International Journal of Mathematics Trends and Technology
ISSN: 2231 – 5373 /doi:10.14445/22315373/IJMTT-V67I1P507

Volume 67 Issue 1, 44-52, January, 2021

Markov Process of Snake And Ladders Board

Zin Mar Tun,

Associate Professor, Department of Mathematics, Mandalar University, Myanmar

Abstract – *We considered Snakes and Ladders Board as the state of Markov chain by looking at the long run behavior of transition. Necessary definitions are described in this paper. And then we derived transition matrix. Finally we presented the probability distribution of Snakes and Ladders game with figures based on the Markov chain.*

Key words: *Markov chain, transition matrix, absorbing state, probability distribution, stochastic process.*

I. INTRODUCTION

Snakes and Ladders board is an ancient Indian board game regarded today as a worldwide classic. It is popular among kids. We study the game mathematically by using basic theory on Markov chain. Markov chain is an important class of probability models. In Snakes and Ladders board, the object of the game is to roll a die and move that many squares up the board until we reach the 100th square and the game is won. The fate of our next term (next 'stats') is only dependent on our current roll (current 'state'). In this paper, we discuss the important definitions on discrete time Markov chain. Finally we constructed transition matrix of order 101 by 101.

II. STOCHASTIC PROCESSES

Definition 2.1 A stochastic process is a family of random variables X_t, where t is a parameter running over a suitable index set T={0, 1, 2,…}. The index t corresponds to discrete units of time and the index set is T. Stochastic processes are distinguished by their state space, or by the range of possible value for the random variable X_t. By their index set T, and by the dependence relations among the random variables X_t. [1]

Definition 2.2 A Markov process (X_t) is a stochastic process with the property that, given the value of X_t, the values of X_s for s > t are not influenced by the values of X_u for u< t. That is, the probability of any particular future behavior of the process, when its current state is known exactly, is not altered by additional knowledge concerning its past behavior. A discrete-time Markov chain is a Markov process whose state space is a finite or countable set, and whose (time) index set is T. In formal terms, the Markov property is that

$$P_r\{X_{n+1} = j|X_0 = i_0, \dots , X_{n-1} = i_{n-1}, X_n = i\} = P_r\{X_{n+1} = j|X_n = i\}.$$

The probability of X_{n+1} being in state j given that X_n is in state i is called the one-step transition probability and is denoted by $P_{ij}^{n,n+1}$. That is

$$P_{ij}^{n,n+1} = P_r\{X_{n+1} = j|X_n = i\}.$$

The notation emphasizes that the transition probabilities are functions not only of the initial and final states but also of the time of transition as well. When the one-step transition probabilities are independent of the time variable n, the Markov chain is said to have stationary transition probabilities.

Then $P_{ij}^{n,n+1} = P_{ij}$ is independent of n, and P_{ij} is the conditional probability that the state value undergoes a transition from i to j in one trial. It is customary to arrange these numbers P_{ij} in a square matrix,

$$P = \begin{pmatrix} P_{00} & P_{01} & P_{02} & P_{03} & \cdots \\ P_{10} & P_{11} & P_{12} & P_{13} & \cdots \\ P_{20} & P_{21} & P_{22} & P_{23} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ P_{i0} & P_{i1} & P_{i2} & P_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

and refer to $\mathrm{P} = P_{ij}$ as the Markov matrix or transition probability matrix of the process.

The i^{th} row of P, for $i = 0, 1, ...$, is the probability distribution of the values of X_{n+1} under the condition that $X_n = i$. If the number of states is finite, then P is a square matrix whose order is equal to the number of states. The quantities P_{ij} satisfy the following

$$P_{ij} \geq 0 \text{ for all } i, j = 0, 1, 2, ..., \sum_{j=0}^{\infty} P_{ij} = 1.$$

A Markov process is completely defined once its transition probability matrix and the probability distribution of the initial state X_0 are specified. [1]

