UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

Facultad de Ciencias Físico Matemáticas

Licenciatura en Ciencias Computacionales

INVESTIGACIÓN DE OPERACIONES

“EXAMEN #3 / PROYECTO FINAL INTEGRADOR”

Maestro: Luis Ángel Gutiérrez Rodríguez **Equipo #10**

Nombre	**Matrícula**
Fuentes Sánchez Jared Evander	1868002
Terrazas Santillana Sebastian	1847317
Garza Reyna Vicente	1847176

San Nicolás de los Garza, Nuevo León el 07 de Mayo de 2021

ÍNDICE

Problema

The Game of Pig

The Game of Pig trata de un juego bastante sencillo de un solo dado, en el cual, dos jugadores deben llegar a 100 puntos. Cada turno, un jugador repetidamente tira un dado hasta que obtenga un 1 o dice parar y anotar la suma de las tiradas (el total del turno). En cualquier momento durante el turno de un jugador, se le presentará dos decisiones:

- **Tirar el dado**: si el jugador obtiene:
    - **1**: el jugador no consigue ningún punto y pasa automáticamente al turno del oponente.
    - **2 - 6**: el número es añadido al turno total al jugador y continua el mismo jugador.
- **Parar**: El total del turno es añadido al puntaje del jugador y pasamos el turno al oponente.

Objetivos a cumplir

(a) Expresar el juego como una matriz que representa una cadena de Markov.
(b) Utilizar la cadena de Markov para intentar encontrar una estrategia que nos permita tener mejores probabilidades de ganar contra otro jugador.

Consideraciones

Con el fin de simplificar el modelo, nos tomamos unas cuantas libertades durante la simulación del juego:

1. En primer lugar, consideramos The Game of Pig con un solo jugador, esto permite analizar un juego sencillo y evita que requiramos expandir la cantidad de estados posibles del juego. Esto es importante porque entre más estados necesitemos, será necesario también “lanzar más dados” si queremos que las probabilidades de distribuyan tan bien como lo harían en la teoría.
2. En caso de sobrepasar el puntaje objetivo (en este caso 100), se considerará que llegamos exactamente al estado objetivo y se reiniciará la cuenta desde 0. Esto también ayuda con la reducción de estados posibles.
3. Ya que el juego termina cuando se alcanza el puntaje objetivo, se considera que la probabilidad de llegar de ese estado a cualquier otro como 0. Por esto mismo, para evitar que no haya filas de la matriz cuya suma no de 1, tenemos que obligar que la probabilidad de llegar del puntaje objetivo a sí mismo es 1. El lector puede pensar en esto como: “No importa cuantas veces lanzamos el dado después de haber ganado, porque ya ganamos”.
4. Se realizaron 100 millones de “lanzamientos del dado”.

