UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE COMPUTACIÓN

ALGORITMOS PARA JUEGOS CON INFORMACIÓN INCOMPLETA Y NO DETERMINISMO

Por
Rubmary Rojas Linárez

TRABAJO DE GRADO
Presentado ante la ilustre Universidad Simón Bolívar
como requisito requisito parcial para optar al título de
Ingeniero de Computación

Sartenejas, Septiembre de 2019

R. ROJAS LINÁREZ
2019

ALGORITMOS PARA JUEGOS CON INFORMACIÓN INCOMPLETA Y NO DETERMINISMO

USB
INGENIERÍA DE COMPUTACIÓN

UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE COMPUTACIÓN

ALGORITMOS PARA JUEGOS CON INFORMACIÓN INCOMPLETA Y NO DETERMINISMO

Por
Rubmary Rojas Linárez

Realizado con la asesoría de:

Blai Bonet

TRABAJO DE GRADO
Presentado ante la ilustre Universidad Simón Bolívar
como requisito requisito parcial para optar al título de
Ingeniero de Computación

Sartenejas, Septiembre de 2019

Página reservada para el acta de evaluación

DEDICATORIA

Debe ser elaborada bajo las mismas normas del desarrollo del trabajo y mismo tipo de letra seleccionado (tamaño 12).

Dedicado a la Universidad Simón Bolívar y a su comunidad universidad.

AGRADECIMIENTOS

Debe ser elaborada bajo las mismas normas del desarrollo del trabajo y mismo tipo de letra seleccionado (tamaño 12).

RESUMEN

Es una exposición clara del tema tratado en el trabajo, de los objetivos, de la metodología utilizada, de los resultados relevantes obtenidos y de las conclusiones. Mismo tipo de fuente seleccionado con tamaño 12 e interlineado sencillo en el párrafo. El resumen no debe exceder de trescientas (300) palabras escritas.
Palabras cláves: palabras, claves, separadas por coma, cinco máximo.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE ACRÓNIMOS

USB Universidad Simón Bolívar

EN Equilibrio de Nash

RM Regret Matching

CFR Conterfactual Regret Minimization

MCCFR Monte Carlo Conterfactual Regret Minimization

GEBR Generalized Expectimax Best Response

OCP One Card Poker

DFS Depth First Search

RPS Rock Paper Scissors

INTRODUCCIÓN

En este documento se muestra el uso apropiado de la clase `clase-usb.cls` en LaTeX para la creación de libros de tabajo final en pregrado y postgrado según las normas del Decanato de Estudios Profesionales y Decanato de Estudios de Postgrado de la Universidad Simón Bolívar. Para más detalle sobre las normas de ambos decanatos visitar las respectivas páginas de internet: `www.profesionales.usb.ve` y `www.postgrado.usb.ve`.

El archivo `tesis-usb.zip` contiene: el archivo de la clase (`tesis-usb.cls`), dos documentos de muestra e instructivos para el uso de la clase (uno para pregrado y otro para postgrado) y las fuentes de ambos documentos instructivos en los archivos comprimidos `ejemplo-pregrado.zip` y `ejemplo-postgrado.zip`, respectivamente.

En el Capítulo 1 de este documento se explica el uso correcto de la clase. En el Capítulo 2 se expone una forma para generar la lista de acrónimos y la lista de simbolos. En el Capítulo 3 se explica la forma de compilar el libro (y cualquier otro documento en general).

Esta es la versión 4.2 de la clase no oficial para trabajos de la Universidad Simón Bolívar, creada y mantenida por:

MSc. Carlos Contreras ccontreras@usb.ve
MSc. Andrés Sajo-Castelli asajo@usb.ve

CAPÍTULO I

JUEGOS EN FORMAL NORMAL O ESTRATÉGICA

En un juego en formal normal los jugadores eligen una única acción (o estrategia) de forma simultánea, obteniendo un pago de acuerdo a las acciones realizada por cada uno de ellos. Estos juegos también se llaman frecuentemente *"one-shot game"* (juegos de un sólo disparo), ya que cada uno de los jugadores realiza una única acción [1] (que puede representar la elección de una estrategia a usar en un juego de múltiples pasos. El ejemplo clásico es el juego de *piedra, papel o tijera*. En este juego cada jugador elige una de las tres opciones mediante un gesto con sus manos: piedra (con un puño cerrado), papel (con la mano extendida) o tijera (con los dedos índice y medio levantados en forma de "V"). La piedra gana contra la tijera, la tijera gana contra el papel y el papel gana contra la piedra. Si los jugadores eligen la misma opción, entonces es un empate.

Definición 1.1 ([2]). *Un juego de N personas en* ***forma normal*** *(o estratégica) es una tupla* $\Gamma = (N, (S_i)_{i\in N}, (u_i)_{i\in N})$*, donde:*

- $N = \{1, 2, \ldots, N\}$ *es el conjunto de jugadores.*
- S_i *es el conjunto de* ***estrategias puras*** *(o acciones) del jugador* i*.*
- $u_i : \Pi_{i\in N} S_i \to \mathbb{R}$ *es la* ***función de pago*** *del jugador* i*.*

1.1. Perfiles Estratégicos, Estrategias Mixtas, y Perfiles Estratégicos Mixtos

A continuación presentamos conceptos básicos que denotan las diversas formas en que los jugadores pueden comportarse para un juego en forma normal. Las estrategias puras son la base a partir de las cuales se construyen las estrategias mixtas. Las estrategias puras se agregan en perfiles estratégicos que denotan el comportamiento de todos los jugadores de forma simultanea, y los perfiles estratégicos mixtos agregan las estrategias mixtas [3].

Tabla 1.1: Tabla de pagos de la forma normal del juego *Piedra, Papel o Tijeras.* *** **EXPLICAR UN POCO LA TABLA. POR LO MENOS UNA ENTRADA** ***

	R (piedra)	P (papel)	S (tijeras)
R (piedra)	$0, 0$	$-1, 1$	$1, -1$
P (papel)	$1, -1$	$0, 0$	$-1, 1$
S (tijeras)	$-1, 1$	$1, -1$	$0, 0$

Definición 1.2. *Un **perfil estratégico** (o perfil de acción) es una N-tupla formada por una estrategia para cada jugador. $S = \Pi_{i \in N} S_i$ es el conjunto de perfiles estratégicos y $s = (s_i)_{i \in N}$ representa un elemento genérico de S.*

Se denotará con s_{-i} la combinación de las estrategias de todos los jugadores excepto la del jugador i, i.e., $s_{-i} = (s_{i'})_{i' \neq i}$. Frecuentemente descomponemos una estrategia s es un par (s_i, s_{-i}) donde la primera componente es una estrategia pura para el jugador i y la segunda componente es un vector de estrategias puras para los otros jugadores.

Piedra, papel o tijeras es un juego para dos jugadores y las acciones (o estrategias puras) son las mismas para cada jugador: $S_1 = S_2 = \{R, P, S\}$ donde R es piedra, P es papel, y S es tijeras. Los juegos en forma normal pueden representarse como una tabla n-dimensional, donde cada dimensión está asociada a un jugador y sus filas/columnas corresponden a las acciones de su jugador correspondiente. Cada una de las entradas de la tabla corresponde a un único perfil estratégico (pues representan la intersección de una única acción de cada jugador) y éstas contienen un vector de pagos para cada jugador [1]. La Tabla 1.1 es la tabla de pagos correspondiente al juego piedra, papel o tijeras.

En vez de realizar siempre la misma acción, un jugador puede elegir su jugada de acuerdo a una distribución de probabilidad la cual se denomina una **estrategia mixta**. Dado un conjunto finito A, se denota con $\Delta(A)$ al conjunto de distribuciones de probabilidad sobre A, es decir $\Delta(A) = \{(x_a)_{a \in A} : \sum_{a \in A} x_a = 1, x_a \geq 0\}$.

Definición 1.3. *Una **estrategia mixta** del jugador i, denotada con σ_i, es una distribución de probabilidad sobre el conjunto S_i; es decir $\sigma_i \in \Delta(S_i)$. Denotamos con $\sigma_i(s_i)$ la probabilidad que el jugador i elija la acción $s_i \in S_i$.*

Definición 1.4. *El **soporte** (support) de una estrategia mixta $\sigma_i \in \Delta(S_i)$ del jugador i es el conjunto de estrategias puras con una probabilidad positiva de ser elegidas:*

$$support(\sigma_i) \;=\; \{s_i : \sigma_i(s_i) > 0\}\,. \tag{1.1}$$

Definición 1.5. *Un **perfil estratégico mixto** σ consiste en una estrategia mixta para cada jugador; es decir, $\sigma \in \Pi_{i\in N}\Delta(S_i)$ es una tupla de forma $\sigma = (\sigma_i)_{i\in N}$.*

Para $\sigma = (\sigma_i)_{i\in N}$ y $s = (s_i)_{i\in N}$, $\sigma(s)$ denota la probabilidad que el perfil estratégico mixto elija la estrategia mixta s; i.e., $\sigma(s) = \prod_{i\in N}\sigma_i(s_i)$. Para un perfil σ y jugador i, descomponemos σ en (σ_i, σ_{-i}) como la combinación de la estrategia para el jugador i y el perfil σ_{-i} para el resto de los jugadores. Similarmente, $\sigma_{-i}(s_{-i}) = \prod_{j\in N, j\neq i}\sigma_j(s_j)$ denota la probabilidad de que los jugadores diferentes al jugador i elijan las estrategias mixtas en el perfil σ_{-i}. Finalmente, si x es una estrategia pura para el jugador i, también utilizamos x para denotar la estrategia mixta σ_i para el jugador i tal que $\sigma_i(x) = 1$.
***** Faltan ejemplos que ilustren estas ideas. Usar RPS. *****

1.2. Ganancia Esperada y Mejor Respuesta

La ganancia esperada del jugador i asociada al perfil estratégico mixto σ denota el el valor promedio que el jugador i obtendría después de jugar el juego infinitas veces cuando todos los jugadores utilizan las estratégias mixtas especificadas en σ.

Definición 1.6. *La **ganancia esperada** del jugador i dado un perfil estratégico mixto σ es*

$$u_i(\sigma) \;=\; \sum_{s\in S} u_i(s)\sigma(s) \;=\; \sum_{s\in S} u_i(s)\prod_{j\in N}\sigma_j(s_j) \;=\; \sum_{s\in S} u_i(s)\sigma_i(s_i)\sigma_{-i}(s_{-i})\,. \tag{1.2}$$

La ganancia esperada del jugador i la podemos descomponer como se muestra a continuación:

Teorema 1.7. *La ganancia esperada $u_i(\sigma)$ del jugador i dado el perfil estratégico σ satisface:*

$$u_i(\sigma) \;=\; \sum_{s_i\in S_i}\sigma_i(s_i)\sum_{s_{-i}\in S_{-i}}\sigma_{-i}(s_{-i})u_i(s_i, s_{-i})\,. \tag{1.3}$$

Dado una perfil estratégico mixto σ, tiene sentido preguntarse si el jugador i está jugando de la mejor forma dadas las estrategias seleccionadas por los otros jugadores. A partir de esta pregunta, definimos el concepto de mejor respuesta para el jugador i dado un perfil σ_{-i} para los otros jugadores.

Definición 1.8. *Sea $i \in N$ un jugador, σ_i una estrategia mixta para el jugador i, y σ_{-i} un perfil estratégico mixto para el resto de los jugadores. Decimos que σ_i es una **mejor***

respuesta *con respecto a* σ_{-i} *si y sólo si* $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$ *para toda estrategia mixta* σ'_i *para el jugador* i*.*

Una mejor respuesta no es necesariamente única. En efecto, salvo el caso extremo en el que hay una única mejor respuesta, que como veremos debe ser una estrategia pura, el número de mejores respuestas es infinito. Cuando el soporte de una estrategia mixta que es mejor respuesta incluye dos o más estrategias puras (acciones), el agente debe ser indiferente a cualquiera de éstas y cualquier mezcla de estas acciones también será mejor respuesta [3].

Teorema 1.9. *Sea* σ_i^* *una estrategia mixta para el jugador* i *que es mejor respuesta a* σ_{-i}*. Cualquier estrategia mixta* σ_i *para el jugador* i *cuyo soporte sea un subconjunto del soporte de* σ_i^* *es también una mejor respuesta a* σ_{-i}*.*

***** Faltan ejemplos que ilustren estas ideas. Usar RPS y ejemplos concretos donde calcular ganancia esperada y mejor respuesta. *****

1.3. Equilibrio de Nash

Cuando cada jugador juega con una mejor respuesta frente a las estrategias del resto de los jugadores se dice que tenemos un Equilibrio de Nash. En un Equilibrio de Nash ningún jugador puede mejorar su ganancia esperada cambiando su estrategia de forma aislada. Por otra parte, si el juego es finito, siempre existe al menos un equilibrio de Nash. Un juego es finito si el número de jugadores es finito, y si el conjunto de estrategias puras para cada jugador es también finito. El concepto de equilibrio de Nash es uno de los conceptos de solución más importantes en el área de teoría de juegos no cooperativos, y es el principal concepto de solución utilizado en el presente trabajo.

Definición 1.10. *Un perfil estratégico mixto* σ *es un* ***equilibrio de Nash*** *si y sólo si para todo jugador* i*, la estrategia* σ_i *es mejor respuesta del jugador* i *para* σ_{-i}*.*

Teorema 1.11 ([3])**.** *Todo juego finito tiene al menos un equilibrio de Nash.*

***** Faltan ejemplos que ilustren estas ideas. Usar RPS. Presentar un equilibro de Nash para RPS. Discutirlo. Argumentar que no puede haber un perfil estratégico que sea un equilibrio de Nash (?). ETC *****

1.4. Equilibrio Correlacionado

Aunque el equilibrio de Nash es uno de los principales conceptos de solución, es importante destacar que éste no garantiza el mejor resultado si los jugadores toman sus decisiones en conjunto. Si a los jugadores se les permite correlacionar sus acciones (es decir, trabajar en grupo), pueden existir estrategias con mayores ganancias para ellos. Este tipo de situaciones son las que considera la noción de equilibrio correlacionado que generaliza al equilibrio de Nash [2]. Todo equilibrio de Nash es un equilibrio correlacionado, pero este último permite otras soluciones importantes [2]. La relación entre los conceptos de equilibro de Nash y correlacionado se muestra en los Teoremas A.4 y A.5.

Definición 1.12. *Una distribución $\psi \in \Delta(S)$ es un* ***equilibrio correlacionado*** *si y sólo si para cualquier jugador i, y para cualesquiera estrategias puras $x, y \in S_i$,*

$$\sum_{s_{-i} \in S_{-i}} \psi(x, s_{-i})[u_i(x, s_{-i}) - u_i(y, s_{-i})] \;\geq\; 0\,. \tag{1.4}$$

Si en la desigualdad (1.4) se cambia el 0 por un $\epsilon > 0$ se obtiene la definición de ϵ-equilibrio correlacionado.

Teorema 1.13. *Si σ es un equilibrio de Nash, entonces σ es un equilibrio correlacionado.*

Teorema 1.14. *Sea $\psi \in \Delta(S)$ un equilibrio correlacionado. Si ψ se factoriza como $\psi = \prod_{i \in N} \sigma_i$ donde $\{\sigma_i\}_{i \in N}$ es un conjunto de estrategias mixtas para cada jugador (i.e., $\psi(s) = \prod_{i \in N} \sigma_i(s_i)$ para todo $s \in S$), entonces ψ es un equilibrio de Nash.*

A diferencia del conjunto de equilibrios de Nash, el cual es un conjunto matemáticamente complejo (un conjunto de puntos fijos), el conjunto de equilibrios correlacionados en un conjunto bastante simple. En particular, el conjunto de equilibrios correlacionado es un politopo (generalización de un polígono en $\mathbb{R}^N$) convexo. Por lo tanto puede esperarse que existan procedimientos simples para calcular equilibrios correlacionados [2].
***** Y esto nos va a llevar a calcular equilibrios de Nash? *****

Teorema 1.15. *Sean σ y σ' dos equilibrios correlacionados, y α un número real en $(0, 1)$. Entonces, la distribución $\alpha\sigma + (1 - \alpha)\sigma'$ es un equilibrio correlacionado.*

CAPÍTULO II

JUEGOS EN FORMA EXTENSA

Muchos juegos constan de una secuencia de acciones realizadas por los jugadores a lo largo del tiempo, haciendo al modelo anterior insatisfactorio debido a que ignora la estructura secuencial de este tipo de problemas de decisión. Estos juegos pueden ser representados en forma de árbol enraizado, donde cada nodo representa un estado del juego y las ramas representan las acciones que se pueden realizar en un nodo (o estado) específico.

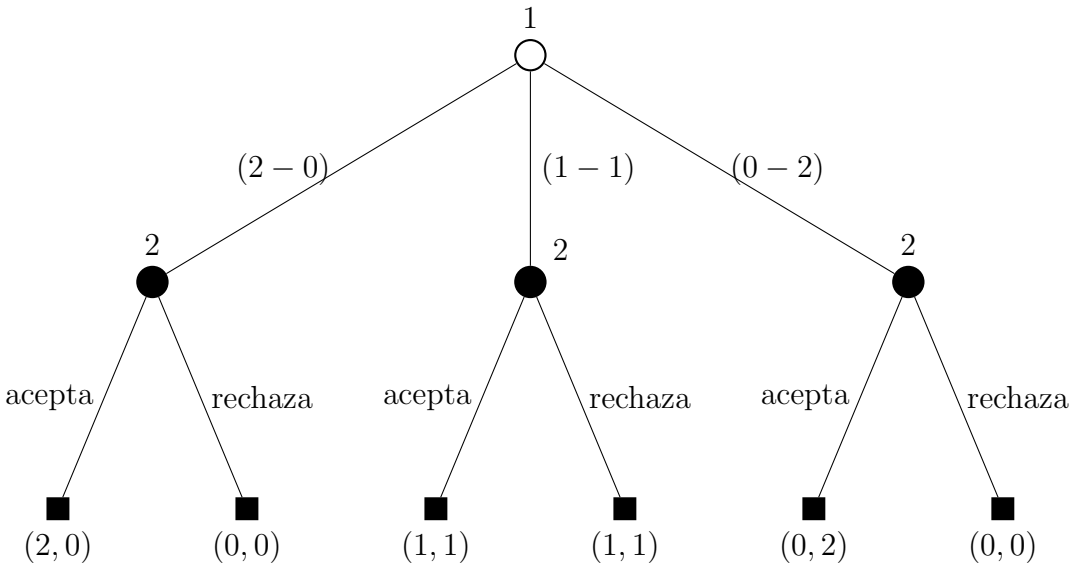
Ejemplo 2.1 ([4, p. 91])**.** *Dos personas utilizan el siguiente procedimiento para compartir dos objetos idénticos e indivisibles. Una de ellas propone una asignación, que la otra persona acepta o rechaza. Si la propuesta es aceptada se lleva a cabo dicha división. En caso de rechazo, ninguna persona recibe ninguno de los dos objetos. Cada persona sólo se preocupa sobre la cantidad de objetos que tiene.*

La Figura 2.1 representa el árbol correspondiente al juego presentado. Cada nodo representa un estado del juego. Los nodos no terminales tienen un jugador asignado, que representa quien debe tomar la decisión en ese estado y las ramas representan las acciones posibles. En este caso la raíz corresponde al primer jugador, el cual tiene 3 opciones posibles: quedarse con los 2 objetos: $(2-0)$, repartir 1 objeto para cada jugador: $(1-1)$, o darle los 2 objetos al jugador 2: $(0-2)$. Los 3 nodos del primer nivel corresponden al jugador 2, en cada uno de ellos tiene dos opciones: aceptar o rechazar la distribución. Las hojas representan los nodos terminales del juego, cada uno con la ganancia respectiva para cada jugador según el caso.

Sin embargo, es importante diferenciar entre dos tipos de juegos: con información completa (o perfecta) y con información incompleta (o imperfecta). En los juegos con información completa los jugadores tienen toda la información sobre las acciones realizadas previamente de todos los jugadores y del estado actual del juego. El Ejemplo 2.1 es un ejemplo de este tipo de juegos; para una definición formal ver [4, pp. 89–90].

En juegos con información incompleta el jugador no tiene toda la información de las acciones tomadas previamente, e incluso pudo haber olvidado las acciones que él u otro

Figura 2.1: Árbol del juego en forma extensiva del Ejemplo 2.1



jugador realizaron previamente. Por lo cual, un jugador puede no tener suficiente información para determinar en qué nodo del árbol se encuentra.

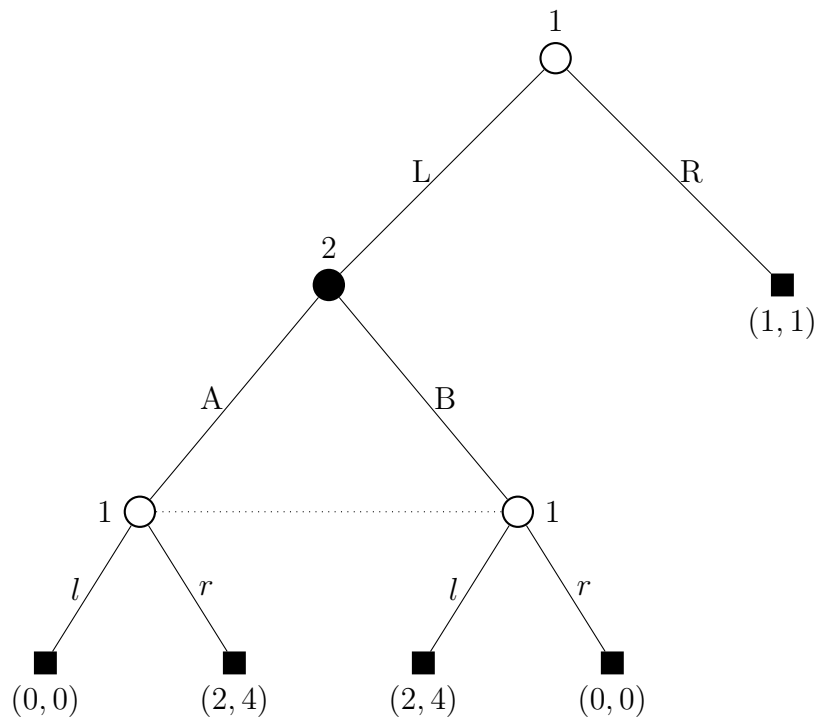
Ejemplo 2.2 ([4, p. 202])**.** *Considere un juego de dos jugadores, el jugador* 1 *y el jugador* 2*, el cual ocurre como sigue: primero, el jugador* 1 *debe elegir una opción entre L y R. Si elige R el juego termina; si elige L se le informa al jugador* 2 *que el jugador* 1 *eligió L y este debe elegir una opción entre A y B. Por último, el jugador* 1 *debe escoger una nueva opción entre l y r, pero sin saber que opción eligió el jugador* 2*. Los pagos son mostrados en las hojas del árbol del juego, presentado en la Figura 2.2*

Se puede observar que los nodos unidos por líneas punteadas son indistinguibles para el jugador 1, pues él no sabe cuál fue la elección del jugador 2. Este tipo de nodos originan los llamados **conjuntos de información**; cf. Definición 2.3 y [4, p. 200]. El concepto de conjunto de información es intrínsica a los juegos en forma extensa y no es necesaria para juegos en forma normal. Nosotros asumimos que los conjuntos de información vienen definidos de forma explícita en el modelo de juego en forma extensa; una definición basada en el árbol del juego puede ser encontrada en [5].

Definición 2.3. *Un juego finito en* ***forma extensa*** *con* ***información incompleta*** *tiene los siguientes componentes:*

- *Un conjunto finito N de* ***jugadores****.*

Figura 2.2: Árbol del juego en forma extensiva presentado en el Ejemplo 2.2



- *Un conjunto finito H de secuencias, las posible* ***historias*** *de acciones, tal que la secuencia vacía está en H, y cada prefijo de una secuencia en H también está en H. $Z \subseteq H$ son las historias terminales (aquellas que no son prefijo de ninguna otra secuencia). $A(h) = \{a : (h, a) \in H\}$ son las acciones disponibles después de una historia no terminal $h \in H$.*

- *Una función P que asigna a cada historia no terminal (cada elemento de $H \setminus Z$) un elemento de $N \cup \{c\}$. P es la* ***función de jugador****. $P(h)$ es el jugador que toma una acción después de la historia h. Si $P(h) = c$ entonces la acción tomada después de la historia h es determinada por el azar. Este tipo de nodos serán denominados* ***nodos de azar****.*

- *Una función f_c que asocia con cada historia h para la cual $P(h) = c$ una medida de probabilidad $f_c(.|h)$ sobre $A(h)$: $f_c(a|h)$ es la probabilidad que la acción a ocurra dado h. Cada medida de probabilidad es independiente de cualquier otra de estas medidas.*

- *Para cada jugador $i \in N$, una partición $\mathcal{I}_i$ de $\{h \in H : P(h) = i\}$ con la propiedad que $A(h) = A(h')$ siempre que h y h' estén en el mismo bloque de la partición. Para $I_i \in \mathcal{I}_i$ denotamos por $A(I_i)$ el conjunto $A(h)$ y por $P(I_i)$ el jugador $P(h)$ para*

cualquier $h \in I_i$. $\mathcal{I}_i$ es la ***partición de información*** *del jugador i, un conjunto $I_i \in \mathcal{I}_i$ es un* ***conjunto de información*** *del jugador i.*

- *Para cada jugador $i \in N$, una función de utilidad u_i de los estados terminales Z a los reales $\mathbb{R}$. Si $N = \{1, 2\}$ y $u_1 = -u_2$, decimos que tenemos un* ***juego de dos jugadores de suma cero en forma extensa****. Definimos $\Delta_{u,i} = \max_z u_i(z) - \min_z u_i(z)$ como el rango de utilidades del jugador i.*

En el Ejemplo 2.2, $H = \{\emptyset, L, R, LA, LB, LAl, LAr, LBl, LBr\}$, note que la cantidad de elementos en H coincide con la cantidad de nodos del árbol. En efecto, en un árbol para cualquier nodo u existe un camino único desde la raíz hasta u. Además, $P(\emptyset) = P(LA) = P(LB) = 1$, y $P(L) = 2$. Las particiones de información son $\mathcal{I}_1 = \{\{\emptyset\}, \{LA, LB\}\}$ y $\mathcal{I}_2 = \{\{L\}\}$. En la definición se incluye un elemento que no está presente en el ejemplo, los **nodos de azar**. Estos nodos corresponden a acciones que no dependen de los jugadores, sino de algún evento externo aleatorio, como el lanzamiento de una moneda, lanzamiento de uno o más dados, o la repartición de cartas en un juego.