Definition 2.3 An absorbing state is one in which, when entered, it is impossible to leave. A state that is not an absorbing state is called a transient state. An absorbing Maokov chain is a Markov chain with absorbing states and with the property that it is possible to absorbing states and with the property that it is possible to transition from any state to absorbing stats in a finite number of transitions.[1]

III. SNAKES and LADDERS GAME'S RULE

A representation of the board is shown in Figure 1. At various locations on the board are placed Snakes and Ladders, each of which connects a pair of sequences. The game of Snakes and Ladders is played on a board with a 10×10 grid, numbered sequentially in a zigzag pattern from 1(the start, in the lower left corner) to 100(the end, in the top left corner). A player start off the board and take turns rolling a single die and moving the corresponding number of squares. If the completion of a move lands you on the head of the Snake, you must go back to the square at the Snake tail. If you land at the bottom of a Ladder, you instantly climb to the top of that Ladder. For example, a player lands on squares 97, they move back to square 58, losing 39 positions.

However, if a player lands on square 8 upon first roll, they instantaneously transition to state 30 gaining 22 squares. If a player rolls a die that would advance them beyond square100, they stay at the same place.

The player must land exactly on square 100 to win.

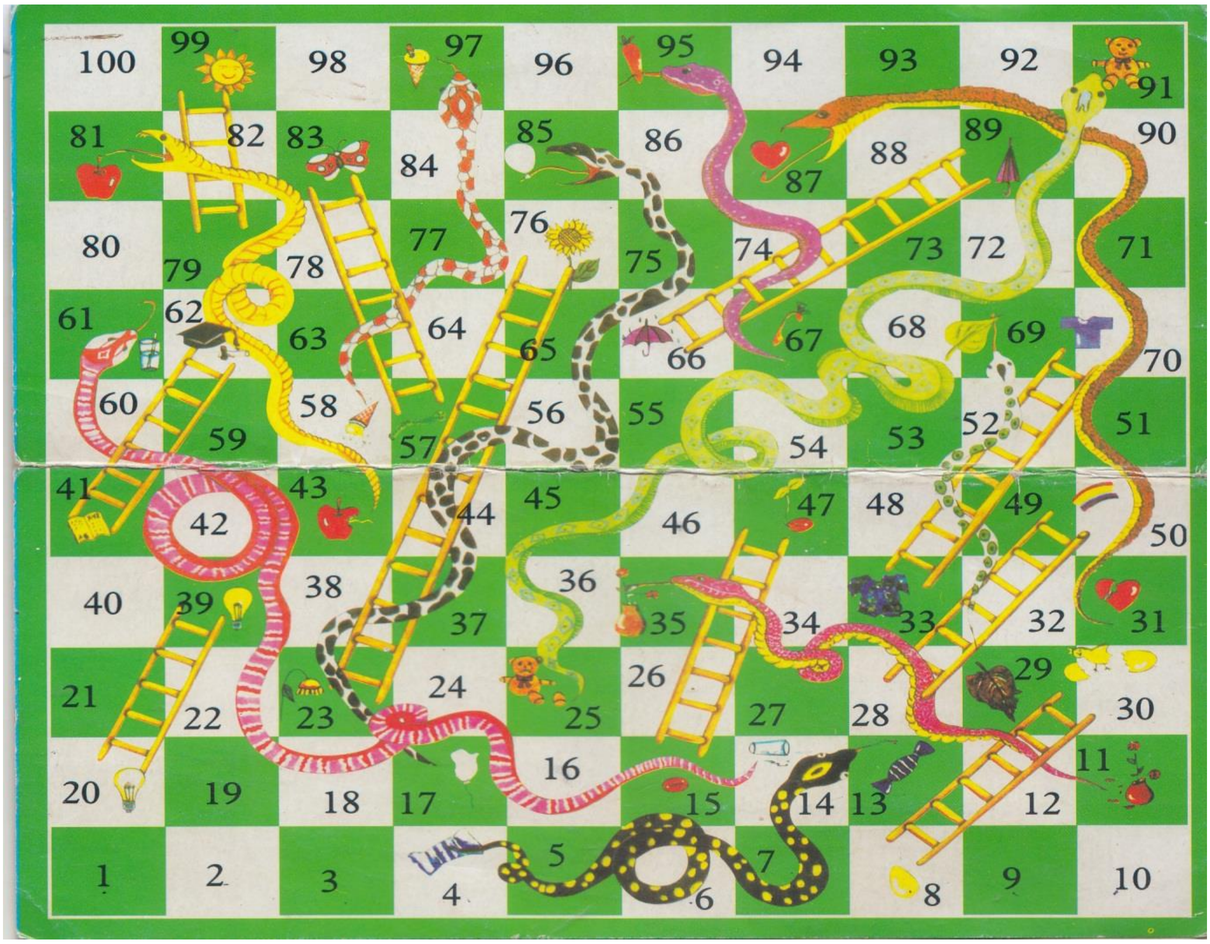

Fig. 1 A standard Snakes and Ladders

IV. MARKOV PROCESS of SNAKES and LADDERS GAME'S

Snakes and Ladders can be represented as an absorbing Markov chain, where each square is a state, with the final square of 100 as the only absorbing state. This absorbing state is recurrent while the remaining 99 states are transient. Given the rule of the game and the board layout, we can create a transition matrix that shows the probability of moving from one transient state to any other state. A sequence of probabilistic transitions from one state to another with no memory of previous states is known as a Markov chain or Markov process. Furthermore, this Markov chain is a regular chain with a limiting probability distribution given by

$$\pi_i = (0, 0, \dots, 1) \quad \text{for states } \ i = 0, 1, \dots, n.$$