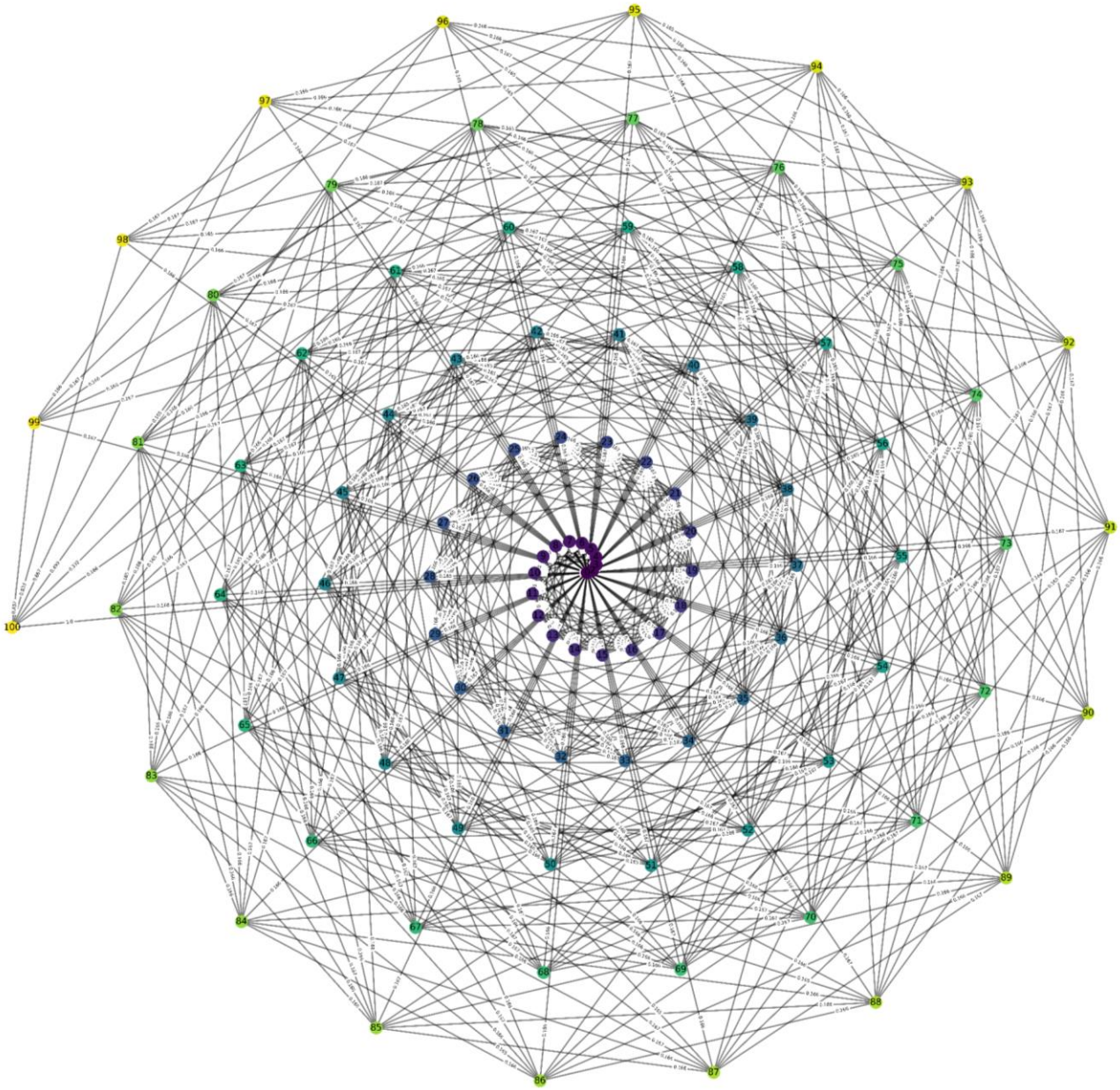
Solución

a) Cadena de Markov

I. Matriz de Transición

Debido al tamaño de la matriz, sería poco viable colocar todos los datos en este documento, así que, por favor, busque dentro del repositorio git el archivo “Matriz_Transicion.xlsx”, para revisar a detalle las probabilidades.

II. Grafo con porcentajes



Para mejor la visualización, se decidió acomodar el grafo en forma de espiral, donde las puntuaciones de menor valor están más cerca del centro.
Para generar el grafo y revisarlo con más detalle, use el archivo “matrix.csv” y el script “PlotMarkovChain.py”; también será necesario tener instalado los paquetes NumPy, NetworkX y Matplotlib.

III. Estado de Equilibrio

Podemos encontrar un tipo de estado de equilibrio: Esto ocurre en el estado 100, en este caso, la probabilidad de llegar a 0 es 0%, al igual que en cualquier otro estado diferente del estado 100, en cuyo caso es 100% (como ya habíamos mencionado en la sección de consideraciones).
En otras palabras, si tenemos la matriz de probabilidades $P \in M_{100\times100}$ con 100 renglones $r_i: i \in 1,2,\cdots,100$ y 100 columnas $c_j: j \in 1,2,\cdots,100$ podemos que ver que r_1 y c_{100} son constantes para P^n con $n \in N$. Y por lo tanto son estados de equilibrio.

IV. Clasificación de cada estado

En este caso, aunque The Game of Pig es relativamente sencillo, no tiene muchos estados que puedan ser clasificados de forma sencilla. Esto se debe a que prácticamente se puede ir de cualquier estado (excepto el 100) de vuelta a 0 y puedes ir de 0 a cualquier otro estado. El único estado que presenta una clasificación simple es el estado 100, el cual es absorbente. Por otra parte, el estado 0 es prácticamente recurrente, excepto porque en caso de estar en el estado 100, la probabilidad de volver al estado 0 es de 0%.

