



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 Simn Bolvar  
como requisito parcial para optar al título de  
Ingeniero de Computación

Sartenejas, Enero de 2020

R. ROJAS LINÁREZ  
2020

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 Simn Bolvar  
como requisito parcial para optar al ttulo de  
Ingeniero de Computación

**Sartenejas, Enero de 2020**

Página reservada para el acta de evaluación

## DEDICATORIA

Dedicado a mis tutores, jurados y todos los que quieran leer (aunque sea parcialmente) este libro. Espero que lo disfruten tanto como yo lo hice escribiéndolo.

## RESUMEN

La teoría de juegos se encarga de estudiar la toma de decisiones estratégicas de individuos racionales en estructuras denominadas juegos. Los juegos no deterministas con información incompleta son aquellos que tienen azar y hay información oculta para los jugadores, como por ejemplo el juego de póker. En este trabajo de grado se estudian los diferentes modelos para este tipo de juegos, los conceptos de solución y los algoritmos para resolverlos. Se limita la investigación a juegos de dos jugadores con suma cero, para los cuales algoritmos como *Regret Matching* y *Conterfactual Regret Minimization* (CFR) permiten calcular un equilibrio de Nash, el principal concepto de solución utilizado. Estos algoritmos fueron implementados para diferentes juegos captados por el modelo. Los juegos en forma normal presentados fueron: piedra, papel o tijera, *matching pennies*, ficha vs. dominó y coronel Blotto, para los cuales se calculó una aproximación a un equilibrio de Nash mediante el algoritmo de *Regret Matching*. Los juegos en forma extensiva estudiados fueron: *One Card Póker* (OCP), dudo (un juego de dados), y dominó para dos personas. Estos juegos fueron parametrizados según el número de cartas, dados o piezas, entre otros elementos; obteniendo múltiples instancias para cada uno de ellos. Para cada instancia se calculó una aproximación a un equilibrio de Nash y se midió el error mediante la explotabilidad. Un juego fue considerado resuelto si la explotabilidad de la estrategia obtenida fue menor que un límite establecido. En el juego OCP fue posible resolver todas las instancias planteadas (utilizando hasta 5.000 cartas). En el juego dudo se resolvieron instancias con dados de hasta 4 caras y 2 dados por jugador. Por último, la instancia más grande resuelta en el juego de dominó incluye fichas que tienen hasta 3 puntos por lado y una distribución inicial de 3 fichas por jugador.

**Palabras claves:** juegos, forma normal, forma extensiva, no determinismo, información incompleta, estrategias.

# ÍNDICE GENERAL

|  |           |
|--|-----------|
| DEDICATORIA  | iii       |
| RESUMEN  | iv        |
| ÍNDICE GENERAL   | v         |
| ÍNDICE DE FIGURAS  | vii       |
| ÍNDICE DE TABLAS   | ix        |
| LISTA DE ACRÓNIMOS   | xi        |
| NOTACIÓN MATEMÁTICA  | xii       |
| INTRODUCCIÓN   | 1         |
| <b>CAPÍTULO I: JUEGOS EN FORMA NORMAL O ESTRATÉGICA</b>                              | <b>4</b>  |
| 1.1. Perfiles Estratégicos, Estrategias Mixtas, y Perfiles Estratégicos Mixtos . . . | 4         |
| 1.2. Ganancia Esperada y Mejor Respuesta . . . . .                                   | 7         |
| 1.3. Equilibrio de Nash . . . . .  | 9         |
| 1.4. Equilibrio Correlacionado . . . . .   | 10        |
| <b>CAPÍTULO II: JUEGOS EN FORMA EXTENSIVA</b>  | <b>13</b> |
| 2.1. Estrategias Puras y Mixtas para Juegos en Forma Extensiva . . . . .             | 17        |
| 2.2. Forma Normal vs. Forma Extensiva . . . . .                                      | 20        |
| 2.3. Estrategias de Comportamiento . . . . .   | 21        |
| 2.4. Perfect Recall . . . . .  | 24        |

|   |           |
|---|-----------|
| <b>CAPÍTULO III: EXPLOTABILIDAD</b>                                   | <b>31</b> |
| 3.1. Juegos de Dos Jugadores de Suma Cero . . . . .                   | 31        |
| 3.2. Explotabilidad . . . . .   | 32        |
| <b>CAPÍTULO IV: REGRET MATCHING</b>                                   | <b>34</b> |
| 4.1. Regret Matching y Equilibrio de Nash . . . . .                   | 37        |
| 4.2. Evaluación Empírica de Regret Matching . . . . .                 | 38        |
| 4.3. Análisis de Experimentos . . . . .                               | 45        |
| 4.3.1. Complejidad de Cada Iteración . . . . .                        | 45        |
| 4.3.2. Número de Iteraciones . . . . .                                | 46        |
| 4.3.3. Tiempo Transcurrido . . . . .                                  | 46        |
| <b>CAPÍTULO V: COUNTERFACTUAL REGRET MINIMIZATION</b>                 | <b>47</b> |
| 5.1. Descomposición del Regret . . . . .                              | 47        |
| 5.2. Regret Minimization . . . . .                                    | 49        |
| 5.3. Counterfactual Regret Minimization (CFR) . . . . .               | 50        |
| 5.4. Monte Carlo Counterfactual Regret Minimization (MCCFR) . . . . . | 51        |
| 5.5. Detalles de Implementación y Ejecución . . . . .                 | 52        |
| <b>CONCLUSIONES Y RECOMENDACIONES</b>                                 | <b>59</b> |
| <b>REFERENCIAS</b>  | <b>61</b> |
| <b>APÉNDICE A: PRUEBAS</b>  | <b>63</b> |
| <b>APÉNDICE B: TEOREMA DE APROXIMACIÓN DE BLACKWELL</b>               | <b>80</b> |
| <b>APÉNDICE C: ESTRATEGIAS MINIMAX Y MAXIMIN</b>                      | <b>82</b> |
| <b>APÉNDICE D: FORMA NORMAL Y PROGRAMACIÓN LINEAL</b>                 | <b>84</b> |
| <b>APÉNDICE E: ALGORITMOS</b>   | <b>88</b> |
| <b>APÉNDICE F: REGRET MATCHING</b>                                    | <b>93</b> |
| <b>APÉNDICE G: COUNTERFACTUAL REGRET MINIMIZATION</b>                 | <b>99</b> |



## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| 2.1. Árbol del juego en forma extensiva del Ejemplo 2.1. Los nodos del primer jugador son representados con círculos sin relleno, los nodos del segundo jugador con círculos con relleno negro y los nodos terminales con cuadrados con relleno negro. . . . .   | 14 |
| 2.2. Árbol del juego en forma extensiva presentado en el Ejemplo 2.2. Los nodos que pertenecen al mismo conjunto de información son unidos por líneas punteadas. . . . .   | 15 |
| 2.3. Árbol completo del juego Kuhn Póker. Los nodos unidos con líneas punteadas o con el mismo diseño y color pertenecen a el mismo conjunto de información. En cada nodo de decisión la acción <i>pasar</i> es representada con una línea discontinua y la acción <i>apostar</i> es representada con una línea doble. . . . . | 17 |
| 2.4. Árbol de la forma extensiva del juego piedra, papel o tijera. . . . .   | 21 |
| 2.5. Árbol correspondiente a la forma normal de la Tabla 2.4. . . . .  | 21 |
| 2.6. Equilibrios de Nash para el juego de Kuhn Poker. Las etiquetas sobre las aristas del árbol indican la probabilidad de escoger dicha acción en cada nodo para un parámetro $\alpha \in [0, \frac{1}{3}]$ . . . . .   | 25 |
| 2.7. Árbol de la forma extensiva del juego con <i>imperfect recall</i> presentado en el Ejemplo 2.11. P representa la acción de parar el juego y C de continuarlo. M representa la acción de mantener las cartas obtenidas inicialmente e I representan la acción de intercambiarlas. . . . .                                  | 26 |
| 2.8. Árbol de la forma extensiva del juego con <i>imperfect recall</i> presentado en el Ejemplo 2.15. Observe que existe una historia que pasa dos veces sobre el mismo conjunto de información. . . . .   | 28 |
| 4.1. Gráficas del <i>regret</i> con respecto al número de iteraciones del juego matching pennies. . . . .  | 40 |
| 4.2. Gráficas del <i>regret</i> con respecto al número de iteraciones del juego piedra, papel o tijera. . . . .  | 41 |
| 4.3. Posibles posiciones de la ficha del segundo jugador en el juego ficha vs. dominó. . . . .   | 42 |

|      |  |     |
|------|--|-----|
| 4.4. | Posibles posiciones de la ficha de dominó que representas las acciones del primer jugador en el juego ficha vs. dominó. . . . .                  | 42  |
| 4.5. | Gráficas del <i>regret</i> con respecto al número de iteraciones del juego ficha vs. dominó. . . . .   | 44  |
| 4.6. | Gráficas del <i>regret</i> con respecto al número de iteraciones del juego Coronel Blotto. . . . .   | 45  |
| 5.1. | Gráficas del <i>regret</i> con respecto al número de iteraciones de los juegos <i>One Card Poker</i> (200) y <i>One Card Poker</i> (3) . . . . . | 54  |
| 5.2. | Juego dudo. Los vasos se utilizan para lanzar los dados y evitar que los oponentes vean el resultado. . . . .                                    | 55  |
| 5.3. | Gráficas del <i>regret</i> con respecto al número de iteraciones de los juego Dudo (6, 1, 1) y Dudo (5, 2, 2). . . . .                           | 57  |
| 5.4. | Gráficas del <i>regret</i> con respecto al número de iteraciones de los juegos Dominó (3, 3) y Dominó (3, 4). . . . .                            | 58  |
| G.1. | Gráficas del regret con respecto al número de iteraciones de los juegos OCP(3) y OCP(12) . . . . .   | 99  |
| G.2. | Gráficas del regret con respecto al número de iteraciones de los juegos OCP(50), OCP(200), OCP(1000) y OCP(5000) . . . . .                       | 100 |
| G.3. | Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(3,1,1), Dudo(3,1,2), Dudo(3,2,1) y Dudo(3,2,2) . . . . .            | 101 |
| G.4. | Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(4,1,1), Dudo(4,1,2), Dudo(4,2,1) y Dudo(4,2,2) . . . . .            | 102 |
| G.5. | Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(5,1,1), Dudo(5,1,2), Dudo(5,2,1) y Dudo(5,2,2) . . . . .            | 103 |
| G.6. | Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(6,1,1), Dudo(6,1,2) y Dudo(6,2,1) . . . . .                         | 104 |
| G.7. | Gráficas del regret con respecto al número de iteraciones de los juegos Domino(2,2), Domino(3,2), Domino(3,3) y Domino(3,4) . . . . .            | 105 |

## ÍNDICE DE TABLAS

|      |  |    |
|------|--|----|
| 1.1. | Tabla de pagos de la forma normal del juego piedra, papel o tijera. . . . .  | 5  |
| 1.2. | Tabla de pagos del juego “batalla de los sexos”. . . . .   | 11 |
| 2.1. | Resumen de las posibles secuencias del juego Kuhn Póker . . . . .  | 16 |
| 2.2. | Estrategias puras para el juego con información incompleta presentado en el Ejemplo 2.2. . . . .   | 18 |
| 2.3. | Ejemplo de una estrategia pura para el jugador 2 en el juego Kuhn Poker. . . . .   | 19 |
| 2.4. | Forma normal de un juego en forma extensiva . . . . .  | 20 |
| 2.5. | Equilibrio de Nash para el juego de Kuhn Poker. Cada fila de la tabla corresponde con uno o varios conjuntos de información que se denotan con enteros (enumerados utilizando un procedimiento de búsqueda en profundidad sobre el árbol del juego). El equilibrio de Nash corresponde con una distribución aleatoria sobre las acciones pasar y apostar la cuál depende de un parámetro $\alpha \in [0, \frac{1}{3}]$ . . . . . | 24 |
| 2.6. | Tabla de la forma normal para un juego con <i>imperfect recall</i> . . . . .   | 29 |
| 2.7. | Probabilidades de las Estrategias Puras . . . . .  | 30 |
| 4.1. | Tabla de pagos del juego matching pennies . . . . .  | 39 |
| 4.2. | Resultados experimentales del juego matching pennies. . . . .  | 40 |
| 4.3. | Resultados experimentales del juego piedra, papel o tijera. . . . .  | 41 |
| 4.4. | Matriz de pagos del juego ficha vs. dominó. . . . .  | 43 |
| 4.5. | Resultados Experimentales del juego ficha vs. dominó. . . . .  | 43 |
| 4.6. | Resultados Experimentales del juego coronel Blotto. . . . .  | 44 |
| 4.7. | Complejidad por iteración de cada uno de los procedimientos. . . . .   | 46 |
| 5.1. | Resultados del algoritmo CFR en el juego OCP( $N$ ) con diferentes números de cartas $N$ . . . . .   | 53 |
| 5.2. | Resultados del algoritmo CFR en el juego dudo. . . . .   | 56 |
| 5.3. | Resultados del algoritmo CFR en el juego dominó. . . . .   | 58 |

|  |    |
|--|----|
| C.1. Tabla de pagos del juego del Ejemplo C.2. . . . .                     | 83 |
| F.1. Estrategias obtenidas en el juego <i>matching pennies</i> . . . . .   | 94 |
| F.2. Resultados experimentales del juego <i>matching pennies</i> . . . . . | 94 |
| F.3. Estrategias obtenidas del juego piedra, papel o tijera. . . . .       | 95 |
| F.4. Resultados experimentales del juego piedra, papel o tijera. . . . .   | 95 |
| F.5. Estrategias obtenidas del juego ficha vs dominó. . . . .              | 96 |
| F.6. Resultados experimentales del juego ficha vs. dominó. . . . .         | 97 |
| F.7. Estrategias obtenidas del juego coronel Blotto. . . . .               | 97 |
| F.8. Resultados experimentales del juego coronel Blotto. . . . .           | 98 |