The standard board of the game can be denoted as R, representing all of the transitions. Specifically

R= {(8, 30), (15, 47), (20, 39), (23, 76), (28, 50), (33, 70), (41, 62), (57, 83), (66, 89),
(79, 99), (13, 4), (35, 11), (61, 14), (69, 32), (81, 43), (85, 17), (87, 31), (91, 25),
(95, 67), (97, 58)}.

V. TRANSITION MATRIX for SNAKES and LADDERS

On any given turn, there are six equally probable options: rolling a 1, 2, 3, 4, 5 or 6. Depending on which space you start on, these leads to six well-defined results. For example, the first turn, the possibilities are the sequences 1, 2, 3, 4, 5 or 6, each with equal probability. We could encode this set of probabilities as a vector of length 101, with $\frac{1}{6}$ in each associated index (where the square zero represents the start of the game, off of the board):

0: $[0, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, 0, \ldots, 0]$.

Each entry in this vector is the probability of going from square zero to the corresponding square. This vector completely describes the first turn of the game.

Similarly, we could construct the vector describing turns from square 1:

1: $[0, 0, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, 0, \ldots, 0]$.

We can be entirely described by a $N \times N$ transition matrix where N is the number of states in the system. The columns of the matrix contain the probability vectors we discussed above, such that element (n, m) is the probability of transitioning to element n when starting from element m.

To apply the theory of Markov chains, we can define the transition matrix. Since figurative square zero is a state, the first row in the transition matrix is the row representing state zero. Below is a summary of a transition matrix of order 101 by 101.

$$
\begin{array}{c c}
 & \begin{array}{cccccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \cdots & 35 & \cdots & 61 & \cdots & & 97 & 98 & 99 & 100 \end{array} \\
P_{101\times 101} = \begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ \vdots \\ 97 \\ 98 \\ 99 \\ 100 \end{array} &
\begin{pmatrix}
0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \frac{4}{6} & \frac{1}{6} & \frac{1}{6} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{5}{6} & \frac{1}{6} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