Ejemplo: Juego de Kuhn Poker

Kuhn Poker es una versión simplificada del juego de Poker con tres cartas y dos jugadores (denominados jugador 1 y jugador 2) definido por Harold W. Kuhn [6]. En este juego se barajan tres cartas marcadas con los números 1, 2 y 3. Posteriormente, cada jugador recibe una de ellas, manteniendo su número como información privada. Es decir, un jugador sabe su propio número pero no sabe el número de su oponente. Al inicio del juego cada jugador apuesta una ficha. El juego ocurre por turnos, los cuales se alternan entre los jugadores comenzando por el jugador 1. En un turno un jugador puede *apostar* o *pasar*. Si un jugador apuesta debe apostar una ficha adicional. Si un jugador pasa después de una apuesta, el oponente gana y toma todas las fichas apostadas. Si hay dos apuestas o dos pases seguidos los jugadores muestran sus cartas y gana el jugador con el número más alto obteniendo todas las fichas apostadas. La Tabla 2.1 presenta un resumen de todas las posibles secuencias con su respectivo pago a cada jugador.

Debido a qué es un juego de suma 0, el jugador perdedor pierde el número de fichas que gana su oponente. El árbol del juego se muestra en la Figura 2.3. La raíz es un *nodo de azar*, que representa la repartición de las cartas, con 6 opciones diferentes, las cuales están representadas con un par ordenado indicando la carta del jugador 1 y la del jugador 2. Cada rama tiene una probabilidad de $\frac{1}{6}$ de ser elegida. Los nodos del primer nivel y tercer nivel corresponden al jugador 1. Este jugador tiene 6 conjuntos de información diferentes,

Tabla 2.1: Resumen de las posibles secuencias del juego Kunh Poker

Secuencia de Acciones			
Jugador 1	Jugador 2	Jugador 1	Pago
pasar	pasar	pasar	+1 al jugador con la carta más alta
	apostar		+1 al jugador 2
	apostar	apostar	+2 al jugador con la carta más alta
apostar	pasar		+1 al jugador 1
	apostar		+2 al jugador con la carta más alta

cada uno con 2 nodos, los cuales se unen mediante las líneas punteadas. Los nodos del segundo nivel corresponden al jugador 2, los conjuntos de información se representan por nodos del mismo color y mismo estilo (relleno de color o no). En cada nodo de decisión (los nodos no terminales sin incluir la raíz), hay dos opciones: pasar, representado con una línea punteada, o apostar, representado con una línea doble. Los nodos terminales tienen la ganancia del jugador 1, según sea el caso.

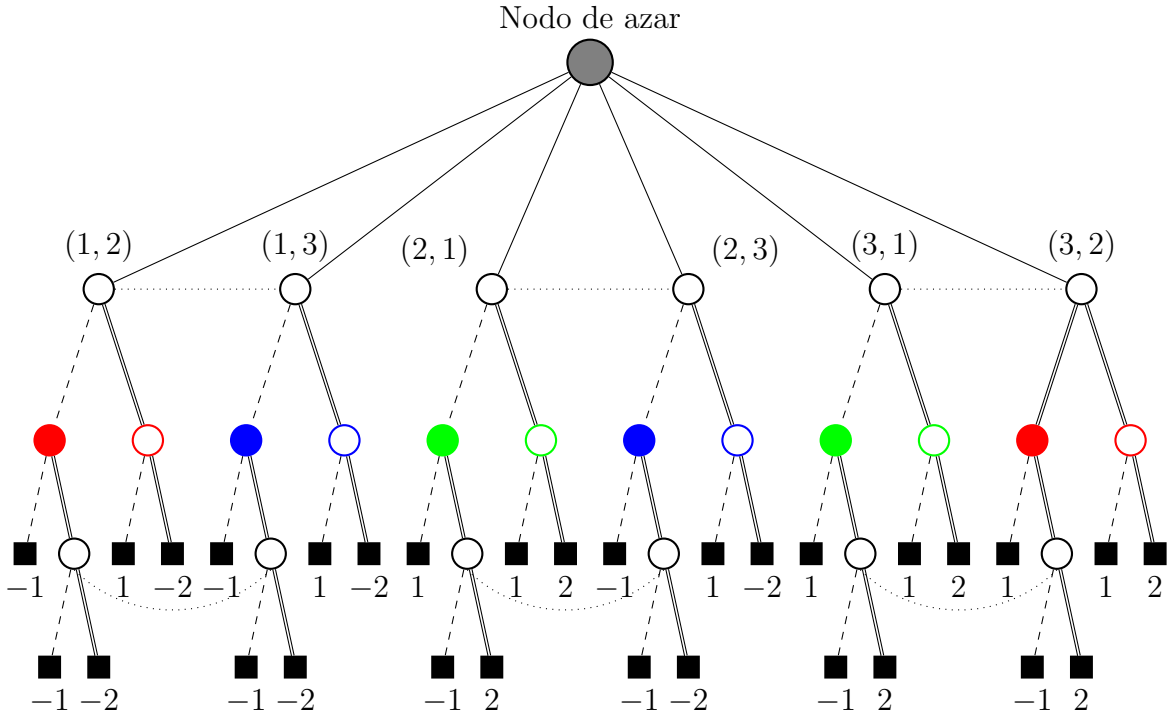
2.1. Estrategias Puras y Mixtas para Juegos en Forma Extensa

Al igual que en juegos en forma normal es necesario establecer las definiciones de estrategias. Las Definiciones 2.4 y 2.5, presentan los conceptos de estrategia pura y estrategia mixta, análogas a las presentadas en los juegos en forma normal. Las definiciones de perfiles estratégicos son equivalentes a las anteriores pero usando los conceptos de estrategias para juegos en forma extensa. Además, se procura utilizar una notación similar a la utilizada en la sección anterior. Sin embargo, se presenta un nuevo concepto, las **estrategias de comportamiento**, que son exclusivas para juegos en forma extensa. A continuación seguimos la formulación de [5] y [4].

Definición 2.4. *Una* ***estrategia pura*** *para el jugador i es una función $s_i : \mathcal{I}_i \to \bigcup_{I_i \in \mathcal{I}_i} A(I_i)$ tal que $s_i(I_i) \in A(I_i)$, donde $A(I_i) = A(h)$ para cualquier $h \in I_i$.*

Note que una estrategia pura consiste en elegir una acción por cada conjunto de información de un jugador en específico. Considere nuevamente el Ejemplo 2.2. En este juego el jugador 1 tiene dos conjuntos de información, $I^1 = \{\emptyset\}$ e $I^2 = \{LA, LB\}$, cada uno con dos posibles elecciones, dando lugar a 4 estrategias puras que son denotadas por s_1, s_2, s_3 y s_4, y presentadas en la Tabla 2.2. En dicha tabla las acciones posibles en el conjunto de información I^1 están representadas por las filas, y las acciones en I^2 por las columnas. De

Figura 2.3: Árbol completo del juego Kunh Poker.



esta forma cada celda representa una única estrategia pura determinada por una acción en cada conjunto de información.

Tabla 2.2: Estrategias puras para el juego con información incompleta presentado en el Ejemplo 2.2.

		I_2	
		l	r
I_1	L	s_1 = elegir L y l	s_2 = elegir L y r
	R	s_3 = elegir R y l	s_4 = elegir R y r

En Kunh Poker una estrategia pura para el jugador 2 puede ser la siguiente: si su carta contiene el número 1 siempre pasa, si su carta contiene el número 2 apuesta si y sólo si el jugador 1 pasa en su primer turno, y si su carta contiene el número 3 siempre apuesta. La Tabla 2.3 presenta cada conjunto de información de forma explícita con su acción correspondiente. Para este juego se caracterizarán los conjuntos de información del jugador 2 por la carta que tiene y la acción realizada por el primer jugador al inicio del juego.

Tabla 2.3: Ejemplo de una estrategia pura para el jugador 2 en el juego Kunh Poker.

Conjunto de Información		Acción del jugador 2
Carta del jugador 2	Acción del jugador 1	
1	pasar	pasar
1	apostar	pasar
2	pasar	apostar
2	apostar	pasar
3	pasar	apostar
3	apostar	apostar

Se denotará, al igual que en los juegos en forma normal, con S_i al conjunto de estrategias puras del jugador i, es decir $S_i = \prod_{I_i \in \mathcal{I}_i} A(I_i)$. Análogamente, se denotará con $S = \prod_{i \in N} S_i$ el conjunto de todas las estrategias puras de todos los jugadores de forma simultánea. Un elemento $s \in S$ es llamado un **perfil estratégico**.

Otra definición de interés es la función de pago para una estrategia pura. Para esto se denotará con $\pi^s(h)$ la probabilidad que $h \in H$ ocurra si todos los jugadores juegan con la estrategia s. Luego, definimos $u_i : S \to \mathbb{R}$ como la esperanza de la función de pago para el jugador i para cada perfil estratégico, la cual viene dada por:

$$u_i(s) \;=\; \sum_{z \in Z} \pi^s(z) u_i(z)\,. \tag{2.1}$$

Definición 2.5. *Una **estrategia mixta** σ_i^m para el jugador i es una distribución de probabilidad sobre S_i. Es decir, $\sigma_i^m \in \Delta(S_i)$.*

Definición 2.6. *Una **perfil estratégico mixto** $\sigma^m \in \prod_{i \in N} \Delta(S_i)$ consiste en una estrategia mixta para cada jugador de forma $\sigma^m = (\sigma_1^m, \sigma_2^m, ..., \sigma_N^m)$.*

Un perfil estratégico mixto indica que cada jugador elige, antes de que el juego comience, un plan completo (es decir, una estrategia pura) de forma aleatoria acorde a cierta distribución de probabilidad (que está dada por su estrategia mixta respectiva).

Si $\sigma^m \in \prod_{i=1}^{N} \Delta(S_i)$ es un perfil estratégico mixto, la ganancia esperada del jugador i, cuando todos los jugadores juegan acorde a σ^m viene dada por:

$$u_i(\sigma^m) \;=\; \sum_{s \in S} \sigma^m(s) u_i(s) \tag{2.2}$$

donde $\sigma^m(s)$ es la probabilidad de que s sea elegida, es decir $\sigma^m(s) = \prod_{i \in N} \sigma_i^m(s_i)$.

Ejemplo: Equilibrio de Nash en el Juego de Kunh Poker

La descripción del juego se encuentra en la sección II. Con respecto a la solución, se tiene que el jugador 2 tiene una ganancia esperada de $\frac{1}{18}$ por mano, como se prueba en [6], si ambos jugadores juegan de forma óptima (es decir, acorde a un equilibrio de Nash). El equilibrio de Nash se resume en la Tabla 2.4, donde los conjuntos de información fueron enumerados en un orden de búsqueda por profundidad (dfs).

I	Equilibrio de Nash	
1	$1-\alpha$	α
$2, 3, 6, 10$	1	0
4	$\frac{2}{3}$	$\frac{1}{3}$
$5, 7, 12$	0	1
8	$\frac{2}{3}$	$\frac{1}{3}$
9	$\frac{2}{3}-\alpha$	$\alpha+\frac{1}{3}$
11	$1-3\alpha$	3α

Tabla 2.4: Equilibrio de Nash para el juego de Kuhn Poker

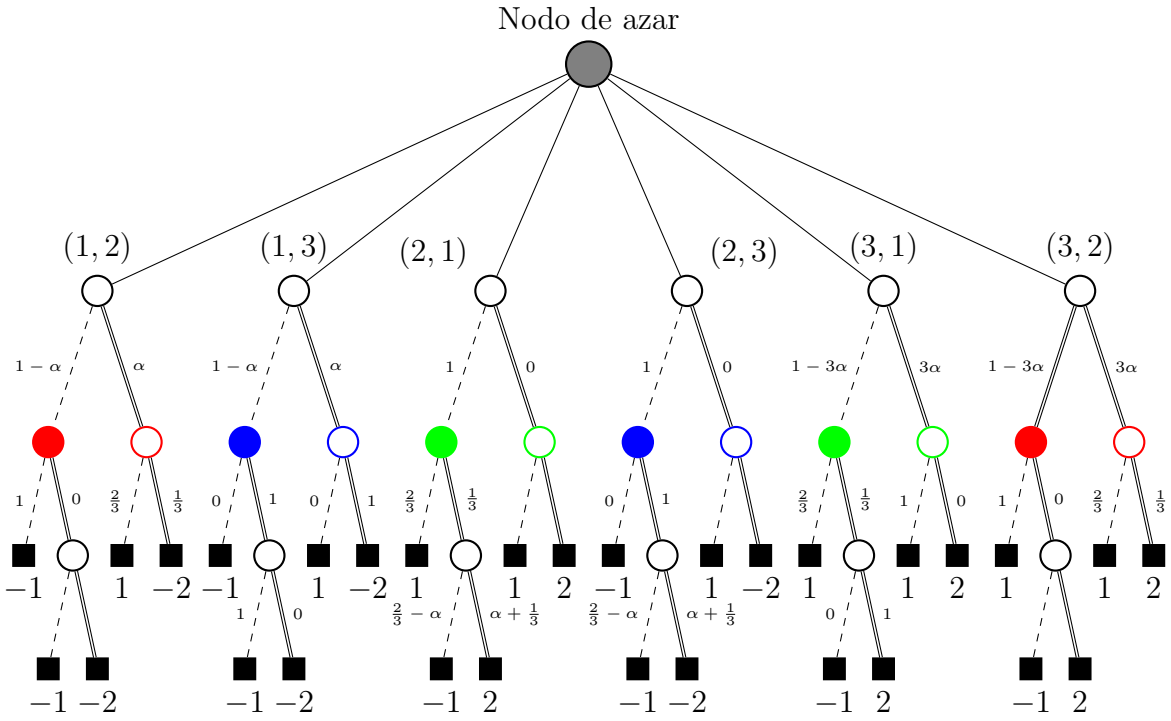
El primer jugador tiene infinitas estrategias óptimas, las cuales pueden ser representadas por la elección de un parámetro $\alpha \in [0, \frac{1}{3}]$. Una vez elegido este parámetro, el primer jugador en su primera jugada debe apostar con probabilidad α cuando su carta tenga el número 1, apuestar con una probabilidad 3α cuando tenga el número 3 y pasar siempre cuando tenga el número 2. Si el primer jugador tiene un segundo turno, debe pasar siempre que tenga el número 1, apostar cuando tiene el número 3, y en el caso que tenga el número 2 debe apostar con probabilidad $\alpha + \frac{1}{3}$.

El segundo jugador tiene una única estrategia mixta óptima: apostar siempre que tenga el número 3. Cuando tenga el número 1, pasar siempre que el primer jugador haya apostado y apostar con probabilidad $\frac{1}{3}$ en caso contrario. Cuando tenga el número 2, debe pasar cuando el oponente haya pasado previamente y apostar con probabilidad $\frac{2}{3}$ en caso contrario. La figura 2.4 muestra el árbol con las distribuciones de probabilidad de las estrategias previamente descritas en cada uno de los nodos alcanzables en el juego.

2.2. Forma Normal vs. Forma Extensa

Un juego en forma normal se caracteriza por el conjunto de estrategias puras S_i y la función de pago u_i para cada jugador $i \in N$. Estos elementos pueden obtenerse a partir de la descripción de un juego en forma extensiva utilizando las definiciones 2.4 y 2.1. De

Figura 2.4: Equilibrio de Nash del juego de Kunh poker



esta forma, es posible asociar un único juego en forma normal a cualquier juego en forma extensa.

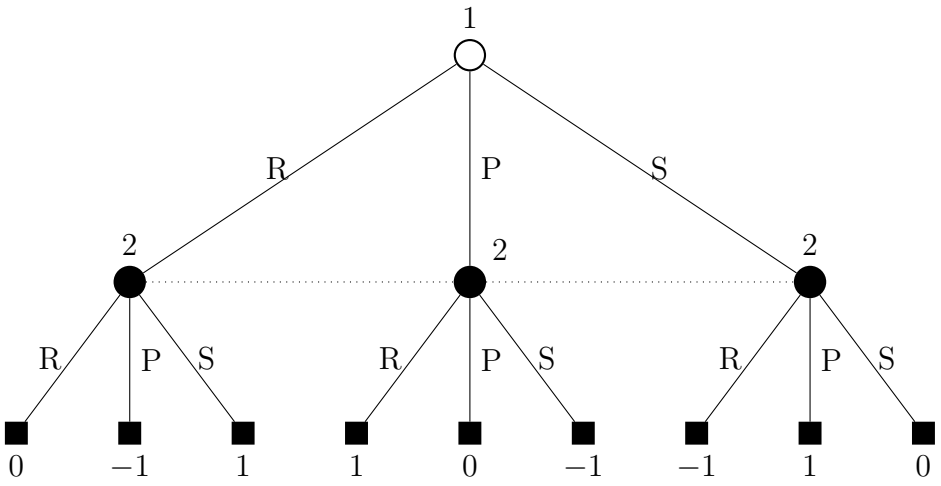
En el Ejemplo 2.2, las estrategias puras para el jugador 1 están definidas en la Tabla 2.2. El jugador dos tiene sólo dos estrategias puras, elegir A o B. Luego, la Tabla 2.5 es la tabla de pagos para juego en forma normal que corresponde al Ejemplo 2.2.

Tabla 2.5: Tabla de pagos de la forma normal correspondiente a la forma extensa del juego presentado en el Ejemplo 2.2.

	Jugador 2	
Jugador 1	Elegir A	Elegir B
Elegir L y l	$0, 0$	$2, 4$
Elegir L y r	$2, 4$	$0, 0$
Elegir R y l	$1, 1$	$1, 1$
Elegir R y r	$1, 1$	$1, 1$

Note que la tabla obtenida tiene 8 configuraciones a pesar que el árbol original tiene sólamente 5 nodos terminales. En general, la forma normal tiene un tamaño exponencial

Figura 2.5: Árbol de la forma extensiva del juego *Piedra, Papel o Tijeras*

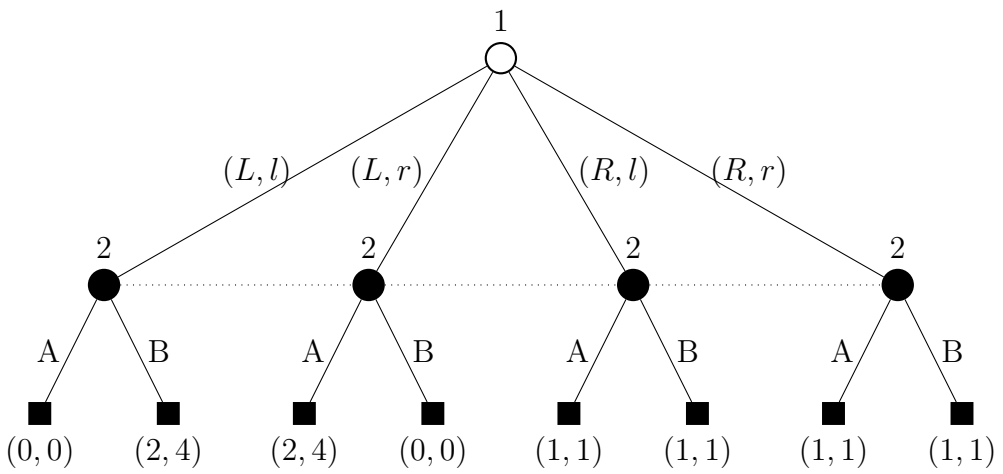


en el tamaño del árbol del juego en forma extensiva. Se observa que más de una celda (o una estrategia pura) lleva al mismo nodo terminal. Esto ocurre cuando el primer jugador elige R, en este caso no importa la segunda elección del primer jugador, ni la estrategia utilizada por el segundo jugador, pues el juego siempre terminará luego que el jugador 1 elija R. Por esto la forma normal de un juego es potencialmente más grande que su forma extensiva.

Por otra parte, dado un juego en forma normal, siempre es posible construir el árbol de una forma extensiva como sigue [5]: se comienza por la raíz, la cual es el único nodo del jugador 1, de ésta salen $|S_1|$ ramas, una para cada estrategia pura $s_1 \in S_1$, estos nodos, los hijos de la raíz, serán los nodos del jugador 2. De cada uno de los nodos del jugador 2 salen $|S_2|$ ramas, una por cada elemento $s_2 \in S_2$ que serán los hijos del jugador 3, y así sucesivamente hasta llegar a los hijos de los nodos del jugador N, que serán los nodos terminales. La Figura 2.5, muestra el árbol para una forma extensiva del juego piedra (R), papel (P) o tijera (S).

Pueden haber diferentes formas extensivas que lleven a la misma forma normal. En efecto, si aplicamos el procedimiento descrito anteriormente a la Tabla 2.5 se obtiene un árbol de 13 nodos (Figura 2.6), en contraste a los 9 nodos del árbol original. En efecto, la forma extensiva proporciona más información sobre los juegos que la forma normal. En especial, la forma extensiva proporciona información acerca del orden y las posibles secuencias de acciones.

Figura 2.6: Árbol correspondiente a la forma normal de la Tabla 2.5



2.3. Estrategias de Comportamiento

En juegos en forma extensiva, el jugador puede utilizar un tipo de estrategia diferente a la presentada anteriormente, y la cual es denominada estrategia de comportamiento (Definición 2.7). Una estrategia de comportamiento para el jugador i especifica una distribución de probabilidad sobre las acciones disponibles en cada conjunto de información del jugador i. Esto difiere a las estrategias mixtas que representan una distribución de probabilidad sobre las estrategias puras de un jugador [4, p. 212].

Definición 2.7. *Una **estrategia de comportamiento** para el jugador i consiste en una distribución de probabilidad para cada conjunto de información $I_i \in \mathcal{I}_i$ sobre el conjunto $A(I_i)$ que pueden ejecutarse en I_i. Es decir, una estrategia de comportamiento es una tupla $(\sigma_i^b(I_i))_{I_i \in \mathcal{I}_i}$ donde $\sigma_i^b(I_i) \in \Delta(A(I_i))$.*

Sea $B^i = \prod_{I_i \in \mathcal{I}_i} \Delta(A(I_i))$ el conjunto de todas las posibles estrategias de comportamiento del jugador i. Si $\sigma_i^b \in B^i$, $\sigma_i^b(I_i) \in \Delta(A(I_i))$ es una distribución de probabilidad sobre $A(I_i)$ mientras que $\sigma_i^b(I_i)(a)$ es la probabilidad de elegir la acción a dada una historia $h \in I_i$.

Definición 2.8. *Una **perfil estratégico de comportamiento** σ^b es una estrategia de comportamiento para cada jugador.*

El conjunto de todos los perfiles estratégicos de comportamiento es $B = \prod_{i \in N} B^i$. Si

$\sigma^b \in B$, la utilidad esperada de la estrategia σ^b para el jugador i es

$$u_i(\sigma^b) \;=\; \sum_{s \in S} \sigma_b(s) u_i(s)$$

donde $\sigma_i^b(s_i) = \prod_{I_i \in \mathcal{I}_i} \sigma_i^b(I_i)(s_i(I_i))$, y $\sigma^b(s) = \prod_{i \in N} \sigma_i^b(s_i)$.

Con las definiciones proporcionadas se puede definir los conceptos de equilibrio de Nash y aproximación de equilibrio de Nash para juegos en forma extensiva.

Definición 2.9. *Sea $\Sigma = \prod_{i \in N} \Sigma_i$ el conjunto de perfiles mixtos o de comportamiento, según sea el caso, para los jugadores en N. Para $\varepsilon \geq 0$, decimos que un perfil estratégico $\sigma \in \Sigma$ es un ε-**equilibrio de Nash** si y sólo si para todo jugador i y perfil $\sigma_i' \in \Sigma_i$,*

$$u_i(\sigma) + \varepsilon \;\geq\; u_i(\sigma_i', \sigma_{-i})\,. \tag{2.3}$$

*El perfil $\sigma \in \Sigma$ es un **equilibrio de Nash** si y sólo si σ es un 0-equilibrio de Nash.*

En el Ejemplo 2.2 se tienen 4 estrategias puras para el jugador 1 (Tabla 2.2). Una estrategia mixta σ_1^m es una distribución de probabilidad sobre el conjunto $\{s_1, s_2, s_3, s_4\}$, donde las probabilidades son $\sigma_1^m(s_1)$, $\sigma_1^m(s_2)$, $\sigma_1^m(s_3)$ y $\sigma_1^m(s_4)$. Por otra parte una estrategia de comportamiento σ_1^b son dos distribuciones de probabilidad, $\sigma_1^b(I^1)$ y $\sigma_1^b(I_2)$, sobre los conjuntos $A(I^1) = \{L, R\}$ y $A(I^2) = \{l, r\}$ respectivamente.

Sea σ un perfil estratégico mixto o de comportamiento. Sea $\sigma_{-i} = (\sigma_j)_{j \neq i}$ la combinación de todas las estrategias de σ excepto σ_i. Sea $\pi^{\sigma_i}(h)$ la probabilidad de alcanzar h dado que el jugador i utiliza la estrategia σ_i y que los demás jugadores juegan para alcanzar h. Note que $\pi^{\sigma_i}(h)$ es la probabilidad de que para todo prefijo propio $h' \sqsubset h$ tal que $P(h') = i$, el i-ésimo jugador elija la acción a correspondiente en h; i.e., $(h', a) \sqsubseteq h$.

Sea $\pi^c(h)$ la probabilidad de alcanzar h asumiendo que todos los jugadores juegan para alcanzar h, es decir:

$$\pi^c(h) = \prod_{(h', a) \sqsubset h : P(h') = c} f_c(a|h')\,. \tag{2.4}$$

Sea $\pi^\sigma(h) = \prod_{i \in N \cup c} \pi^{\sigma_i}(h)$ la probabilidad de que la historia h ocurra si todos los jugadores eligen las acciones acorde al perfil estratégico σ. Sea $\pi^{\sigma_{-i}}(h) = \frac{\pi^\sigma(h)}{\pi^{\sigma_i}(h)}$ el producto de todas las contribuciones de los jugadores (incluyendo las elecciones en las historias de azar) excepto el jugador i. Sean $\pi^\sigma(I) = \sum_{h \in I}, \pi^\sigma(h)$, $\pi^{\sigma_i}(I) = \sum_{h \in I}, \pi^{\sigma_i}(h)$ y $\pi^{\sigma_{-i}}(h) = \sum_{h \in I}, \pi^{\sigma_{-i}}(h)$ la probabilidad de alcanzar el conjunto de información I dado la estrategia σ, σ_i y σ_{-i}, respectivamente.