## LISTA DE ACRÓNIMOS

**USB** Universidad Simón Bolívar

**EN** Equilibrio de Nash

**RM** Regret Matching

**CFR** Counterfactual Regret Minimization

**MCCFR** Monte Carlo Counterfactual Regret Minimization

**GEBR** Generalized Expectimax Best Response

**OCP** One Card Poker

**DFS** Depth First Search

**RPS** Rock Paper Scissors

**AWS** Amazon Web Services

**EC2** Elastic Compute Cloud

## NOTACIÓN MATEMÁTICA

|                            |  |
|----------------------------|--|
| $S_i$                      | Conjunto de estrategias puras del jugador $i$ .  |
| $s$                        | Perfil estratégico puro.   |
| $s_i$                      | Estrategia pura del jugador $i$ .  |
| $s_{-i}$                   | Perfil estratégico $s$ excluyendo la estrategia del jugador $i$ .  |
| $ A $                      | Cardinalidad (número de elementos) del conjunto $A$ .  |
| $\Delta(A)$                | Conjunto de distribuciones de probabilidad del conjunto $A$ .  |
| $\Delta^n$                 | Simplex $n$ -dimensional.  |
| $\sigma_i$                 | Estrategia mixta del jugador $i$ .   |
| $\sigma_i(s_i)$            | Probabilidad de elegir $s_i$ dada la estrategia mixta $\sigma_i$ .   |
| $\sigma$                   | Perfil estratégico mixto.  |
| $\sigma(s)$                | Probabilidad de que se elija el perfil estratégico $s$ bajo $\sigma$ .   |
| $u_i$                      | Función de pago (utilidad) del jugador $i$ .   |
| $u_i(\sigma)$              | Ganancia esperada del jugador $i$ dado $\sigma$ .  |
| $h \sqsubset h'$           | La historia (secuencia) $h$ es prefijo de la historia $h'$ .   |
| $\varepsilon_\sigma$       | Explotabilidad del perfil estratégico mixto $\sigma$ .   |
| $\sigma_{I \rightarrow a}$ | Perfil estratégico idéntico a $\sigma$ pero en el conjunto de información $I$ siempre es elegida la acción $a$ . |
| $R_i^T$                    | <i>Regret</i> promedio general del jugador $i$ a tiempo $T$ .  |
| $\bar{\sigma}_i^T$         | Estrategia promedio del tiempo 1 al tiempo $T$ del jugador $i$ .   |
| $u_i(\sigma, I)$           | Utilidad contrafactual del jugador $i$ , dado el conjunto de información $I$ y el perfil estratégico $\sigma$ .  |
| $R_{i,imm}^T(I)$           | <i>Regret</i> contrafactual inmediato del conjunto de información $I$ .  |

|                     |  |
|---------------------|--|
| $\pi^\sigma(h)$     | Probabilidad de alcanzar la historia $h$ dado el perfil estratégico $\sigma$ .   |
| $\pi^{\sigma_i}(h)$ | Probabilidad de alcanzar la historia $h$ dado que todos los jugadores juegan para alcanzar $h$ (incluyendo los nodos de azar), con excepción del jugador $i$ que utiliza $\sigma$ .  |
| $\pi^c(h)$          | Probabilidad de alcanzar la historia $h$ dado que todos los jugadores juegan para alcanzar $h$ .   |
| $\pi^{\sigma-i}(h)$ | Probabilidad de alcanzar la historia $h$ dado que todos los jugadores utilizan el perfil estratégico $\sigma$ , menos el jugador $i$ que juega para alcanzar $h$ . Incluye también las probabilidades de los nodo de azar                |
| $\pi^\sigma(I)$     | Probabilidad de alcanzar el conjunto de información $I$ dado el perfil estratégico $\sigma$ .  |
| $\pi^{\sigma_i}(I)$ | Probabilidad de alcanzar el conjunto de información $I$ dado que todos los jugadores juegan para alcanzar $h$ (incluyendo los nodos de azar), con excepción del jugador $i$ que utiliza $\sigma$ .                                       |
| $\pi^c(I)$          | Probabilidad de alcanzar el conjunto de información $I$ dado que todos los jugadores juegan para alcanzar $h$ .  |
| $\pi^{\sigma-i}(I)$ | Probabilidad de alcanzar el conjunto de información $I$ dado que todos los jugadores utilizan el perfil estratégico $\sigma$ , menos el jugador $i$ que juega para alcanzar $h$ . Incluye también las probabilidades de los nodo de azar |
| $\pi^\sigma(h, h')$ | Probabilidad de que ocurra la historia $h'$ , dado que ocurrió la historia $h$ .   |

# INTRODUCCIÓN

La teoría de juegos puede ser definida como el estudio de modelos matemáticos de conflicto y cooperación entre agentes que deben tomar decisiones de forma racional e inteligente [1, p. 1]; estos modelos se denominarán **juegos**. Esta disciplina tiene aplicaciones en diversas áreas, incluyendo ciencias sociales, economía, matemática y ciencias de la computación.

Uno de los principales pioneros de esta disciplina fue John von Neumann, con su publicación *Zur Theorie der Gesellschaftsspiele* (Sobre la Teoría de Juegos) en el año 1928 [2]. Asimismo, John Forbes Nash Jr. con su publicación *Non-cooperative Games* (Juegos no Cooperativos) en el año 1951 [3], introduce importantes conceptos, entre los cuales se encuentra el concepto de solución que hoy en día se conoce como equilibrio de Nash.

Aunque hay diferentes tipos de juegos, este trabajo se enfoca en juegos no deterministas con información incompleta. Con no determinismo se hace referencia a que los juegos incluyen incertidumbre probabilística, esta incertidumbre puede ocurrir, por ejemplo, al lanzar una moneda, repartir cartas de forma aleatoria o lanzar dados. Por otra parte, un juego con información incompleta permite modelar situaciones donde los jugadores tienen información parcial sobre algunas de las acciones que ya han sido tomadas [4, p. 199].

El juego de póker (con sus diferentes versiones) es uno de los juegos más estudiados en esta categoría. Note que es un juego no determinista ya que se reparten cartas de forma aleatoria al inicio del mismo. Por otra parte, cada jugador desconoce las cartas que poseen los demás jugadores, por lo que poseen información parcial de la distribución inicial de las cartas. En contraste, juegos como el ajedrez, las damas o *go*, son todos juegos deterministas (no hay elementos de azar) y además con información completa, pues todos los jugadores saben lo que ha ocurrido durante el juego y no hay información oculta entre ellos.

Uno de los retos para esta categoría de modelos consiste en determinar qué significa que un juego sea resuelto o que un jugador juegue de forma óptima. Para esto es necesario introducir el concepto de estrategias, las cuales indican las acciones o planes de acción que tomarán los jugadores en un momento determinado [5, p. 24]. Luego, resolver un juego



puede tener diferentes significados acorde al concepto de solución que se utilice, siendo el equilibrio de Nash uno de los más importantes y el utilizado en el presente trabajo. Es importante destacar que en un equilibrio de Nash las acciones de los jugadores no son necesariamente deterministas, es decir, un jugador puede tomar decisiones diferentes ante el mismo escenario.

Por otra parte, Hart y Mas-Colell (2000) introducen el concepto de *regret matching* [6], en el cual los jugadores alcanzan un equilibrio teniendo en cuenta el “arrepentimiento” de sus jugadas previas, el cual se mide con una métrica denominada *regret*, y haciendo las futuras jugadas proporcionales al *regret* positivo. Este concepto es la base para el algoritmo *Counterfactual Regret Minimization* (CFR), propuesto por Zinkevich, Johanson, Bowling y Piccione (2007) que permite encontrar una aproximación del equilibrio de Nash en cierto tipo de juegos con información incompleta, que sean de dos jugadores con suma cero [7].

Dentro de este contexto, el objetivo de este proyecto de grado es comprender los conceptos en el área de juegos de dos personas que involucran información incompleta y no determinismo, así como implementar los algoritmos de *Regret Matching* y CFR, realizando experimentos sobre distintos juegos que son capturados por el modelo. Con el fin de alcanzar el objetivo general se proponen los siguientes objetivos específicos:

- Comprender los diferentes modelos de juegos y los elementos que los componen. Incluyendo juegos en forma normal y juegos en forma extensiva.
- Comprender los diferentes conceptos de solución para el tipo de juegos, como equilibrio correlacionado y equilibrio de Nash.
- Comprender los resultados teóricos más relevantes, y sus demostraciones, en relación a los modelos de juegos estudiados y los algoritmos implementados.
- Implementar los algoritmos *Regret Matching* y *Counterfactual Regret Minimization* que permiten encontrar equilibrios de Nash para el tipo de juego planteado.
- Implementar una clase general que permita representar los juegos que se quieren estudiar (independientemente de las reglas específicas de cada juego), así como diferentes juegos concretos que sean capturados por el modelo.
- Realizar experimentos sobre los juegos propuestos utilizando los algoritmos implementados.
- Evaluar las estrategias obtenidas en cada uno de los juegos implementados.

Este libro se estructura en 5 capítulos. El Capítulo I contiene el marco teórico de los juegos en forma normal o estratégica. Se presenta una definición formal de este tipo de juegos y los elementos que los componen. También se presentan dos conceptos de solución importantes: equilibrio de Nash y equilibrio correlacionado. El Capítulo II contiene el marco teórico de los juegos en forma extensiva, se presentan los elementos en este tipo de juegos y se comparan con los juegos en forma normal. Además, se introduce una clasificación dentro de este tipo de juegos: juegos con *perfect recall* o con *imperfect recall*. Ambos capítulos contienen diversos ejemplos que ilustran los conceptos introducidos.

El Capítulo III presenta las propiedades que tienen los juegos de dos jugadores de suma cero, y explica por qué el equilibrio de Nash es importante en este tipo de juegos. Además, introduce dos nuevos conceptos de solución: estrategias *minimax* y *maximin*. Por último, se explica el concepto de explotabilidad que es la métrica que se utiliza para evaluar las estrategias obtenidas de forma experimental en los juegos.

El Capítulo IV presenta tres procedimientos que utilizan *Regret Matching*, los cuales conducen a un equilibrio de Nash cuando los juegos son de dos jugadores de suma cero. Además, se presentan 4 juegos en forma normal y los resultados experimentales que se obtienen al aplicar los procedimientos sobre ellos. El Capítulo V presenta el algoritmo CFR y una familia de este tipo de algoritmo, denominada *Monte Carlo CFR* (MCCFR). Este capítulo también incluye 3 clases de juegos en forma extensiva y los resultados obtenidos al aplicar una versión de MCCFR sobre ellos. Finalmente, se presentan las conclusiones y las recomendaciones de este proyecto para las investigaciones futuras en el área.

La implementación de los procedimientos de *Regret Matching*, junto con los resultados experimentales reportados en esta tesis se encuentran de forma pública <https://github.com/rubmary/regret-matching>. Similarmente, la implementación de los juegos y el algoritmo CFR, junto a las estrategias obtenidas y resultados experimentales se encuentran en <https://github.com/rubmary/cfr>.

Adicionalmente se desarrolló una aplicación web que permite observar la estrategia obtenida, la implementación se encuentra en <https://github.com/rubmary/domino-app>. Es importante destacar y agradecer la contribución de Samuel Nacache para el desarrollo de la interfaz gráfica de la aplicación.