It is important to note that the states representing the bottom of Ladders or the head of the Snakes have zero probability associated with them, since upon landing there, the player instantaneously transitions to another state. Thus, in order to be considered a proper transition matrix, the matrix needs to be reduced to a 81×81dimension. Computationally, though, the two would lead to the same results. Moreover, defining a native transition matrix of 101×101 makes interpretation of state easier, as they match the game board squares.

```
function snakes_probability ( )
  timestamp ( )
  N = 100;P = toeplitz(zeros(1,N+1),[0 ones(1,6) zeros(1,N-6)]);
  for k = N - 4 : N + 1
    P(k,k) = k - N + 5;
  end
  P = P / 6;
  top = [ 8  15  20 23 28 33 41 57 66 79] + 1;
  bot = [13 35 61  69 81 85 87 91 95 97] + 1;
  for k = 1 : length(top)
    r = top(k);
    s = bot(k);      % Chute or ladder from r to s.
    P(:,s) = P(:,s) + P(:,r);    % Add column r to column s.
  end
  P(top,:) = []; P(:,top) = [];
  figure ( 2 )
  grid on;
  spy ( P )
    xlabel ( 'number of turns' ); ylabel ( 'fraction of game completed' );
  filename = 'snakes_transition.png';
  print ( '-dpng', filename );
  M = 100;
  cumprob = zeros ( M, 1 );
  v = P(1,:);
  for n = 2 : M,
    v = v * P;
    cumprob(n) = v(end);
  end
  return
  end
  function timestamp ( )
   t = now;
   c = datevec ( t );
   s = datestr ( c, 0 );
   fprintf ( 1, '%s\n', s );
   return
  end
```

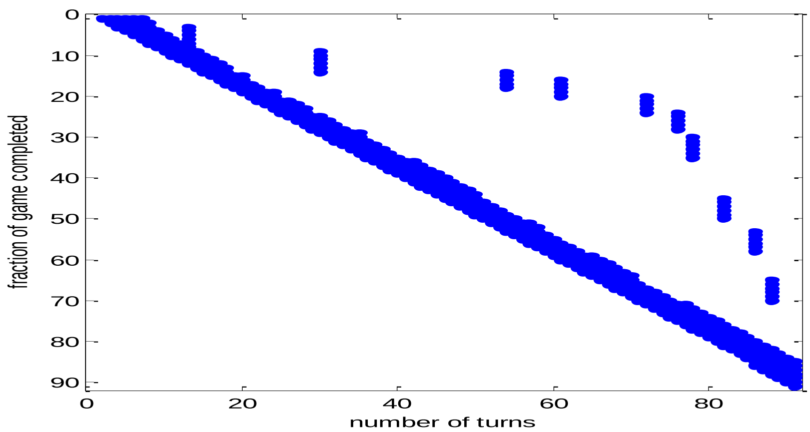

Fig. 2 Transition matrix for Snakes and Ladders.

```
function snakes_probability ( )
 N = 100;   P = toeplitz(zeros(1,N+1),[0 ones(1,6) zeros(1,N-6)]);
```

```
    for k = N - 4 : N + 1
      P(k,k) = k - N + 5;
    end
    P = P / 6;
    top = [ 8  15  20 23 28 33 41 57 66 79] + 1;
   bot = [13 35 61  69 81 85 87 91 95 97] + 1;
    for k = 1 : length(top)
      r = top(k);   s = bot(k);      % Chute or ladder from r to s.
      P(:,s) = P(:,s) + P(:,r);    % Add column r to column s.
    end
    P(top,:) = [];  P(:,top) = [];  M = 100;
    cumprob = zeros ( M, 1 );    v = P(1,:);
    for n = 2 : M,
      v = v * P;    cumprob(n) = v(end);
    end
    figure ( 3 )
    colormap ( [0.8,0.4,0.4] )
    bar ( diff ( [0;cumprob] ) )
    xlabel ( 'number of turns', 'FontSize', 12, 'FontWeight', 'Bold' );
    ylabel ( 'fraction of game completed', 'FontSize', 12, 'FontWeight', 'Bold' );
    xlim ( [ 0 M ] )
    filename = 'snakes_gamelength_pdf.png'; print ( '-dpng', filename );
    return
  end
  function timestamp ( )
    t = now;   c = datevec ( t );    s = datestr ( c, 0 );
    fprintf ( 1, '%s\n', s );
    return
  end
```

