

# IIC3675: Tarea 2

Bruno Cerdá Mardini

a)

P.d.q:  $v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \cdot q_\pi(s, a)$

$$\begin{aligned}
 v_\pi(s) &\doteq \mathbb{E}_\pi[G_t | S_t = s] \\
 &= \mathbb{E}_\pi[\mathbb{E}_\pi[G_t | S_t = s, A_t]] && \text{(Esperanzas iteradas, notar que } S_t = s \text{ no es una variable aleatoria, ya que tiene asignado un valor.)} \\
 &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) \cdot \mathbb{E}_\pi[G_t | S_t = s, A_t = a] && \text{(Definición de esperanza)} \\
 &= \sum_{a \in \mathcal{A}(s)} \pi(a|s) \cdot q_\pi(s, a) && \text{(Definición de } q_\pi(s, a) \text{)}
 \end{aligned}$$

■

b)

Parto la demostración desde el otro lado. P.d.q:

$$\begin{aligned}
 \sum_{s',r} p(s', r | s, a) (r + \gamma v_\pi(s')) &= q_\pi(s, a) \\
 \sum_{s',r} p(s', r | s, a) (r + \gamma v_\pi(s')) &= \sum_{s',r} p(s', r | s, a) \cdot r + \sum_{s',r} p(s', r | s, a) \cdot \gamma v_\pi(s') \\
 &= \mathbb{E}_\pi[R_{t+1} | s, a] + \gamma \sum_{s',r} p(s', r | s, a) \cdot v_\pi(s') && \text{(Definición de esperanza)} \\
 &= \mathbb{E}_\pi[R_{t+1} | s, a] + \gamma \sum_{s',r} p(s', r | s, a) \cdot \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] && \text{(Definición de } v_\pi(s) \text{)} \\
 &= \mathbb{E}_\pi[R_{t+1} | s, a] + \gamma \mathbb{E}_\pi[\mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] | s, a] && \text{(Definición de esperanza)} \\
 &= \mathbb{E}_\pi[R_{t+1} | s, a] + \gamma \mathbb{E}_\pi[G_{t+1} | s, a] && \text{(Esperanzas iteradas)} \\
 &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | s, a] \\
 &= \mathbb{E}_\pi[G_t | s, a] && \text{(Definición de } G_t \text{)} \\
 &\doteq q_\pi(s, a) && \text{(Definición de } q_\pi(s, a) \text{)}
 \end{aligned}$$

■

c)

Dado que las políticas son deterministas y las probabilidades de moverse son todas igual a 1, la ecuación de Bellman se simplifica. Pasamos de la fórmula general:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')]$$

se reduce a una versión sin sumatorias, ya que solo hay una acción posible con probabilidad 1 y solo un resultado posible con probabilidad 1:

$$v_\pi(s) = r + \gamma v_\pi(s')$$

Para  $\pi_l(s_0) = a_l$ , tenemos el siguiente sistema de ecuaciones:

$$\begin{aligned} v_{\pi_l}(s_0) &= 1 + \gamma \cdot v_{\pi_l}(s_l) \\ v_{\pi_l}(s_l) &= 0 + \gamma \cdot v_{\pi_l}(s_0) \end{aligned}$$

La segunda ecuación se sustituye en la primera:

$$\begin{aligned} v_{\pi_l}(s_0) &= 1 + \gamma(\gamma \cdot v_{\pi_l}(s_0)) \\ v_{\pi_l}(s_0) &= 1 + \gamma^2 v_{\pi_l}(s_0) \\ v_{\pi_l}(s_0) - \gamma^2 v_{\pi_l}(s_0) &= 1 \\ v_{\pi_l}(s_0)(1 - \gamma^2) &= 1 \\ v_{\pi_l}(s_0) &= \frac{1}{1 - \gamma^2} \end{aligned}$$

Para  $\pi_r(s_0) = a_r$ , tenemos el siguiente sistema de ecuaciones:

$$\begin{aligned} v_{\pi_r}(s_0) &= 0 + \gamma \cdot v_{\pi_r}(s_r) \\ v_{\pi_r}(s_r) &= 2 + \gamma \cdot v_{\pi_r}(s_0) \end{aligned}$$

La segunda ecuación se sustituye en la primera:

$$\begin{aligned} v_{\pi_r}(s_0) &= \gamma(2 + \gamma \cdot v_{\pi_r}(s_0)) \\ v_{\pi_r}(s_0) &= 2\gamma + \gamma^2 v_{\pi_r}(s_0) \\ v_{\pi_r}(s_0) - \gamma^2 v_{\pi_r}(s_0) &= 2\gamma \\ v_{\pi_r}(s_0)(1 - \gamma^2) &= 2\gamma \\ v_{\pi_r}(s_0) &= \frac{2\gamma}{1 - \gamma^2} \end{aligned}$$

Ahora considerando la selección de  $\gamma = 0, 0.5$  y  $0.9$ , Remplazamos en las fórmulas:  $v_{\pi_l}(s_0) = \frac{1}{1-\gamma^2}$  y  $v_{\pi_r}(s_0) = \frac{2\gamma}{1-\gamma^2}$ .