V. Estimación calculando el comportamiento de la cadena en un $T > 5$

En primer lugar, podemos observar que las probabilidades de obtener 100 puntos (empezando de 0 puntos) en un lanzamiento son de 0 hasta el 16to lanzamiento y empiezan a ser mayores (aunque también increíblemente pequeñas, en el orden de 10^{-11}) desde el lanzamiento 17. Podemos comprobar teóricamente que esto corresponde con la realidad.

También podemos encontrar que las probabilidades de obtener 0 disminuyen con la cantidad de lanzamientos, aunque esto no empieza a ser más o menos evidente hasta el lanzamiento 22. Esto se debe a que entre mayor sea la probabilidad de llegar a 100, más disminuye la probabilidad de poder volver a 0, ya que 100 es un estado absorbente, como mencionamos anteriormente.

En el repositorio, puede encontrar el archivo “Matrix_Power_10”, puede encontrar más detalles de este comportamiento. Por si no quedó claro con el nombre, a nuestra matriz de transición la elevamos a la décima potencia.

b) Estrategia propuesta

Una forma de aproximar el juego de una forma probabilística con la que podemos encontrar una estrategia que nos permita determinar cuándo dejar de lanzar el dado y guardar nuestra puntuación es la siguiente:

1. En primer lugar, se supone que tenemos un puntaje $x_t \in 0,2,3,\ldots,99$, o sea que en el momento t nos encontramos en el estado x_t.
2. Proseguimos a calcular la esperanza de lanzar de nuevo el dado, esto nos sirve para saber cuántos puntos esperar en $t+1$. Para calcular la esperanza tenemos:

   $$P_{x_t,0} \times 0 + P_{x_t,x_t+2} \times (x_t + 2) + P_{x_t,x_t+3} \times (x_t + 3) + \cdots + P_{x_t,x_t+6} \times (x_t + 6)$$

   Pero, ya que la probabilidad de llegar a cualquier otro de esos estados es $1/6$, por ser el resultado de un dado justo, podemos simplificar la esperanza a: $X_{t+1} = 1/6 \times (5x_t + 20)$.
3. En este caso queremos minimizar la diferencia entre el x_t y la esperanza en $t+1$ (esto es X_{t+1}). Tenemos entonces: $Min: |x_t - X_{t+1}|$. Pero el menor valor que puede tomar el valor absoluto de cualquier número es 0, así que podemos usar la siguiente ecuación: $|x_t - X_{t+1}| = 0 \rightarrow |x_t - 1/6 \times (5x_t + 20)| = 0$. Aplicando las propiedades del valor absoluto y resolviendo conforfe a $x_t$obtenemos que:

   $$|x_t - 1/6 \times (5x_t + 20)| = 0$$
   $$x_t - 1/6 \times (5x_t + 20) = 0$$
   $$x_t - 5/6 \times x_t - 20/6 = 0$$
   $$x_t(1 - 5/6) = 20/6$$
   $$x_t(1/6) = 20/6$$
   $$x_t = 20$$

   Con esto podemos concluir de forma teórica que el mejor momento para terminar tu turno cuando se juega The Game of Pig es al obtener un mínimo de 20 puntos. Esto se puede verificar con el programa “GetExpectancies.py”, encontrado en el repositorio.

En el caso de jugar contra un segundo jugador, puede ser conveniente no apegarse tanto a la “regla de 20” en ciertos casos, sobre todo en caso de que se tenga una ventaja o desventaja considerable. En situación de desventaja, es mejor arriesgar más mientras que en casos de tener la ventaja la estrategia puede cambiar de varias formas. Uno puede escoger ir a la segura, o si así lo desea puede intentar arriesgarse e ir directamente por la victoria. Otra estrategia propuesta, es tomar una ventaja que incentive al oponente a que arriesgue. Esto, sin embargo, es más difícil de modelar y no está incluido en el alcance de este proyecto.

Link para repositorio Git

https://github.com/VSJ-Operations-Research-Team/PIA.git