# CAPÍTULO I

## JUEGOS EN FORMA NORMAL O ESTRATÉGICA

En un juego en forma normal cada uno de los jugadores eligen una única “acción” (que puede representar una estrategia completa para un juego complejo) de forma simultánea, y cada jugador obtiene un pago de acuerdo a las acciones realizadas por todos los jugadores. Frecuentemente, estos juegos también se llaman *one-shot game* (juegos de un sólo disparo) ya que cada uno de los jugadores realiza una única acción [8]. El ejemplo clásico es el juego piedra, papel o tijera (RPS por sus siglas en inglés). En este juego cada uno de los dos jugadores elige una de 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 ambos jugadores eligen la misma opción, entonces es un empate.

**Definición 1.1** ([6]). *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:*

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

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

A continuación se presentan 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 simultánea, y los perfiles estratégicos mixtos agregan las estrategias mixtas [9].

**Definición 1.2.** Un *perfil estratégico* (o *perfil de acción*) es una  $N$ -tupla formada por una estrategia pura para cada jugador.  $S = \prod_{i \in N} S_i$  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 se descompone una estrategia  $s$  en 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. Además, 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 para cada jugador) y éstas contienen un vector de pagos para cada jugador [8].

Piedra, papel o tijera es un juego para dos jugadores y las acciones (o estrategias puras) son las mismas para cada jugador:  $S_1 = S_2 = \{\mathcal{R}, \mathcal{P}, \mathcal{S}\}$  donde  $\mathcal{R}$  es piedra,  $\mathcal{P}$  es papel, y  $\mathcal{S}$  es tijera. La Tabla 1.1 es la tabla de pagos correspondiente a este juego, en el cual  $N = 2$ , por lo que la tabla es de 2 dimensiones. Las filas representan las acciones del jugador 1 y las columnas las del jugador 2, además cada celda corresponde a un perfil estratégico. Por ejemplo, si el primer jugador elige tijera y el segundo jugador elige papel, la estrategia pura es  $s = (\mathcal{S}, \mathcal{P})$  y la celda correspondiente es la que se encuentra en la posición  $(3, 2)$  (tercera fila y segunda columna). Esta celda contiene el vector  $(1, -1)$  ya que en este caso el primer jugador gana obteniendo una utilidad de 1, mientras que el segundo jugador pierde obteniendo una utilidad de  $-1$ .

Tabla 1.1: Tabla de pagos de la forma normal del juego piedra, papel o tijera.

|                        | $\mathcal{R}$ (piedra) | $\mathcal{P}$ (papel) | $\mathcal{S}$ (tijera) |
|------------------------|------------------------|-----------------------|------------------------|
| $\mathcal{R}$ (piedra) | 0, 0                   | -1, 1                 | 1, -1                  |
| $\mathcal{P}$ (papel)  | 1, -1                  | 0, 0                  | -1, 1                  |
| $\mathcal{S}$ (tijera) | -1, 1                  | 1, -1                 | 0, 0                   |

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  $\Delta(A) = \{(x_a)_{a \in A} : \sum_{a \in A} x_a = 1, x_a \geq 0\}$ . También se denotará a  $\Delta^n$ , como el simplex  $n$ -dimensional, i.e.,  $\Delta^n = \{x \in \mathbb{R}^{n+1} : \sum_{0 \leq i \leq n} x_i = 1, x_i \geq 0\}$ .

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

$$\text{support}(\sigma_i) = \{s_i : \sigma_i(s_i) > 0\}. \quad (1.1)$$

**Definición 1.5.** Un *perfil estratégico mixto*  $\sigma$  consiste en una estrategia mixta para cada jugador; es decir,  $\sigma \in \prod_{i \in N} \Delta(S_i)$  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 de que el perfil estratégico mixto elija la estrategia mixta  $s$ ; i.e.,  $\sigma(s) = \prod_{i \in N} \sigma_i(s_i)$ . Para un jugador  $i$ , un perfil  $\sigma$  se descompone en un par  $(\sigma_i, \sigma_{-i})$  donde  $\sigma_i$  es una estrategia mixta del jugador  $i$  y  $\sigma_{-i}$  es un perfil 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 en  $\{j\}_{j \neq i}$  elijan las estrategias  $\{s_j\}_{j \neq i}$  especificadas en el perfil  $s_{-i}$ . Finalmente, si  $x$  es una estrategia pura para el jugador  $i$ , también se utiliza  $x$  para denotar la estrategia mixta  $\sigma_i$  para el jugador  $i$  que elige  $x$  con probabilidad 1 y elige las otras estrategias puras del jugador  $i$  con probabilidad 0; i.e.,  $\sigma_i(x) = 1$  y  $\sigma_i(x') = 0$  para toda  $x' \in S_i$  con  $x' \neq x$ .

En el juego piedra, papel o tijera una posible estrategia mixta para el jugador  $i$  es elegir piedra o papel con probabilidad  $\frac{1}{2}$  y nunca elegir tijera. Si se denota a dicha estrategia con  $\sigma_i$ , entonces  $\sigma_i(\mathcal{R}) = \sigma_i(\mathcal{P}) = \frac{1}{2}$  y  $\sigma_i(\mathcal{S}) = 0$ . Esta estrategia también puede ser representada por  $\sigma_i = (\frac{1}{2}, \frac{1}{2}, 0)$ , donde la primera componente representa la probabilidad del jugador de elegir piedra, la segunda la probabilidad de elegir papel y la última la de elegir tijera. Otra posible estrategia mixta  $\sigma'_i$  consiste en elegir piedra con probabilidad  $\frac{1}{3}$ , papel con probabilidad  $\frac{1}{2}$  y tijera con probabilidad  $\frac{1}{6}$ , i.e.  $\sigma'_i = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$ . Note que el soporte de  $\sigma_i$  es igual a  $\text{support}(\sigma_i) = \{\mathcal{R}, \mathcal{P}\}$  y el soporte de  $\sigma'_i$  es  $\text{support}(\sigma'_i) = S_i$ , pues en esta última estrategia todas las acciones tienen una probabilidad positiva de ser elegidas. Luego, si el jugador 1 decide utilizar la estrategia  $\sigma_i$  y el jugador 2 la estrategia  $\sigma'_i$ , entonces  $\sigma_1 = (\frac{1}{2}, \frac{1}{2}, 0)$ ,  $\sigma_2 = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$  y  $\sigma = (\sigma_1, \sigma_2)$  es un perfil estratégico mixto. Sea  $s = (\mathcal{P}, \mathcal{S})$  el perfil estratégico (puro), donde el primer jugador elige papel y el segundo jugador elige tijera, luego  $\sigma(s) = \sigma(\mathcal{P}) \cdot \sigma(\mathcal{S}) = \frac{1}{2} \cdot \frac{1}{6} = \frac{1}{12}$ . Por otra parte, si  $s' = (\mathcal{S}, \mathcal{P})$  entonces  $\sigma(s') = 0 \cdot \frac{1}{2} = 0$ .

## 1.2. Ganancia Esperada y Mejor Respuesta

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

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

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

La ganancia esperada del jugador  $i$  se puede descomponer como se muestra a continuación. (La demostración de este Teorema y otros contenidos en la tesis se encuentran en el Apéndice A).

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

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

Dado un perfil estratégico mixto  $\sigma$ , 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, se define el concepto de mejor respuesta para el jugador  $i$  dado un perfil  $\sigma_{-i}$  para los otros jugadores.

**Definición 1.8.** Sea  $i \in N$  un jugador,  $\sigma_i$  una estrategia mixta para el jugador  $i$ , y  $\sigma_{-i}$  un perfil estratégico mixto para el resto de los jugadores. Se dice que  $\sigma_i$  es una **mejor respuesta** con respecto a  $\sigma_{-i}$  si y sólo si  $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$  para toda estrategia mixta  $\sigma'_i$  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 se verá 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 [9].

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

En el juego piedra, papel o tijera, la ganancia esperada del jugador  $i$  cuando utiliza la estrategia  $\sigma = (\sigma_1, \sigma_2)$ , viene dada por:

$$\begin{aligned} u_i(\sigma) = & \sigma_i(\mathcal{R}) (\sigma_{-i}(\mathcal{R})u_i(\mathcal{R}, \mathcal{R}) + \sigma_{-i}(\mathcal{P})u_i(\mathcal{R}, \mathcal{P}) + \sigma_{-i}(\mathcal{S})u_i(\mathcal{R}, \mathcal{S})) \\ & + \sigma_i(\mathcal{P}) (\sigma_{-i}(\mathcal{R})u_i(\mathcal{P}, \mathcal{R}) + \sigma_{-i}(\mathcal{P})u_i(\mathcal{P}, \mathcal{P}) + \sigma_{-i}(\mathcal{S})u_i(\mathcal{P}, \mathcal{S})) \\ & + \sigma_i(\mathcal{S}) (\sigma_{-i}(\mathcal{R})u_i(\mathcal{S}, \mathcal{R}) + \sigma_{-i}(\mathcal{P})u_i(\mathcal{S}, \mathcal{P}) + \sigma_{-i}(\mathcal{S})u_i(\mathcal{S}, \mathcal{S})) . \end{aligned} \quad (1.4)$$

Al sustituir las utilidades de las estrategias puras y eliminar los términos nulos, se obtiene:

$$\begin{aligned} u_i(\sigma) = & \sigma_i(\mathcal{R}) (\sigma_{-i}(\mathcal{S}) - \sigma_{-i}(\mathcal{P})) \\ & + \sigma_i(\mathcal{P}) (\sigma_{-i}(\mathcal{R}) - \sigma_{-i}(\mathcal{S})) \\ & + \sigma_i(\mathcal{S}) (\sigma_{-i}(\mathcal{P}) - \sigma_{-i}(\mathcal{R})) . \end{aligned} \quad (1.5)$$

En particular, para la estrategia presentada previamente  $\sigma = (\sigma_1, \sigma_2)$ , con  $\sigma_1 = (\frac{1}{2}, \frac{1}{2}, 0)$  y  $\sigma_2 = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$ , las ganancias esperadas del primer y segundo jugador son iguales a:

$$u_1(\sigma) = \frac{1}{2} \left( \frac{1}{6} - \frac{1}{2} \right) + \frac{1}{2} \left( \frac{1}{3} - \frac{1}{6} \right) + 0 \left( \frac{1}{2} - \frac{1}{6} \right) = -\frac{1}{12} \quad (1.6)$$

$$u_2(\sigma) = \frac{1}{3} \left( 0 - \frac{1}{2} \right) + \frac{1}{2} \left( \frac{1}{2} - 0 \right) + \frac{1}{6} \left( \frac{1}{2} - \frac{1}{2} \right) = \frac{1}{12} . \quad (1.7)$$

Calculemos ahora las mejores respuesta a  $\sigma_1$  y  $\sigma_2$ . Supongamos que el jugador 1 utiliza  $\sigma_1$  y sea  $\sigma_2^* = (x, y, z)$  la mejor respuesta a  $\sigma_1$ . Entonces la ganancia esperada del jugador 2 viene dada por:

$$u_2(\sigma_1, \sigma_2^*) = x \left( 0 - \frac{1}{2} \right) + y \left( \frac{1}{2} - 0 \right) + z \left( \frac{1}{2} - \frac{1}{2} \right) \quad (1.8)$$

$$u_2(\sigma_1, \sigma_2^*) = -\frac{1}{2}x + \frac{1}{2}y \quad (1.9)$$

Luego, la mejor respuesta a  $\sigma_1$  se obtiene al maximizar  $f(x, y) = -\frac{1}{2}x + \frac{1}{2}y$ , con  $x + y + z = 1$  y  $x, y, z \geq 0$ . Es claro que la función se maximiza en dicho dominio cuando  $x = z = 0$  y  $y = 1$ . Luego  $\sigma_2^* = (0, 1, 0)$ ; i.e., la estrategia en el que el jugador 2 siempre

elige papel. En este caso existe una única mejor respuesta a  $\sigma_1 = (\frac{1}{2}, \frac{1}{2}, 0)$ , cuyo soporte tiene un único elemento:  $\text{support}(\sigma_2^*) = \{\mathcal{P}\}$ .