- $\gamma = 0$ :

- $v_{\pi_l}(s_0) = \frac{1}{1-0} = 1$
- $v_{\pi_r}(s_0) = \frac{0}{1-0} = 0$
- $\pi_l$  es la política óptima.

- $\gamma = 0.5$ :

- $v_{\pi_l}(s_0) = \frac{1}{1-0.25} = \frac{1}{0.75} \approx 1.33$
- $v_{\pi_r}(s_0) = \frac{2(0.5)}{1-0.25} = \frac{1}{0.75} \approx 1.33$
- Ambas son políticas óptimas.

- $\gamma = 0.9$ :

- $v_{\pi_l}(s_0) = \frac{1}{1-0.81} = \frac{1}{0.19} \approx 5.26$
- $v_{\pi_r}(s_0) = \frac{2(0.9)}{1-0.81} = \frac{1.8}{0.19} \approx 9.47$
- $\pi_r$  es la política óptima.

d)

Para reportar los valores de los estados iniciales, y el tiempo que el algoritmo tomo para resolver el problema, puse los outputs en las tablas 1, 2 y 3.

Table 1: Resultados para GridProblem ( $\gamma = 1.0$ )

Tamaño Grilla	Valor Estado Inicial	Tiempo (s)
3	-8.000	0.005
4	-18.000	0.031
5	-37.333	0.175
6	-60.231	0.228
7	-93.496	0.332
8	-131.193	0.617
9	-180.001	1.005
10	-233.879	1.721

Table 2: Resultados para CookieProblem ( $\gamma = 0.99$ )

Tamaño Grilla	Valor Estado Inicial	Tiempo (s)
3	0.787	0.101
4	0.612	0.835
5	0.452	2.557
6	0.325	4.825
7	0.231	9.851
8	0.164	17.858
9	0.117	34.129
10	0.083	49.686

Table 3: Resultados para GamblerProblem ( $\gamma = 1.0$ )

Probabilidad Cara	Valor Estado Inicial	Tiempo (s)
0.25	0.067	0.218
0.40	0.284	0.309
0.55	0.612	0.435

e)

Existen problemas que toman más tiempo que otros debido al distinto tamaño del espacio de estados que tiene cada problema. El problema más lento en solucionar es el CookieProblem, en donde un estado determinado es de la forma (posicion-agente, posicion-galleta). Notar que el agente puede estar en ( $\text{TamañoGrilla} * \text{TamañoGrilla}$ ) distintas posiciones, y la galleta también, por lo que habrá un total de  $(\text{TamañoGrilla})^4$  distintos estados, lo cual escala de muy mala manera.

Luego, el GridProblem es similar al CookieProblem, con la diferencia de que el estado solo es la posición del agente, por lo que la cantidad de estados distintos es  $(\text{TamañoGrilla})^2$ , lo cual tampoco escala bien, pero es mejor que lo que ocurre con el CookieProblem.

Finalmente, el GamblerProblem es el más rápido de todos, principalmente debido a que solo puede tener un número de 101 estados distintos. También vale la pena mencionar que en GridProblem y CookieProblem, mientras más grande es el tamaño de la grilla, más lento es el cálculo de valores, lo cual es esperable, ya que existirán más estados y, por lo tanto, demorará más en converger. Esto se puede visualizar en los gráficos de convergencia (Figuras 1, 2 y 3).

En el GamblerProblem, mientras más uniforme sea la probabilidad (cercana a 0.5), más se demorará. Esto se debe a que el agente se quedará apostando por más tiempo, ya que perderá y ganará de manera más balanceada. Pero si la probabilidad está más cargada para cualquiera de los dos lados, entonces el tiempo es menor, ya que perderá o ganará de manera más determinista.