Para un perfil estratégico σ, sea $u_i(\sigma) = \sum_{z \in Z} u_i(z)\pi^\sigma(z)$ la ganancia esperada del jugador i cuando todos los jugadores utilizan el perfil estratégico σ. Sea $I(h)$ el conjunto de información al que pertenece h, como los conjuntos de información son una partición de las historias correspondientes a cada jugador $I(h)$ está bien definida. La notación presentada es la utilizada en [7].

2.4. Perfect Recall

El concepto de *perfect recall* hace referencia a juegos en los cuales, en cualquier punto cualquier jugador *recuerda* lo que sabía previamente [4, p. 203]. En particular, cada jugador recuerda los movimientos que hizo previamente. La definición de *perfect recall* puede ser dada mediante el árbol del juego [5] o mediante la subsecuencia correspondiente a los nodos de un jugador [4, p. 203] y [8, p. 44]. Sin embargo, se utiliza una definición equivalente, proporcionada en la Definición 2.10.

Definición 2.10. *Se dice que el jugador i tiene* ***perfect recall*** *en el juego Γ (en forma extensiva) si para cualquier par de historias h_1, h_2 con $P(h_1) = P(h_2) = i$, tales que $I(h_1) = I(h_2)$ las siguientes condiciones se cumplen*

$$h \sqsubseteq h_1 \;\Rightarrow\; (\exists h' \sqsubseteq h_2 : I(h) = I(h')) \tag{2.5}$$

$$(h_1, a) \sqsubseteq h \;\wedge\; (h_2, b) \sqsubseteq h' \;\wedge\; a \neq b \;\Rightarrow\; I(h) \neq I(h') \tag{2.6}$$

Intuitivamente, las condiciones presentadas representan las siguientes propiedades del jugador i:

1. *El jugador i recuerda lo que sabía* (Ecuación 2.5): en cualquier momento el jugador i recuerda si pasó o no por un conjunto de información específico. En efecto, si dos secuencias, digamos h_1 y h_2 pertenecen al mismo conjunto de información y para llegar a h_1 se debe pasar por h, entonces, para llegar a h_2, se debe pasar por algún h' tal que h' y h pertenezcan al mismo conjunto de información.
2. *El jugador i recuerda lo que eligió* (Ecuación 2.6): si desde una historia h el jugador elige a estará siempre en un conjunto de información diferente si en ese punto hubiese elegido la acción $b \neq a$.

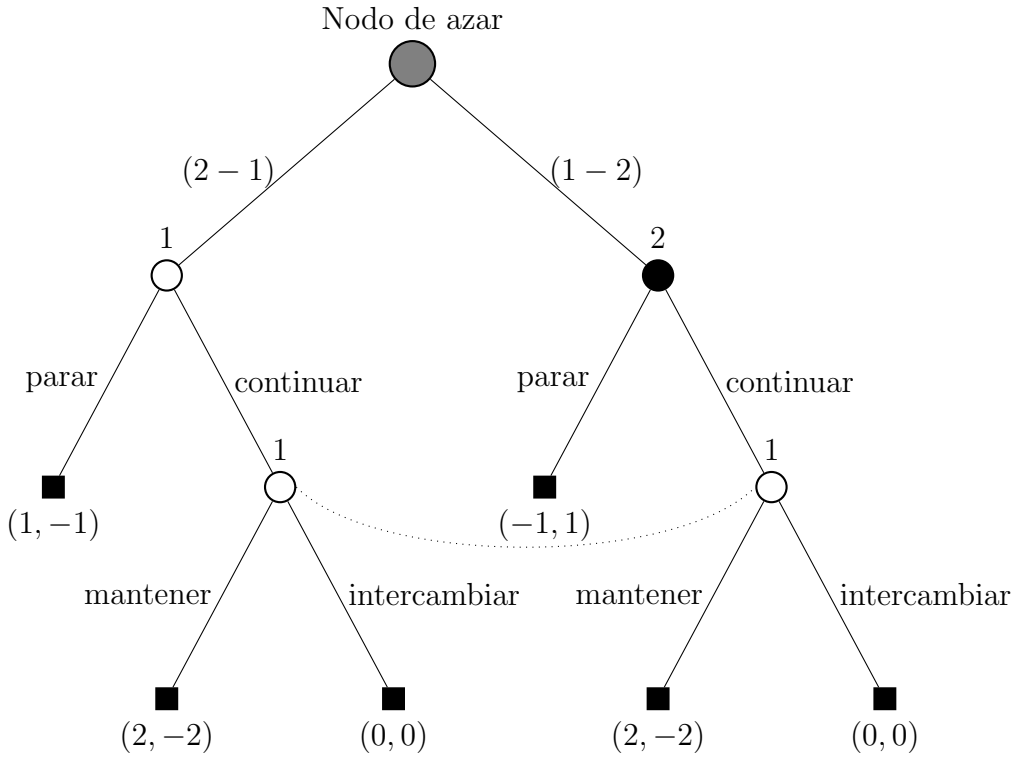
Los juegos presentados previamente: Ejemplo 2.1, Ejemplo 2.2 y Kunh Poker son todos juegos con *perfect recall*. El Ejemplo 2.11 muestra un juego con *imperfect recall*.

Ejemplo 2.11. *([5]) Considere un juego de dos jugadores de suma cero en el cual el jugador 1 consta de 2 jugadores: Alice y su esposo Bill, y el jugador 2 consta de una sóla*

persona: Zeno. Se tienen dos cartas con los números 1 *y* 2 *y son repartidas aleatoriamente entre Alice y Zeno. La persona con la carta más alta recibe 1$ de la persona con la carta más baja, y ésta decide si seguir jugando o no. Si el juego continúa, Bill, sin saber el resultado de la repartición inicial de las cartas, decide si Alice y Zeno intercambian sus cartas o no. Nuevamente, quien posea la carta más alta recibe 1$ de quien posea la carta más baja.*

La Figura 2.7 representa el juego en forma extensiva. Note que cuando es el turno de Bill, él no sabe quien tiene la carta más alta, cosa que su esposa sí sabía en el turno anterior. Al considerar a la pareja como un sólo jugador, se obtiene que el jugador 1 *olvidó* como fueron repartidas las cartas. En efecto, el jugador 1 tiene dos conjuntos de información $I_1^1 = \{(2-1)\}$ e $I_1^2 = \{(2-1, \mathit{continuar}), (1-2, \mathit{continuar})\}$. En particular, no se cumple la primera condición (Ecuación 2.5), pues la secuencia $(2-1) \sqsubset (2-1, \mathit{continuar})$, pero no existe una subsecuencia de $(1-2, \mathit{continuar})$ que pertenezca a I_1^1.

Figura 2.7: Árbol de la forma extensiva del juego con *imperfect recall* presentado en el Ejemplo 2.11



Una pregunta de interés es si es posible sustituir una estrategia mixta por una estrategia de comportamiento o viceversa. Para esto, es necesario establecer la definición equivalencia

entre estrategias (Definición 2.12) y alcanzabilidad de una historia bajo una estrategia pura (Definición 2.13).

Definición 2.12. *Dadas dos estrategias σ y σ', se dice que son equivalentes si la probabilidad de alcanzar cualquier historia terminal es la misma, es decir si $\pi^{\sigma}(z) = \pi^{\sigma'}(z)$ para todo $z \in Z$.*

Definición 2.13. *Sea $s_i \in S_i$ y $I_i \in \mathcal{I}$, diremos que I_i es alcanzable bajo s_i si $\exists h \in H$ tal que $h \in I_i$ y para toda historia $h' \sqsubset h$, con $P(h') = i$, se tiene que $(h', s_i(h'))$ también es prefijo de h.*

Note que la definición anterior puede ser aplicada tanto a perfiles estratégicos como a estrategias para un jugador en particular, utilizando la definición de $\pi^{\sigma}(z)$ correspondiente.

Luego, las preguntas que se desean responder son las siguientes: (i) ¿Dada una estrategia mixta σ^m, existe una estrategia de comportamiento σ^b tal que σ^m y σ^b son equivalentes? (ii) ¿Dada una estrategia de comportamiento σ^b existe una estrategia mixta σ^m tal que σ^b y σ^m son equivalentes?. Los Teoremas A.7 y A.8 ofrecen respuestas a estas interrogantes.

El Teorema A.7 establece lo siguiente: si para cualquier camino de la raíz a un nodo no se atraviesa 2 veces o más el mismo conjunto de información, entonces para cualquier estrategia de comportamiento existe una estrategia mixta equivalente. Por otra parte, el Teorema A.8 estable que si todos los jugadores tienen *perfect-recall* entonces para toda estrategia mixta existe una estrategia de comportamiento equivalente.

En particular, si se tiene *perfect-recall* entonces ningún camino pasa por el mismo conjunto de información más de una vez y, por lo tanto, para cualquier estrategia de comportamiento también existe una estrategia de comportamiento equivalente. En efecto, las estrategias de comportamiento es una forma compacta de representar las estrategias en este tipo de juegos.

Teorema 2.14. *Dado un juego en forma extensa y un jugador i, tal que: si $h' \sqsubset h$ y $P(h') = P(h) = i$, entonces $I(h') \neq I(h)$. Luego, para cualquier estrategia de comportamiento $\sigma_i^b \in B^i$, la estrategia mixta σ_i^m dada por:*

$$\sigma_i^m(s_i) := \prod_{I_i \in \mathcal{I}_i} \sigma_i^b(I_i)(s_i(I_i)) \tag{2.7}$$

es equivalente a la estrategia σ_i^b.

Antes de presentar la demostración, se mostrará un ejemplo de un juego en el que no se cumple la condición del Teorea A.7, es decir un juego en el que una historia atraviese más de una vez el mismo conjunto de información. En este tipo de juegos se obtiene que

el poder expresivo de una estrategia mixta y el de una estrategia de comportamiento no son comparables. El Ejemplo 2.15 [8, p. 44] muestra esta situación.

Ejemplo 2.15. *Considere un juego de dos jugadores tal que:*

$$H = \{\emptyset, (L), (R), (L, L), (L, R), (R, U), (R, D)\} \tag{2.8}$$

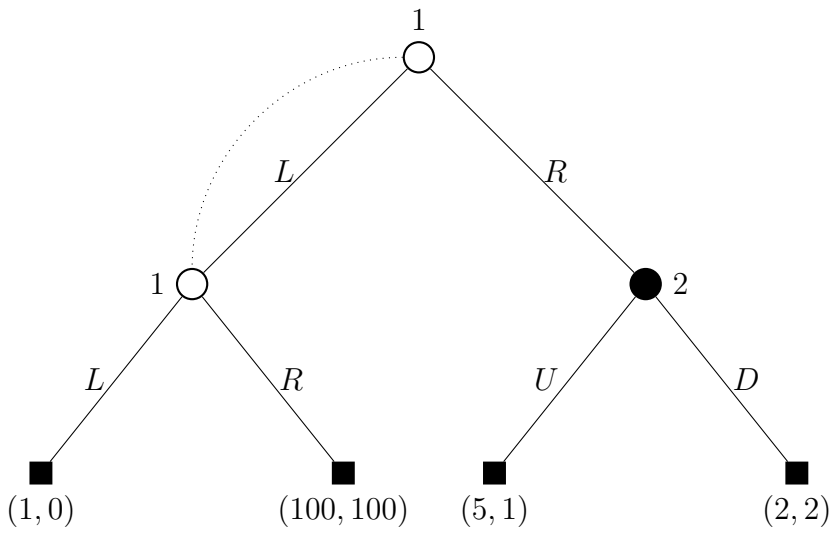
$$P(\emptyset) = P(L) = 1,\ P(R) = 2 \tag{2.9}$$

$$f(L, L) = (1, 0),\ f(L, R) = (100, 100),\ f(R, U) = (5, 1),\ f(R, D) = (2, 2) \tag{2.10}$$

$$\mathcal{I}_1 = \{\{\emptyset, (L)\}\},\ \mathcal{I}_2 = \{\{R\}\} \tag{2.11}$$

Este juego de imperfect recall corresponde al árbol de la Figura 2.8.

Figura 2.8: Árbol de la forma extensiva del juego con *imperfect recall* presentado en el Ejemplo 2.15



Note que, en efecto, la historia (L) atraviesa 2 veces el único conjunto de información del jugador 1, este jugador *olvida* si ya hizo la elección entre L o R previamente. En este juego el jugador 1 tiene 2 posibles estrategias puras, elegir L o R. Por lo tanto, en una estrategia mixta el elige una de estas 2 acciones según alguna distribución de probabilidad, si embargo, luego de la elección siempre realizará la misma jugada cuando pase por ese conjunto de información. En particular, la historia (L, R) no puede ocurrir y el pago de 100 es irrelevante en el contexto de estrategias mixtas.

En este juego en particular, se tiene que la estrategia R es mejor para el jugador 1, independientemente de la elección del jugador 2 y la estrategia pura D del jugador 2 es

mejor respuesta ante cualquier estrategia de 1. Luego, el único equilibrio de Nash (de estrategias mixtas) es (σ_1, σ_2), donde $\sigma_1(L) = 0$, $\sigma_1(R) = 1$, $\sigma_2(D) = 1$ y $\sigma_2(U) = 0$, cuya ganancia es igual a 2 para ambos jugadores.

Por otra parte, si consideramos estrategias de comportamiento, se debe elegir una distribución $(p, 1-p)$ para elegir L y R. En este caso la historia (L, R) tiene una probabilidad de $p(1-p)$ de ser elegida y su pago juega un papel relevante al momento de elegir la estrategia óptima. De esta forma, la estrategia mencionada previamente ya no es un equilibrio en estrategias de comportamientos y se obtiene el equilibrio cuando $p = \frac{98}{198}$ y el jugador 2 elige D. Note que en el Ejemplo 2.15, para esta estrategia de comportamiento, no existe una estrategia mixta equivalente, sin embargo, cuando se cumple la condición del Teorema A.7, sí lo es. La prueba se presenta a continuación.

El Teorema A.8 indica cuando es posible encontrar una estrategia mixta equivalente a una estrategia de comportamiento, lo cual ocurre cuando el juego tiene *perfect recall*.

Antes de presentar el siguiente teorema considere nuevamente el Ejemplo 2.11, el cual no tiene *perfect recall*. Las estrategias puras para el jugador 1 en este juego son: (parar, intercambiar), (parar, mantener), (continuar, intercambiar), (continuar, mantener). Las estrategias puras para el jugador 2 son: (parar) y (continuar). Luego, la Tabla 2.6 es la tabla correspondiente a la forma normal del juego, la cual incluye sólo la función del pago del jugador 1, ya que al ser un juego de suma 0, el pago del jugador 2 se obtiene trivialmente.

Tabla 2.6: Tabla de la forma normal para el Juego 2.11 con imperfect recall

	(parar)	(continuar)
(parar, mantener)	0	$-\frac{1}{2}$
(parar, intercambiar)	0	$\frac{1}{2}$
(continuar, mantener)	$\frac{1}{2}$	0
(continuar, intercambiar)	$-\frac{1}{2}$	0

Note que el pago que proporciona la estrategia (parar, intercambiar) al jugador 1 es no menor que el pago que le proporciona la estrategia (parar, mantener), sin importar lo que juegue el jugador 2. Asimismo, para éste jugador, la estrategia (continuar, mantener) es mejor que la estrategia (continuar, intercambiar). Esto puede motivar al jugador 1 a elegir una estrategia mixta en la que la probabilidad asignada a las estrategias (parar, mantener) y (continuar, intercambiar) sea 0. Supongamos que la estrategia consiste en elegir las estrategias (parar, intercambiar) y (continuar, mantener) con una probabilidad de $\frac{1}{2}$ cada una. La interrogante planteada es la siguiente ¿Existirá una estrategia de comportamiento

equivalente a esta estrategia mixta?

La respuesta a la interrogante es no. Para observarlo, considere una estrategia de comportamiento en la que se elige parar con probabilidad α y mantener con una probabilidad β. La probabilidad de elegir cada una de las estrategias puras se observa en la Tabla 2.7

Tabla 2.7: Probabilidades de cada estrategia pura dada una estrategia de comportamiento para el jugador 1 del Ejemplo 2.11

	parar (α)	continuar ($1-\alpha$)
mantener (β)	$\alpha\beta$	$(1-\alpha)\beta$
intercambiar ($1-\beta$)	$\alpha(1-\beta)$	$(1-\alpha)(1-\beta)$

En la estrategia mixta deseada la estrategia pura (parar, mantener) tiene probabilidad 0, por lo que $\alpha = 0$ o $\beta = 0$ debería ser 0. Sin embargo, si $\alpha = 0$ la estrategia pura (parar, intercambiar) tiene una probabilidad 0 de ser elegida y si $\beta = 0$ entonces es imposible elegir la estrategia (continuar, mantener). Luego, no existe una estrategia de comportamiento equivalente a la estrategia mixta deseada. Sin embargo, el Ejemplo 2.11 no tiene *perfect recall*, el Teorema A.8 enuncia que siempre que se tenga *perfect recall* sí es posible.

Teorema 2.16. *Dado un juego finito de N personas en el que el jugador i tiene "perfect recall". Entonces, para cada estrategia mixta $\sigma_i^m \in \Delta(S_i)$ del jugador i, existe una estrategia de comportamiento $\sigma_i^b \in B^i$, equivalente a σ_i^m.*

Se puede observar que cuando se tiene un juego con *perfect recall*, se cumplen las condiciones de los Teoremas A.8 y A.7. Por lo tanto se pueden intercambiar estrategias mixtas por estrategias de comportamiento y viceversa sin perder poder expresivo. Esto se enuncia con el Teorema 2.17 [8, p. 45].

Teorema 2.17. *En un juego con perfect recall, cualquier estrategia mixta de un agente dado puede ser remplazada por una estrategia de comportamiento equivalente, y cualquier estrategia de comportamiento puede ser remplazada por una estrategia mixta equivalente. Dos estrategias son equivalentes en el sentido en que inducen los mismos resultados de probabilidades, para cualquier perfil estratégico fijo (mixto o de comportamiento) del resto de los agentes.*

Como corolario del teorema anterior se obtiene que el conjunto de los equilibrios de Nash o cambia si el estudio se restringe a estrategias de comportamiento. Los juegos estudiados en este trabajo presentan *perfect recall*, por lo tanto, en las próximas secciones se

restringirá el estudio a estrategias de comportamientos. Se asumirá que todas las estrategias son de comportamiento y por lo tanto se denotarán únicamente por σ (en vez de σ^b). Sin embargo, es importante resaltar nuevamente, que esta equivalencia es cierto solamente si el juego tiene *perfect recall*. En juegos generales con información incompleta, estrategias mixtas y de comportamiento mantienen conjuntos de equilibrio no comparables [8, p. 45].

CAPÍTULO III

EVALUACIÓN DE ESTRATEGIAS Y EXPLOTABILIDAD

En un juego de suma cero el **valor del juego** es igual a la ganancia esperada del primer jugador cuando los jugadores utilizan un Equilibrio de Nash $\sigma^* = (\sigma_1^*, \sigma_2^*)$. Es decir, el valor del juego es igual a $u = u_1(\sigma^*)$. Supongamos ahora que el jugador 1 usa una estrategia σ_1, que es una ligera modificación de σ^*, entonces el jugador 2 puede usar una estrategia que sea mejor respuesta a σ_1, digamos σ_2'. Luego,

$$u_2(\sigma_1, \sigma_2') \;\geq\; u_2(\sigma_1, \sigma_2^*) \;\geq\; u_2(\sigma_1^*, \sigma_2^*)\,. \tag{3.1}$$

La primera desigualdad se obtiene porque σ_2' es mejor respuesta del jugador 2 a σ_1 y la segunda desigualdad ocurre porque σ_1^* es mejor respuesta del jugador 1 a σ_2^*. Luego $u_2(\sigma_1, \sigma_2') = u_2(\sigma_1^*, \sigma_2^*) + \varepsilon_1$ para algún $\varepsilon_1 \geq 0$. Por lo tanto, la estrategia del jugador 1 se volvió *explotable* por una cantidad ε_1. De forma análoga se puede obtener que, si el jugador 2 utiliza una estrategia σ_2 ligeramente alejada del equilibrio de Nash, esta estrategia será explotable por una cantidad no negativa ε_2.

La **explotabilidad** ε_σ de una estrategia $\sigma = (\sigma_1, \sigma_2)$ es definida por la expresión $\varepsilon_\sigma = \varepsilon_1 + \varepsilon_2$. La explotabilidad es usada frecuentemente para medir la distancia de una estretegia al equilibrio de Nash [9, p. 7]. Si definimos $v_i = u_i(\sigma_i, \sigma_{-i}')$, entonces por lo anterior $v_i = u_i(\sigma^*) + \varepsilon_i$. Si $u = u_1(\sigma^*)$ es el valor del juego, note que $v_1 = u_1(\sigma^*) + \varepsilon_1 = u + \varepsilon_1$ y $v_2 = u_2(\sigma^*) + \varepsilon_2 = -u + \varepsilon_2$. Entonces, $\varepsilon_\sigma = u + \varepsilon_1 - u + \varepsilon_2 = v_1 + v_2$.

Queremos encontrar una forma sencilla de calcular la cantidad v_i. Para lograr esto utilizamos el hecho que para cualquier estrategia de cualquier jugador siempre existe una mejor respuesta cuyo soporte tiene un único elemento (Corolario del Teorema A.3). Este resultado permite obtener la siguiente expresión para v_i que permite calcular la explotabilidad ε_σ de una estrategia dada $\sigma = (\sigma_1, \sigma_2)$:

$$v_i \;=\; \max_{s_{-i} \in S_{-i}} u_i(\sigma_i, s_{-i})\,. \tag{3.2}$$

****** EJEMPLOS: RPS? ******

Estas fórmulas se usan para calcular la explotabilidad de las estrategias obtenidas al ejecutar cada uno de los procedimientos implementados en cada uno de los juegos en forma normal que se utilizan en los experimentos. ****** Qué pasa para los juegos en forma extensa? ******

CAPÍTULO IV

REGRET MATCHING

¿Hay procedimientos adaptativos simples que permitan calcular un equilibrio correlacionado? A continuación se describen tres procedimientos, dos de los cuales llevan a equilibrios correlacionados [2].

Procedimiento A: Regret condicional

Sea Γ un juego en forma normal el cual es jugado repetidamente a través del tiempo $t = 1, 2, \ldots$. Sea $h_t = (s^\tau)_{\tau=1}^t \in \prod_{\tau=1}^t S$ la historia del juego al inicio del tiempo $t+1$. El jugador $i \in N$ elige su estrategia con una distribución de probabilidad $p_{t+1}^i \in \Delta(S_i)$, definida de la siguiente manera.

Para cada par de estrategias $j, k \in S_i$, supongamos que el jugador i remplaza la estrategia j (cada vez que la jugó en el pasado) por la estrategia k. Luego, su ganancia a tiempo $1 \leq \tau \leq t$ hubiera sido:

$$W_i^\tau(j,k) \;=\; \begin{cases} u_i(k, s_{-i}^\tau) & \text{si } s_i = j\,, \\ u_i(s^\tau) & \text{en otro caso.} \end{cases} \tag{4.1}$$

La diferencia resultante en el promedio de la función de pago, denotada con $D_i^t(j,k)$, para el jugador i sería:

$$D_i^t(j,k) \;=\; \frac{1}{t}\sum_{\tau=1}^{t} W_i^\tau(j,k) - \frac{1}{t}\sum_{\tau=1}^{t} u_i(s^\tau) \;=\; \frac{1}{t}\sum_{\substack{1\leq\tau\leq t \\ s_i^\tau = j}} u_i(k, s_{-i}^\tau) - u_i(s^\tau)\,. \tag{4.2}$$

Finalmente, definimos

$$R_i^t(j,k) \;=\; [D_i^t(j,k)]^+ \;=\; \text{máx}(0, D_i^t(j,k))\,. \tag{4.3}$$

La expresión (4.3) se puede interpretar como una medida de "arrepentimiento" del jugador i de haber elegido la acción j en vez de la acción k en el pasado, y por lo tanto, dicha medida es denominada *regret*.

Fijemos un número $\mu > 0$ suficientemente grande. Sea $j \in S_i$ la última estrategia jugada por el jugador i, es decir $j = s_i^t$. Luego, la distribución de probabilidad $p_{t+1}^i \in \Delta(S_i)$ usada por el jugador i a tiempo $t+1$ es definida como:

$$\begin{cases} p_{t+1}^i(k) \;:=\; \frac{1}{\mu} R_t^i(j,k) & \text{si } k \neq j\,, \\ p_{t+1}^i(j) \;:=\; 1 - \sum_{k \in S_i, k \neq j} p_{t+1}^i(k)\,. \end{cases} \tag{4.4}$$

La distribución inicial $p_1^i \in \Delta(S_i)$, a tiempo $t = 1$, es elegida de forma arbitraria.

Para cada tiempo t, sea $z_t \in \Delta(S)$ la distribución empírica de las N-tuplas jugadas hasta tiempo t, es decir: $z_t(s) := \frac{1}{t}|\{1 \leq \tau \leq t : s^\tau = s\}|$. El siguiente teorema enuncia que el procedimiento arriba descrito produce un equilibrio correlacionado.

Teorema 4.1 ([2])**.** *Si cada jugador juega de acuerdo al procedimiento descrito por* (4.4)*, entonces la distribución empírica del juego z_t converge (a.s.) cuando $t \to \infty$ al conjunto de equilibrios correlacionado del juego Γ.*

Es importante destacar que z_t no tiene que converger necesariamente a un punto equilibrio correlacionado. El Teorema 4.1 es equivalente al siguiente enunciado: para todo $\varepsilon > 0$, existe un tiempo $T_0 = T_0(\varepsilon)$ tal que para todo $t \geq T_0$ podemos encontrar un equilibrio correlacionado ψ_t que está distancia menor que ε de z_t.

En el procedimiento descrito cada jugador tiene dos opciones en cada período: continuar jugando con la última estrategia, o cambiarla por otra estrategia cuyas probabilidades son proporcionales a cuanto mayor hubiese sido su ganancia acumulada si hubiese hecho ese cambio en el pasado. El procedimiento planteado es simple, tanto de entender y explicar, como de implementar. Además en cada período no sólo se elige la mejor respuesta, todas las respuestas mejores a la actual pueden ser escogidas con probabilidades que son proporcionales a sus ganancias aparentes (medidas por el *regret*). Este tipo de procedimientos son llamados procedimientos de *regret matching*. Por último, el procedimiento tiene inercia: la estrategia jugada previamente importa, siempre hay una probabilidad positiva de continuar jugando la misma estrategia, y más aún, sólo se cambiará de estrategia si hay una razón para hacerlo.