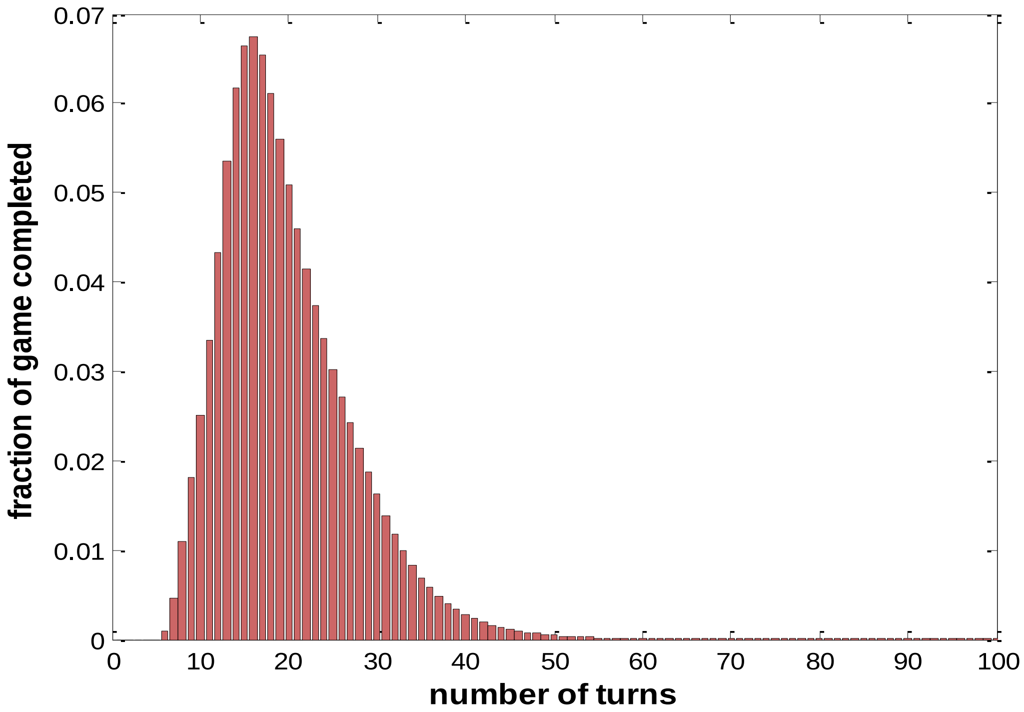

Fig. 3 Probability for Game Length.

```
function snakes_probability ( )
  N = 100;P = toeplitz(zeros(1,N+1),[0 ones(1,6) zeros(1,N-6)]);
  for k = N - 4 : N + 1
    P(k,k) = k - N + 5;
  end
  P = P / 6;
  top = [ 8  15  20 23 28 33 41 57 66 79] + 1;
  bot = [13 35 61  69 81 85 87 91 95 97] + 1;
  for k = 1 : length(top)
    r = top(k);    s = bot(k);      % Chute or ladder from r to s.
    P(:,s) = P(:,s) + P(:,r);       % Add column r to column s.
  end
  P(top,:) = [];    P(:,top) = [];
  M = 100;cumprob = zeros ( M, 1 );   v = P(1,:);
  for n = 2 : M,
    v = v * P;   cumprob(n) = v(end);
  end
  figure (4)
  colormap ( [0.8,0.4,0.4] )
  bar ( cumprob )
  xlabel ( 'number of turns', 'FontSize', 12, 'FontWeight', 'Bold' );
  ylabel ( 'fraction of game completed', 'FontSize', 12, 'FontWeight', 'Bold' );
  xlim ( [0 M] )
  filename = 'snakes_gamelength_cdf.png'; print ( '-dpng', filename );
  return
end
function timestamp ( )
  t = now;    c = datevec ( t );    s = datestr ( c, 0 );
  fprintf ( 1, '%s\n', s );
  return
end
```