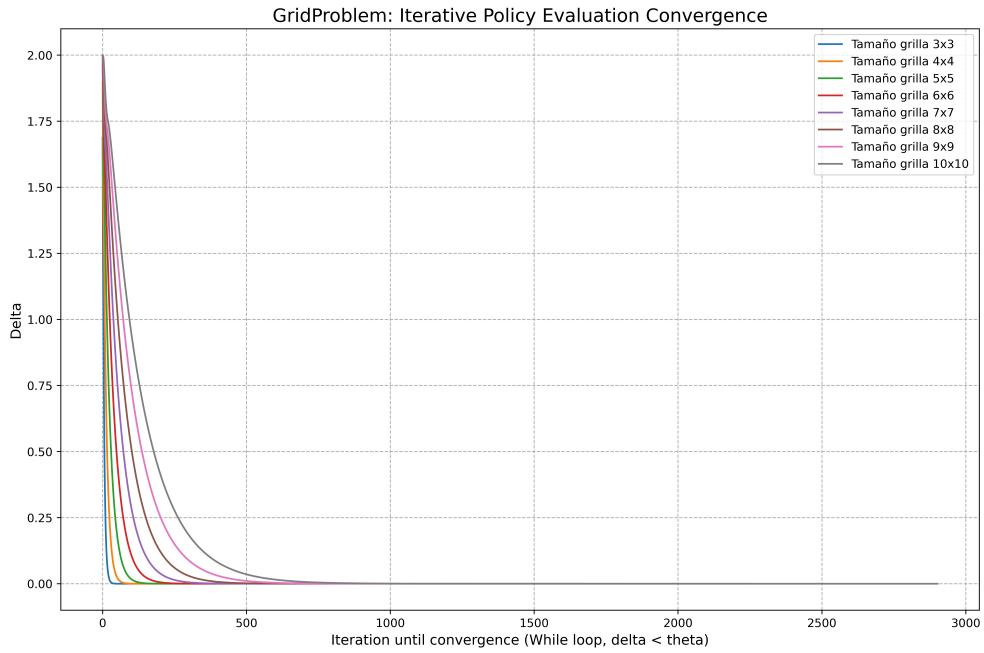


Figure 1: Convergencia para GridProblem.

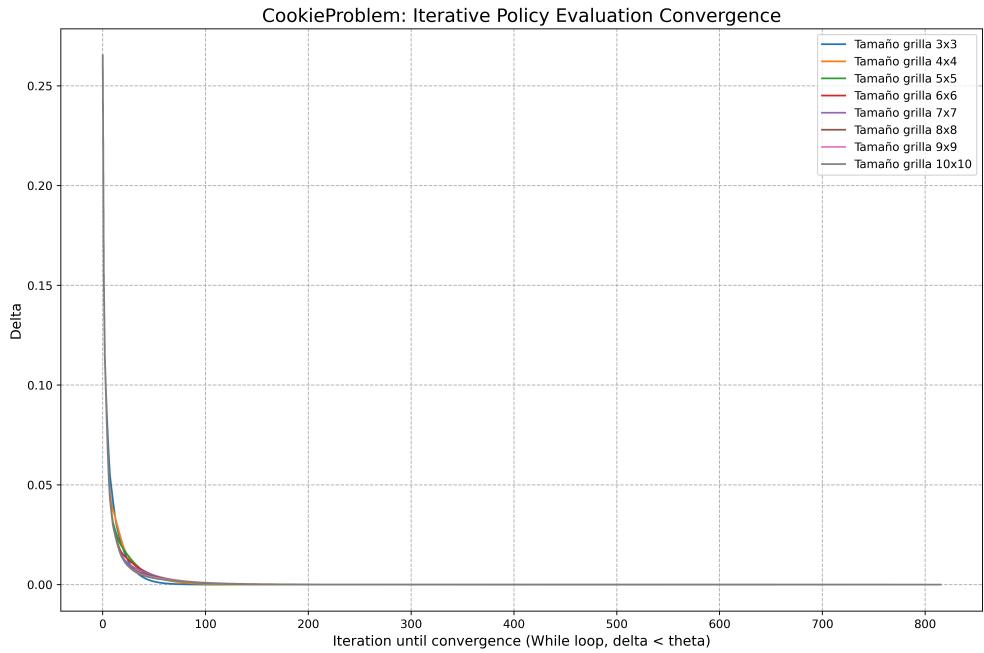


Figure 2: Convergencia para CookieProblem.