El *regret* juega un papel importante para la elección de la siguiente distribución de probabilidad, lo cual conlleva a la siguiente pregunta: ¿Cuál es la relación entre los *regrets* y el equilibrio correlacionado? Una condición necesaria y suficiente para que la distribución

empírica converja al conjunto de equilibrio correlacionado es que todos los *regrets* converjan a cero (Proposición A.9).

Teorema 4.2. *Sea $(s_t)_{t=1,2,...}$ una secuencia de juegos de Γ. Entonces, $R_i^t(j,k)$ converge a 0 para cada i y cada $j,k \in S_i$, con $j \neq k$, si y sólo si la secuencia de distribuciones empíricas z_t converge al conjunto de equilibrio correlacionado.*

Procedimiento B: Vector invariante de probabilidad

Este procedimiento es una variación del anterior. Sin embargo, a tiempo $t+1$ las probabilidades de transición de la estrategia utilizada por el jugador i son determinadas por la matriz estocástica (derecha) M_t^i definida en (4.4); i.e., $M_t^i(j,k) = \frac{1}{\mu} R_t^i(j,k)$ si $k \neq j$, y $M_t^i(j,j) = 1 - \frac{1}{\mu}\sum_{k \in S_i, k \neq j} R_t^i(j,k)$.

Considere un vector (fila) invariante de probabilidad q_t^i, donde $q_t^i \in \Delta(S_i)$, para la matriz M^t. Es decir, q_t^i satisface $q_t^i \times M_t^i = q_t^i$ (dicho vector siempre existe):

$$q_t^i(j) \;=\; \sum_{k \in S_i} q_t^i(k) M_t^i(k,j) \;=\; \left[\sum_{k \in S_i, k \neq j} q_t^i(k)\frac{1}{\mu} R_t^i(k,j)\right] + q_i^t(j)\left[1 - \frac{1}{\mu}\sum_{k \in S_i, k \neq j} R_t^i(j,k)\right] \tag{4.5}$$

para todo $j \in S_i$. Definamos $R_t^i(j,j) = 0$, luego:

$$q_t^i(j) \;=\; \left[\sum_{k \in S_i} q_t^i(k)\frac{1}{\mu} R_t^i(k,j)\right] + q_t^i(j)\left[1 - \sum_{k \in S_i} \frac{1}{\mu} R_t^i(j,k)\right] \tag{4.6}$$

$$\Rightarrow \quad \mu q_t^i(j) \;=\; \left[\sum_{k \in S_i} q_t^i(k) R_t^i(k,j)\right] + q_t^i(j)\left[\mu - \sum_{k \in S_i} R_t^i(j,k)\right] \tag{4.7}$$

$$\Rightarrow \quad \mu q_t^i(j) \;=\; \left[\sum_{k \in S_i} q_t^i(k) R_t^i(k,j)\right] + \mu q_t^i(j) - q_t^i(j)\sum_{k \in S_i} R_t^i(j,k)\,. \tag{4.8}$$

Por lo tanto,

$$q_t^i(j)\sum_{k \in S_i} R_t^i(j,k) \;=\; \sum_{k \in S_i} q_t^i(k) R_i^t(k,j)\,. \tag{4.9}$$

Teorema 4.3. *Supongamos que a cada período $t+1$, el jugador i elige las estrategias acorde a un vector de distribución de probabilidad q_t^i que satisface* (4.9)*. Entonces, $R_t^i(j,k)$ converge a cero (a. s.) para todo $j,k \in S_i$ con $j \neq k$.*

Procedimiento C: Regret incondicional

El tercer procedimiento no conduce necesariamente a un equilibrio correlacionado. Sin embargo es considerado "universalmente consistente" (Definición 4.4). En este procedimiento, el pago promedio del jugador i, en el límite, no es peor a el pago si él hubiese jugado cualquier estrategia constante k, para todo $\tau \leq t$.

Definición 4.4. *Un procedimiento adaptativo es* ***universalmente consistente*** *para el jugador i si:*

$$\limsup_{t\to\infty} \left[\max_{k\in S_i} \frac{1}{t} \sum_{\tau=1}^{t} u_i(k, s_{-i}^{\tau}) - \frac{1}{t} \sum_{\tau=1}^{t} u_i(s_\tau) \right] \leq 0 \quad (a.s.) \tag{4.10}$$

El procedimiento es definido a continuación. A tiempo t, definimos

$$D_i^t(k) = \frac{1}{t} \sum_{\tau=1}^{t} u_i(k, s_{-i}^{\tau}) - u_i(s_\tau)\,, \tag{4.11}$$

$$R_i^t(k) = [D_i^t(k)]^+ = \max(0, D_i^t(k))\,. \tag{4.12}$$

Luego, la distribución de probabilidad a tiempo $t+1$, $p_{t+1}^i \in \Delta(S_i)$, es definida como sigue:

$$p_{t+1}^i(k) = \frac{R_i^t(k)}{\sum_{k'\in S_i} R_i^t(k')} \tag{4.13}$$

si el denominador es positivo, y de forma arbitraria en caso contrario. Note que las probabilidades son elegidas de forma proporcional a $R_i^t(k)$ que será denominado *regret* incondicional (en contraste al *regret* condicional definido previamente).

Teorema 4.5. *El procedimiento adaptativo definido en* (4.13) *es universalmente consistente para el jugador i.*

4.1. Regret Matching y Equilibrio de Nash

En el capítulo anterior se describieron procedimientos universalmente consistente, algunos de los cuales permiten obtener equilibrios correlacionados. Sin embargo, éstos no garantizan obtener un equilibrio de Nash, surgiendo la siguiente interrogante: ¿Bajo que condiciones se puede garantizar que un procedimiento universalmente consistente conduce a un equilibrio de Nash? El Teorema A.12 responde esta pregunta.

Teorema 4.6. *Sea* Γ *un juego de dos jugadores de suma cero y sea* $(s^t)_{t=1,2,\ldots,T}$ *una secuencia de juegos de* Γ*, tales que, para todo* $s_i \in S_i$*, para todo* $i \in 1, 2$*:*

$$\frac{1}{T}\sum_{t=1}^{T} u_i(s_i, s_{-i}^t) - \frac{1}{T}\sum_{t=1}^{T} u_i(s^t) \leq \varepsilon \tag{4.14}$$

para algún $\varepsilon > 0$*. Sea* $\bar{\sigma}^T = (\bar{\sigma_1}^T, \bar{\sigma_2}^T)$*, donde:*

$$\bar{\sigma}_i^T(s_i) = \frac{|\{1 \leq T : s_i^t = s_i\}|}{T} = \frac{\#(s_i)}{T} \tag{4.15}$$

es decir, $\bar{\sigma}^T$*, es la distribución empírica de probabilidad. Entonces* $\bar{\sigma}^T$ *es un* 2ε*-equilibrio de Nash.*

En juegos de dos jugadores de suma cero, si un procedimiento es universalmente consistente, su distribución empírica llevará a un equilibrio de Nash. Los procedimientos propuestos en la sección anterior son universalmente consistentes, por lo que se pueden utilizar para encontrar un equilibrio de Nash en un juego particular. Estos algoritmos fueron implementados y probados en diferentes juegos de suma cero para encontrar una aproximación a un equilibrio de Nash en cada uno de ellos.

4.2. Evaluación Empírica de Regret Matching

Los algoritmos fueron probados en cuatro juegos diferentes en forma normal: *matching pennies*, piedra, papel y tijeras, ficha vs dominó y coronel blotto. Estos juegos son de suma cero para dos persona, por lo tanto es suficiente con determinar el pago para del primer jugador para que el juego esté bien definido. Por otra parte, cualquier juego con estas características puede modelarse como un problema de programación lineal [10, pp. 228-233] y resolverse mediante procedimientos destinados para esto.

Cada uno de los juegos es descrito mediante sus reglas. Además, si el juego tiene un tamaño fijo y es lo suficientemente pequeños, se muestra su matriz de pagos y el problema de programación lineal asociado con una solución.

Juego de Matching Pennies

En este juego cada jugador tiene una moneda y selecciona cara o sello de forma secreta. Si las elecciones son iguales gana el jugador 1, en caso contrario gana el jugador 2. La matriz de pagos de este juego se muestra en la Tabla 4.1.

Tabla 4.1: Tabla de pagos del juego *matching pennies*

	cara	sello
cara	1	-1
sello	-1	1

El problema de programación lineal asociado es el siguiente

$$\begin{array}{rl} \text{máx} & z \\ \text{sujeto a} & \\ & \begin{array}{rcrcrcl} & & x_1 & + & x_2 & = & 1 \\ z & - & x_1 & + & x_2 & \leq & 0 \\ z & + & x_1 & - & x_2 & \leq & 0 \\ & & x_1, & & x_2 & \geq & 0 \end{array} \end{array} \tag{4.16}$$

Cuya solución primal viene dada por $(z^*, x_1^*, x_2^*) = (0, \frac{1}{2}, \frac{1}{2})$ y su solución dual por $(w^*, y_1^*, y_2^*) = (0, \frac{1}{2}, \frac{1}{2})$. Luego el equilibrio de Nash se obtiene cuando ambos jugadores eligen cara o sello con probabilidad $\frac{1}{2}$ y el valor del juego es igual a 0.

Juego de Piedra, Papel o Tijeras

Este juego es descrito en la sección I y su matriz de pago se muestra en la Tabla 1.1. El problema de programación lineal asociado es el siguiente:

$$\begin{array}{rl} \text{máx} & z \\ \text{sujeto a} & \\ & \begin{array}{rcrcrcrcl} & & x_1 & + & x_2 & + & x_3 & = & 1 \\ z & & & + & x_2 & - & x_3 & \leq & 0 \\ z & - & x_1 & & & + & x_3 & \leq & 0 \\ z & + & x_1 & - & x_2 & & & \leq & 0 \\ & & x_1, & & x_2, & & x_3 & \geq & 0 \end{array} \end{array} \tag{4.17}$$

La solución primal y dual de este problema son $(z^*, x_1^*, x_2^*, x_3^*) = (0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ y $(w^*, y_1^*, y_2^*, y_3^*) = (0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, respectivamente. Luego, el equilibrio de Nash se obtiene cuando ambos jugadores eligen cada una de las acciones con probabilidad igual a $\frac{1}{3}$.

Juego de Ficha vs Dominó

En este juego cada jugador tiene un tablero de tamaño 2×3. El primer jugador tiene una ficha de dominó (que ocupa dos casillas con un lado en común) que puede colocar de 7 formas diferentes, cada forma es mostrada en la figura 4.1, con su respectiva etiqueta. El segundo jugador posee una ficha que ocupa una sóla casilla de su tablero y la ubica en una de las 6 casillas, las cuales se numeran en la Figura 4.2. Luego se superponen los tableros y si la ficha es cubierta por el dominó entonces el segundo jugador gana, en caso contrario gana el primer jugador [10, p. 237].

Figura 4.1: Posibles posiciones de la ficha de dominó

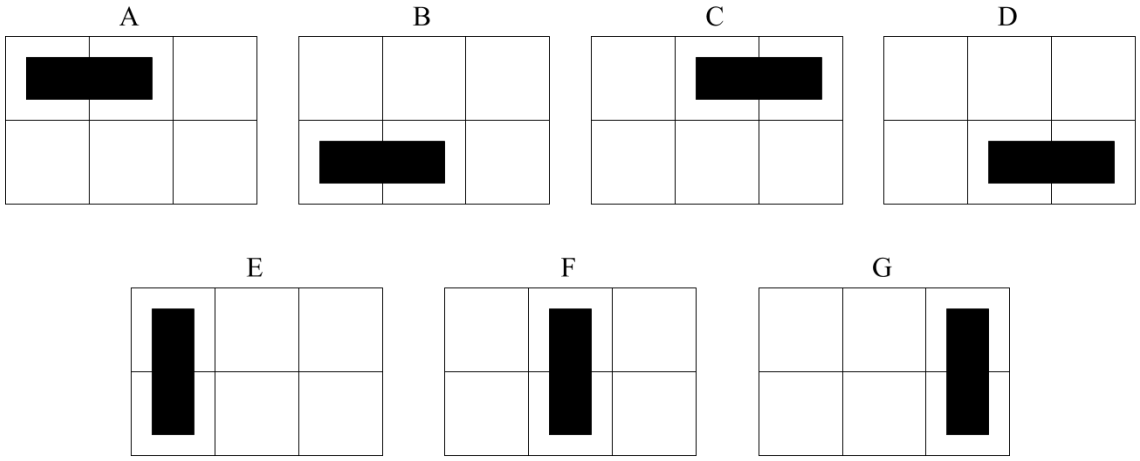


Figura 4.2: Posibles posiciones de la ficha del segundo jugador

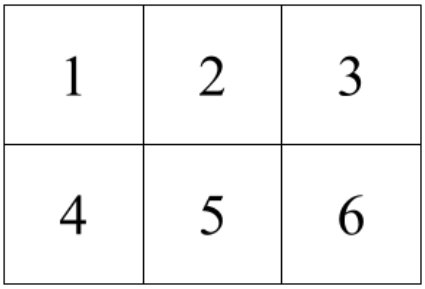


Tabla 4.2: Matriz de pagos del juego Ficha vs Dominó

	1	2	3	4	5	6
A	-1	-1	1	1	1	1
B	1	1	1	-1	-1	1
C	1	-1	-1	1	1	1
D	1	1	1	1	-1	-1
E	-1	1	1	-1	1	1
F	1	-1	1	1	-1	1
G	1	1	-1	1	1	-1

El problema de programación lineal asociado es:

$$
\begin{array}{rl}
max & z \\
\text{sujeto a} &
\end{array}
\qquad
\begin{array}{rcccccccccccccc}
 & & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & = & 1 \\
z & + & x_1 & - & x_2 & - & x_3 & - & x_4 & + & x_5 & - & x_6 & - & x_7 & \leq & 0 \\
z & + & x_1 & - & x_2 & + & x_3 & - & x_4 & - & x_5 & + & x_6 & - & x_7 & \leq & 0 \\
z & - & x_1 & - & x_2 & + & x_3 & - & x_4 & - & x_5 & - & x_6 & + & x_7 & \leq & 0 \\
z & - & x_1 & + & x_2 & - & x_3 & - & x_4 & + & x_5 & - & x_6 & - & x_7 & \leq & 0 \\
z & - & x_1 & + & x_2 & - & x_3 & + & x_4 & - & x_5 & + & x_6 & - & x_7 & \leq & 0 \\
z & - & x_1 & - & x_2 & - & x_3 & + & x_4 & - & x_5 & - & x_6 & + & x_7 & \leq & 0 \\
 & & x_1, & & x_2, & & x_3, & & x_4, & & x_5, & & x_6, & & x_7, & \geq & 0
\end{array}
\tag{4.18}
$$

Este problema no tiene solución única (lo que implica que el juego no tiene un equilibrio de Nash único), una solución viene dada por $(z^*, x_1^*, x_2^*, x_4^*, x_5^*, x_6^*, x_6^*, x_7^*) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0, \frac{1}{3})$ y $(w^*, y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*) = (\frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3}, 0, \frac{1}{3}, 0)$, esta solución corresponde a la estrategia en la que el jugador 1 elige las posiciones A, B y G con probabilidad $\frac{1}{3}$ cada una, y el jugador 2 elige las posiciones 1, 3, y 5 con probabilidad $\frac{1}{3}$ cada una.

Juego de Coronel Blotto

En este juego cada uno de los jugadores tiene S soldados en total que debe ubicar en N campos de batallas. Cada soldado puede ser asignado a un único campo, pero cualquier número de soldados puede ser colocado en cualquier campo, incluyendo el cero. Un jugador obtiene un campo de batalla si asigna más soldados que su oponente en ese campo de batalla. El juego es ganado por el jugador que obtenga un mayor número de

campos y su pago es igual a la diferencia entre el número de campos obtenidos por cada uno de los jugadores [11].

Formalmente el juego puede ser descrito de la siguiente manera. Cada jugador debe elegir N números enteros, digamos $(a_1, a_2, ..., a_N)$ y $(b_1, b_2, ..., b_N)$, para el jugador 1 y 2 respectivamente, tales que $a_1 + a_2 + ... + a_N = S$ y $b_1 + b_2 + ... + b_N = S$, con $N < S$, donde a_i y b_i es la cantidad de soldados ubicados el i-ésimo campo por el primer y segundo jugador, respectivamente. Para estas distribuciones, la ganancia del jugador 1 viene dada por:

$$|\{1 \leq i \leq N : a_i > b_i\}| - |\{1 \leq i \leq N : a_i < b_i\}| \tag{4.19}$$

Este juego depende de dos parámetros: el número de soldados S y el número de campos de batallas N, por lo que la matriz de pagos no es constante y por eso no es presentada como en los juegos anteriores. La matriz para de un juego con S soldados y N es una matriz cuadrada de tamaño $\binom{N+S-1}{S-1}$.

4.3. Detalles de Implementación y Ejecución

Los algoritmos fueron implementados en el lenguaje de programación C++, utilizando la librería estándar y una librería adicional llamada *Eigen*, para factorizar matrices y resolver sistemas de ecuaciones.

Se implementó una clase para encontrar un equilibrio de Nash mediante el algoritmo de *Regret Matching*. En cada iteración la actualización de las estrategias depende de cada procedimiento según las fórmulas propuestas en la sección anterior.

En el juego Coronel Blotto la matriz de pagos no tiene un tamaño fijo y además no es proporcionada de forma explícita, por lo que es necesario generarla dependiendo de los parámetros. Para esto se creó un programa que, dado el número de campos de batalla (N) y el número de soldados (S), genera todas las posibles distribuciones de cada uno de los jugadores mediante un algoritmo de *backtracking* y calcula el pago para cada juego posible, obteniendo como salida del programa la matriz deseada. De esta forma se generó la matriz de pagos para este juego cuando $N = 3$ y $S = 5$.

Las ejecuciones de estos algoritmos se realizaron en una máquina personal, con las siguientes características:

- Procesador: Intel® Core™ i5-8250U CPU @ 1.60GHz
- 8CPUs
- 8GB de memoria RAM

- Sistema Operativo: Ubuntu 18.04.3 LTS

4.4. Resultados Experimentales

En esta sección se presenta un resumen de los resultados experimentales obtenidos utilizando el algoritmo *Regret Matching* en los juegos descritos. Cada procedimiento fue probado 10 veces por juego, finalizando cada corrida cuando se obtenía un regret máximo menor que 0.005.

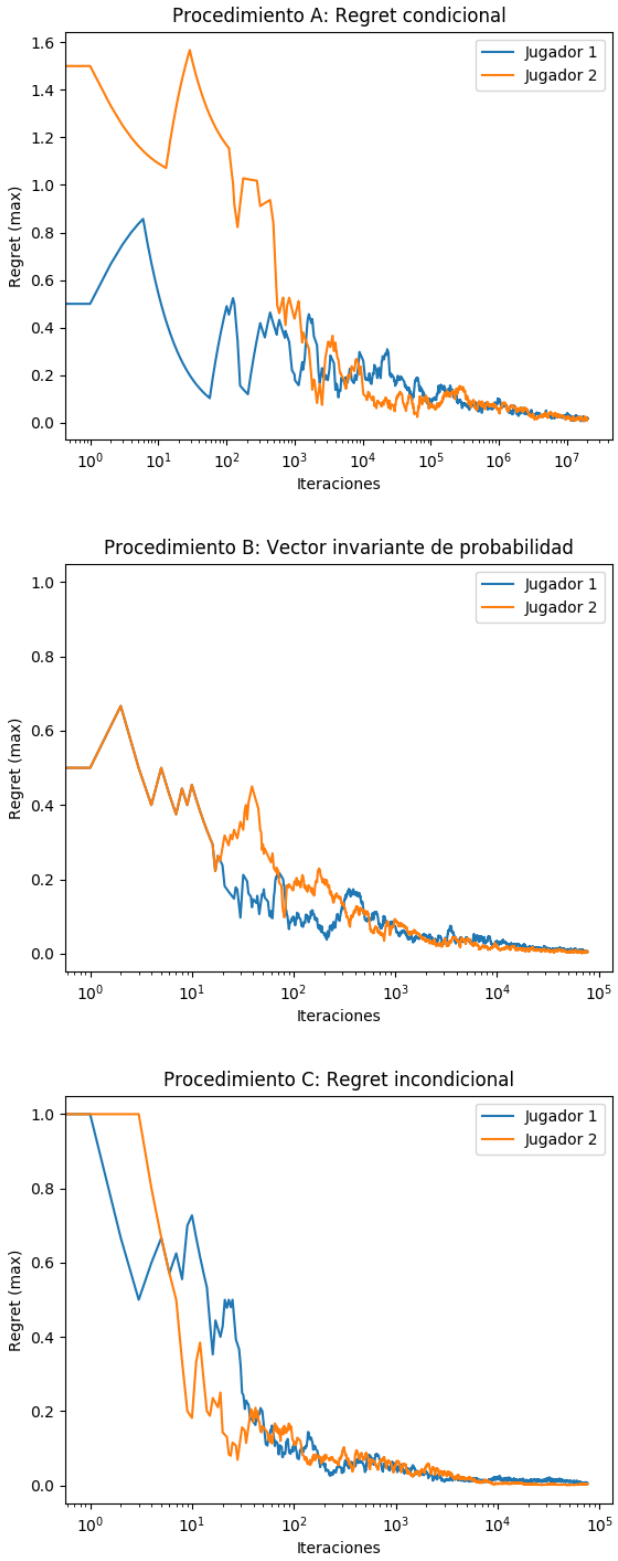
La tabla 4.3 muestra un resumen de los resultados. En esta tabla se muestra, por cada juego, el tamaño de la matriz de pagos, el valor teórico del juego (u_t), el valor del juego utilizando la estrategia obtenida en la última corrida del algoritmo (u_e) y la explotabilidad (ε_σ). Las dos últimas métricas se presentan para cada uno de los procedimientos (A, B y C). En todos los casos se observa que la utilidad esperada para las estrategias obtenidas es cercana al valor del juego, además, se obtuvo una explotabilidad menor o igual que 0.011, por lo que las estrategias obtenidas representan buenas aproximaciones al equilibrio de Nash en cada juego.

		u_e			ε_σ		
Juegos	u_t	A	B	C	A	B	C
Matching Pennies	0	0	0	0	0.006	0.006	0.008
Piedra, Papel o Tijeras	0	−0.000012	0.000004	0.000022	0.006	0.01	0.009
Ficha vs. Dominó	$\frac{1}{3}$	0.333	0.334	0.334	0.01	0.007	0.004
Coronel Blotto	0	0.000219	−0.000502	0.000024	0.01	0.011	0.009

Tabla 4.3: Resumen de los resultados y evaluación de las estrategias obtenidas usando el algoritmo de Regret Matching en juegos en forma normal

Para evaluar la convergencia se midió el tiempo necesario para alcanzar el regret desea y el número de iteraciones, en la Tabla 4.4 se presenta el tiempo (T), el número de iteraciones (I) y el tiempo por iteración (T/I) obtenido para cada uno de los juegos para cada procedimiento. Estos resultados son el promedio de las 10 corridas realizadas por juego y procedimiento. Además, se crearon gráficas del regret por iteración para observar como disminuye a medida que corre el algoritmo, la Figura 4.3 muestra las gráficas para el juego Coronel Blotto y los 3 procedimientos. Estas gráficas son mostradadas con una escala logarítmica en el eje x para apreciar mejor los resultados. En el Apéndice A se muestran las tablas detalladas con los resultados en cada una de las corridas y las gráficas de cada juego con los 3 procedimientos.

Figura 4.3: Gráficas del regret con respecto al número de iteraciones del juego Coronel Blotto



Juegos	Proc.	T	I	T/I
Matching Pennies	A	10.276	3892550.4	2.64×10^{-06}
	B	0.777	25616.6	3.03×10^{-05}
	C	0.042	16260.5	2.58×10^{-06}
Piedra, Papel o Tijeras	A	12.198	4519054.1	2.70×10^{-06}
	B	0.345	6601.3	5.23×10^{-05}
	C	0.049	19321.1	2.54×10^{-06}
Ficha vs. Dominó	A	319.179	108319272.4	2.95×10^{-06}
	B	11.275	75250.2	1.50×10^{-04}
	C	0.237	84318.5	2.81×10^{-06}
Coronel Blotto	A	875.533	190222305.3	4.60×10^{-06}
	B	79.358	66378.4	1.20×10^{-03}
	C	0.166	48613.5	3.41×10^{-06}

Tabla 4.4: Resumen de los resultados y evaluación de las estrategias obtenidas usando el algoritmo de Regret Matching en juegos en forma normal

A continuación se se analiza el desempeño de los procedimientos, comparándolos entre sí, observando la rapidez de convergencia de cada uno de ellos.

4.4.1. Complejidad de cada iteración

Los procedimientos cambian en la forma en que se elige la siguiente estrategia en cada iteración. En los procedimientos A y B se utiliza un regret condicional, en el que se mide el *arrepentimiento* de cambiar una estrategia por otra en específica. Esta métrica se debe mantener a lo largo de todas las iteraciones, por lo que cada iteración necesita memoria adicional de complejidad $\mathcal{O}(N^2 + M^2)$, donde N y M es el número de acciones posibles para el jugador 1 y 2, respectivamente. En el procedimiento C se utiliza únicamente el regret incondicional, por lo que la cantidad de memoria adicional es del orden $\mathcal{O}(N + M)$.

Con respecto a la complejidad de tiempo se tiene que los procedimientos de regret condicional e incondicional (A y C), son lineales al número de acciones. Sin embargo, en el procedimiento B es necesario resolver un sistema de ecuaciones lineales para elegir cada estrategia nueva, del tamaño del número de acciones del jugador respectivo, obteniendo que la complejidad total es $\mathcal{O}(N^3 + M^3)$. La Tabla 4.5 muestra un resumen de la complejidad es tiempo y memoria adicional.

Por lo anterior, se observa que la velocidad de las iteraciones del procedimiento que calcula el vector invariante de probabilidad es más lenta en todos los casos, estando uno o dos órdenes de magnitud por encima, según el tamaño de la matriz. Por lo que, si la

Procedimiento	Memoria	Tiempo
A	$\mathcal{O}(N^2 + M^2)$	$\mathcal{O}(N + M)$
B	$\mathcal{O}(N^2 + M^2)$	$\mathcal{O}(N^3 + M^3)$
C	$\mathcal{O}(N + M)$	$\mathcal{O}(N + M)$

Tabla 4.5: Complejidad por iteración de cada uno de los procedimientos

matriz es sumamente grande, el segundo método sería el menos adecuado.