De forma similar se obtiene que, si  $\sigma_1^* = (x, y, z)$  es mejor respuesta a  $\sigma_2$ , entonces  $u_1(\sigma_1^*, \sigma_2) = -\frac{1}{3}x + \frac{1}{6}y + \frac{1}{6}z$ . Note que en este caso se maximiza la función cuando  $y = \alpha$  y  $z = 1 - \alpha$ , para cualquier  $\alpha \in [0, 1]$ . Luego, el jugador es indiferente ante las estrategias  $\mathcal{P}$  y  $\mathcal{S}$ , por lo que existen infinitas estrategias que son mejor respuesta a  $\sigma_1$ . Finalmente se obtiene que  $\sigma_1^*$  es mejor respuesta si y sólo si  $\sigma_1^* = (0, \alpha, 1 - \alpha)$  para cualquier  $\alpha \in [0, 1]$ , i.e.  $\sigma_1^*$  es mejor respuesta si y sólo si  $\text{support}(\sigma_1^*) \subseteq \{\mathcal{P}, \mathcal{S}\}$ .

### 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 tiene 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  $\sigma$  es un **equilibrio de Nash** si y sólo si para todo jugador  $i$ , la estrategia  $\sigma_i$  es mejor respuesta del jugador  $i$  para  $\sigma_{-i}$ .*

**Teorema 1.11** ([9]). *Todo juego finito tiene al menos un equilibrio de Nash.*

Es importante destacar que el Teorema 1.11 es cierto al considerar perfiles estratégicos mixtos, pero no al considerar únicamente perfiles estratégicos puros. No todos los juegos tienen algún perfil estratégico puro que sea equilibrio de Nash, como por ejemplo, el juego RPS. Para ver esto, supongamos que el primer jugador juega con alguna estrategia pura, digamos  $\mathcal{R}$ , luego, la única mejor respuesta a esta estrategia es jugar con la estrategia pura  $\mathcal{P}$ . Pero esto no es un equilibrio de Nash, pues ahora la mejor respuesta a la estrategia del segundo jugador es que el primer jugador juegue con la estrategia pura  $\mathcal{S}$ . Análogamente, cuando el primer o segundo jugador juegan con cualquier estrategia pura no se puede obtener un equilibrio de Nash.

En RPS existe un único equilibrio de Nash, que ocurre cuando  $\sigma_1^* = \sigma_2^* = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ . En efecto, note que si el jugador  $i$  utiliza  $\sigma_i^*$ , para cualquier estrategia  $\sigma_{-i}$ , la ganancia del



jugador  $-i$  viene dada por:

$$u_{-i}(\sigma_i^*, \sigma_{-i}) = \sigma_{-i}(\mathcal{R}) \left( \frac{1}{3} - \frac{1}{3} \right) + \sigma_{-i}(\mathcal{P}) \left( \frac{1}{3} - \frac{1}{3} \right) + \sigma_{-i}(\mathcal{S}) \left( \frac{1}{3} - \frac{1}{3} \right) = 0. \quad (1.10)$$

Luego, el jugador  $-i$  es indiferente ante cualquier estrategia mixta, pues su ganancia esperada siempre es igual a 0, y por lo tanto cualquier estrategia  $\sigma_{-i}$  es mejor respuesta a  $\sigma_i^*$ . En particular  $\sigma_1^*$  es mejor respuesta a  $\sigma_2^*$  y  $\sigma_2^*$  es mejor respuesta a  $\sigma_1^*$  y el perfil estratégico  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  es un equilibrio de Nash.

#### 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. Todo equilibrio de Nash es un equilibrio correlacionado, pero este último permite otras soluciones importantes [6]. La relación entre los conceptos de equilibrio de Nash y correlacionado se muestra en los Teoremas 1.13 y 1.14.

**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  $x, y \in S_i$ ,

$$\sum_{s_{-i} \in S_{-i}} \psi(x, s_{-i}) [u_i(x, s_{-i}) - u_i(y, s_{-i})] \geq 0. \quad (1.11)$$

Si en la desigualdad (1.11) se cambia el 0 por un  $\epsilon > 0$  se obtiene la definición de  $\epsilon$ -equilibrio correlacionado.

**Teorema 1.13.** Si  $\sigma$  es un equilibrio de Nash, entonces  $\sigma$  es un equilibrio correlacionado.

**Teorema 1.14.** Sea  $\psi \in \Delta(S)$  un equilibrio correlacionado. Si  $\psi$  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  $\psi$  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 (ver Teorema 1.15 abajo). Por lo tanto puede esperarse que existan procedimientos simples para calcular equilibrios correlacionados [6]. En el Capítulo IV se presentan algunos procedimientos que permiten calcular equilibrios correlacionados, y estos procedimientos permiten calcular un equilibrio de Nash cuando los juegos cumplen con ciertas condiciones.

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

**Ejemplo 1.16** ([5, p. 67]). *Juego “batalla de los sexos”. Considere una pareja María y José; ellos tendrán una cita por lo que deben elegir un evento para la cita. A María le gusta el ballet y a José el béisbol. Ellos prefieren ir juntos al mismo evento que ir a eventos diferentes. Sin embargo, cada uno se sentiría más feliz si deciden ir al evento de su preferencia. La Tabla 1.2 es la tabla de pagos correspondiente.*

Tabla 1.2: Tabla de pagos del juego “batalla de los sexos”.

|       |         | José   |         |
|-------|---------|--------|---------|
|       |         | ballet | béisbol |
| María | ballet  | 2, 1   | 0, 0    |
|       | béisbol | 0, 0   | 1, 2    |

En este caso sí existen equilibrios de Nash con estrategias puras. El perfil estratégico (ballet, ballet) es un equilibrio de Nash. En efecto, si José sabe que María siempre elige ballet, lo mejor que él puede hacer es ir al ballet con su compañera, asimismo, si María sabe que José siempre elegirá ballet, lo mejor que puede hacer ella es elegir también ballet. De forma análoga se obtiene que la estrategia (béisbol, béisbol) también es un equilibrio de Nash.

Un tercer equilibrio de Nash ocurre cuando cada jugador utiliza una estrategia tal que su oponente sea indiferente ante la elección de su propia estrategia (en relación a su utilidad). Es decir, José utiliza una estrategia tal que María obtenga siempre la misma ganancia esperada indiferentemente de la estrategia que ella utilice. Análogamente María utiliza una estrategia para la cual José siempre obtiene la misma ganancia sin importar lo que él elija.

Suponga que María (que será considerada el primer jugador) utiliza una estrategia  $\sigma_1 = (x, 1 - x)$  (donde la primera componente corresponde a la probabilidad de elegir

ballet). Si José utiliza una estrategia  $\sigma_2 = (\beta_1, \beta_2)$  entonces su ganancia esperada es igual a  $u_2(\sigma_1, \sigma_2) = x\beta_1 + 2(1-x)\beta_2$ . Luego, José es indiferente a los valores  $\beta_1$  y  $\beta_2$  cuando  $x = 2(1-x)$ , lo que ocurre cuando  $x = \frac{2}{3}$ . En efecto, si  $\sigma_1 = (\frac{2}{3}, \frac{1}{3})$ , entonces la ganancia esperada de José siempre es igual a:

$$u_2(\sigma_1, \sigma_2) = \frac{2}{3}\beta_1 + 2\left(1 - \frac{2}{3}\right)\beta_2 = \frac{2}{3}(\beta_1 + \beta_2) = \frac{2}{3}. \quad (1.12)$$

Por otra parte, si José utiliza una estrategia  $\sigma_2 = (y, 1-y)$  y María utiliza una estrategia  $\sigma_1 = (\theta_1, \theta_2)$  la ganancia esperada para María es igual a  $u_2(\sigma_1, \sigma_2) = 2y\theta_1 + (1-y)\theta_2$  y ella será indiferente a la elección de  $\theta_1$  y  $\theta_2$  cuando  $2y = 1-y$ , i.e., cuando  $y = \frac{1}{3}$ . Note que cuando  $\sigma_2 = (\frac{1}{3}, \frac{2}{3})$ , entonces:

$$u_2(\sigma_1, \sigma_2) = \frac{2}{3}\beta_1 + 2\left(1 - \frac{2}{3}\right)\beta_2 = \frac{2}{3}(\beta_1 + \beta_2) = \frac{2}{3}. \quad (1.13)$$

Luego se tiene que  $\sigma = ((\frac{2}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{2}{3}))$  es un equilibrio de Nash. Sin embargo, ninguna de las 3 soluciones parece realmente satisfactoria. Las 2 primeras son claramente más beneficiosas para uno de los jugadores y la última, aunque podría parecer más justa, proporciona una ganancia esperada menor que las estrategias anteriores para ambos jugadores. ¿Será posible que los jugadores cooperen entre sí para obtener mejores resultados?

Los jugadores podrían realizar lo siguiente: lanzar una moneda, si el resultado es cara van al ballet y si es sello van al béisbol. En este caso ya no se limitan a estrategias mixtas y están eligiendo una distribución de probabilidad sobre todos los perfiles estratégicos. La distribución propuesta es  $\psi \in \Delta(S)$ , tal que  $\psi(\text{ballet}, \text{ballet}) = \psi(\text{beisbol}, \text{beisbol}) = \frac{1}{2}$  y  $\psi(\text{ballet}, \text{beisbol}) = \psi(\text{beisbol}, \text{ballet}) = 0$ .  $\psi$  es un equilibrio correlacionado y ambos jugadores obtendrían una ganancia esperada de  $\frac{3}{2}$ .

# CAPÍTULO II

## JUEGOS EN FORMA EXTENSIVA

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 ( $A$ ) o rechaza ( $R$ ). 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.

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].

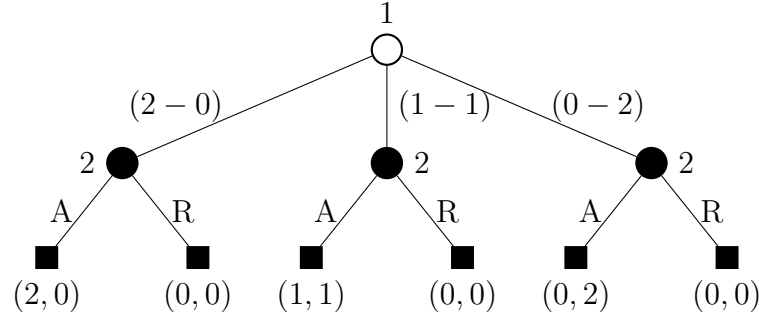


Figura 2.1: Árbol del juego en forma extensiva del Ejemplo 2.1. Los nodos del primer jugador son representados con círculos sin relleno, los nodos del segundo jugador con círculos con relleno negro y los nodos terminales con cuadrados con relleno negro.

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 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 qué 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 intrínseco a los juegos en forma extensiva y no es necesario para juegos en forma normal. Se asumirá que los conjuntos de información vienen definidos de forma explícita en el modelo de juego en forma extensiva; una definición basada en el árbol del juego puede ser encontrada en [10].

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

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

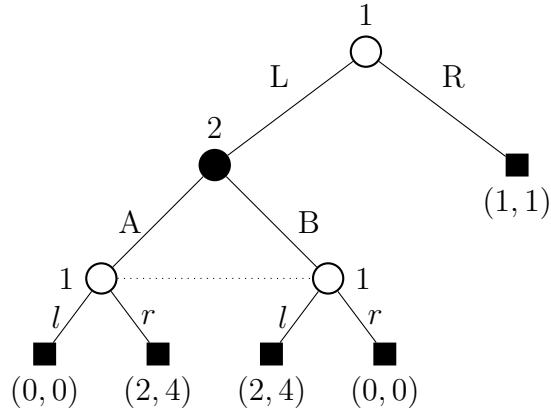


Figura 2.2: Árbol del juego en forma extensiva presentado en el Ejemplo 2.2. Los nodos que pertenecen al mismo conjunto de información son unidos por líneas punteadas.

$H$ .  $Z \subseteq H$  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  $f_c(\cdot|h)$  sobre  $A(h)$ :  $f_c(a|h)$  es la probabilidad de 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 de que  $A(h) = A(h')$  siempre que  $h$  y  $h'$  estén en el mismo bloque de la partición. Para  $I_i \in \mathcal{I}_i$  se denota 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  $u_1 = -u_2$ , se dice que se tiene un **juego de dos jugadores de suma cero en forma extensiva**. Se define  $\Delta_{u,i} = \max_z u_i(z) - \min_z u_i(z)$  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  $P(L) = 2$  indican a cuál jugador le toca jugar en cada nodo del árbol. 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.

### Juego de Kuhn Poker

Kuhn Poker es una versión simplificada del juego de Póker con tres cartas y dos jugadores (denominados jugador 1 y jugador 2) definido por Harold W. Kuhn [11]. 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.

Tabla 2.1: Resumen de las posibles secuencias del juego Kuhn Póker.