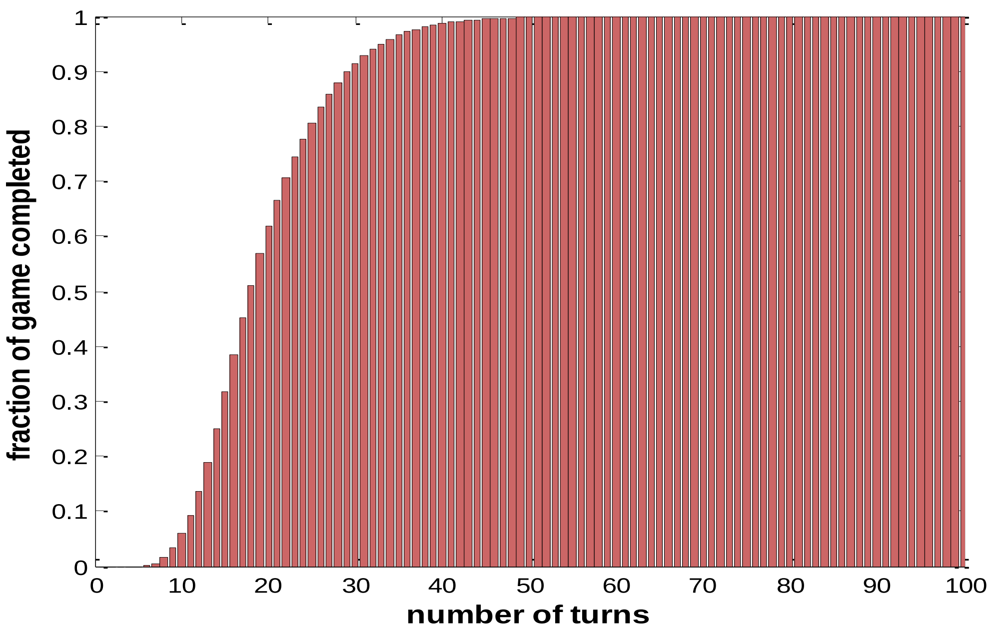

Fig.4 Cumulative Probability for Game Length

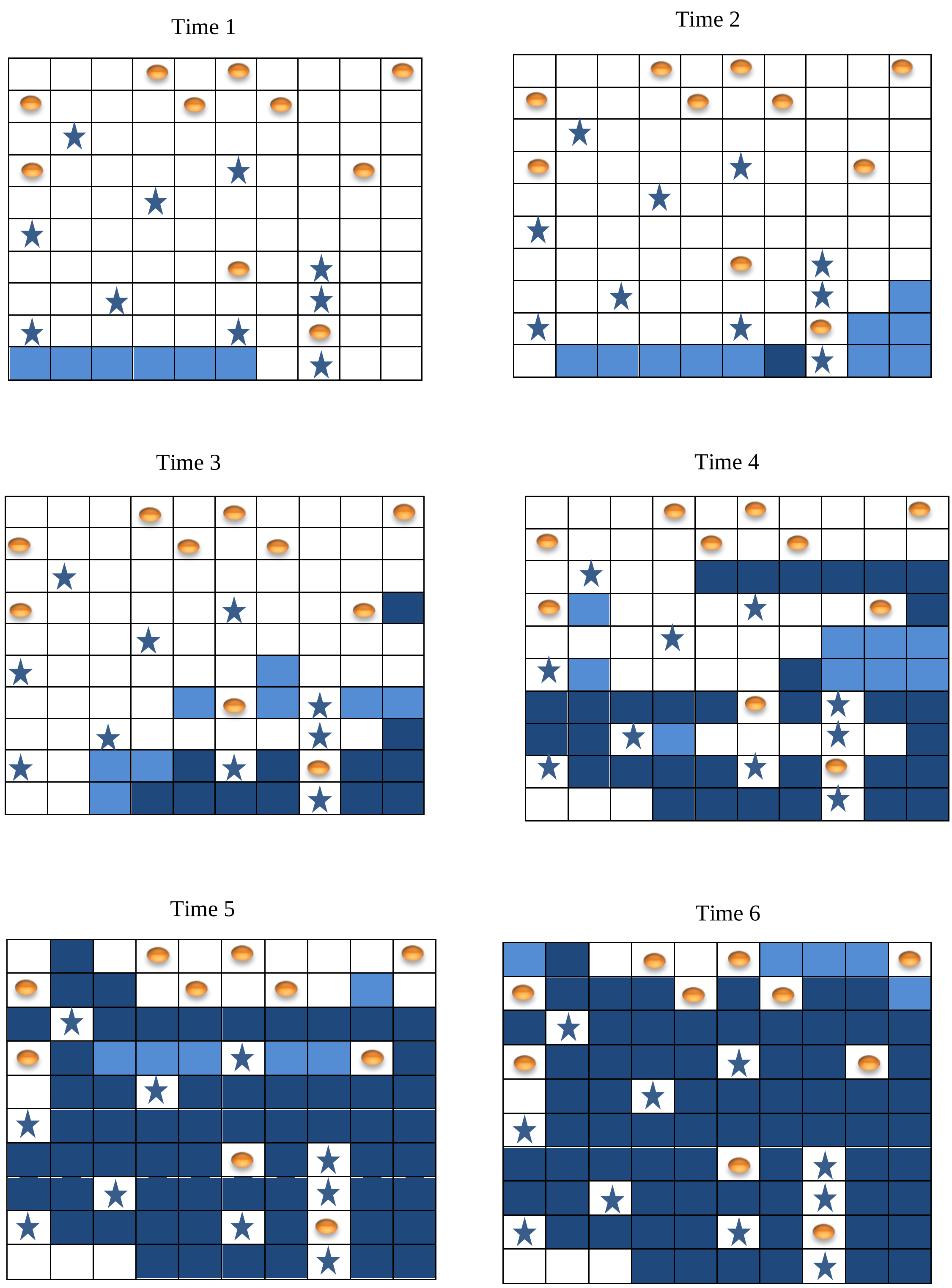

Fig. 5 The probability distribution over all 100 states for times 1 to 6. Times denote turns, or rolls of the die. The eggs and the stars represent the Snakes and Ladders on the board while the shaded squares represent a player lands at that state at the specific time.

VI. RESULTS

At time 1, the process is distributed over states 1, 2, 3, 4, 5 and 6. Specifically, the probability of being in the transient states gets smaller (fainter squares), while the probability of being at the transient states gets larger (darker squares). We drive the minimum number of turns needed to reach state 100 is 6. The probability of the game over at time 6, however, is very low. We analyze how the Markov chain behaves over time for a single player shown in Figure 5.

VII. CONCLUSIONS

According to the rules of this game, when a player is at the bottom of a Ladder, he will automatically at the top. Therefore, that time is like the time when is full of luck in life. As life is not always full of good things, in this game, when a player land on a Snake's head, he will be swallowed. That situation in the game can be compared with the time when a person is the most difficult situation in life, faces depressing experience and his life is ruined. Snakes and Ladders game can be considered and compared with a life which a person must break through. Throughout one's life, it is moving along with full of ups and downs in accordance with fortune and misfortune. Honestly, I would like everyone to notice the above-mentioned facts. In brief, I should like to conclude by presenting this interesting and beneficial paper which can be thought from the point of view of Mathematics and life as well.

ACKNOWLEDGEMENTS

I am deeply thankful to Dr. Thet Khine, Head and Professor of Mandalar University, who has given me a great help for preparing this paper. Finally, I would like to express my warm gratitude to my friends, involved in the successful compilation.

REFERENCES

[1] L.A. Cheteyan, S. Hengeveld, M,A,Jones, and "Chutes and Ladders for the impatient", vol.42, No1, the mathematical association of America, The College Mathematics Journal, January (2011).

[2] M. ROSS,S., "Introduction to Probability Model", 10td Edition, India, (2010).