4.4.2. Número de iteraciones

En la Tabla 4.4 se puede observar un resumen de las iteraciones promedio de los tres procedimientos en cada uno de los juegos. Se nota que el procedimiento A, regret incondicional es el que necesita muchas más iteraciones para converger. Con respecto a los procedimientos B y C, se observa que en algunos casos el promedio en el procedimiento B fue menor y en otros el promedio del procedimiento C. También es importante destacar que en el juego de piedra, papel o tijera se tienen varios casos donde se obtiene la convergencia en menos de 10 iteraciones (ver apéndice A), esos son casos donde se obtiene el equilibrio de Nash de forma exacta en pocas iteraciones.

4.4.3. Tiempo transcurrido

Observando el tiempo promedio de los procedimientos en la tabla 4.4, se nota que el procedimiento A es el que emplea más tiempo en todos los casos, esto ocurre porque necesita muchas más iteraciones que los otros dos procedimientos. Por otra parte el procedimiento C es también más rápido que el procedimiento B, ya que la complejidad en cada iteración para resolver el sistema de ecuaciones enlentece el tiempo total necesario, incluso, si la matriz es muy grande este procedimiento podría ser más lento que el procedimiento A y no sería factible.

Aunque el procedimiento donde se aplica regret matching al regret incondicional (procedimiento C), es el más sencillo de implementar y el más rápido en converger, este procedimiento tiene una desventaja con respecto a los otros dos. Al utilizar el regret condicional, los dos primeros procedimientos garantizan que el regret condicional tiende a cero para cualquier par de estrategias de cada jugador y por lo tanto, conducen siempre a un equilibrio correlacionado. El tercer procedimiento sólo minimiza el regret incondicional y por lo tanto, si el juego es de más de dos jugadores o no es de suma cero, entonces ya no es

de utilidad para hallar alguna solución del juego.

CAPÍTULO V

COUNTERFACTUAL REGRET MINIMIZATION

El objetivo de esta sección es presentar un algoritmo que permita encontrar un equilibrio de Nash en un juego en forma extensiva no determinista con información incompleta. Aunque todo juego en forma extensiva puede ser representado en forma normal, esto no es de mucho interés, pues la forma normal puede tener un tamaño exponencialmente más grande al tamaño del árbol. Se verá como el concepto de *regret minimization* puede ser extendido a juegos secuenciales, sin necesidad de la forma normal explícita. Los conceptos, procedimientos y teoremas mostrados en esta sección, son presentados en [7].

5.1. Regret Minimization

La primera definición clave es el *regret*. Para esto, es necesario considerar jugar repetidamente un juego en forma extensiva. Sea σ_i^t la estrategia usada por el jugador i a tiempo t. La Definición 5.1, presenta el concepto de *average overall regret*.

Definición 5.1. *El* ***average overall regret*** *del jugador i a tiempo T es:*

$$R_i^T \;=\; \max_{\sigma^* \in B_i} \frac{1}{T} \sum_{t=1}^{T} u_i(\sigma_i^*, \sigma_{-i}^t) - u_i(\sigma^t)\,. \tag{5.1}$$

Se denotará con $\bar{\sigma}_i^T$ la estrategia promedio del jugador i del tiempo 1 al tiempo T. En particular, para cada conjunto de información $I \in \mathcal{I}_i$ y para cada acción $a \in A(I)$ se define:

$$\bar{\sigma}_i^T(I)(a) \;=\; \frac{\sum_{t=1}^{T} \pi^{\sigma_i^t}(I)\sigma_i^t(I)(a)}{\sum_{t=1}^{T} \pi^{\sigma_i^t}(I)}\,. \tag{5.2}$$

Esta estrategia es el promedio ponderado de las probabilidades $\sigma^t(I)(a)$ con respecto a que tan probable es alcanzar I dado σ_i^t.

La relación entre el *average overall regret* y el concepto de solución se muestra en el Teorema A.13 [7].

Teorema 5.2. *En un juego de 2 jugadores de suma cero si el average overall regret a tiempo T es menor que ε entonces σ^{-T} es un 2ε-equilibrio de Nash.*

Como consecuencia del teorema anterior se obtiene que un algoritmo que lleve el *average overall regret* a cero, conducirá a un equilibrio de Nash. La idea fundamental del enfoque presentado a continuación, propuesto en [7], consiste en descomponer el *average overall regret* en un conjunto de términos aditivos de *regret* que puedan ser minimizados independientemente. En particular es necesario introducir un par de nuevo concepto, la utilidad contrafactual (Definición 5.3) y el *regret* contrafactual inmediato (Definición 5.4).

Definición 5.3. *La **utilidad contrafactual** es la ganancia esperada dado que el conjunto I es alcanzado y todos los jugadores juegan con la estrategia σ con excepción del jugador i que juega para alcanzar I. Formalmente, si $\pi^{\sigma}(h, h')$ es la probabilidad de ir de la historia h a la historia h', entonces:*

$$u_i(\sigma, I) \;=\; \frac{\sum_{h \in I, z \in Z} \pi^{\sigma_{-i}(h)} \pi^{\sigma}(h, z) u_i(z)}{\pi^{\sigma_{-i}}(I)} \,. \tag{5.3}$$

Para toda acción $a \in A(I)$, se define $\sigma|_{I \to a}$ como el perfil estratégico idéntico a σ excepto que el jugador i siempre elige a en el conjunto de información I.

Definición 5.4. *El **regret contrafactual inmediato** es:*

$$R^T_{i,imm}(I) \;=\; \max_{a \in A(I)} \frac{1}{T} \sum_{t=1}^{T} \pi^{\sigma^t_{-i}}(I) \left[u_i(\sigma^t|_{I \to a}, I) - u_i(\sigma^t, I) \right] . \tag{5.4}$$

Intuitivamente, el *regret* contrafactual inmediato es el arrepentimiento del jugador i en su decisión en el conjunto de información I, en términos de la utilidad contrafactual, con un término de ponderación adicional para la probabilidad contrafactual que I alcanzaría en esa ronda si el jugador hubiera intentado hacer eso. Usualmente, es de mayor interés el *regret* cuando es positivo, por lo que se define $R^{T,+}_{i,\mathrm{imm}}(I) = max(R^T_{i,\mathrm{imm}}(I), 0)$. Luego, se tiene el siguiente resultado:

Teorema 5.5.

$$R^T_i \;\leq\; \sum_{I \in \mathcal{I}_i} R^{T,+}_{i,imm}(I) \,. \tag{5.5}$$

Debido a que minimizar cada *regret* contrafactual inmediato minimiza el *average overall regret* arrepentimiento general promedio, es posible enfocarse en minimizar los primeros para obtener un equilibrio de Nash.

5.2. Counterfactual Regret Minimization

Antes de mostrar el algoritmo principal, denominado *Counterfactual Regret Minimization* para los juegos en forma extensiva, es necesario introducir el algoritmo de *Regret Matching* general. Este algoritmo puede ser descrito en un dominio donde hay un conjunto fijo de acciones A, una función $u^t : A \to \mathbb{R}$ y en cada ronda una distribución de probabilidad p^t es elegida.

Definición 5.6. *El regret de no haber elegido la acción $a \in A$ hasta tiempo T, se define como:*

$$R_i^T(a) = \frac{1}{T}\sum_{t=1}^{T}\left[u_i(a) - \sum_{a' \in A} p^t(a)u^t(a)\right] \tag{5.6}$$

Se define $R^{t,+}(a) = \text{máx}(R^t(a), 0)$. Luego la distribución p^{t+1} es elegida de la siguiente manera:

$$p^t(a) = \begin{cases} \frac{R_i^{t,+}}{\sum_{a' \in A} R^{t,+}(a)} & \text{si} \quad \sum_{a' \in A} R^{t,+}(a) > 0 \\ \frac{1}{|A|} & \text{en otro caso} \end{cases} \tag{5.7}$$

Teorema 5.7. *Si $|u| = \text{máx}_{t \in \{1,2,\ldots T\}} \text{máx}_{a,a' \in A}(u^t(a) - u^t(a'))$ entonces el regret del algoritmo de regret matching está acotado por:*

$$max_{a \in A} R^t(a) \;\leq\; \frac{|u|\sqrt{|A|}}{\sqrt{T}}\,. \tag{5.8}$$

Luego, el algoritmo de *Counterfactual Regret Minimization* es una aplicación del algoritmo *Regret Minimization* de forma independiente a cada conjunto de información. En particular, se mantiene, para cada $I \in \mathcal{I}_i$ y para todo $a \in A(I)$:

$$R_i^T(I, a) \;=\; \frac{1}{T}\sum_{t=1}^{T} \pi^{\sigma^t_{-i}}(I)[u_i(\sigma^t|_{I \to a}, I) - u_i(\sigma^t, I)] \tag{5.9}$$

Se define $R_i^{T,+}(I,a) = \text{máx}(R_i^T(I,a), 0)$, luego a tiempo $T+1$ la estrategia elegida es:

$$\sigma_i^{T+1}(I)(a) \;=\; \begin{cases} \frac{R_i^{T,+}(I,a)}{\sum_{a' \in A(I)} R_i^{T,+}(I,a')} & \text{si } \sum_{a' \in A(I)} R_i^{T,+}(I,a') > 0\,, \\ \frac{1}{|A(I)|} & \text{en otro caso.} \end{cases} \tag{5.10}$$

Este algoritmo consiste en seleccionar las acciones de forma proporcional a la cantidad del *regret* contrafactual positivo de no haber elegido esa acción. Si ninguna de estas cantidades es positiva, entonces la acción se elige con una distribución uniforme. Luego, como cota de convergencia, se tiene el siguiente teorema:

Teorema 5.8. *Si el jugador i selecciona las acciones de acuerdo al procedimiento anterior, entonces $R_{i,imm}^T(I) \leq \Delta_{u,i}\frac{\sqrt{|A_i|}}{\sqrt{T}}$ y por lo tanto $R_i^T \leq \Delta_{u,i}|\mathcal{I}_i|\frac{|A_i|}{\sqrt{T}}$, donde $|A_i| = max_{h:P(h)=1}|A(h)|$.*

5.3. Monte Carlo Conterfactual Regret Minimization

En la sección V se explicó el algoritmo de CFR, utilizado para resolver juegos en forma extensiva. Sin embargo, en la versión presentada es necesario recorrer el árbol completo en cada iteración, esta versión suele conocerse como *vanilla* CFR. En [12] se describe una familia general de algoritmos CFR (basados en muestreo) denominada **Monte Carlo Conterfactual Regret Minimization** (MCCRF), para evitar recorrer el árbol completo en cada iteración.

La idea general es restringir los estados terminales alcanzados en cada iteración, pero manteniendo el mismo valor esperado para la utilidad contrafactual. Dada la definición 2.3, sea $\mathcal{Q} = \{Q_1, Q_2, ..., Q_r\}$, un conjunto de subconjuntos de Z tal que su unión sea igual a Z. Cada uno de estos conjuntos será llamado un bloque. Sea $q_j > 0$ la probabilidad de considerar el bloque Q_j para la iteración actual (donde $\sum_{j=1}^r q_j = 1$). Sea $q(z) = \sum_{j|j \in Q_i}$, es decir, $q(z)$ es la probabilidad de considerar z en la iteración actual. La utilidad contrafactual muestreada, cuando se actualiza el bloque j es:

$$\tilde{u}_i(\sigma, I|j) \;=\; \sum_{h \in I, z \in Q_j} \frac{\pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z)}{q(z)\pi^{\sigma_{-i}}(I)} \tag{5.11}$$

Lema 5.9. $E_{j\sim q_j}[\tilde{u}_i(\sigma, I|j)] \;=\; u_i(\sigma, I)$

Demostración.

$$E_{j\sim q_j}[\tilde{u}_i(\sigma, I|j)] = \sum_j q_j u_i(\sigma, I) \tag{5.12}$$

$$= \sum_j q_j \frac{\sum_{h\in I, z\in Q_j} \pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z)}{q(z)\pi^{\sigma_{-i}}(I)} \tag{5.13}$$

$$= \sum_j \sum_{\substack{h\in I \\ z\in Q_j}} \frac{q_j \pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z)}{q(z)\pi^{\sigma_{-i}}(I)} \tag{5.14}$$

$$= \sum_{\substack{h\in I \\ z\in Z}} \sum_{j|z\in Q_j} \frac{q_j \pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z)}{q(z)\pi^{\sigma_{-i}}(I)} \tag{5.15}$$

$$= \sum_{\substack{h\in I \\ z\in Z}} \left(\frac{\sum_{j|z\in Q_j} q_j}{q(z)} \right) \frac{\pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z)}{\pi^{\sigma_{-i}}(I)} \tag{5.16}$$

$$= \sum_{\substack{h\in I \\ z\in Z}} \frac{\pi^{\sigma_{-i}}(h)\pi^{\sigma}(h,z)u_i(z)}{\pi^{\sigma_{-i}}(I)} = u_i(\sigma, I) \tag{5.17}$$

La ecuación 5.14 se obtiene de la definición de $\tilde{u}_i(\sigma, I|j)$. 5.15 y 5.16 se obtienen al reordenar las sumatorias y considerando que la unión de los bloques generan a Z. La ecuación 5.17 □

Si se elige $\mathcal{Q} = Z$, es decir un único bloque con todas las historias terminales y $q_1 = 1$, la utilidad contrafactual es igual a la utilidad contrafactual muestreada y se obtiene el algoritmo *vanilla* CFR. Si se eligen los bloques para incluir todas las historias terminales con la misma secuencia de acciones en los nodos de azar se obtiene el *chance-sampled* CFR, siendo esta última versión la utilizada para estudiar los juegos presentados en este trabajo de grado. Se implementa el algoritmo como es detallado en [1] que se presenta en el ***apéndice X***.

5.4. Evaluación de las Estrategias y Explotabilidad

Para evaluar la convergencia de los algoritmos y la estrategia obtenida se utilizaron las métricas de *regret* y explotabilidad, respectivamente.

La explotabilidad se obtiene al calcular la mejor respuesta de la estrategia de cada jugador y sumar los resultados, como se explicó en la sección ***Explicar explotabilidad en el capítulo 2***. Sin embargo, la diferencia es que en los juegos de forma extensiva no se pueden listar todas las estrategias fácilmente como en los juegos en forma normal, ya

que esta tarea es exponencial en el tamaño del árbol.

Para calcular la explotabilidad en estos juegos se utilizó el algoritmo propuesto en [9], denominado *Generalized Expectimax Best Response* (GEBR), descrito en el apéndice ***X***. La complejidad de este algoritmo es $\mathcal{O}(ND)$ donde N es el número de nodos del árbol y D es la profundidad del árbol. Note que el algoritmo tiene una alta complejidad, por lo que se usará únicamente para calcular la explotabilidad de la estrategia final.

5.5. Detalles de implementación

Los algoritmos y la representación de los juegos fueron implementados en el lenguaje de programación C++. Para la representación de los juegos se utilizó una clase abstracta llamada *Game*, que recibe como template los tipos para el estado, las acciones, las propiedades, los conjunto de información y el Hash del juego específico.

Esta clase contiene las funciones virtuales necesarias para recorrer el árbol del juego de forma **implícita**, tales como: *actions*, que retornan las acciones del juegos en el estado actual, *update_state*, que actualiza el estado del juego dada una acción a realizar, *terminal_state* que indica si un estado es terminal o no, *utiliy* que retorna la utilidad en un estado terminal, entre otras. Los algoritmos CFR y GEBR utilizan esta clase abstracta en su implementación.

Para cada tipo de juego, se creó una clase derivada de la clase *Game*, donde se implementaron las funciones según las reglas de cada juego. De esta forma se puede utilizar la misma implementación de los algoritmos para todos los juegos.

Cabe destacar que todos los juegos fueron representados mediantes árboles con la raíz como único nodo de azar. Algunos juegos tienen esta representación de forma natural, por ejemplo, el Kuhn Poker, ya que las cartas se reparten al inicio y luego se juega acorde a esa distribución, sin volver a introducir ninguna jugada aleatoria. Otros juegos pueden tener nodos de azar distintos a la raíz, sin embargo siempre es posible transformarlos a un árbol que represente el mismo juego donde todos los nodos de azar son condensados en la raíz y cada hijo de la raíz representa una elección por cada uno de los nodos de azar del árbol original. En esta representación se asume que todas las decisiones aleatorias se toman al inicio del juego.

La clase *Game* y todos los algoritmos se implementados suponiendo la raíz como único nodo de azar del juego.

5.6. Descripción de los juegos

Para probar los algoritmos se implementaron tres tipos de juegos diferentes: *One Card Poker* (OCP), *Dudo*, un juego de dados, y una versión del juego de dominó para 2 personas. La descripción detallada de las reglas de los juegos se describen en esta sección.

Juego de One-Card Poker

One-Card Poker, abreviado OCP(N), es la versión generalizada del juego Kuhn Póker, explicado en la sección II. En este juego, cada jugador recibe una carta de un mazo de N cartas, y luego pueden apostar o retirarse según las mismas reglas del Kuhn Póker. Note que OCP(3) es equivalente al Kuhn Poker. El árbol de este juego tiene $9N(N-1)+1$ nodos (incluyendo el nodo inicial, que es el nodo de azar) y hay $4N$ conjuntos de información entre ambos jugadores.

Juego de Dudo

Dudo, también conocido como *Bluff*, *Liar's Dice* o Perudo, es un juego de dados y apuestas. Usualmente se juega entre 2 y 6 jugadores. Los jugadores se ubican en forma circular y cada uno de ellos tiene un número de dados. De forma simultánea, todos lanzan sus dados, cada jugador puede ver el resultado de sus propios dados, pero no puede ver el resultado de los dados de los otros jugadores. Una vez hecho esto, los jugadores empiezan a apostar sobre el número de veces que apareció una cara en específico en todos los dados que hay en la mesa.

Una apuesta consiste en decir 2 números (x, y), esto indica que el jugador apuesta que hay, al menos, x dados cuyo resultado fue el número y. El primer jugador (que se elige previamente mediante el lanzamiento de 1 dado o de alguna otra forma), realiza la primera apuesta y, en sentido horario, cada jugador puede hacer una apuesta más alta o decir "dudo" y retar al jugador anterior. Una apuesta es más alta que otra si el número de dados que se anuncian en la apuesta (x) es mayor, o si el número de dados es igual, pero la cara apostada (y) es mayor. Por ejemplo $(3, 1)$ es mayor que $(2, 5)$, y ambas apuestas son mayores que $(2, 3)$.

Por otra parte, si un jugador reta al jugador previo, se descubren todos los dados de todos los jugadores. Si la cantidad de dados con la cara y es mayor o igual a x, donde (x, y) fue la apuesta realizada por el jugador, el jugador que hizo el reto pierde un dado. En

caso contrario, el jugador que hizo la apuesta pierde un dado. Luego, todos los jugadores lanzan sus dados nuevamente y una nueva ronda de apuestas empieza por el jugador que perdió la ronda anterior. Un jugador pierde cuando se queda sin dados, el ganador es el último jugador con al menos un dado restante. La figura 5.1 muestra una foto del juego, de *Perudo*, una versión comercial de este juego, que está diseñada para 6 jugadores, donde cada jugador empieza con 5 dados. En la figura se observan los vasos que se utilizan para lanzar los dados y evitar que cada jugador vea los dados de los demás.

Figura 5.1: Juego Dudo. Los vasos se utilizan para lanzar los dados y evitar que los oponentes vean el resultado

En este trabajo de grado consideraremos este juego para 2 jugadores únicamente. Dudo(K, D_1, D_2) hará referencia a una única ronda de apuestas de 2 jugadores, donde el primer jugador tiene D_1 dados, el segundo jugador tiene D_2 dados y cada dado tiene K caras. El juego completo consiste en múltiples rondas, donde D_1 o D_2 disminuye en una unidad al finalizar cada ronda. Cuando uno de los jugadores pierde todos los dados obtiene una utilidad de -1, mientras que su oponente obtiene una utilidad de 1. En este juego cada ronda se considerará un subjuego y se representará con un árbol independiente, donde los valores esperados para los juegos Dudo$(K, D_1 - 1, D_2)$ y Dudo$(K, D_1, D_2 - 1)$ se precalculan y se utilizan como utilidad para las hojas del árbol Dudo(K, D_1, D_2). Note que, en el juego estándar, K siempre tiene un valor de 6.

Cuando el jugador i lanza D_i dados hay $\binom{D_i+K-1}{K-1}$ resultados posibles diferentes, ya que cada resultado puede ser representado con una tupla $(a_1, a_2, ..., a_k)$, donde a_j representa el

número de dados con la cara j, por lo que $\sum_j^K = D_i$ y $a_j \geq 0$. Por otra parte cada secuencia de apuestas puede ser representada por una secuencia binaria de longitud $K(D_1 + D_2)$, donde el i-ésimo bit es 1 si la i-ésima secuancia más fuerte fue dicha durante la ronda y 0 en caso contrario. Por ejemplo, si $D_1 = D_2 = 1$, las apuestas $(1,1) - (1,3) - (1,6) - (2,4) - (2,5) - (1,6)$ se representa con la secuencia binaria 101001000110, por lo que hay $2^{K(D_1+D_2)}$ secuencias diferentes. Cada secuencia pertenece a un jugador en específico, por lo que si $D_1 = D_2$, el número de conjuntos de información es igual a $\binom{D_i+K-1}{K-1}2^{K(D_1+D_2)}$.

Para contar el número total de nodos, se puede considerar el lanzamiento de los dados de forma independiente, pues las secuencias posibles de apuestas no dependen del resultado de los dados. Por lo expuesto anteriormente el número posible de apuestas es igual a $2^{K(D_1+D_2)}$, pero después de cada secuencia siempre se puede decir "dudo", salvo para la secuencia vacía. Luego el número total de nodos (incluyendo nodos terminales y no terminales) es igual a $\binom{D_1+K-1}{K-1}\binom{D_1+K-1}{K-1}(2^{K(D_1+D_2)+1} - 1) + 1$.

Juego de Dominó

En este trabajo se utilizó una versión de este juego para 2 jugadores. Al inicio del juego cada jugador toma una cantidad específica de piezas de forma aleatoria, las piezas restantes se dejan sin descubrir para ser usadas en turnos posteriores. Como en el juego tradicional de dominó, los jugadores juegan por turnos alternados (el primero jugador se elige de forma arbitraria), cada uno debe colocar una ficha válida acorde a las reglas *estándares* en Venezuela del juego (ver apéndice $\boldsymbol{X}$). Si un jugador no puede colocar una ficha toma una ficha de las que no están descubiertas (si todavía hay disponibles), el jugador verifica si puede colocar la ficha tomada y en caso contrario pasa el turno y juega el oponente.

El juego termina cuando alguno de los jugadores usa todas las piezas o cuando ambos jugadores no pueden jugar ni tomar piezas nuevas, en este último caso se dice que el juego está bloqueado. El ganador es el jugador que se queda sin piezas o, en caso de bloqueo, el jugador que acumule menos puntos en todas las piezas que quedaron en su mano. La utilidad obtenida es el número de puntos que el jugador perdedor acumuló en las piezas que quedaron en su mano (con signo positivo para el jugador ganador y signo negativo para el perdedor). Cabe destacar que sólo se puede tomar una pieza o pasar, si no se puede realizar una jugada con la mano actual.

Usualmente se utilizan 28 piezas, donde las piezas pueden tener entre 0 y 6 puntos en cada extremo, y cada jugador recibe 7 piezas al inicio del juego. En este trabajo se parametriza el número máximo de puntos que puede tener una ficha, así como la cantidad de piezas repartidas inicialmente. De esta forma se hará referencia a Domino(M, N) an

un juego donde las piezas tienen entre 0 y M puntos (con un total de $(M+1)(M+2)/2$ piezas) y cada jugador recibe N piezas al inicio del juego.

En este juego no es fácil calcular el tamaño del árbol y el número de conjuntos de información, principalmente porque las acciones posibles en un estados dependen tanto de la mano del jugador, como de las piezas en la mesa. En el Kuhn Póker simpre hay 2 acciones posibles *pasar*, *apostar* y en el Dudo las acciones disponibles dependen únicamente de la última apuesta y no dependen de los dados que tengan los jugadores. Así que se decidió estimar estos parámetros recorriendo el árbol del juego mediante DFS. La tabla 5.1, muestra el número de nodos del árbol y el número de conjuntos de información por cada juego de dominó que se presenta.

	Conjutos de Información	Nodos
Domino$(1,1)$	3	13
Domino$(2,2)$	441	7321
Domino$(3,2)$	844437	46534657
Domino$(3,3)$	1082290	246760993
Domino$(3,4)$	902218	1547645185

Tabla 5.1: Número de nodos y conjuntos de Información en diferentes juegos de Dominó

5.7. Resultados experimentales

Se crearon varias instancias de los 3 juegos explicados en la sección 5.6 con diferentes parámetros. Para cada instancia se utilizó el algoritmo de CFR y se iteró sobre el árbol durante 10 (***número tentativo***) horas (se excluye el tiempo que se calcula el regret ya que no forma parte del algoritmo y esto se hace únicamente para obtener las gráficas). Una vez terminado el tiempo asignado se calcula la mejor respuesta para cada jugador y la explotabilidad. Una instancia de un juego se considerará resuelta si la explotabilidad de la estrategia obtenida es menor que el 1 % de la mínima unidad de utilidad posible según cada juego.

La tabla 5.2 resume los resultados, donde N representa el número de nodos del árbol I el número de conjuntos de información, u_σ el valor del juego usando la estrategia obtenida y ε_σ la explotabilidad. También se agrega el número de iteraciones y la última columna indica si el juego fue resuelto o no, según lo establecido en el párrafo anterior.

La gráfica ***Mostrar una o dos gráficas interesantes y decir algo al respecto***. En el apéndice ***X*** se pueden observar todas las gráficas del regret con respecto al número

Juego	N	I	Iteraciones	u	ε	Resuelto
OCP(3)	55	12				✓
OCP(12)	1.189	48				✓
OCP(50)	22.051	200				✓
OCP(200)	358.201	800				✓
OCP(1000)	8.991.001	4.000				✓
OCP(4000)	143.964.001	16.000				✓
Dudo$(3,1,1)$	1144	192				✓
Dudo$(3,2,1)$	18415	2304				✓
Dudo$(3,1,2)$	18415	2304				✓
Dudo$(3,2,2)$	294877	24576				✓
Dudo$(4,1,1)$	8177	1024				✓
Dudo$(4,2,1)$	327641	28672				✓
Dudo$(4,1,2)$	327641	28672				✓
Dudo$(4,2,2)$	13107101	655360				✗
Dudo$(5,1,1)$	51176	5120				✓
Dudo$(5,2,1)$	4915126	327680				✓
Dudo$(5,1,2)$	4.915.126	327680				✓
Dudo$(5,2,2)$	471.858.976	15728640				✗
Dudo$(6,1,1)$	294.877	24576				✓
Dudo$(6,2,1)$	66.060.163	3538944				✓
Dudo$(6,1,2)$	66.060.163	3538944				✓
Dudo$(6,2,2)$	14.797.504.071	352321536				✗
Domino$(2,2)$	441	7321				✓
Domino$(3,2)$	844437	46534657				✓
Domino$(3,3)$	1082290	246760993				✓
Domino$(3,4)$	902218	1547645185				✓
Domino$(4,2)$						✗

Tabla 5.2: Resultados del algortimo CFR en los diferentes juegos

de iteraciones, se nota que el regret tiende a 0 en todos los casos (***Agregar algún otro detalle interesante que se vea en las gráficas***)

CONCLUSIONES

Sugerencias y revisiones de esta clase enviarlos a los correos ccontreras@usb.ve y asajo@usb.ve.