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

Debido a que 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, 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 discontinua, o apostar, representado con una línea doble. Los nodos terminales tienen la ganancia del jugador 1, según sea el caso.

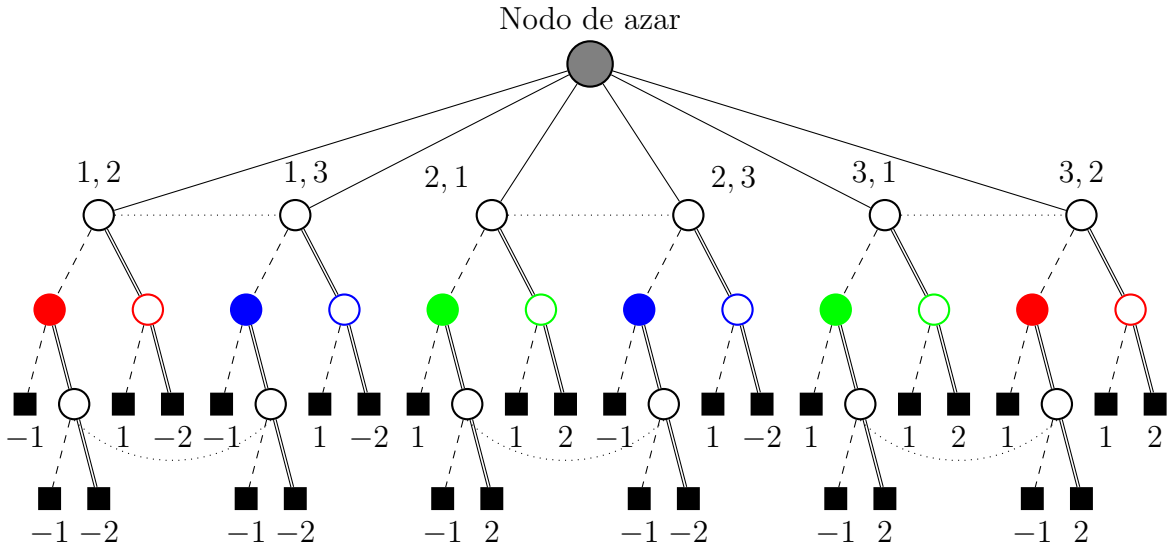


Figura 2.3: Árbol completo del juego Kuhn Póker. Los nodos unidos con líneas punteadas o con el mismo diseño y color pertenecen a el mismo conjunto de información. En cada nodo de decisión la acción *pasar* es representada con una línea discontinua y la acción *apostar* es representada con una línea doble.

## 2.1. Estrategias Puras y Mixtas para Juegos en Forma Extensiva

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 extensiva. 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 extensiva. A continuación, se sigue la formulación de [10] y [4].

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

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 mostradas 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 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 | $s_1 = \text{elegir L y l}$ | $s_2 = \text{elegir L y r}$ |
|       | R | $s_3 = \text{elegir R y l}$ | $s_4 = \text{elegir R y r}$ |

En Kuhn Póker 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.

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  $S = \prod_{i \in N} S_i$

Tabla 2.3: Ejemplo de una estrategia pura para el jugador 2 en el juego Kuhn 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              |

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 de que  $h \in H$  ocurra si todos los jugadores juegan con la estrategia  $s$ . Luego, se define  $u_i : S \rightarrow \mathbb{R}$  como la esperanza de la función de pago para el jugador  $i$  para cada perfil estratégico, la cual viene dada por:

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

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

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

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  $\sigma^m$  viene dada por:

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

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

## 2.2. Forma Normal vs. Forma Extensiva

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

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 2.4 es la tabla de pagos para juego en forma normal que corresponde al Ejemplo 2.2.

Tabla 2.4: Tabla de pagos de la forma normal correspondiente a la forma extensiva del juego presentado en el Ejemplo 2.2.

|           |                  | Jugador 2  |            |
|-----------|------------------|------------|------------|
|           |                  | Elegir $A$ | Elegir $B$ |
| Jugador 1 | 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 únicamente 5 nodos terminales. En general, la forma normal tiene un tamaño exponencial 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 las elecciones posteriores pues el juego termina inmediatamente. 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 [10]: 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 2.4, 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. Si se

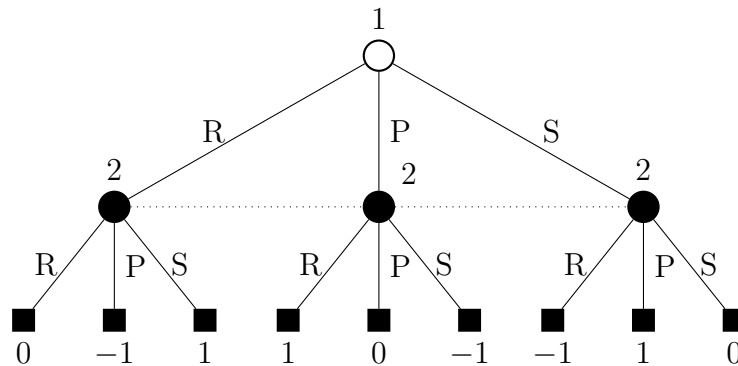


Figura 2.4: Árbol de la forma extensiva del juego piedra, papel o tijera.

aplica el procedimiento descrito anteriormente a la Tabla 2.4 se obtiene un árbol de 13 nodos (Figura 2.5), en contraste a los 9 nodos del árbol original. En efecto, la forma extensiva proporciona más información sobre los juegos que la forma normal. Particularmente, la forma extensiva proporciona información acerca del orden y las posibles secuencias de acciones.

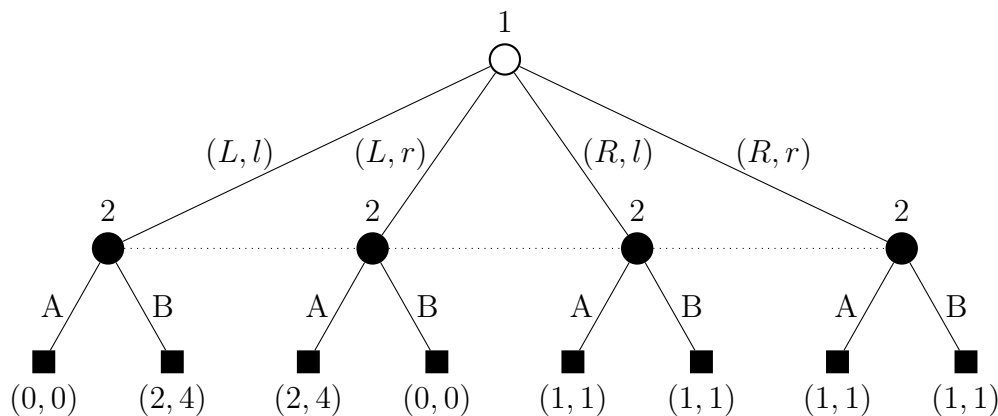


Figura 2.5: Árbol correspondiente a la forma normal de la Tabla 2.4.

### 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 dis-

tribución 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.** Un **perfil estratégico de comportamiento**  $\sigma^b$  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  $\sigma^b$  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  $\varepsilon \geq 0$ , se dice que un perfil estratégico  $\sigma \in \Sigma$  es un  **$\varepsilon$ -equilibrio de Nash** si y sólo si para todo jugador  $i$  y perfil  $\sigma'_i \in \Sigma_i$ ,

$$u_i(\sigma) + \varepsilon \geq u_i(\sigma'_i, \sigma_{-i}). \quad (2.3)$$

El perfil  $\sigma \in \Sigma$  es un **equilibrio de Nash** si y sólo si  $\sigma$  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  $\sigma_1^m$  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  $\sigma_1^b$  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  $\sigma$  un perfil estratégico mixto o de comportamiento. Sea  $\sigma_{-i} = (\sigma_j)_{j \neq i}$  la combinación de todas las estrategias de  $\sigma$  excepto  $\sigma_i$ . Sea  $\pi^{\sigma_i}(h)$  la probabilidad de alcanzar  $h$  dado que el jugador  $i$  utiliza la estrategia  $\sigma_i$  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:

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

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  $\sigma$ . 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  $\pi^{\sigma_{-i}}(I) = \sum_{h \in I} \pi^{\sigma_{-i}}(h)$  la probabilidad de alcanzar el conjunto de información  $I$  dado la estrategia  $\sigma$ ,  $\sigma_i$  y  $\sigma_{-i}$ , respectivamente.

Para un perfil estratégico  $\sigma$ , 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  $\sigma$ . 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].

## Equilibrio de Nash en el Juego de Kuhn Poker

En el juego de Kuhn Poker si ambos jugadores juegan de forma óptima, es decir, acorde a un equilibrio de Nash, entonces el jugador 2 tiene una ganancia esperada de  $\frac{1}{18}$  por mano, como se prueba en [11]. El conjunto de equilibrios de Nash se resume en la Tabla 2.5, donde los conjuntos de información fueron enumerados en un orden de búsqueda por profundidad (DFS).

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  $\alpha$  cuando su carta tenga el número 1, apostar con una probabilidad  $3\alpha$  cuando tenga el número 3 y pasar siempre cuando tenga el número 2. Si el primer jugador tiene un segundo turno, debe pasar siempre

Tabla 2.5: Equilibrio de Nash para el juego de Kuhn Poker. Cada fila de la tabla corresponde con uno o varios conjuntos de información que se denotan con enteros (enumerados utilizando un procedimiento de búsqueda en profundidad sobre el árbol del juego). El equilibrio de Nash corresponde con una distribución aleatoria sobre las acciones pasar y apostar la cuál depende de un parámetro  $\alpha \in [0, \frac{1}{3}]$ .

| Conjunto de información | Equilibrio de Nash     |                        |
|-------------------------|------------------------|------------------------|
|                         | pasar                  | apostar                |
| 1                       | $1 - \alpha$           | $\alpha$               |
| 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                      | $1 - 3\alpha$          | $3\alpha$              |

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 2.6 muestra el árbol con las distribuciones de probabilidad de las estrategias previamente descritas en cada uno de los nodos alcanzables en el juego.

## 2.4. Perfect Recall

El concepto de *perfect recall* hace referencia a juegos en los cuales, en cualquier punto todo jugador *recuerda* lo que sabía previamente [4, p. 203]. En particular, cada jugador recuerda los movimientos públicos que se han hecho durante el juego. La definición de *perfect recall* puede ser dada mediante el árbol del juego [10] o mediante la subsecuencia correspondiente a los nodos de un jugador [4, p. 203] y [12, 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  $\Gamma$  (en forma extensiva) si para cualquier par de historias  $h_1, h_2$  con  $P(h_1) = P(h_2) = i$ , tales que*

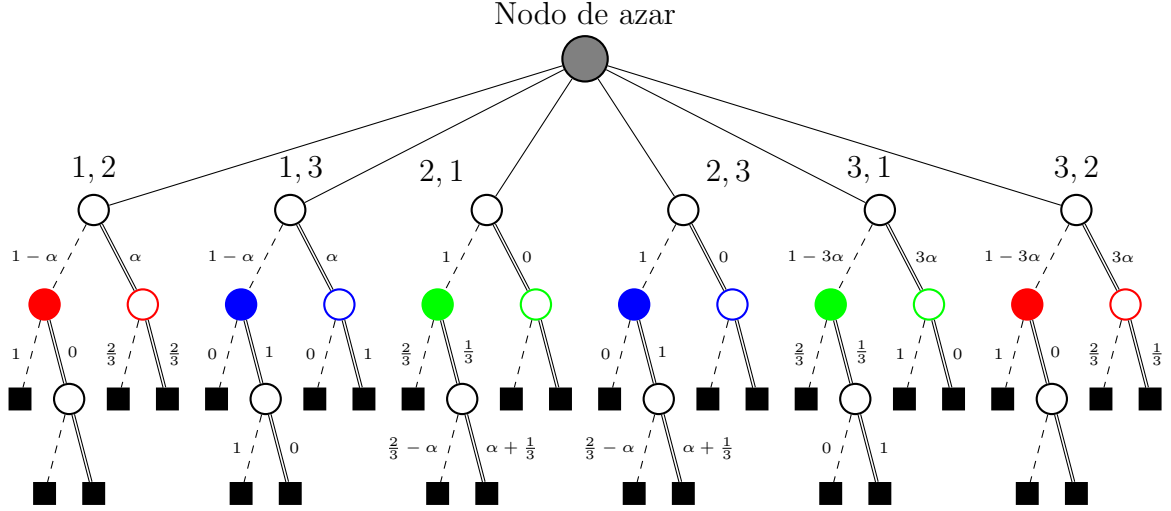


Figura 2.6: Equilibrios de Nash para el juego de Kuhn Poker. Las etiquetas sobre las aristas del árbol indican la probabilidad de escoger dicha acción en cada nodo para un parámetro  $\alpha \in [0, \frac{1}{3}]$ .