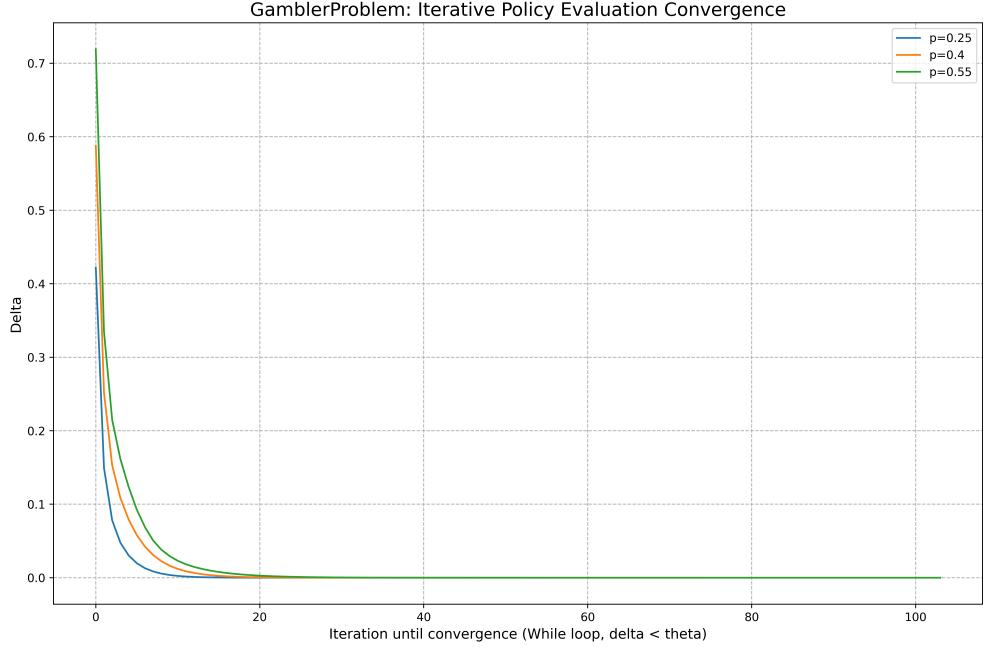


Figure 3: Convergencia para GamblerProblem.

f)

Si bajamos el valor del factor de descuento, deberíamos esperar que el tiempo de cómputo sea menor, ya que los valores deberían converger de manera más rápida. Probé todos los problemas con un  $\gamma = 0.5$  y las soluciones se encuentran mucho más rápido.

Esto ocurre ya que con un gamma pequeño, las recompensas serán más relevantes en la fórmula de Bellman, por lo que los valores de los estados estarán más ligados a las recompensas inmediatas que a las recompensas futuras. De esta manera, el vínculo entre los valores de los estados se reduce, por lo que la convergencia ocurre más rápido.

g)

Los resultados obtenidos luego de evaluar la política greedy que resulta de los valores calculados en la parte (d) se encuentran en las tablas de abajo (Tablas 4, 5 y 6). La columna "Valor Greedy" contiene los resultados luego de evaluar la política greedy a partir de los valores de la parte (d). La columna "Valor Óptimo" contiene los valores resultantes al ejecutar el algoritmo "Policy Iteration" del capítulo 4.3 del libro de Sutton y Barto (notar que estos valores están redondeados al tercer decimal). La columna "¿Greedy Óptima?" simplemente establece si la política greedy es la óptima o no. Finalmente, la columna "Tiempo (s)" contiene el tiempo que se demoró cada ejecución.

En todos los casos, la política greedy es la óptima, ya que se retorna el mismo valor que el que se obtiene con la política óptima, la cual fue encontrada con el algoritmo dicho anteriormente.

Solamente en el GamblerProblem con  $p = 0.55$  la política greedy no es la óptima, esto debido a que con la política óptima el valor converge a 1.

Table 4: Resultados para GridProblem con Política Greedy

Tamaño	Valor Greedy	Valor Óptimo	¿Greedy Óptima?	Tiempo (s)
3	-2.000	-2.000	Sí	0.006
4	-2.000	-2.000	Sí	0.023
5	-4.000	-4.000	Sí	0.068
6	-4.000	-4.000	Sí	0.177
7	-6.000	-6.000	Sí	0.307
8	-6.000	-6.000	Sí	0.588
9	-8.000	-8.000	Sí	1.072
10	-8.000	-8.000	Sí	1.638

Table 5: Resultados para CookieProblem con Política Greedy

Tamaño	Valor Greedy	Valor Óptimo	¿Greedy Óptima?	Tiempo (s)
3	0.970	0.970	Sí	0.103
4	0.951	0.951	Sí	0.565
5	0.932	0.932	Sí	1.881
6	0.914	0.914	Sí	4.648
7	0.895	0.895	Sí	9.658
8	0.878	0.878	Sí	17.348
9	0.860	0.860	Sí	31.722
10	0.843	0.843	Sí	48.779

Table 6: Resultados para GamblerProblem con Política Greedy

$p$	Valor Greedy	Valor Óptimo	¿Greedy Óptima?	Tiempo (s)
0.25	0.250	0.250	Sí	0.199
0.40	0.400	0.400	Sí	0.285
0.55	0.730	0.999	No	0.456

h)

Para reportar los valores optimos de los estados iniciales, y el tiempo que el algoritmo tomo para resolver el problema, puse los outputs en las tablas 7, 8 y 9.