REFERENCIAS

[1] Neller, Todd W. y Marc Lanctot: *An Introduction to Counterfactual Regret Minimization*. Informe técnico, Gettysburg College, 2000.

[2] Hart, Sergiu y Andreu Mas-Colell: *A simple adaptative procedure leading to correlated equilibrium*. Econometrica, 68(5):1127–1150, Septiembre 2000.

[3] Jiang, Albert Xin y Kevin Leyton-Brown: *A Tutorial on the Proof of the Existence of Nash Equilibria*. Informe técnico, Department of Computer Science, University of British Columbia, 2007.

[4] Osborne, Martin J. y Ariel Rubinstein: *A Course in Game Theory*. The MIT Press, Cambridge, Massachusetts, 1994.

[5] Hart, Sergiu: *Games in Extensive and Strategic Forms*. En R. J. Auman and S. Hart (editor): *Handbook of Game Theory*, volumen 1, capítulo 2, páginas 19–40. Elsevier Science Publisher B.V., Noviembre 1992.

[6] Kuhn, Harold W.: *Simplified two-person poker*. En Kuhn, Harold W. y Albert W. Tucker (editores): *Contributions to the Theory of Games*, volumen 1, páginas 97–103. Princeton University Press, 1950.

[7] Zinkevich, Martin, Michael Johanson, Michael Bowling y Carmelo Piccione: *Regret Minimization in Games with Incomplete Information*. En *Advances in Neuronal Information Processing System 20 (NIPS)*, 2007.

[8] Leyton-Brown, Kevin y Yoav Shoham: *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. Morgan & Claypool, 2008.

[9] Lanctot, Marc: *Monte Carlo Sampling and Regret Minimization for Equilibrium Computation and Decision-Making in Large Extensive Form Games*. Tesis de Doctorado, University of Alberta, 2003.

[10] Chvátal, Vašek: *Linear Programming*. W. H. Freeman and Company, 1983.

[11] Arad, Ayala y Ariel Rubinstein: *Multi-Dimensional Iterative Reasoning in Action: The Case of the Colonel Blotto Game*. En *Journal of Economic Behavior & Organization*, volumen 84, páginas 571–585. Elsevier, 2012.

[12] Lanctot, Marc, Kevin Waugh, Martin Zinkevich y Michael Bowling: *Monte Carlo Sampling for Regret Minimization in Extensive Games*. En *Advances in Neuronal Information Processing System 22 (NIPS)*, 2009.

[13] Blackwell, David: *An analog of the Minimax Theorem for Vector Payoffs*. Pacific Journal of Mathematics, 6(1), Noviembre 1956.

[14] Koller, Daphne, Nimrod Megid y Bernhard von Sten: *Fast Algorithms for Finding Randomized Strategies in Game Trees*. En *The 26th Annual ACM Symposium on the Theory of Computing*, 1994.

APÉNDICE A

PRUEBAS

A.1. Capítulo I

Teorema A.1. *La ganancia esperada $u_i(\sigma)$ del jugador i dado el perfil estratégico σ satisface:*

$$u_i(\sigma) \;=\; \sum_{s_i \in S_i} \sigma_i(s_i) \sum_{s_{-i} \in S_{-i}} \sigma_{-i}(s_{-i}) u_i(s_i, s_{-i})\,. \tag{A.1}$$

Demostración. Partiendo de la Definición 1.6 se obtiene

$$u_i(\sigma) \;=\; \sum_{s \in S} u_i(s)\sigma_i(s_i)\sigma_{-i}(s_{-i}) \;=\; \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} u_i(s_i, s_{-i})\sigma_i(s_i)\sigma_{-i}(s_{-i}) \tag{A.2}$$

$$= \sum_{s_i \in S_i} \sigma_i(s_i) \sum_{s_{-i} \in S_{-i}} \sigma_{-i}(s_{-i}) u_i(s_i, s_{-i})\,. \tag{A.3}$$

□

Lema A.2. *Sea σ_i^* una estrategia mixta para el jugador i que es mejor respuesta a σ_{-i}, y sea $x \in S_i$ una estrategia pura para el jugador i. Entonces, para toda estrategia pura $y \in S_i$ diferente de x,*

$$\sigma_i^*(x) \sum_{s_{-i}} u_i(x, s_{-i})\sigma_{-i}(s_{-i}) \;\geq\; \sigma_i^*(x) \sum_{s_{-i}} u_i(y, s_{-i})\sigma_{-i}(s_{-i})\,. \tag{A.4}$$

Demostración. Considere la estrategia mixta σ_i' definida por:

$$\sigma_i'(s_i) \;=\; \begin{cases} 0 & \text{si } s_i = x \\ \sigma_i^*(x) + \sigma_i^*(y) & \text{si } s_i = y \\ \sigma_i^*(s_i) & \text{en otro caso} \end{cases} \tag{A.5}$$

Utilizando el Lema A.1 y el hecho que σ_i^* es mejor respuesta a σ_{-i}:

$$\begin{aligned} u_i(\sigma_i^*, \sigma_{-i}) &\geq u_i(\sigma_i', \sigma_{-i}) && \text{(A.6)} \\ &= \sum_{z \in S_i} \sigma_i'(z) \sum_{s_{-i}} u_i(z, s_{-i}) \sigma_{-i}(s_{-i}) && \text{(A.7)} \\ &= \sum_{z \neq x} \sigma_i^*(z) \sum_{s_{-i}} u_i(z, s_{-i}) \sigma_{-i}(s_{-i}) + \sigma_i^*(x) \sum_{s_{-i}} u_i(y, s_{-i}) \sigma_{-i}(s_{-i}) \,. && \text{(A.8)} \end{aligned}$$

Por el Lema A.1, $u_i(\sigma_i^*, \sigma_{-i}) = \sum_{z \in S_i} \sigma_i^*(z) \sum_{s_{-i}} u_i(z, s_{-i}) \sigma_{-i}(s_{-i})$. Entonces,

$$\sigma_i^*(x) \sum_{s_{-i}} u_i(x, s_{-i}) \sigma_{-i}(s_{-i}) \;\geq\; \sigma_i^*(x) \sum_{s_{-i}} u_i(y, s_{-i}) \sigma_{-i}(s_{-i}) \,. \qquad \text{(A.9)}$$

□

Teorema A.3. *Sea σ_i^* una estrategia mixta para el jugador i que es mejor respuesta a σ_{-i}. Cualquier estrategia mixta σ_i para el jugador i cuyo soporte sea un subconjunto del soporte de σ_i^* es también una mejor respuesta a σ_{-i}.*

Demostración. Sea $x \in S_i$ una *estrategia pura* perteneciente al soporte de σ_i^*, y sea $y \in S_i$ una estrategia pura *diferente* de x.

Por el Lema A.2,

$$\sum_{s_{-i}} u_i(x, s_{-i}) \sigma_{-i}(s_{-i}) \;\geq\; \sum_{s_{-i}} u_i(y, s_{-i}) \sigma_{-i}(s_{-i}) \,. \qquad \text{(A.10)}$$

En particular, si x y x' son distintos, y ambos pertenecen al soporte de σ_i,

$$\sum_{s_{-i}} u_i(x, s_{-i}) \sigma_{-i}(s_{-i}) \;=\; \sum_{s_{-i}} u_i(x', s_{-i}) \sigma_{-i}(s_{-i}) \;=\; C \qquad \text{(A.11)}$$

donde C es una constante que sólo depende de σ_{-i}. Luego, para cualquier estrategia σ_i, tal que $\mathit{support}(\sigma_i) \subseteq \mathit{support}(\sigma_i^*)$, se tiene:

$$u_i(\sigma_i, \sigma_{-i}) \;=\; \sum_{x \in S_i} \sigma_i(x) \sum_{s_{-i}} u_i(x, s_{-i}) \sigma_{-i}(s_{-i}) \;=\; \sum_{x \in S_i} \sigma_i(x) C \;=\; C \,. \qquad \text{(A.12)}$$

En particular, $u_i(\sigma_i^*, \sigma_{-i}) = C$, y σ_i es también mejor respuesta a σ_{-i}. □

Teorema A.4. *Si σ es un equilibrio de Nash, entonces σ es un equilibrio correlacionado.*

Demostración. Sea σ un equilibrio de Nash, sean $x, y \in S_i$ estrategias puras distintas cualesquiera para el jugador i, y sea σ_i' una estrategia mixta cualquiera para el jugador i.

Por el Lema A.2,

$$\sigma_i(x)\sum_{s_{-i}} u_i(x,s_{-i})\sigma_{-i}(s_{-i}) \;\geq\; \sigma_i(x)\sum_{s_{-i}} u_i(y,s_{-i})\sigma_{-i}(s_{-i})\,. \tag{A.13}$$

Es decir,

$$0 \;\leq\; \sigma_i(x)\sum_{s_{-i}} \sigma_{-i}(s_{-i})[u_i(x,s_{-i}) - u_i(y,s_{-i})] \;=\; \sum_{s_{-i}} \sigma(x,s_{-i})[u_i(x,s_{-i}) - u_i(y,s_{-i})]\,. \tag{A.14}$$

Luego, σ es un equilibrio correlacionado. □

Teorema A.5. *Sea $\psi \in \Delta(S)$ un equilibrio correlacionado. Si ψ se factoriza como $\psi = \prod_{i\in N}\sigma_i$ donde $\{\sigma_i\}_{i\in N}$ es un conjunto de estrategias mixtas para cada jugador (i.e., $\psi(s) = \prod_{i\in N}\sigma_i(s_i)$ para todo $s \in S$), entonces ψ es un equilibrio de Nash.*

Demostración. Sea $\psi = \prod_{i\in N}\sigma_i$ un equilibrio correlacionado en forma factorizada. Debemos mostrar que para cualquier jugador i y estrategia mixta σ_i' para el jugador i, se cumple $u_i(\sigma) \geq u_i(\sigma_i', \sigma_{-i})$.

Sean x y y estrategias puras para el jugador i. Como σ es un equilibrio correlacionado,

$$0 \;\leq\; \sigma_i(x)\sum_{s_{-i}} \sigma_{-i}(s_{-i})[u_i(x,s_{-i}) - u_i(y,s_{-i})]\,. \tag{A.15}$$

Al sumar sobre $x \in S_i$ obtenemos,

$$0 \;\leq\; \sum_{x\in S_i}\sum_{s_{-i}} \sigma(x,s_{-i})[u_i(x,s_{-i}) - u_i(y,s_{-i})] \;=\; \sum_{s} \sigma(s)[u_i(s) - u_i(y,s_{-i})]\,. \tag{A.16}$$

Si $x^* \in S_i$ es tal que $\sigma_i(x^*) > 0$, obtenemos de (A.15) al multiplicar por $\sigma_i'(y)$ y sumar sobre $y \in S_i$:

$$\sum_{y\in S_i}\sigma_i'(y)\sum_{s_{-i}} \sigma_{-i}(s_{-i})[u_i(x^*,s_{-i}) - u_i(y,s_{-i})] \;=\; \sum_{s} \sigma'(s)[u_i(x^*,s_{-i}) - u_i(s)] \;\geq\; 0 \tag{A.17}$$

donde σ' denota la estrategia $\sigma' = (\sigma_i', \sigma_{-i})$. Al sumar (A.16) y (A.17), obtenemos que para cualquier y y x^* tal que $\sigma_i(x^*) > 0$:

$$\sum_{s\in S} u_i(s)[\sigma(s) - \sigma'(s)] - \sum_{s\in S}\sigma(s)u_i(y,s_{-i}) + \sum_{s\in S}\sigma'(s)u_i(x^*,s_{-i}) \;\geq\; 0\;. \tag{A.18}$$

Por otra parte, note que:

$$\sum_{s\in S}\sigma(s)u_i(x^*,s_{-i}) - \sum_{s\in S}\sigma'(s)u_i(x^*,s_{-i}) \tag{A.19}$$

$$= \sum_{s_{-i}} u_i(x^*,s_{-i})\sigma_{-i}(s_{-i})\sum_{z\in S_i}[\sigma_i(z)-\sigma_i'(z)] \tag{A.20}$$

$$= \sum_{s_{-i}} u_i(x^*,s_{-i})\sigma_{-i}(s_{-i})\left[\sum_{z\in S_i}\sigma_i(z) - \sum_{z\in S_i}\sigma_i'(z)\right] \tag{A.21}$$

$$= 0\,. \tag{A.22}$$

Luego, al tomar $y = x^*$ en (A.18),

$$\sum_{s\in S} u_i(s)[\sigma(s)-\sigma'(s)] \;=\; \sum_{s\in S}u_i(s)\sigma(s) - \sum_{s\in S}u_i(s)\sigma'(s) \;=\; u_i(\sigma) - u_i(\sigma_i',\sigma_{-i}) \;\geq\; 0\,. \tag{A.23}$$

Como σ_i' es una estrategia cualquiera para el jugador i, σ es un equilibrio de Nash. □

Teorema A.6. *Sean σ y σ' dos equilibrios correlacionados, y α un número real en $(0,1)$. Entonces, la distribución $\alpha\sigma + (1-\alpha)\sigma'$ es un equilibrio correlacionado.*

Demostración. Como σ y σ' son equilibrios correlacionados y $\alpha, 1-\alpha \in (0,1)$ se cumple que para cualesquiera x e y:

$$\alpha \sum_{s_{-i}\in S_{-i}} \sigma(x,s_{-i})[u_i(x,s_{-i}) - u_i(y,s_{-i})] \geq 0 \text{ y} \tag{A.24}$$

$$(1-\alpha) \sum_{s_{-i}\in S_{-i}} \sigma'(x,s_{-i})[u_i(x,s_{-i}) - u_i(y,s_{-i})] \geq 0 \tag{A.25}$$

Sumando las ecuaciones anteriores y factorizando se obtiene:

$$\sum_{s_{-i}\in S_{-i}} [\alpha\sigma(x,s_{-i}) + (1-\alpha)\sigma'(x,s_{-i})][u_i(x,s_{-i}) - u_i(y,s_{-i})] \geq 0 \tag{A.26}$$

Por lo que $\alpha\sigma + (1-\alpha)\sigma'$ es un equilibrio correlacionado. □

A.2. Capítulo II

Teorema A.7. *Dado un juego en forma extensa y un jugador i, tal que: si $h' \sqsubset h$ y $P(h') = P(h) = i$, entonces $I(h') \neq I(h)$. Luego, para cualquier estrategia de comportamiento*

$\sigma_i^b \in B^i$, la estrategia mixta σ_i^m dada por:

$$\sigma_i^m(s_i) := \prod_{I_i \in \mathcal{I}_i} \sigma_i^b(I_i)(s_i(I_i)) \tag{A.27}$$

es equivalente a la estrategia σ_i^b.

Demostración. Se quiere probar que para todo $z \in Z$, se tiene que $\pi^{\sigma_i^m}(z) = \pi^{\sigma_i^b}(z)$. Para cualquier estrategia se denotará con $\sigma_i(s)$ la probabilidad de elegir la estrategia s_i bajo σ_i. Además, la probabilidad de elegir una estrategia s_i bajo la estrategia σ_i^b es exactamente el lado derecho de la Ecuación A.27, la cual, por definición es la probabilidad de elegir s_i bajo σ_i^m. Luego se tiene que $\sigma_i^b(s_i) = \sigma_i^m(s_i)$ para cualquier estrategia pura $s_i \in S_i$.

Por otra parte, como ninguna historia atraviesa más de una vez el mismo conjunto de información, se tiene que para cualquier estrategia σ_i (mixta o de comportamiento):

$$\pi^{\sigma_i}(z) = \sum_{\substack{s_i \in S_i \\ z \text{ esalcanzable} \\ \text{por } s_i}} \sigma_i(s_i) \tag{A.28}$$

Luego, $\pi^{\sigma_i^b}(z) = \pi^{\sigma_i^m}(z)$ para todo $z \in Z$, obteniendo el resultado deseado. □

Teorema A.8. *Dado un juego finito de N personas en el que el jugador i tiene "perfect recall". Entonces, para cada estrategia mixta $\sigma_i^m \in \Delta(S_i)$ del jugador i, existe una estrategia de comportamiento $\sigma_i^b \in B^i$, equivalente a σ_i^m.*

Demostración. Se denotará por $\pi^{\sigma_i}(I_i, a)$ la probabilidad, bajo σ_i, que I_i sea alcanzable y se elija la acción a. De forma más general se denotará con $\pi^{\sigma_i}(I_i, a_1, a_2, ..., a_k)$ la probabilidad que I_i sea alcanzable y que luego el jugador i elija las acciones $a_1, a_2, ...a_k$. Luego se elige la siguiente estrategia de comportamiento:

$$\begin{aligned}
\sigma_i^b(I_i)(a) &= P[\text{se elija } a \text{ bajo } \sigma_i^m | I_i \text{ es alcanzable bajo } \sigma_i^m] && \text{(A.29)} \\
&= \frac{P[I_i \text{ sea alcanzable bajo } \sigma_i^m \text{ y se elija la opción } a \text{ bajo } \sigma_i^m]}{P[I_i \text{ es alcanzable bajo } \sigma_i^m]} && \text{(A.30)} \\
&= \frac{\pi^{\sigma_i^m}(I_i, a)}{\pi^{\sigma_i^m}(I_i)} && \text{(A.31)}
\end{aligned}$$

En caso que $\pi^{\sigma_i^m}(I_i) > 0$ y de forma arbitraria en caso contrario.

Se demostrará que $\pi^{\sigma_i^b}(z) = \pi^{\sigma_i^m}(z)$, cuando $\pi^{\sigma_i^m}(z) > 0$.

Dado $z \in Z$, sean $a_1, a_2, ..., a_k$ las acciones elegidas por el jugador i (en ese orden), y sean $I_i^1, I_i^2, ..., I_i^k$ los conjuntos de información respectivos. Note que $\pi^{\sigma_i^m}(I_i^j, a_i^j) =$

$\pi^{\sigma_i^m}(I_i^{j+1})$, luego:

$$\pi^{\sigma_i^b}(z) \;=\; \prod_{j=1}^{k} \sigma_i^b(I_i^j)(a_i^j) \;=\; \prod_{j=1}^{k} \frac{\pi^{\sigma_i^m}(I_i^j, a_j)}{\pi^{\sigma_i^m}(I_i^j)} \;=\; \frac{\pi^{\sigma_i^m}(I_i^k, a_k)}{\pi^{\sigma_i^m}(I_i^1)} \;=\; \pi^{\sigma_i^m}(I_i^k, a_k) \tag{A.32}$$

Además, usando inducción, se obtiene que para cualquier $k' < k$ se tiene:

$$\pi^{\sigma_i^m}(I_i^k, a_k) \;=\; \pi^{\sigma_i^m}(I_i^{k'}, a_{k'}, a_{k'+1}, ..., a_k) \tag{A.33}$$

Entonces

$$\pi^{\sigma_i^m}(I_i^k, a_k) = \pi^{\sigma_i^m}(I_i^1, a_1, a_2, ..., a_k) = \pi^{\sigma_i^m}(z) \tag{A.34}$$

Obteniendo $\pi^{\sigma_i^b}(z) = \pi^{\sigma_i^m}(z)$, que era lo que se quería demostrar. □

A.3. Capítulo IV

Teorema A.9. *Sea $(s_t)_{t=1,2,...}$ una secuencia de juegos de Γ. Entonces, $R_i^t(j,k)$ converge a 0 para cada i y cada $j,k \in S_i$, con $j \neq k$, si y sólo si la secuencia de distribuciones empíricas z_t converge al conjunto de equilibrio correlacionado.*

Demostración. Note que:

$$D_i^t(j,k) \;=\; \frac{1}{t} \sum_{\substack{1 \leq \tau \leq t \\ s_i^\tau = j}} u_i(k, s_{-i}^\tau) - u_i(s^\tau) \tag{A.35}$$

$$= \sum_{\substack{s \in S \\ s_i = j}} \frac{1}{t} |\{1 \leq \tau \leq t : s^\tau = s\}| \left[u_i(k, s_{-i}) - u_i(s)\right] \tag{A.36}$$

$$= \sum_{\substack{s \in S \\ s_i = j}} z_t(s) \left[u_i(k, s_{-i}) - u_i(s)\right]. \tag{A.37}$$

Dado $\varepsilon > 0$, $R_i^t(j,k) \leq \varepsilon$ si y sólo si:

$$\sum_{s \in S : s_i = j} z_t(s) \left[u_i(k, s_{-i}) - u_i(s)\right] = D_i^t(j,k) \;\leq\; \varepsilon \tag{A.38}$$

Obteniendo que $R_i^t(j,k) \leq \varepsilon$ para todo $i \in N$ y todo $j,k \in S_i$ si y sólo si z_t es un ε-equilibrio correlacionado. Por lo tanto, todos los *regrets* convergen a cero si y sólo si z_t converge al conjunto de equilibrio correlacionado. □

Teorema A.10. *Supongamos que a cada período $t+1$, el jugador i elige las estrategias acorde a un vector de distribución de probabilidad q_t^i que satisface* (4.9)*. Entonces, $R_t^i(j,k)$ converge a cero (a. s.) para todo $j,k \in S_i$ con $j \neq k$.*

Demostración. La prueba es una aplicación directa del Teorema de Aproximación de Blackwell con L, v y $\mathcal{C}$ definidos de la siguiente manera:

- $L = \{(j,k) \in S_i \times S_i : j \neq k\}$
- $v(s_i, s_{-i}) \in \mathbb{R}^L$ dado por

$$[v(s_i, s_{-i})](j,k) \;=\; \begin{cases} u_i(k, s_{-i}) - u_i(j, s_{-i}) & \text{si } s_i = j \\ 0 & \text{en otro caso,} \end{cases} \tag{A.39}$$

- $\mathcal{C} = \mathbb{R}_-^L = \{x \in \mathbb{R}^L : x_i \leq 0 \ \forall i \in L\}$ es decir, el ortante negativo.

Demostraremos que $\mathcal{C}$ es alcanzable por i. Note que:

$$w_{\mathcal{C}}(\lambda) \;=\; \sup\{\lambda \cdot c : c \in \mathcal{C}\} \;=\; \sup\left\{\sum_{i \in L} \lambda_i c_i : c_i \leq 0\right\}. \tag{A.40}$$

Luego, si $\lambda_i \geq 0$, $\forall i \in L$, entonces $\lambda \cdot c \leq 0$ para todo $c \in \mathcal{C}$, y $w_{\mathcal{C}}(\lambda) = 0$. Por otra parte, si $\lambda_i < 0$ para algún $i \in N$, entonces $c_i\lambda_i$ no está acotado superiormente y $w_{\mathcal{C}}(\lambda) = \infty$. Luego,

$$w_{\mathcal{C}} \;=\; \begin{cases} 0 & \text{si } \lambda \in \mathbb{R}_+^L, \\ \infty & \text{en caso contrario.} \end{cases} \tag{A.41}$$

Por otra parte, se tiene que:

$$\lambda \cdot v(q_\lambda, s_{-i}) \;=\; \sum_{(j,k) \in L} \lambda(j,k) \cdot [v(q_\lambda, s_{-i})](j,k) \tag{A.42}$$

$$= \sum_{(j,k) \in L} \lambda(j,k) \left[\sum_{s_i \in S_i} q_\lambda(s_i) v(s_i, s_{-i})\right](j,k) \tag{A.43}$$

$$= \sum_{(j,k) \in L} \lambda(j,k) q_\lambda(j) [v(j, s_{-i})](j,k) \tag{A.44}$$

$$= \sum_{(j,k) \in L} \lambda(j,k) q_\lambda(j) [u_i(k, s_{-i}) - u_i(j, s_{-i})] \tag{A.45}$$

$$= \sum_{(j,k) \in L} \lambda(j,k) q_\lambda(j) u_i(k, s_{-i}) - \sum_{(j,k) \in L} \lambda(j,k) q_\lambda(j) u_i(j, s_{-i}) \tag{A.46}$$

$$= \sum_{k\in S_i} u_i(k, s_{-i}) \sum_{j\in S_i} \lambda(j,k) q_\lambda(j) - \sum_{j\in S_i} q_\lambda(j) u_i(j, s_{-i}) \sum_{k\in S_i} \lambda(j,k) \quad (A.47)$$

$$= \sum_{j\in S_i} u_i(j, s_{-i}) \sum_{k\in S_i} \lambda(k,j) q_\lambda(k) - \sum_{j\in S_i} q_\lambda(j) u_i(j, s_{-i}) \sum_{k\in S_i} \lambda(j,k) \quad (A.48)$$

$$= \sum_{j\in S_i} u_i(j, s_{-i}) \left[\sum_{k\in S_i} \lambda(k,j) q_\lambda(k) - q_\lambda(j) \sum_{k\in S_i} \lambda(j,k) \right] . \quad (A.49)$$

Defina

$$\alpha(j) \;=\; \sum_{k\in S_i} \lambda(k,j) q_\lambda(k) - q_\lambda(j) \sum_{k\in S_i} \lambda(j,k) \,. \quad (A.50)$$

Entonces, $\lambda \cdot v(q_\lambda, s_{-i}) = \sum_{j\in S_i} u_i(j, s_{-i}) \alpha(j)$. Luego, en este caso, la condición del Teorema C.4 es equivalente a:

$$\sum_{j\in S_i} u_i(j, s_{-i}) \alpha(j) \;\leq\; 0 \,. \quad (A.51)$$

Si se elige q_λ que cumpla:

$$q_\lambda(j) \sum_{k\in S_i} \lambda(j,k) \;=\; \sum_{k\in S_i} \lambda(k,j) q_\lambda(k) \quad (A.52)$$

para todo $j \in S_i$, entonces $\alpha(j) = 0$ para $j \in S_i$, y la condición del Teorema C.4 se cumple como igualdad cuando $\mathcal{C} = \mathbb{R}^L_-$.