$I(h_1) = I(h_2)$  las siguientes condiciones se cumplen:

$$h \sqsubseteq h_1 \implies (\exists h' \sqsubseteq h_2 : I(h) = I(h')), \quad (2.5)$$

$$(h_1, a) \sqsubseteq h \wedge (h_2, b) \sqsubseteq h' \wedge a \neq b \implies I(h) \neq I(h'). \quad (2.6)$$

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, por ejemplo  $h_1$  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  $a$ , entonces el jugador estaría 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 Kuhn Poker son todos juegos con *perfect recall*. El Ejemplo 2.11 muestra un juego con *imperfect recall*.

**Ejemplo 2.11.** ([10]) Considere un juego de dos jugadores de suma cero en el cual el



jugador 1 consta de 2 personas: Alicia y su esposo Bernardo, y el jugador 2 consta de una sólo persona: Zoe. Se tienen dos cartas con los números 1 y 2 que son repartidas aleatoriamente entre Alicia y Zoe. La persona con la carta más alta recibe \$1 de la persona con la carta más baja, y ésta decide si sigue jugando (C) o para el juego (P). Si el juego continúa, Bernardo, sin saber el resultado de la repartición inicial de las cartas, decide si Alicia y Zoe intercambian (I) sus cartas o mantienen las mismas cartas (M). Nuevamente, quien posea la carta más alta recibe \$1 de quien posea la carta más baja, y el juego termina.

La Figura 2.7 representa el juego en forma extensiva. Note que cuando es el turno de Bernardo, 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  $I_1^2 = \{(2-1, C), (1-2, C)\}$ . En particular, no se cumple la primera condición (Ecuación 2.5), pues la secuencia  $(2-1) \sqsubset (2-1, C)$ , pero no existe una subsecuencia de  $(1-2, C)$  que pertenezca a  $I_1^1$ .

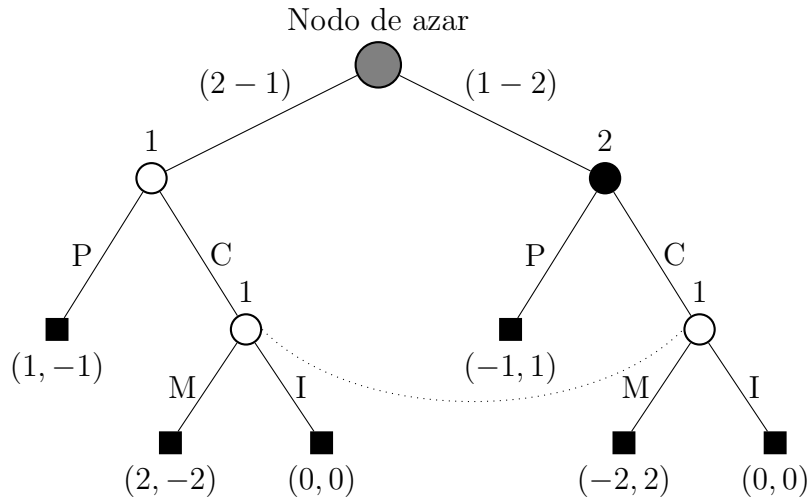


Figura 2.7: Árbol de la forma extensiva del juego con *imperfect recall* presentado en el Ejemplo 2.11. P representa la acción de parar el juego y C de continuarlo. M representa la acción de mantener las cartas obtenidas inicialmente e I representan la acción de intercambiarlas.

Una pregunta de interés es si es posible sustituir una estrategia mixta por una estrategia de comportamiento o viceversa. Para esto, es necesario establecer una definición de equivalencia entre estrategias, y la alcanzabilidad de una historia bajo una estrategia pura.

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

**Definición 2.13.** Sea  $s_i \in S_i$  una estrategia pura del jugador  $i$  e  $I_i \in \mathcal{I}$  un conjunto de información de dicho jugador. Se dice que  $I_i$  es alcanzable bajo  $s_i$  si existe una historia  $h \in H$  tal que  $h \in I_i$  y para toda historia (prefijo)  $h' \sqsubset h$  se cumple que: si  $P(h') = i$ , entonces  $(h', s_i(I(h')))) \sqsubset h$  y si  $P(h') = c$  entonces existe una acción  $a$  tal que  $(h', a) \sqsubset h$  y la probabilidad de elegir la acción  $a$  en  $h'$  es positiva, i.e.,  $f_c(a|h') > 0$ .

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. Las preguntas que se desean responder son las siguientes: (i) ¿Dada una estrategia mixta  $\sigma^m$ , existe una estrategia de comportamiento  $\sigma^b$  tal que  $\sigma^m$  y  $\sigma^b$  son equivalentes? (ii) ¿Dada una estrategia de comportamiento  $\sigma^b$ , existe una estrategia mixta  $\sigma^m$  tal que  $\sigma^b$  y  $\sigma^m$  son equivalentes?. Los Teoremas 2.14 y 2.16 responden estas interrogantes.

El Teorema 2.14 establece que si para cualquier camino de la raíz a un nodo no se pasa 2 o más veces el mismo conjunto de información, entonces para cualquier estrategia de comportamiento existe una estrategia mixta equivalente. Por otra parte, el Teorema 2.16 establece 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 mixtas en este tipo de juegos.

**Teorema 2.14.** Si para el jugador  $i$  se cumple que  $I(h') \neq I(h)$  para cualquier par de historias  $h$  y  $h'$  tal que  $h' \sqsubset h$  y  $P(h) = P(h') = i$ , entonces para cualquier estrategia de comportamiento  $\sigma_i^b \in B^i$  para el jugador  $i$ , existe una estrategia mixta  $\sigma_i^m$  que es **equivalente** a  $\sigma_i^b$ . En particular, la estrategia mixta  $\sigma_i^m$  viene dada por:

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

El Ejemplo 2.15 muestra un juego en el que no se cumple la condición del Teorema 2.14, es decir, un juego en el que una historia pasa más de una vez sobre mismo conjunto de información. En este tipo de juegos se obtiene que el poder expresivo de las estrategia mixtas y las estrategias de comportamiento no son comparables.

**Ejemplo 2.15** ([12, p. 44]). Considere un juego en forma extensiva de dos jugadores definido por:

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

$$P(\emptyset) = P(L) = 1, \quad P(R) = 2, \quad (2.9)$$

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

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

El árbol de este juego, con *imperfect recall*, se muestra en la Figura 2.8.

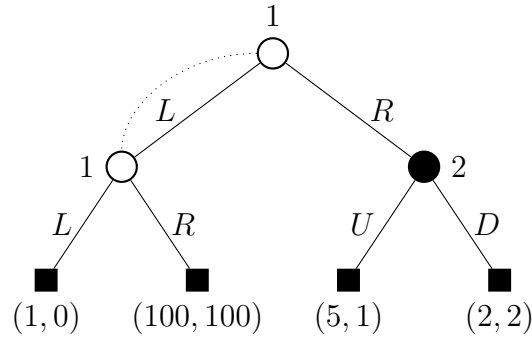


Figura 2.8: Árbol de la forma extensiva del juego con *imperfect recall* presentado en el Ejemplo 2.15. Observe que existe una historia que pasa dos veces sobre el mismo conjunto de información.

Note que la historia  $(L, L)$  atraviesa 2 veces el único conjunto de información del jugador 1. La situación se corresponde con que el jugador 1 *olvida* la elección entre  $L$  o  $R$  que hizo previamente cuando elige  $L$  en la raíz. En este juego el jugador 1 tiene 2 posibles estrategias puras, elegir  $L$  o  $R$ . Por lo tanto, en una estrategia mixta él elige una de estas 2 acciones según alguna distribución de probabilidad. Sin embargo, luego de la elección siempre realizará la misma jugada cuando tenga que tomar una decisión. En particular, la historia  $(L, R)$  no puede ocurrir y el pago de  $(100, 100)$  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  $\sigma = (\sigma_1, \sigma_2)$ , 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 se consideran 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. La estrategia mencionada previamente ya no es un equilibrio de Nash con respecto al conjunto de estrategias de comportamientos. De hecho, el equilibrio se obtiene cuando  $p = \frac{98}{198}$  y el jugador 2 siempre elige  $D$ . Note que para esta estrategia de comportamiento, no existe una estrategia mixta equivalente; sin embargo, en juegos que cumplen la condición del Teorema 2.14 para el jugador  $i$ , toda estrategia de comportamiento para  $i$  tienen una estrategia mixta para  $i$  que es equivalente.

Se considerará nuevamente el Ejemplo 2.11, el cual no tiene *perfect recall*. Las estrategias puras para el jugador 1 son (P, I), (P, M), (C, I), (C, M), y las estrategias puras para el jugador 2 son (P) y (C). Luego, la Tabla 2.6 es la tabla correspondiente a la forma normal del juego, la cual incluye sólo la función de pago del jugador 1, ya que al ser un juego de suma 0, el pago del jugador 2 está completamente determinado.

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

|        | (P)  | (C)  |
|--------|------|------|
| (P, M) | 0    | -0.5 |
| (P, I) | 0    | 0.5  |
| (C, M) | 0.5  | 0    |
| (C, I) | -0.5 | 0    |

Note que el pago que proporciona la estrategia (P, I) al jugador 1 es siempre mayor o igual al pago que le proporciona la estrategia (P, M), sin importar lo que haga el jugador 2. En este caso se dice que la estrategia (P, I) domina a la estrategia (P, M). Asimismo, para este jugador, la estrategia (C, M) domina a la estrategia (C, I). Por lo tanto, el jugador 1 se puede enfocar en encontrar una estrategia mixta que no considere las estrategias (P, M) y (C, I). Supongamos que la estrategia del jugador 1 consiste en elegir las estrategias (P, I) y (C, M) 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 el juego con probabilidad  $\alpha$  y mantener las cartas con una probabilidad  $\beta$ . La probabilidad de elegir cada una de las estrategias puras del jugador 1 se observa en la Tabla 2.7. En la estrategia mixta deseada la estrategia pura (P, M) tiene probabilidad 0, por lo que  $\alpha$  o  $\beta$  debería ser 0. Sin embargo, si  $\alpha = 0$  la estrategia pura (P, I) tiene una probabilidad 0 de ser elegida y si  $\beta = 0$  entonces es imposible elegir

la estrategia (C, M). Luego, no existe una estrategia de comportamiento equivalente a la estrategia mixta deseada. Sin embargo, esto no es un contraejemplo al Teorema 2.16 ya que el juego no tiene *perfect recall*.

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

|                   | P ( $\alpha$ )      | C ( $1 - \alpha$ )        |
|-------------------|---------------------|---------------------------|
| M ( $\beta$ )     | $\alpha\beta$       | $(1 - \alpha)\beta$       |
| I ( $1 - \beta$ ) | $\alpha(1 - \beta)$ | $(1 - \alpha)(1 - \beta)$ |

**Teorema 2.16.** *Considere un juego finito de  $N$  personas. Si 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  $\sigma_i^m$ .*

Se puede observar que cuando se tiene un juego con *perfect recall*, se cumplen las condiciones de los Teoremas 2.16 y 2.14. 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.

**Teorema 2.17** ([12, p. 45]). *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 al teorema anterior se obtiene que el conjunto de los equilibrios de Nash no 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, el estudio es restringido, sin perder generalidad, a las estrategias de comportamientos que se denotarán simplemente por  $\sigma$  (en vez de  $\sigma^b$ ). Sin embargo, es importante resaltar nuevamente que esta equivalencia es cierta únicamente si el juego tiene *perfect recall*. En juegos generales con información incompleta, estrategias mixtas y de comportamiento mantienen conjuntos de equilibrio no comparables [12, p. 45].

# CAPÍTULO III

## EXPLOTABILIDAD

### 3.1. Juegos de Dos Jugadores de Suma Cero

Dado un juego (en forma normal o extensiva) de dos jugadores, se dice que el juego es de suma cero si  $u_1 = -u_2$ . Estos juegos representan competición pura, un jugador debe ganar a expensas del otro [12, p. 5]. De los juegos presentados, piedra, papel o tijera, Kuhn Poker y el Ejemplo 2.11 son juegos de suma cero. Por otra parte, el juego de “la batalla de los sexos” (Ejemplo 1.16) y los ejemplos 2.1, 2.2 y 2.15, no lo son.