Table 7: Resultados para GridProblem ( $\gamma = 1.0$ )

Tamaño Grilla	Valor Óptimo Inicial	Tiempo (s)
3	-2.000	0.000
4	-2.000	0.000
5	-4.000	0.001
6	-4.000	0.001
7	-6.000	0.002
8	-6.000	0.002
9	-8.000	0.003
10	-8.000	0.005

 Table 8: Resultados para CookieProblem ( $\gamma = 0.99$ )

Tamaño Grilla	Valor Óptimo Inicial	Tiempo (s)
3	0.970	0.002
4	0.951	0.008
5	0.932	0.025
6	0.914	0.064
7	0.895	0.141
8	0.878	0.277
9	0.860	0.507
10	0.843	0.894

 Table 9: Resultados para GamblerProblem ( $\gamma = 1.0$ )

Probabilidad Cara ( $p_h$ )	Valor Óptimo Inicial	Tiempo (s)
0.25	0.250	0.054
0.40	0.400	0.063
0.55	1.000	6.083

i)

El set de todas las políticas óptimas para el GamberProblem se pueden visualizar en las figuras 4, 5 y 6.

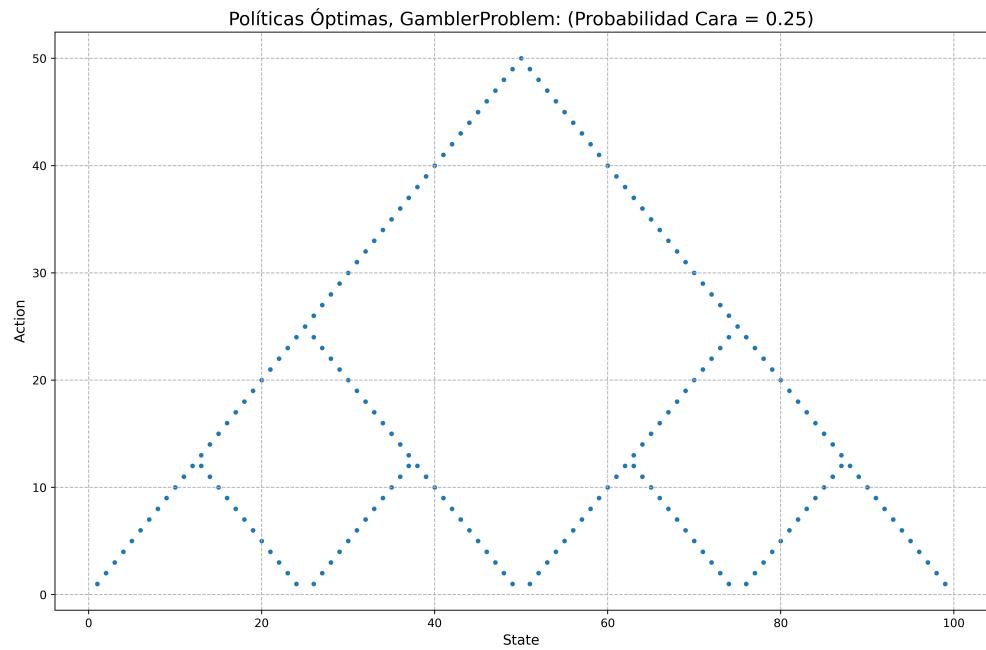


Figure 4: Políticas Óptimas GamblerProblem,  $p_h = 0.25$ .