Por otra parte, sea $D_t = \frac{1}{t} \sum_{\tau=1}^{t} v(s_\tau)$ el promedio de los vectores de pago a tiempo t. Entonces,

$$D_t[j,k] \;=\; \sum_{\tau=1}^{t} v(s_\tau)[j,k] \;=\; \sum_{1\leq\tau\leq t, s_i^\tau = j} u_i(k, s_{-i}^\tau) - u_i(j, s_{-i}^\tau) \;=\; D_i^t(j,k) \,. \quad (A.53)$$

Para $x \notin \mathbb{R}^-$, $F(x) = x^-$ y $\lambda(x) = x - x^- = x^+$, obteniendo

$$\lambda(D_t) \;=\; (R_t^i(j,k))_{(j,k)\in L} \,. \quad (A.54)$$

Luego, usar una estrategia que cumpla

$$q_\lambda(j) \sum_{k\in S_i} \lambda(j,k) \;=\; \sum_{k\in S_i} q_\lambda(k) \lambda(k,j) \quad (A.55)$$

cuando $\lambda(j,k) = [D_i^t(j,k)]^+ = R_t^i(j,k)$ es equivalente que la estrategia $p_{t+1}^i \in \Delta(S_i)$

cumpla con

$$p_{t+1}^i(j)\sum_{k\in S_i} R_i^t(j,k) \;=\; \sum_{k\in S_i} R_i^t(k,j)p_{t+1}^i(k) \tag{A.56}$$

Aplicando el Teorema C.4 se tiene que al usar dicha estrategia, D_t alcanza a $\mathbb{R}^-$ que es equivalente a que $R_i^t(j,k)\to 0$ para todo $j,k\in S_i$. □

Teorema A.11. *El procedimiento adaptativo definido en* (4.13) *es universalmente consistente para el jugador* i.

Demostración. La prueba es similar a la del procedimiento anterior. Se definen L, v y $\mathcal{C}$ del Teorema C.4 de la siguiente manera:

- $L = S_i$,
- $v = v(s_i, s_{-i}) \in \mathbb{R}^L$ dada por: $[v(s_i,s_{-i})](k) = u_i(k,s_{-i}) - u_i(s_i,s_{-i})$,
- $\mathcal{C} = \mathbb{R}^L_- = \{x\in\mathbb{R}^L : x_i \le 0\ \forall i\in L\}$ (i.e. el ortante negativo).

Se demostrará que $\mathcal{C}$ es alcanzable por i. Al igual que antes, se tiene que:

$$w_{\mathcal{C}} \;=\; \begin{cases} 0 & \text{si } \lambda\in\mathbb{R}^L_+\,, \\ \infty & \text{en caso contrario.}\end{cases} \tag{A.57}$$

Por otra parte,

$$\begin{aligned}
\lambda\cdot v(q_\lambda, s_{-i}) &= \sum_{k\in L}\lambda(k)\cdot[v(q_\lambda,s_{-i})](k) && \text{(A.58)}\\
&= \sum_{k\in S_i}\lambda(k)\cdot\sum_{j\in S_i} q_\lambda(j)[v(j,s_{-i})](k) && \text{(A.59)}\\
&= \sum_{k\in S_i}\lambda(k)\cdot\sum_{j\in S_i} q_\lambda(j)[u_i(k,s_{-i}) - u_i(j,s_{-i})] && \text{(A.60)}\\
&= \sum_{\substack{k\in S_i\\ j\in S_i}}\lambda(k)q_\lambda(j)[u_i(k,s_{-i}) - u_i(j,s_{-i})] && \text{(A.61)}\\
&= \sum_{\substack{k\in S_i\\ j\in S_i}}\lambda(k)q_\lambda(j)u_i(k,s_{-i}) - \sum_{\substack{k\in S_i\\ j\in S_i}}\lambda(k)q_\lambda(j)u_i(j,s_{-i}) && \text{(A.62)}\\
&= \sum_{\substack{j\in S_i\\ k\in S_i}} u_i(j,s_{-i})\lambda(j)q_\lambda(k) - \sum_{\substack{j\in S_i\\ j\in S_i}} u_i(j,s_{-i})\lambda(k)q_\lambda(j) && \text{(A.63)}\\
&= \sum_{\substack{j\in S_i\\ k\in S_i}} u_i(j,s_{-i})[\lambda(j)q_\lambda(k) - \lambda(k)q_\lambda(j)] && \text{(A.64)}
\end{aligned}$$

$$= \sum_{j \in S_i} u_i(j, s_{-i}) \left[\lambda(j) \sum_{k \in S_i} q_\lambda(k) - q_\lambda(j) \sum_{k \in S_i} \lambda(k) \right] \tag{A.65}$$

$$= \sum_{j \in S_i} u_i(j, s_{-i}) \left[\lambda(j) - q_\lambda(j) \sum_{k \in S_i} \lambda(k) \right] . \tag{A.66}$$

La última igualdad porque $\sum_{k \in S_i} q_\lambda(k) = 1$. Luego, si se define:

$$\alpha(j) \;=\; \lambda(j) - q_\lambda(j) \sum_{k \in S_i} \lambda(k) \,, \tag{A.67}$$

obtenemos $\lambda \cdot v(q_\lambda, s_{-i}) = \sum_{j \in S_i} u_i(j, s_{-i}) \alpha(j)$. Note que si $q_{\lambda(j)} = \frac{\lambda(j)}{\sum_{k \in S_i} \lambda(k)}$, entonces $\alpha(j) = 0$ para todo $j \in S_i$ y se cumple la condición del Teorema C.4 en forma de igualdad. Además, para $D_t = \frac{1}{t} \sum_{\tau=1}^{t} v(s_\tau)$, tenemos

$$D_t[k] \;=\; \sum_{\tau=1}^{t} v(s_\tau)[k] \;=\; \sum_{\tau \leq t} [u_i(k, s^\tau_{-i}) - u_i(s_\tau)] \;=\; D_i^t(k) \,. \tag{A.68}$$

Luego $F(D_t) = D_t^-$ y $\lambda(D_t) = D_t^+ = (R_i^t(k))_{k \in S_i}$, obteniendo:

$$q_{\lambda(D_t)} \;=\; \frac{[\lambda(D_t)](j)}{\sum_{k \in S_i} [\lambda(D_t)](k)} \;=\; \frac{R_i^t(j)}{\sum_{k \in S_i} R_i^t(k)} \tag{A.69}$$

Al elegir $p_{t+1}(j) = q_{\lambda(D_t)}(j) = \frac{R_i^t(j)}{\sum_{k \in S_i} R_i^t(k)}$, se obtiene que D_t alcanza a $\mathbb{R}^-$, lo cual es equivalente a que $R_i^t(j) \to 0$ para todo $j \in S_i$. □

Teorema A.12. *Sea Γ un juego de dos jugadores de suma cero y sea $(s^t)_{t=1,2,\ldots,T}$ una secuencia de juegos de Γ, tales que, para todo $s_i \in S_i$, para todo $i \in 1, 2$:*

$$\frac{1}{T} \sum_{t=1}^{T} u_i(s_i, s^t_{-i}) - \frac{1}{T} \sum_{t=1}^{T} u_i(s^t) \leq \varepsilon \tag{A.70}$$

para algún $\varepsilon > 0$. Sea $\bar{\sigma}^T = (\bar{\sigma_1}^T, \bar{\sigma_2}^T)$, donde:

$$\bar{\sigma}_i^T(s_i) = \frac{|\{1 \leq T : s_i^t = s_i\}|}{T} = \frac{\#(s_i)}{T} \tag{A.71}$$

es decir, $\bar{\sigma}^T$, es la distribución empírica de probabilidad. Entonces $\bar{\sigma}^T$ es un 2ε-equilibrio de Nash.

Demostración. Por hipótesis del teorema, se tiene que:

$$\frac{1}{T}\sum_{t=1}^{T} u_i(s_i, s_{-i}^t) - \frac{1}{T}\sum_{t=1}^{T} u_i(s^t) \leq \varepsilon \tag{A.72}$$

Reordenado la sumatoria del primer término y utilizando la definición de $\bar{\sigma}$, se obtiene:

$$\frac{1}{T}\sum_{s_{-i}\in S_{-i}} \#(s_{-i}) u_i(s_i, s_{-i}) - \frac{1}{T}\sum_{t=1}^{T} u_i(s^t) \leq \varepsilon \tag{A.73}$$

$$\Rightarrow \sum_{s_{-i}\in S_{-i}} \bar{\sigma}_{-i}^T(s_{-i}) u_i(s_i, s_{-i}) - \frac{1}{T}\sum_{t=1}^{T} u_i(s^t) \leq \varepsilon \tag{A.74}$$

Sea $\sigma_i \in \Delta(S_i)$ cualquier estrategia del jugador i, luego

$$\sum_{s_i\in S_i} \sigma_i(s_i) \left[\sum_{s_{-i}\in S_{-i}} \bar{\sigma}_{-i}^T(s_{-i}) u_i(s_i, s_{-i}) - \frac{1}{T}\sum_{t=1}^{T} u_i(s^t) \right] \leq \sum_{s_i\in S_i} \sigma_i(s_i)\varepsilon \tag{A.75}$$

$$\Rightarrow \sum_{s_i\in S_i}\sum_{s_{-i}\in S_{-i}} \sigma_i(s_i)\bar{\sigma}_{-i}^T(s_{-i}) u_i(s_i, s_{-i}) - \sum_{s_i\in S_i} \sigma_i(s_i) u_i(s^t) \leq \varepsilon \tag{A.76}$$

$$\Rightarrow u_i(\sigma_i, \bar{\sigma}_{-i}^T) - \frac{1}{T}\sum_{t=1}^{T} u_i(s^t) \leq \varepsilon \tag{A.77}$$

En particular, se tiene que, para estrategias cualesquiera $\sigma_1 \in \Delta(S_1)$ y $\sigma_2 \in \Delta(S_2)$

$$u_1(\sigma_1, \bar{\sigma}_2^T) - \frac{1}{T}\sum_{t=1}^{T} u_1(s^t) \leq \varepsilon \tag{A.78}$$

$$u_2(\bar{\sigma}_1^T, \sigma_2) - \frac{1}{T}\sum_{t=1}^{T} u_2(s^t) \leq \varepsilon \tag{A.79}$$

Además, como Γ es un juego de suma cero, se tiene que $u_2(\bar{\sigma}_1^T, \sigma_2) = -u_1(\bar{\sigma}_1^T, \sigma_2)$ y $u_2(s^t) = -u_1(s^t)$, luego:

$$u_2(\bar{\sigma}_1^T, \sigma_2) - \frac{1}{T}\sum_{t=1}^{T} u_2(s^t) = -u_1(\bar{\sigma}_1^T, \sigma_2) - \frac{1}{T}\sum_{t=1}^{T} -u_1(s^t) \leq \varepsilon \tag{A.80}$$

En particular, si $\sigma_2 = \bar{\sigma_2}^T$ entonces:

$$-u_1(\bar{\sigma}_1^T, \bar{\sigma_2}^T) + \frac{1}{T}\sum_{t=1}^{T} u_1(s^t) \leq \varepsilon \tag{A.81}$$

Al sumar las desigualdades A.78 y A.81 se obtiene que:

$$u_1(\sigma_1, \bar{\sigma}_2^T) - u_1(\bar{\sigma}_1^T, \bar{\sigma}_2^T) \leq 2\varepsilon \tag{A.82}$$

$$\Rightarrow u_1(\bar{\sigma}^T) + 2\varepsilon \geq u_1(\sigma_1, \bar{\sigma}_2^T) \tag{A.83}$$

Análogamente se tiene que $u_2(\bar{\sigma}^T) + 2\varepsilon \geq u_2(\bar{\sigma_1}^T, \sigma_2)$, con lo que se concluye que $\bar{\sigma}^T$ es un 2ε-equilibrio de Nash. □

A.4. Capítulo V

Teorema A.13. *En un juego de 2 jugadores de suma cero si el average overall regret a tiempo T es menor que ε entonces σ^{-T} es un 2ε-equilibrio de Nash*

Demostración. Se probará que la probabilidad de alcanzar z bajo $\bar{\sigma}_i^T$ viene dada por el promedio de alcanzar z en cada estrategia. Sean $h_1 \sqsubset h_2 \sqsubset h_3 \sqsubset ... \sqsubset h_m \sqsubset z$ todos los prefijos de z correspondientes al jugador i, es decir $P(h_k) = i \ \forall k : 1 \leq k \leq m$ y sean $a_1, a_2, ..., a_m$ las acciones correspondientes en z en cada historia respectiva. Luego:

$$\pi^{\bar{\sigma}_i^T}(z) = \prod_{k=1}^{m} \bar{\sigma}_i^T(I(h_k))(a_k) \tag{A.84}$$

$$= \prod_{k=1}^{m} \frac{\sum_{t=1}^{T} \pi^{\sigma_i^t}(I(h_k))\sigma_i^t(I(h_k))(a)}{\sum_{t=1}^{T} \pi^{\sigma_i^t}(I(h_k))} \tag{A.85}$$

Por otra parte, note que $\pi^{\sigma_i^t}(I)\sigma_i^t(I(h_k))(a_k) = \pi^{\sigma_i^t}(I(h_{k+1}))$. Entonces:

$$\pi^{\bar{\sigma}_i^T}(z) = \frac{\sum_{t=1}^{T} \pi^{\sigma_i^t}(I_m)\sigma_i^t(I_m)(a_m)}{\sum_{t=1}^{T} \pi^{\sigma_i^t}(I_1)} \tag{A.86}$$

$$= \frac{\sum_{t=1}^{T} \pi^{\sigma_i^t}(z)}{\sum_{t=1}^{T} 1} \tag{A.87}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \pi^{\sigma_i^t}(z) \tag{A.88}$$

Además, se tiene que, para cualquier jugador i y cualquier estrategia de σ_i:

$$\frac{1}{T}\sum_{t=1}^{T} u_i(\sigma_i', \sigma_{-i}^t) = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{z\in Z}\pi^{\sigma_i'}(z)\pi^{\sigma_{-i}^t}(z)\pi^c(z)\right) \tag{A.89}$$

$$= \sum_{z\in Z} u_i(z)\pi^{\sigma_i'}(z)\pi^c(z)\left(\frac{1}{T}\sum_{t=1}^{T}\pi^{\sigma_{-i}^t}(z)\right) \tag{A.90}$$

$$= \sum_{z\in Z} u_i(z)\pi^{\sigma_i'}(z)\pi^{\bar{\sigma}_i^T}(z)\pi^c(z) \tag{A.91}$$

$$= u_i(\sigma_i', \bar{\sigma}_{-i}^T) \tag{A.92}$$

Por otra parte, como $R_2^T \leq \varepsilon$, para todo $\sigma_2' \in B_2$ se tiene que:

$$\frac{1}{T}\sum_{t=1}^{T}[u_2(\sigma_1^t, \sigma_2') - u_2(\sigma^t)] \leq \varepsilon \tag{A.93}$$

$$\Rightarrow \frac{1}{T}\sum_{t=1}^{T} u_2(\sigma^t) + \varepsilon \geq \frac{1}{T}\sum_{t=1}^{T} u_2(\sigma_1^t, \sigma_2') \tag{A.94}$$

$$\Rightarrow \frac{1}{T}\sum_{t=1}^{T} u_2(\sigma^t) + \varepsilon \geq u_2(\bar{\sigma}_1^T, \sigma_2') \tag{A.95}$$

Luego, como se cumple para cualquier σ_2', se cumple para σ_2^t para $t = 1, 2, ..., T$, obteniendo:

$$\frac{1}{T}\sum_{t=1}^{T} u_2(\sigma^t) + \varepsilon \geq \frac{1}{T}\sum_{t=1}^{T} u_2(\bar{\sigma}_1^T, \sigma_2^t) \tag{A.96}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\sum_{z\in Z} u_2(z)\pi^{\bar{\sigma}_1^T}\pi^{\sigma_2^t}(z)\pi^c(z) \tag{A.97}$$

$$= \sum_{z\in Z} u_2(z)\pi^{\bar{\sigma}_1^T}\pi^c(z)\left(\frac{1}{T}\sum_{t=1}^{T}\pi^{\sigma_2^t}(z)\right) \tag{A.98}$$

$$= \sum_{z\in Z} u_2(z)\pi^{\bar{\sigma}_1^T}\pi^{\bar{\sigma}_2^T}\pi^c(z) \tag{A.99}$$

$$= u_2(\bar{\sigma}^T) \tag{A.100}$$

Como Γ es un juego de suma cero, se tiene que $u_2 = -u_1(\sigma)$ para toda estrategia σ,

luego:

$$\frac{1}{T}\sum_{t=1}^{T} u_2(\sigma^t) + \varepsilon \;\geq\; u_2(\bar{\sigma}^T) \tag{A.101}$$

$$\Rightarrow \frac{1}{T}\sum_{t=1}^{T} -u_1(\sigma^t) + \varepsilon \;\geq\; -u_1(\bar{\sigma}^T) \tag{A.102}$$

$$\Rightarrow u_1(\bar{\sigma}^T) \;+\varepsilon \;\geq\; \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma^t) \tag{A.103}$$

$$\Rightarrow u_1(\bar{\sigma}^T) \;+2\varepsilon \;\geq\; \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma^t) + \varepsilon \tag{A.104}$$

Por otra parte, como $R_i^t \leq \varepsilon$ se tiene que, para cualquier $\sigma_1' \in B_1$:

$$\frac{1}{T}\sum_{t=1}^{T} u_1(\sigma^t) + \varepsilon \;\geq\; \frac{1}{T}\sum_{t=1} u_1(\sigma_1', \sigma_2^t) \;=\; u_1(\sigma_1', \bar{\sigma}_2^T) \tag{A.105}$$

Luego, se obtiene que:

$$u_1(\bar{\sigma}^T) \;+2\varepsilon \;\geq\; \frac{1}{T}\sum_{t=1}^{T} u_1(\sigma^t) + \varepsilon \;\geq\; u_1(\sigma_1', \bar{\sigma}_2^T) \tag{A.106}$$

$$\Rightarrow u_1(\bar{\sigma}^T) \;+2\varepsilon \;\geq\; u_1(\sigma_1', \bar{\sigma}_2^T) \tag{A.107}$$

Análogamente, se demuestra que:

$$u_2(\bar{\sigma}^T) \;+2\varepsilon \;\geq\; u_2(\bar{\sigma}_1^T, \sigma_2') \tag{A.108}$$

Concluyendo que $\bar{\sigma}^T$ es un 2ε-equilibrio correlacionado. □

APÉNDICE B

RESULTADOS EXPERIMENTALES, REGRET MATCHING EN JUEGOS EN FORMA NORMAL

En este apéndice se presentan las tablas y gráficas detalladas para los juegos en forma normal descritos en la sección **??**. Para cada juego se muestra una tabla con la estrategia obtenida en la última corrida de cada uno de los procedimientos y, en caso de conocerse, el equilibrio de Nash. Para cada estrategia se muestra la utilidad de cada jugador si utilizan una mejor respuesta frente a la estrategia calculada para el oponente v_1 y v_2, así como la explotabilidad ε_σ (ver la Sección III para definiciones formales).

Además, se presenta una tabla que indica el tiempo de cada ejecución (T), el número de iteraciones para alcanzar la cota deseada (I) y el tiempo promedio de cada iteración en cada una de las ejecuciones (T/I), así como el promedio del tiempo y del número de iteraciones para cada procedimiento. También se muestran las gráficas del regret por iteraciones, para observar su convergencia. Estas gráficas son mostradas con una escala logarítmica en el eje x para apreciar mejor los resultados.

B.1. Matching Pennies

En este juego, si un jugador elige cada acción con una probabilidad de 0.5, entonces su ganancia esperada es igual a 0, sin importar la estrategia de su oponente, obteniendo el equilibrio de Nash cuando ambos jugadores utilizan esta estragia. Las estrategias obtenidas no corresponden al equilibrio de Nash, sin embargo, garantizan una utilidad cercana a 0 en todos los casos, obteniendo una explotabilidad no mayor a 0.008, como se muestra en la Tabla B.1. Por lo que todas las estrategias obtenidas son un ε- equilibrio de Nash, con $\varepsilon < 0.008$.

La Tabla B.2 muestra los resultados obtenidos relacionados al tiempo y número de iteraciones de los procedimientos. El procedimiento A, regret condicional, tuvo una duración promedio de 10.276 segundos, con un número promedio de iteraciones de 3892550.4, obteniendo un promedio de 2.64×10^{-6} segundos por iteración. Con el procedimiento B,

	E.N.	A	B	C
σ_1	$(0.500, 0.500)$	$(0.500, 0.500)$	$(0.500, 0.500)$	$(0.500, 0.500)$
σ_2	$(0.500, 0.500)$	$(0.497, 0.503)$	$(0.503, 0.497)$	$(0.504, 0.496)$
(v_1, v_2)	$(0.000, 0.000)$	$(0.006, 0.000)$	$(0.006, 0.000)$	$(0.008, 0.000)$
ε_σ	0	0.006	0.006	0.008

Tabla B.1: Estrategias obtenidas del juego Matching Pennies

que utiliza un vector invariante de probabilidad, se obtuvo un tiempo, número de iteraciones y tiempo por iteración promedios de 3.777 segundos, 25616.6 iteraciones y 3.03×10^{-5} segundos por iteración, respectivamente. Por último, el procedimiento C, regret incondicional, se obtuvo un tiempo promedio de 0.042, el número de iteraciones promedio fue de 16260.5, obteniendo un promedio de 2.58×10^{-6} segundos por iteración.

A			B			C		
T	I	T/I	T	I	T/I	T	I	T/I
7.663	3068341	2.50×10^{-06}	0.985	32510	3.03×10^{-05}	0.002	955	2.53×10^{-06}
9.650	3857071	2.50×10^{-06}	1.748	56946	3.07×10^{-05}	0.064	24968	2.55×10^{-06}
23.313	8950013	2.60×10^{-06}	0.552	18401	3.00×10^{-05}	0.061	23854	2.57×10^{-06}
11.757	4240611	2.77×10^{-06}	0.309	10197	3.03×10^{-05}	0.025	9724	2.57×10^{-06}
2.377	877335	2.71×10^{-06}	0.747	24892	3.00×10^{-05}	0.011	4188	2.59×10^{-06}
5.062	1818992	2.78×10^{-06}	0.848	28142	3.01×10^{-05}	0.025	9666	2.60×10^{-06}
4.281	1557496	2.75×10^{-06}	0.132	4405	3.01×10^{-05}	0.045	16951	2.64×10^{-06}
22.110	8230100	2.69×10^{-06}	1.307	43116	3.03×10^{-05}	0.021	8155	2.64×10^{-06}
3.691	1432846	2.58×10^{-06}	0.639	21311	3.00×10^{-05}	0.093	35270	2.64×10^{-06}
12.853	4892699	2.63×10^{-06}	0.500	16246	3.08×10^{-05}	0.076	28874	2.64×10^{-06}
10.276	3892550.4	2.64×10^{-06}	0.777	25616.6	3.03×10^{-05}	0.042	16260.5	2.58×10^{-06}

Tabla B.2: Resultados del juego Matching Pennies

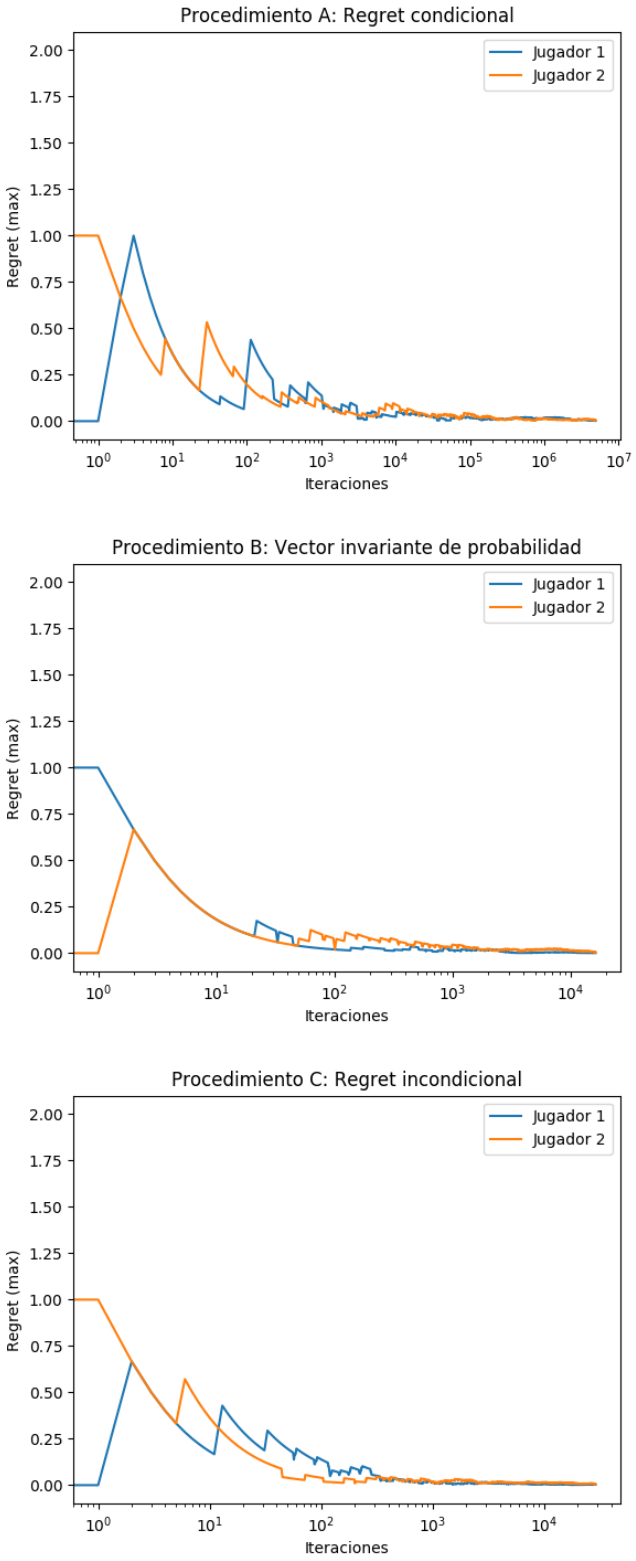
La Figura B.1 muestra el regret incondicional con respecto al tiempo de la última corrida, para los 3 procedimientos. Se observa que en todos los casos el *regret* total de cada jugador converge a cero.

B.2. Piedra, Papel o Tijeras

En este juego, al igual que en el anterior, ambos jugadores pueden garantizar una utilidad esperada de 0 sin importar la estrategia utilizada por su oponente, que se obtiene al elegir cada acción con igual probabilidad. Las estrategias obtenidas son presentadas en la tabla B.3. No todas corresponden al equilibrio de Nash exacto, sin embargo, cada una de ellas es un ε-equilibrio de Nash con $\varepsilon < 0.01$.