¿Qué propiedades importantes tienen este tipo de juegos? La importancia principal es que en estos juegos el equilibrio de Nash es una solución satisfactoria en varios aspectos. Para ver esto, note lo siguiente. Considere un equilibrio de Nash  $\sigma^* = (\sigma_1^*, \sigma_2^*)$ . Se tiene que  $\sigma_2^*$  es mejor respuesta a  $\sigma_1^*$ , lo que es equivalente a que  $u_2(\sigma^*) \geq u_2(\sigma_1^*, \sigma_2)$ , para cualquier estrategia  $\sigma_2$  del segundo jugador. Como  $u_2 = -u_1$ , sustituyendo y multiplicando por  $-1$  la desigualdad, se obtiene que  $u_1(\sigma^*) \leq u_1(\sigma_1^*, \sigma_2)$ . Esto nos dice que si  $u_1(\sigma^*) = u$ , entonces el jugador 1 tendrá una ganancia esperada de al menos  $u$ , **indiferentemente** de la estrategia que utilice su oponente. Análogamente, el jugador 2 puede garantizar una ganancia esperada de al menos  $u_2(\sigma^*) = -u$ .

**Teorema 3.1.** *Sea  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  un equilibrio de Nash de un juego de dos jugadores de suma cero, tal que  $u_1(\sigma^*) = u$ . Entonces  $u_i(\sigma^*) \leq u_i(\sigma_i^*, \sigma_{-i})$ , para cualquier estrategia  $\sigma_{-i}$ .*

Como consecuencia del Teorema 3.1 se obtiene que, dado el jugador  $i$ ,  $u_i(\sigma^*)$  tendrá el mismo valor para cualquier estrategia  $\sigma^*$  que sea un equilibrio de Nash. Además las estrategias de los jugadores son intercambiables y siempre se obtendrá también un equilibrio de Nash. Finalmente, se puede definir el **valor del juego** como  $u_1(\sigma^*)$  con  $\sigma^*$  cualquier equilibrio de Nash (se elige el jugador 1 por convención) [12, p. 17].

**Teorema 3.2.** Sean  $\sigma = (\sigma_1, \sigma_2)$  y  $\sigma' = (\sigma'_1, \sigma'_2)$  equilibrios de Nash en un juego de dos jugadores con suma cero. Entonces  $\sigma'' = (\sigma_1, \sigma'_2)$  y  $\sigma''' = (\sigma'_1, \sigma_2)$  son también equilibrios de Nash. Además,  $u_i(\sigma) = u_i(\sigma') = u_i(\sigma'') = u_i(\sigma''')$ , para  $i \in \{1, 2\}$ .

El Teorema 3.2 no es cierto para juegos que no son de suma cero. Por ejemplo, en “la batalla de los sexos”, cuando ambos jugadores siempre eligen ir al ballet José obtiene una ganancia de 1 y María de 2, cuando los dos siempre eligen ir al béisbol José tiene una ganancia de 2 y María de 1, finalmente, cuando utilizan la estrategia mixta  $\sigma = ((\frac{2}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{2}{3}))$  cada uno obtiene una ganancia esperada de  $\frac{2}{3}$ . Note que se obtienen valores diferentes en ambos casos, aún cuando ambos representan equilibrios de Nash. Además, el perfil estratégico mixto  $\sigma = ((1, 0), (\frac{1}{3}, \frac{2}{3}))$  que no es un equilibrio de Nash da mejor utilidad a ambos jugadores de forma simultanea.

### 3.2. Explotabilidad

Aunque idealmente nos gustaría calcular algún equilibrio de Nash, en la práctica no siempre es posible y usualmente se obtiene alguna aproximación. Por lo cual, es de interés medir que tan alejada se encuentra una estrategia en particular del equilibrio de Nash.

Sea  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  un equilibrio de Nash en un juego de dos jugadores de suma cero. Supongamos ahora que el jugador 1 usa una estrategia  $\sigma_1$ , que es una ligera modificación de  $\sigma^*$ , entonces el jugador 2 puede usar una estrategia que sea mejor respuesta a  $\sigma_1$ , digamos  $\sigma'_2$ . Luego,

$$u_2(\sigma_1, \sigma'_2) \geq u_2(\sigma_1, \sigma_2^*) \geq u_2(\sigma_1^*, \sigma_2^*). \quad (3.1)$$

La primera desigualdad se obtiene porque  $\sigma'_2$  es mejor respuesta del jugador 2 a  $\sigma_1$ , y la segunda desigualdad ocurre porque  $\sigma_1^*$  es mejor respuesta del jugador 1 a  $\sigma_2^*$ . 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  $\varepsilon_1$ . De forma análoga se puede obtener que si el jugador 2 utiliza una estrategia  $\sigma_2$  ligeramente alejada del equilibrio de Nash, esta estrategia será explotable por una cantidad no negativa  $\varepsilon_2$ .

La **explotabilidad**  $\varepsilon_\sigma$  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 estrategia al equilibrio de Nash [13, p. 7]. Si se define  $v_i = u_i(\sigma_i, \sigma'_{-i})$ , entonces por lo anterior  $v_i = u_i(\sigma^*) + \varepsilon_i$ . Si  $u = u_1(\sigma^*)$  es el valor del juego,  $v_1 = u_1(\sigma^*) + \varepsilon_1 = u + \varepsilon_1$  y

$v_2 = u_2(\sigma^*) + \varepsilon_2 = -u + \varepsilon_2$ . Luego,  $\varepsilon_\sigma = u + \varepsilon_1 - u + \varepsilon_2 = v_1 + v_2$ . Note que la explotabilidad puede ser calculada conociendo las mejores respuestas a las estrategias, aún sin conocer el valor del juego.

En los juegos en forma normal se puede calcular el valor  $v_i$  de forma sencilla. Para esto, se utilizará el hecho que para cualquier estrategia de cualquier jugador siempre existe una mejor respuesta cuyo soporte tiene un único elemento (Corolario del Teorema 1.9). Este resultado permite obtener la siguiente expresión para  $v_i$  para calcular la explotabilidad  $\varepsilon_\sigma$  de una estrategia dada  $\sigma = (\sigma_1, \sigma_2)$ :

$$v_i = \max_{s_i \in S_i} u_i(s_i, \sigma_{-i}). \quad (3.2)$$

Considere nuevamente el juego piedra, papel o tijera, y la estrategia  $\sigma = (\sigma_1, \sigma_2)$  con  $\sigma_1 = (0, 33 \ 0, 33 \ 0, 34)$  y  $\sigma_2 = (0, 34 \ 0, 33 \ 0, 33)$ . Calculemos la explotabilidad de  $\sigma_1$ ,  $\sigma_2$  y  $\sigma$ . Se sabe que el valor del juego para este caso es igual a 0. Por otra parte:

$$u_1(\mathcal{R}, \sigma_2) = 0,33(0) + 0,33(-1) + 0,34(0) = 0.01, \quad (3.3)$$

$$u_1(\mathcal{P}, \sigma_2) = 0,33(1) + 0,33(0) + 0,34(-1) = -0.01, \quad (3.4)$$

$$u_1(\mathcal{S}, \sigma_2) = 0,33(-1) + 0,33(1) + 0,34(0) = 0.00. \quad (3.5)$$

Luego,  $v_1 = \max\{0,01; -0,01; 0,00\} = 0,01$  y  $\varepsilon_1 = v_1 - u = 0,01$ . De forma análoga se tiene que  $v_2 = \varepsilon_2 = 0,01$ , y finalmente, se concluye que  $\varepsilon_\sigma = 0,02$ .

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 describen en el Capítulo IV. Sin embargo, la expresión 3.2 no es práctica para juegos en forma extensiva, ya que no es factible listar todos los perfiles estratégicos, como en los juegos en forma normal. Para calcular la explotabilidad en los juegos en forma extensiva (descritos en el capítulo 5) se utiliza el algoritmo propuesto en [13] para calcular la mejor respuesta a una estrategia.



# CAPÍTULO IV

## REGRET MATCHING

En la sección 1.4 se introdujo el concepto de equilibrio correlacionado (Definición 1.12). Además, se afirma que el conjunto de equilibrios correlacionado es un conjunto simple. A continuación se describen tres procedimientos, dos de los cuales llevan a equilibrios correlacionados [6].

### Procedimiento A: Regret Condicional

Sea  $\Gamma$  un juego en forma normal el cual es jugado repetidamente a través del tiempo  $t = 1, 2, \dots$ . Sea  $h_t = (s^\tau)_{\tau=1}^t \in \prod_{\tau=1}^t S$  la historia del juego al inicio del tiempo  $t + 1$ ; i.e., el  $s^\tau$  es el perfil estratégico a tiempo  $\tau$  que contiene las acciones realizadas por cada jugador. El jugador  $i \in N$  elige la estrategia a utilizar a tiempo  $t + 1$  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  $i$  reemplaza 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:

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

La diferencia resultante en el promedio de la función de pago, denotada con  $D_i^t(j, k)$ , para el jugador  $i$  viene dada por (4.2). Por otra parte, 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*.

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

$$R_i^t(j, k) = [D_i^t(j, k)]^+ = \max\{0, D_i^t(j, k)\}. \quad (4.3)$$

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:

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

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** ([6]). *Si cada jugador juega de acuerdo al procedimiento descrito por (4.4), entonces la distribución empírica del juego  $z_t$  converge (a.s.) cuando  $t \rightarrow \infty$  al conjunto de equilibrios correlacionado del juego  $\Gamma$ .<sup>1</sup>*

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 en 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 (Teorema 4.2).

---

<sup>1</sup>Convergencia a.s. (almost sure) indica que el conjunto de secuencias  $\{z_t\}_t$  para las cuales la convergencia no se cumple es un conjunto de probabilidad 0.

**Teorema 4.2.** Sea  $(s_t)_{t=1,2,\dots}$  una secuencia de juegos de  $\Gamma$ . Entonces,  $R_t^i(j, k)$  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  $M_t^i(j, j) = 1 - \frac{1}{\mu} \sum_{k \in S_i, k \neq j} R_t^i(j, k)$  si  $k = j$ .

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

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

**Teorema 4.3.** Sea  $R_t^i(j, j) = 0$ . El vector  $q_t^i$ , definido en 4.5, cumple que:

$$q_t^i(j) \sum_{k \in S_i} R_t^i(j, k) = \sum_{k \in S_i} q_t^i(k) R_t^i(k, j). \quad (4.6)$$

**Teorema 4.4.** 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 (4.6). 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 4.5). En este procedimiento, el pago promedio del jugador  $i$ , en el límite, no es peor al pago si él hubiese jugado cualquier estrategia constante  $k$ , para todo  $\tau \leq t$ .

**Definición 4.5** ([6, p. 1139]). Un procedimiento adaptativo es **universalmente consis-**

*tente* para el jugador  $i$  si:

$$\limsup_{t \rightarrow \infty} \left[ \max_{k \in S_i} \frac{1}{t} \sum_{\tau=1}^t u_i(k, s_{-i}^\tau) - \frac{1}{t} \sum_{\tau=1}^t u_i(s_\tau) \right] \leq 0 \quad (a.s.). \quad (4.7)$$

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

$$D_i^t(k) = \frac{1}{t} \sum_{\tau=1}^t u_i(k, s_{-i}^\tau) - u_i(s_\tau), \quad (4.8)$$

$$R_i^t(k) = [D_i^t(k)]^+ = \max\{0, D_i^t(k)\}. \quad (4.9)$$

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

$$p_{t+1}^i(k) = \frac{R_i^t(k)}{\sum_{k' \in S_i} R_i^t(k')} \quad (4.10)$$

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 4.6.** *El procedimiento adaptativo definido en (4.10) es universalmente consistente para el jugador  $i$ .*

#### 4.1. Regret Matching y Equilibrio de Nash

¿Bajo qué condiciones se puede garantizar que un procedimiento universalmente consistente conduzca a un equilibrio de Nash? El Teorema 4.7 permite concluir que en juegos de dos jugadores de suma cero, si un procedimiento es universalmente consistente, su distribución empírica llevará a un equilibrio de Nash (si  $\mathcal{A}$  es un conjunto se utilizará  $|\mathcal{A}|$  para denotar su cardinalidad).