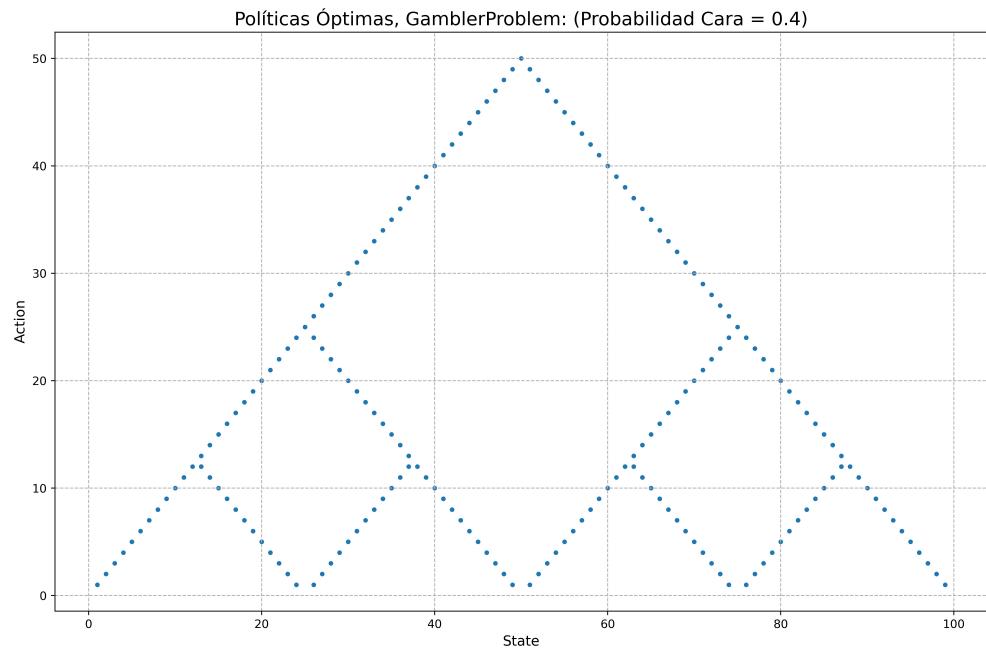


Figure 5: Políticas Óptimas GamblerProblem,  $p_h = 0.4$ .

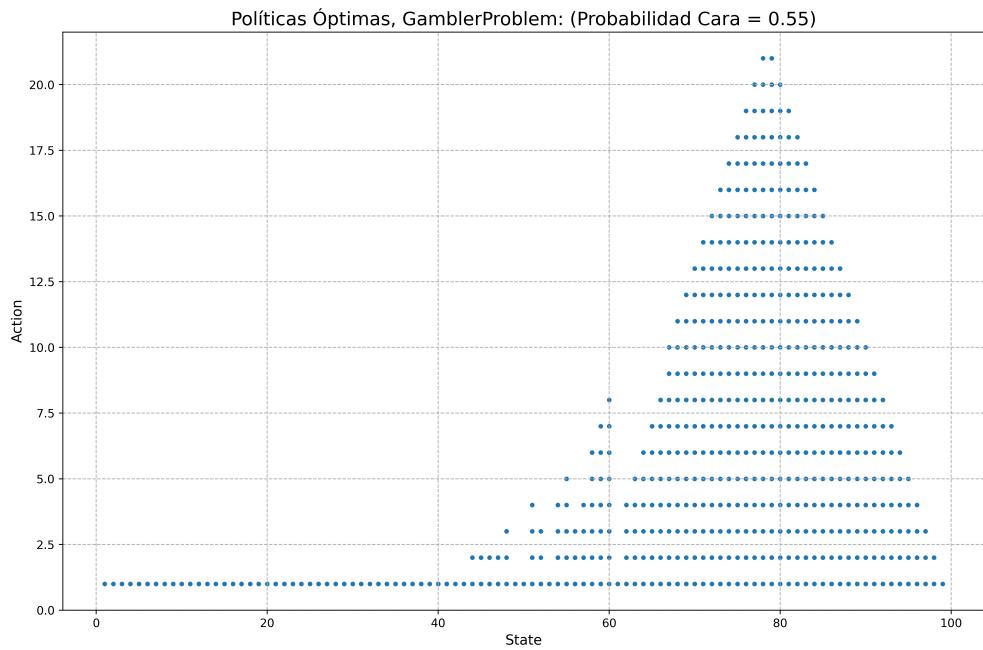


Figure 6: Políticas Óptimas GamblerProblem, con  $p_h = 0.55$ .

j)

Los resultados los reporto mediante dos gráficos, en donde cada uno contiene los resultados, luego de ejecutar On-Policy every visit MC Control para BlackJack y Cliff Walking.

