Number of chips have a significant effect on game length
  - Average game length was 32-37 rolls with 20 chips.
  - 41-47 rolls with unlimited chips

Conclusion

- Evaluated expected length of sequence dice game on one-dimensional board and full board.
- Used coupon collector model to theoretically simplify the playing board.
- Analysed expected time it would take to win the game with a restricted set of rules.
- Proposed and implemented three different kinds of decisions for rules 10 and 11 and three placement strategies.
- Evaluated effects of rules decisions and strategies on expected length of game.
- Offensive strategy outperformed defensive and random strategies.
- Expected length of one-dimensional board version of game when two players applied original rules is higher than expected length of full board version.
- Average game length is approximately 32-37 rolls when two players competed with 20 chips and applied original rules, and 42-47 rolls when number of chips is unlimited.

Future Work

- Theoretical study to examine the expected length of the sequence dice game in different variations.
- Study to investigate the expected length of the game in both multiplayer (more than 2 players) and team settings.
- Study to determine the actual number of chips that can be used by two players for the game to not end in a draw.

Thank you for your attention

A digital version of this presentation can be found here:

https://github.com/olalekanlasisi/sequence-dice-game/Sequence-Dice-Game-Presentation.pdf