La Tabla B.4 muestra los resultados obtenidos relacionados al tiempo y número de iteraciones de los procedimientos. El procedimiento A, *regret* condicional, tuvo una duración promedio de 25.715 segundos, con un número promedio de iteraciones de 4519054.1,

Figura B.1: Gráficas del regret con respecto al número de iteraciones del juego Matching Pennies



		Estrategias	v_1/v_2	ε_σ
EN	σ_1	$(0.333, 0.333, 0.333)$	0.000	0.000
	σ_2	$(0.333, 0.333, 0.333)$	0.000	
A	σ_1	$(0.332, 0.335, 0.332)$	0.003	0.006
	σ_2	$(0.331, 0.334, 0.335)$	0.003	
B	σ_1	$(0.330, 0.334, 0.336)$	0.006	0.010
	σ_2	$(0.329, 0.335, 0.337)$	0.004	
C	σ_1	$(0.333, 0.337, 0.330)$	0.005	0.009
	σ_2	$(0.336, 0.330, 0.335)$	0.004	

Tabla B.3: Estrategias obtenidas del juego Piedra, Papel o Tijeras

obteniendo un promedio de 2.7×10^{-6} segundos por iteración. Con el procedimiento B, que utiliza un vector invariante de probabilidad, se obtuvo un tiempo, número de iteraciones y tiempo por iteración promedios de 0.345 segundos, 6601.3 iteraciones y 5.23×10^{-5} segundos por iteración, respectivamente. Por último, el procedimiento C, *regret* incondicional, se obtuvo un tiempo promedio de 0.049, el número de iteraciones promedio fue de 19321.1, obteniendo un promedio de 2.54×10^{-6} segundos por iteración.

A			B			C		
25.715	9107389	2.82×10^{-06}	0.724	13750	5.26×10^{-05}	0.034	12967	2.64×10^{-06}
29.494	10951479	2.69×10^{-06}	0.692	13257	5.22×10^{-05}	0.041	16096	2.57×10^{-06}
7.015	2641656	2.66×10^{-06}	0.000	6	4.36×10^{-05}	0.063	24423	2.56×10^{-06}
4.610	1748365	2.64×10^{-06}	0.849	16255	5.22×10^{-05}	0.048	18613	2.56×10^{-06}
8.051	3033028	2.65×10^{-06}	0.000	3	4.28×10^{-05}	0.082	32222	2.55×10^{-06}
9.870	3717278	2.66×10^{-06}	0.000	3	4.28×10^{-05}	0.084	33042	2.54×10^{-06}
2.749	1037895	2.65×10^{-06}	0.000	3	4.06×10^{-05}	0.049	19316	2.55×10^{-06}
11.971	4517546	2.65×10^{-06}	0.556	10644	5.23×10^{-05}	0.024	9601	2.54×10^{-06}
14.974	5606070	2.67×10^{-06}	0.000	3	3.74×10^{-05}	0.014	5621	2.55×10^{-06}
7.532	2829835	2.66×10^{-06}	0.631	12089	5.22×10^{-05}	0.054	21310	2.55×10^{-06}
12.198	4519054.1	2.70×10^{-06}	0.345	6601.3	5.23×10^{-05}	0.049	19321.1	2.54×10^{-06}

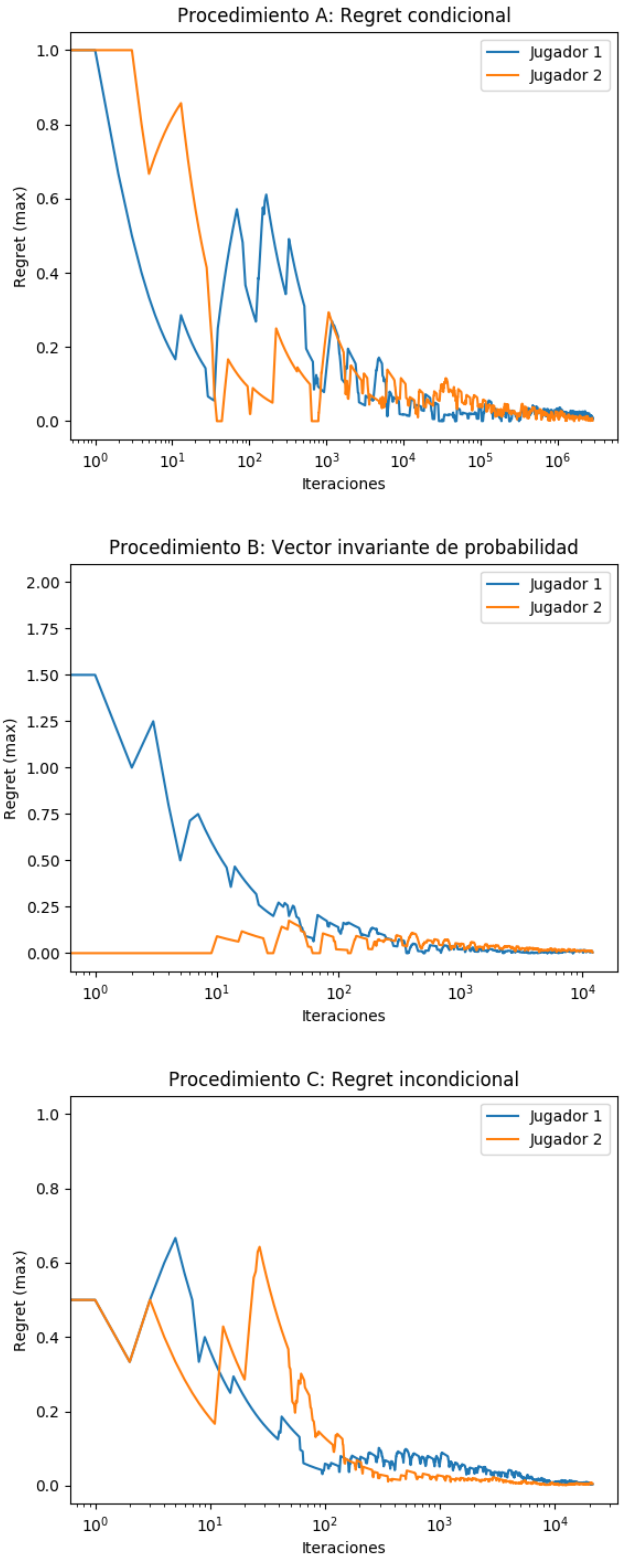
Tabla B.4: Resultados del juego Piedra, Papel o Tijeras

La Figura B.2 muestra el regret incondicional con respecto al tiempo de la última corrida para los procedimientos A, B y C. Se observa como el *regret* total de ambos jugadores converge a cero.

B.3. Ficha vs. Dominó

El primer jugador puede garantizar una ganancia esperada de, al menos 1/3, por lo que el segundo jugador puede garantizar no perder más de 1/3. A diferencia de los juegos anteriores, la matriz de pagos de este juegos no es simétrica y el primer jugador tiene ventaja sobre el segundo. Además, este juego no tiene un equilibrio de Nash único. En

Figura B.2: Gráficas del regret con respecto al número de iteraciones del juego Piedra, Papel o Tijeras



la Tabla B.3 se observa que las estrategias obtenidas para el primer jugador le permiten obtener una ganancia esperada al menos de 0.330, 0.326 y 0.329, respectivamente para los procedimientos A, B y C. Todos estos valores son menores que 1/3, pero con una diferencia menor que 0.01. Por otra parte el segundo jugador puede garantizar un valor esperado no menor que -0.338 con cualquiera de los procedimientos.

		Estrategias	v_1/v_2	ε_σ
EN	σ_1	$(0.333, 0.333, 0.000, 0.000, 0.000, 0.000, 0.333)$	0.333	0.000
	σ_2	$(0.333, 0.000, 0.333, 0.000, 0.333, 0.000)$	-0.333	
A	σ_1	$(0.136, 0.137, 0.116, 0.118, 0.198, 0.081, 0.214)$	0.338	0.010
	σ_2	$(0.165, 0.171, 0.163, 0.166, 0.166, 0.169)$	-0.328	
B	σ_1	$(0.121, 0.118, 0.135, 0.137, 0.214, 0.078, 0.198)$	0.335	0.007
	σ_2	$(0.157, 0.178, 0.156, 0.177, 0.157, 0.175)$	-0.331	
C	σ_1	$(0.128, 0.128, 0.129, 0.134, 0.208, 0.073, 0.202)$	0.334	0.004
	σ_2	$(0.169, 0.165, 0.168, 0.164, 0.169, 0.165)$	-0.330	

Tabla B.5: Estrategias obtenidas del juego Ficha vs Dominó

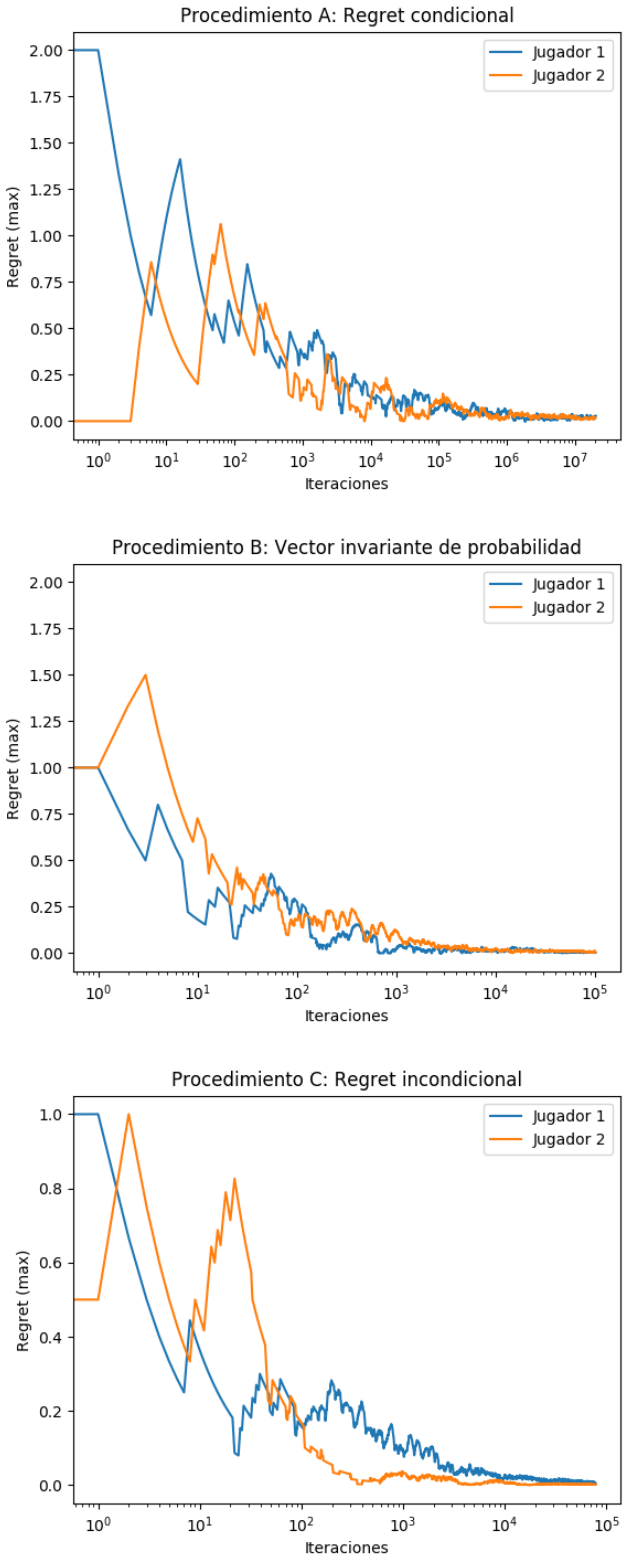
La Tabla B.6 muestra los resultados obtenidos relacionados al tiempo y número de iteraciones de los procedimientos de este juego. El procedimiento A, regret condicional, tuvo una duración promedio de 319.179 segundos, con un número promedio de iteraciones de 108319272.4, obteniendo un promedio de 2.95×10^{-6} segundos por iteración. Con el procedimiento B, que utiliza un vector invariante de probabilidad, se obtuvo un tiempo, número de iteraciones y tiempo por iteración promedios de 11.275 segundos, 75250.2 iteraciones y 1.5×10^{-4} segundos por iteración, respectivamente. Por último, el procedimiento C, regret incondicional, se obtuvo un tiempo promedio de 0.237, el número de iteraciones promedio fue de 84318.5, obteniendo un promedio de 2.81×10^{-6} segundos por iteración.

A			B			C		
669.839	215859538	3.10×10^{-06}	4.458	29721	1.50×10^{-04}	0.188	66700	2.81×10^{-06}
309.685	117568373	2.63×10^{-06}	9.019	60333	1.49×10^{-04}	0.260	92401	2.82×10^{-06}
399.170	152612646	2.62×10^{-06}	3.646	24338	1.50×10^{-04}	0.212	75674	2.81×10^{-06}
131.570	38097125	3.45×10^{-06}	12.996	86898	1.50×10^{-04}	0.145	51776	2.80×10^{-06}
263.482	96741015	2.72×10^{-06}	4.516	30170	1.50×10^{-04}	0.134	47862	2.80×10^{-06}
203.854	77156602	2.64×10^{-06}	15.420	103021	1.50×10^{-04}	0.385	136950	2.81×10^{-06}
201.267	76467409	2.63×10^{-06}	17.399	115935	1.50×10^{-04}	0.351	124882	2.81×10^{-06}
316.007	97849871	3.23×10^{-06}	17.266	115056	1.50×10^{-04}	0.203	72315	2.81×10^{-06}
383.736	110341861	3.48×10^{-06}	12.805	85532	1.50×10^{-04}	0.271	96438	2.81×10^{-06}
313.177	100498284	3.12×10^{-06}	15.227	101498	1.50×10^{-04}	0.220	78187	2.81×10^{-06}
319.179	108319272.4	2.95×10^{-06}	11.275	75250.2	1.50×10^{-04}	0.237	84318.5	2.81×10^{-06}

Tabla B.6: Resultados del juego Ficha vs Dominó

La Figura B.3 muestra el regret incondicional con respecto al tiempo de la última corrida, para los procedimientos A, B y C. Se observa como el *regret* máximo converge a

Figura B.3: Gráficas del regret con respecto al número de iteraciones del juego Ficha vs. Dominó



cero para ambos jugadores en cada uno de los procedimientos.

B.4. Coronel Blotto

En este juego no se posee un equilibrio de Nash como referencia. Sin embargo, como la matriz de pagos es simétrica, el valor del juego debe ser 0, así que las estrategias obtenidas, se mostradas en la Tabla B.7, deben garantizar un valor esperado cercano a 0. En esta tabla, también se observa que cada una de las estrategias tienen una explotabilidad menor o igual que 0.011.

Estrategias
Procedimiento A
$(0, 0, 0.126, 0.113, 0, 0, 0, 0.080, 0, 0.100, 0, 0.131, 0, 0.001, 0.111, 0.118, 0.094, 0.124, 0, 0, 0)$
$(0, 0, 0.101, 0.109, 0, 0, 0, 0.116, 0, 0.139, 0, 0.132, 0, 0.002, 0.076, 0.076, 0.141, 0.106, 0, 0, 0)$
$(v_1, v_2) = (0.002, 0.008)$
$\varepsilon_\sigma = 0.01$
Procedimiento B
$(0, 0.002, 0.093, 0.110, 0.001, 0, 0.002, 0.111, 0.001, 0.128, 0.001, 0.126, 0, 0.001, 0.076, 0.112, 0.088, 0.145, 0.001, 0.001, 0)$
$(0, 0.001, 0.102, 0.107, 0.001, 0, 0.0, 0.154, 0.001, 0.099, 0, 0.055, 0.001, 0, 0.156, 0.113, 0.140, 0.069, 0.002, 0.001, 0)$
$(v_1, v_2) = (0.004, 0.007)$
$\varepsilon_\sigma = 0.011$
Procedimiento C
$(0, 0, 0.119, 0.106, 0, 0, 0, 0.110, 0, 0.107, 0, 0.108, 0, 0, 0.122, 0.122, 0.117, 0.1, 0, 0, 0)$
$(0, 0, 0.148, 0.096, 0, 0, 0, 0.099, 0, 0.095, 0, 0.093, 0, 0, 0.155, 0.126, 0.117, 0.070, 0, 0, 0)$
$(v_1, v_2) = (0.004, 0.005)$
$\varepsilon_\sigma = 0.009$

Tabla B.7: Estrategias obtenidas del juego Coronel Blotto

Los resultados obtenidos relacionados al tiempo y número de iteraciones de cada procedimiento son mostrados en la Tabla B.8.

A			B			C		
940.377	197127165	4.77×10^{-06}	90.239	75420	1.20×10^{-03}	0.047	13559	3.50×10^{-06}
532.020	109697363	4.85×10^{-06}	74.886	62704	1.19×10^{-03}	0.192	56383	3.41×10^{-06}
396.583	82924728	4.78×10^{-06}	56.735	47416	1.20×10^{-03}	0.046	13664	3.39×10^{-06}
362.203	80521418	4.50×10^{-06}	41.290	34596	1.19×10^{-03}	0.162	47742	3.40×10^{-06}
967.890	207963652	4.65×10^{-06}	69.359	58123	1.19×10^{-03}	0.090	26547	3.40×10^{-06}
1016.540	245737655	4.14×10^{-06}	64.457	53560	1.20×10^{-03}	0.118	34715	3.41×10^{-06}
553.971	112170109	4.94×10^{-06}	80.789	67624	1.19×10^{-03}	0.261	76657	3.40×10^{-06}
966.339	204832370	4.72×10^{-06}	138.294	115846	1.19×10^{-03}	0.358	105149	3.40×10^{-06}
1787.020	384044065	4.65×10^{-06}	84.924	70978	1.20×10^{-03}	0.121	35434	3.42×10^{-06}
1232.380	277204528	4.45×10^{-06}	92.610	77517	1.19×10^{-03}	0.260	76285	3.41×10^{-06}
875.533	190222305.3	4.60×10^{-06}	79.358	66378.4	1.20×10^{-03}	0.166	48613.5	3.41×10^{-06}

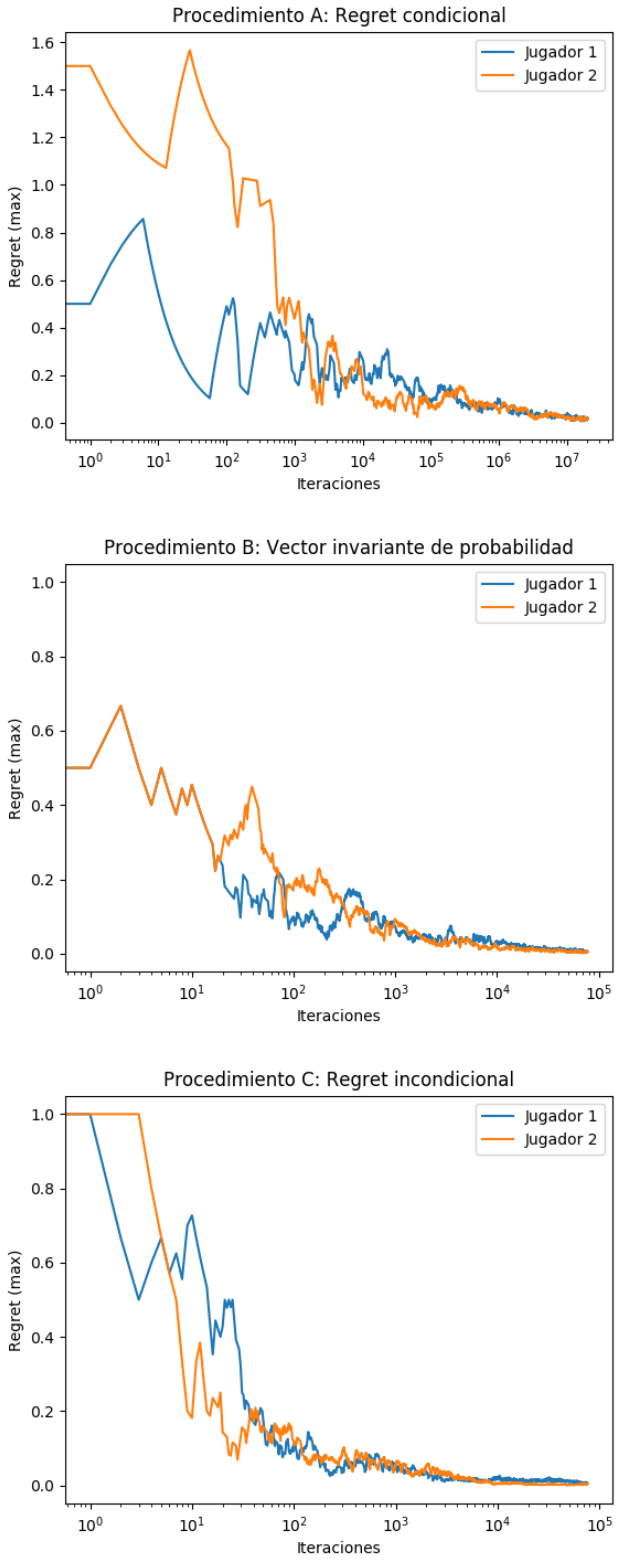
Tabla B.8: Resultados del juego Coronel Blotto

El procedimiento A, regret condicional, tuvo una duración promedio de 875.533 segundos, con un número promedio de iteraciones de 190222305.3, obteniendo un promedio de

4.60×10^{-6} segundos por iteración. Con el procedimiento B, que utiliza un vector invariante de probabilidad, se obtuvo un tiempo, número de iteraciones y tiempo por iteración promedios de 79.358 segundos, 66378.4 iteraciones y 1.2×10^{-3} segundos por iteración, respectivamente. Por último, el procedimiento C, regret incondicional, se obtuvo un tiempo promedio de 0.166, el número de iteraciones promedio fue de 48613.5, obteniendo un promedio de 3.41×10^{-6} segundos por iteración.

La Figura B.4 muestra el regret incondicional con respecto al tiempo de la última corrida para los tres procedimientos. Se observa que el regret máximo tiende a cero para cada uno de los jugadores en todos los procedimientos.

Figura B.4: Gráficas del regret con respecto al número de iteraciones del juego Coronel Blotto



APÉNDICE C

TEOREMA DE APROXIMACIÓN DE BLACKWELL

Los procedimientos que calculan equilibrios correlacionados se basan en el método de aproximación de Blackwell [2].

El marco teórico en el cual se aplica el teorema está conformado por: (1) un **decididor** i que toma decisiones de un conjunto finito de acciones S_i, (2) un **oponente** $-i$ que toma decisiones de un conjunto finito de acciones S_{-i}, (3) un **conjunto indexado** denotado por L, y (4) un **vector de pagos** $v(s_i, s_{-i}) \in \mathbb{R}^{|L|}$. El decididor y oponente toman decisiones $s_t = (s_i^t, s_{-i}^t) \in S_i \times S_{-i}$ indexadas en tiempo $t \geq 1$. El problema planteado consiste en ver si el decididor puede garantizar que el promedio de pagos D_t a tiempo t, definido por

$$D_t \;=\; \frac{1}{t}\sum_{\tau=1}^{t} v(s_\tau) \;=\; \frac{1}{t}\sum_{\tau=1}^{t} v(s_i^\tau, s_{-i}^\tau) \tag{C.1}$$

alcanza el conjunto $\mathbb{R}^{|L|}$. Antes de enunciar el teorema es necesario presentar las definiciones de distancia de un punto a un conjunto (Definición C.1), un conjunto alcanzable (Definición C.2), y de función de soporte (Definición C.3).

Definición C.1. *Sea A un conjunto cerrado y convexo en $\mathbb{R}^n$, y $x \in \mathbb{R}^n$ un punto cualquiera. La **distancia** de x a A es definida por*

$$dist(x, A) \;=\; \text{mín}\{\|x - a\| : a \in A\} \tag{C.2}$$

donde $\|\cdot\|$ denota la distancia euclidiana en $\mathbb{R}^n$.

Definición C.2. *Sea $\mathcal{C}$ un conjunto convexo y cerrado en $\mathbb{R}^{|L|}$. El conjunto $\mathcal{C}$ es **alcanzable** por el decididor i si hay un procedimiento para i que garantiza que D_t alcanza a $\mathcal{C}$; es decir. $dist(D_t, \mathcal{C}) \to 0$ (a.s.) sin importar la elección del oponente $-i$.*

Definición C.3. *Sea $\mathcal{C} \in \mathbb{R}^n$ un conjunto. La **función de soporte** $w_{\mathcal{C}}$ para el conjunto*

$\mathcal{C}$, es definida por

$$w_{\mathcal{C}}(\lambda) \;=\; \sup\{\lambda \cdot c : c \in \mathcal{C}\} \tag{C.3}$$

donde $\cdot$ denota el producto interno en $\mathbb{R}^n$.

Dado un conjunto convexo y cerrado $\mathcal{C}$ denotaremos con $F(x)$ el punto (único) más cercano a x de C, y con $\lambda(x) = x - F(x)$. El Teorema de Aproximación de Blackwell establece una condición necesaria y suficiente para el problema planteado previamente.

Teorema C.4 (Aproximación de Blackwell)**.** *Sea $\mathcal{C} \subseteq \mathbb{R}^{|L|}$ un conjunto convexo y cerrado con función de soporte $w_{\mathcal{C}}$. Entonces, $\mathcal{C}$ es alcanzable por i si y sólo si para todo $\lambda \in \mathbb{R}^{|L|}$, existe una estrategia mixta $q_\lambda \in \Delta(S_i)$ para el decididor i tal que para todo $s_{-i} \in S_{-i}$:*

$$\lambda \cdot v(q_\lambda, s_{-i}) \;\leq\; w_{\mathcal{C}}(\lambda)\,. \tag{C.4}$$

En esta expresión, $v(q, s_{-i})$ denota $\sum_{s_i \in S_i} q(s_i) u_i(s_i, s_{-i})$. Además, el siguiente procedimiento garantiza que $dist(D_t, \mathcal{C}) \to 0$ (a.s.) cuando $t \to \infty$: en el tiempo $t+1$, jugar $q_{\lambda(D_t)}$ si $D_t \notin \mathcal{C}$, y jugar arbitrariamente si $D_t \in \mathcal{C}$.

***** Hasta aquí tenemos la descripción teórica del modelo y soluciones. Ahora vienen algoritmos. Comenzar un nueva capítulo. *****