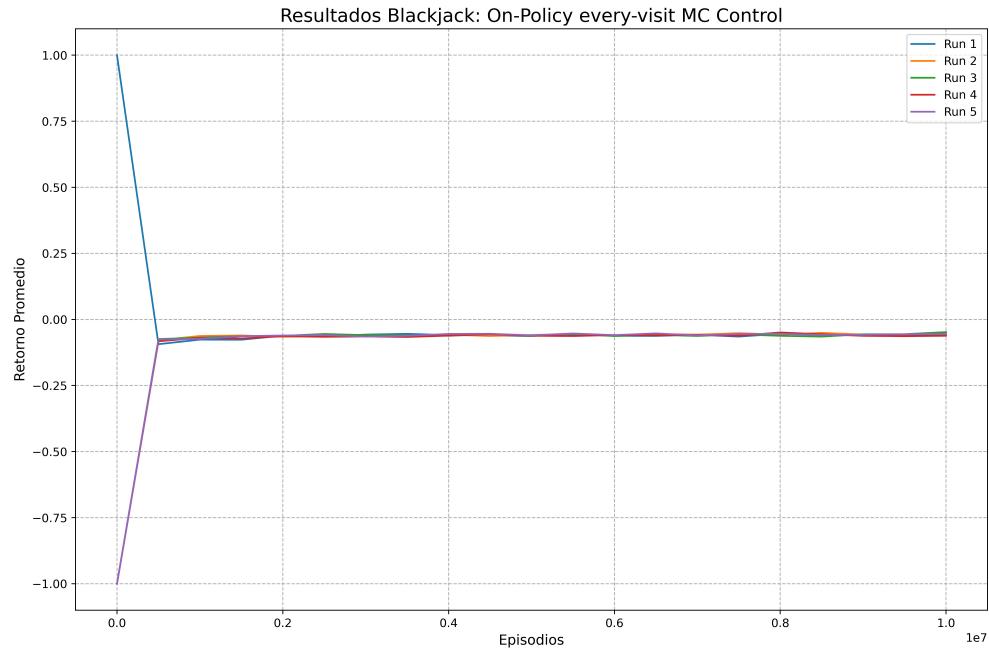


Figure 7: Se puede observar que el retorno promedio converge en las 5 ejecuciones, alcanzando un valor aproximado de -0.05.

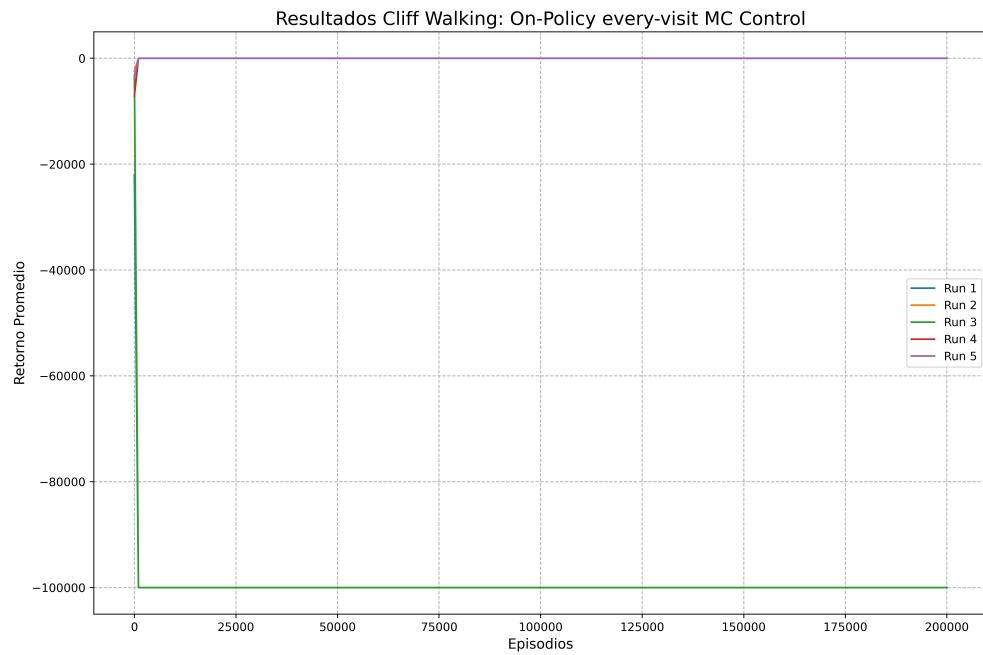


Figure 8: A diferencia de la figura 7, en este gráfico se puede observar mayor inestabilidad, en donde distintas ejecuciones convergen a distintos resultados.

k)

Para responder esta pregunta, adjunto una figura del libro de Sutton y Barto que muestra la política óptima para Blackjack (Figura 9 en el informe, Figura 5.2 en el libro).

Observándola, podemos notar que la estrategia depende de dos cosas: Si tenemos un "Ace" y cuál es la carta del dealer que podemos observar.

- **Si tenemos un "Ace":** La política óptima recomienda ser más agresivos, ya que, en la práctica, indica que se deben pedir cartas hasta alcanzar una suma de 18. Sin embargo, si la suma es mayor a 18, la política indica que es recomendable dejar de pedir cartas.
- **Sin no tenemos un "Ace":** La política óptima recomienda ser más cuidadosos, ya que indica que se deben pedir cartas hasta alcanzar una suma de 16, pero si la suma es mayor a 16 entonces hay que dejar de pedir cartas. También hay que considerar que si el jugador tiene entre 12 y 16 puntos, lo recomendable es dejar de pedir cartas si y solo si el dealer muestra una carta de menor valor (de 2 a 6).

Sobre si siempre se llega a la misma conclusión, sí, siempre se llega a la misma conclusión en las 5 corridas. Como se puede observar en la Figura 7 del informe, todas las ejecuciones convergen al mismo valor de retorno promedio. Esto nos sugiere que este algoritmo para este juego es bastante estable.

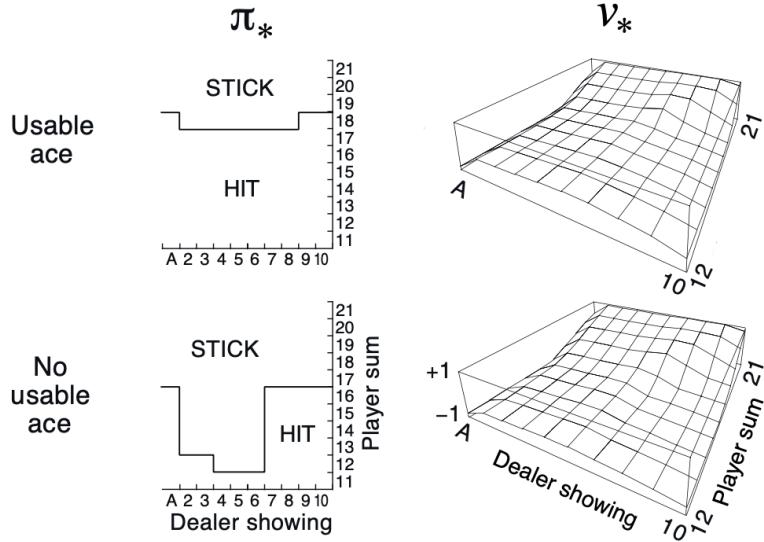


Figure 9: Política óptima para Blackjack, extraída de Sutton & Barto (Figura 5.2).

l)

El mejor curso de acción es irse por arriba, ya que permite asegurarse de mejor manera que el agente no llegue a la sección de "The Cliff", en donde recibe una recompensa negativa catastrófica. Sin embargo, hay que considerar que este algoritmo es epsilon-greedy, por lo

que de manera inevitable van a haber situaciones en donde el agente va a tocar la sección de "The Cliff".

Observando la figura 8, podemos afirmar que el resultado descrito anterior no es consistente en las 5 corridas, de manera específica, hay corridas en donde el retorno promedio alcanza valores bajísimos.

La razón por la cual en algunas corridas llega al retorno promedio de -100.000 es debido a que el agente está actuando con una política que tiene un loop, por lo que el agente comienza a dar pasos sin llegar al fin, por lo que en cada uno recibe -1 de recompensa, y como hay un límite de 100.000 pasos, converge en esa cantidad.

**m)**

Considerando los resultados del algoritmo para ambos problemas (Figura 7 y 8), yo diría que Monte Carlo es un algoritmo estable solamente para algunos casos.

Para Blackjack, el algoritmo si es estable, esto se ve evidenciado en la figura 7, y ocurre principalmente debido a que en blackjack cada episodio de manera determinista va a terminar.

Para Cliff Walking, el algoritmo no es estable, esto también se ve evidenciado en la figura 8, y esto ocurre debido a que existen situaciones en donde el agente se puede quedar en un loop infinito.

**n)**

Lo que ocurre es que el retorno promedio de manera consistente se convierte en -100.001. Esto debido a que el espacio es mucho más grande ahora, entonces el agente simplemente no puede encontrar una solución antes de que se le "termine el tiempo". Técnicamente si lo puede resolver, pero con la suposición de que tenga mucha suerte y tome las acciones buenas para poder llegar al fin, sin embargo, dado que no tiene tiempo para encontrar una política óptima, es bien difícil que esto ocurra.