**Teorema 4.7.** *Sea  $\Gamma$  un juego de dos jugadores de suma cero y sea  $(s^t)_{t=1,2,\dots,T}$  una secuencia de juegos de  $\Gamma$ , tales que, para todo  $s_i \in S_i$ , para todo  $i \in 1, 2$ :*

$$\frac{1}{T} \sum_{t=1}^T u_i(s_i, s_{-i}^t) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \leq \varepsilon \quad (4.11)$$

para algún  $\varepsilon > 0$ . Sea  $\bar{\sigma}^T = (\bar{\sigma}_1^T, \bar{\sigma}_2^T)$ , donde:

$$\bar{\sigma}_i^T(s_i) = \frac{|\{t \leq T : s_i^t = s_i\}|}{T}, \quad (4.12)$$

es decir,  $\bar{\sigma}^T$  es la distribución empírica de probabilidad, note que  $|\{t \leq T : s_i^t = s_i\}|$  es igual al número de veces que se eligió  $s_i$  hasta el tiempo  $T$ . Entonces,  $\bar{\sigma}^T$  es un  $2\varepsilon$ -equilibrio de Nash.

Por otra parte, los procedimientos A y B también son universalmente consistentes (corolario del Teorema 4.8), por lo que los tres procedimientos pueden ser utilizados para calcular una aproximación de un equilibrio de Nash en cualquier juego de dos jugadores de suman cero.

**Teorema 4.8.** *En un procedimiento adaptativo de Regret Matching, si el regret condicional converge a 0, entonces el procedimiento es universalmente consistente.*

## 4.2. Evaluación Empírica de Regret Matching

Los algoritmos propuestos fueron probados en 4 juegos diferentes en forma normal: *matching pennies*, piedra, papel y tijera, ficha vs. dominó, y coronel Blotto. Todos estos juegos son de dos jugadores y, debido a que los algoritmos son universalmente consistentes, pueden ser utilizados para encontrar un equilibrio de Nash en cada uno de ellos. Además, es suficiente con definir el pago para el primer jugador para que el juego esté bien definido.

Por otra parte, un equilibrio de Nash en juegos de dos jugadores de suma cero puede modelarse como un problema de programación lineal [14, pp. 228-233] (ver Apéndice D) y resolverse mediante procedimientos destinados para esto. Esto nos permite encontrar por otro procedimiento un equilibrio de Nash para juegos suficientemente pequeños, y así verificar la correctitud de los algoritmos de *Regret Matching*.

Cada uno de los juegos es descrito mediante sus reglas y, cuando sea factible, mostraremos la matriz de pagos explícitamente. Los problemas de programación lineal que permiten obtener un equilibrio de Nash en cada juego se encuentran en el Apéndice D.

La implementación fue realizada en el lenguaje C++ utilizando la librería estándar y una librería adicional llamada *Eigen* [15] para factorizar matrices y resolver sistemas de ecuaciones. También se implementó una clase para encontrar un equilibrio de Nash mediante el algoritmo de *Regret Matching*. En cada iteración del algoritmo, la actualización

de las estrategias depende de cada procedimiento según las fórmulas propuestas anteriormente. La evaluación experimental de estos algoritmos se realizó en una máquina personal con las siguientes características: procesador Intel® Core™ i5-8250U @ 1.60GHz, 8CPUs y 8GB de memoria. RAM.

Cada procedimiento fue probado 10 veces para cada juego, finalizando cada ejecución al obtener un regret máximo menor a 0,005. Para evaluar la convergencia se midió el tiempo necesario para alcanzar el *regret* deseado y el número de iteraciones. Por cada juego se muestra una tabla con los resultados obtenidos que incluye para cada uno de los procedimientos, la ganancia esperada para el primer jugador al utilizar la estrategia obtenida en la última corrida del algoritmo ( $u(\sigma)$ ) y la explotabilidad ( $\varepsilon_\sigma$ ) de dicha estrategia, así como también el promedio sobre las 10 ejecuciones del algoritmo del tiempo de ejecución en segundos ( $T$ ), el número de iteraciones ( $I$ ) y el tiempo por iteración ( $T/I$ ). También se muestra las gráficas del *regret* con respecto al número de iteraciones de cada uno de los procedimientos. Estas gráficas son mostradas con una escala logarítmica en el eje  $x$  para apreciar mejor los resultados. En el Apéndice F se muestran tablas con resultados adicionales en cada una de las corridas.

## 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 es presentado en la Ecuación D.10 (Apéndice D), cuya solución primal y dual vienen dadas por la Ecuación 4.13. 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.

$$(z^*, x_1^*, x_2^*) = (w^*, y_1^*, y_2^*) = \left(0, \frac{1}{2}, \frac{1}{2}\right). \quad (4.13)$$

Tabla 4.2: Resultados experimentales del juego matching pennies.

|                                     | A                     | B                     | C                     |
|-------------------------------------|-----------------------|-----------------------|-----------------------|
| Ganancia esperada $u(\sigma)$       | 0,000                 | 0,000                 | 0,000                 |
| Explotabilidad $\varepsilon_\sigma$ | 0,006                 | 0,006                 | 0,008                 |
| Tiempo $T$                          | 10,276                | 0,777                 | 0,042                 |
| Iteraciones $I$                     | 3.892.550,4           | 25.616,6              | 16.260,5              |
| $T/I$                               | $2,64 \times 10^{-6}$ | $3,03 \times 10^{-5}$ | $2,58 \times 10^{-6}$ |

La Tabla 4.2 muestra los resultados experimentales obtenidos. Note que la explotabilidad siempre es menor que 0,008. La Figura 4.1 muestra el *regret* con respecto al número de iteraciones en cada juego. Se observa como el *regret* tiende a cero en cada una de las gráficas.

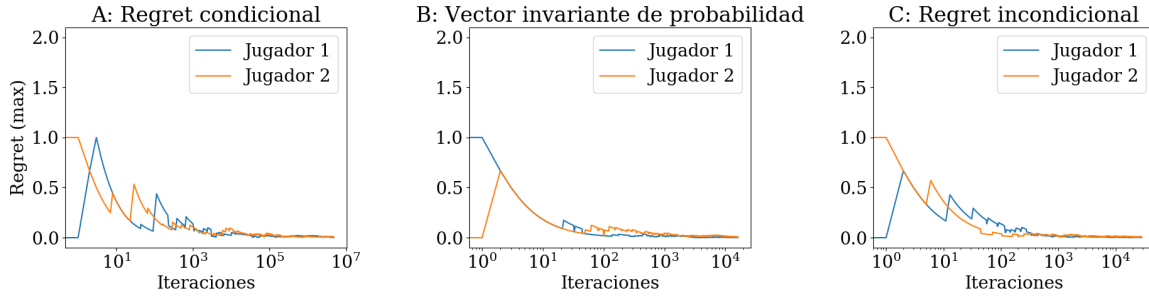


Figura 4.1: Gráficas del *regret* con respecto al número de iteraciones del juego matching pennies.

## Piedra, Papel o Tijera

Este juego es descrito en el Capítulo I y su matriz de pago se muestra en la Tabla 1.1. El problema de programación lineal asociado se encuentra en la Ecuación D.12 y su solución (primal y dual) es:

$$(z^*, x_1^*, x_2^*, x_3^*) = (w^*, y_1^*, y_2^*, y_3^*) = \left(0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \quad (4.14)$$

Luego, el equilibrio de Nash se obtiene cuando ambos jugadores eligen cada una de las acciones con probabilidad igual a  $\frac{1}{3}$  y el valor del juego es igual a 0. Es importante destacar que en todo juego simétrico el valor del juego es 0 y las estrategias óptimas son iguales para ambos jugadores.

Tabla 4.3: Resultados experimentales del juego piedra, papel o tijera.

|                                     | A                     | B                     | C                     |
|-------------------------------------|-----------------------|-----------------------|-----------------------|
| Ganancia esperada $u(\sigma)$       | -0,000012             | 0,000004              | 0,000022              |
| Explotabilidad $\varepsilon_\sigma$ | 0,006                 | 0,010                 | 0,009                 |
| Tiempo $T$                          | 12,198                | 0,345                 | 0,049                 |
| Iteraciones $I$                     | 4.519.054,1           | 6.601,3               | 19.321,1              |
| $T/I$                               | $2,70 \times 10^{-6}$ | $5,23 \times 10^{-5}$ | $2,54 \times 10^{-6}$ |

La Tabla 4.3 muestra el resumen de los resultados para piedra, papel o tijera. Note que la explotabilidad siempre es menor o igual que 0,01. La gráfica 4.2 muestra el *regret* con respecto al número de iteraciones para cada uno de los procedimientos.

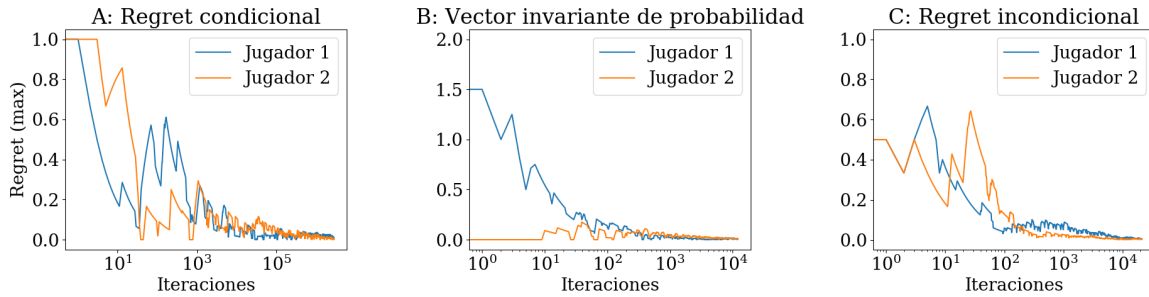


Figura 4.2: Gráficas del *regret* con respecto al número de iteraciones del juego piedra, papel o tijera.

## Ficha vs. Dominó

En este juego cada jugador tiene un tablero de tamaño  $2 \times 3$ . El primer jugador tiene una ficha de dominó que puede colocar de 7 formas diferentes. Cada una de las formas es mostrada en la Figura 4.4, con su respectiva etiqueta. El segundo jugador posee una ficha que ocupa una única casilla de su tablero y la ubica en una de las 6 casillas, las cuales se numeran en la Figura 4.3. 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 [14, p. 237]. La matriz de pagos de este juego se muestra en la Tabla 4.4.

El problema de programación lineal asociado se encuentra en la Ecuación D.15. Este problema no tiene solución única (lo que implica que el juego no tiene un equilibrio de Nash único), una solución (primal y dual) viene dada por:



|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

Figura 4.3: Posibles posiciones de la ficha del segundo jugador en el juego ficha vs. dominó.

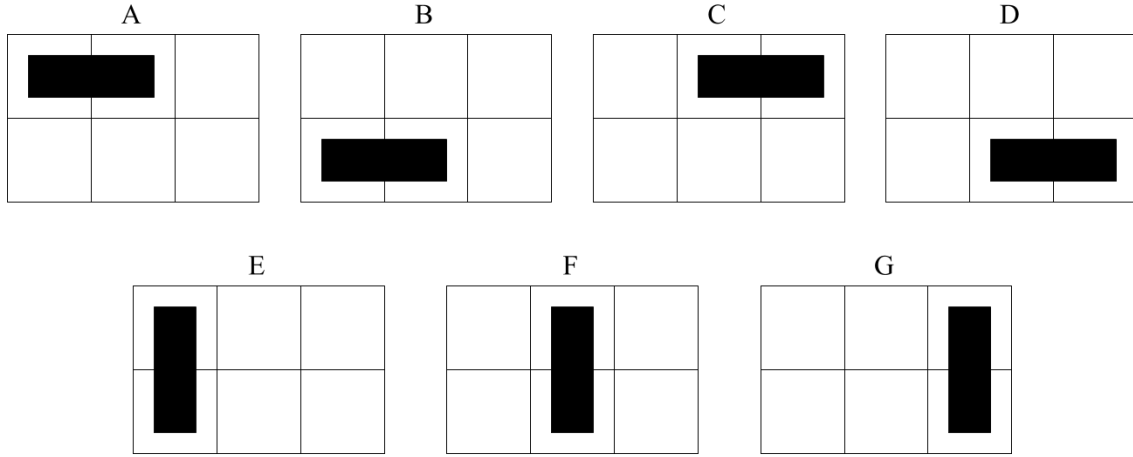


Figura 4.4: Posibles posiciones de la ficha de dominó que representas las acciones del primer jugador en el juego ficha vs. dominó.

$$(z^*, x_1^*, x_2^*, x_4^*, x_5^*, x_6^*, x_6^*, x_7^*) = \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0, \frac{1}{3} \right), \quad (4.15)$$

$$(w^*, y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*) = \left( \frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3}, 0, \frac{1}{3}, 0 \right). \quad (4.16)$$

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. Además, el valor del juego es igual a  $\frac{1}{3}$ . La Tabla 4.5 muestra el resumen de los resultados experimentales. Note que la máxima explotabilidad es igual a 0,01. Por otra parte, la Figura 4.5 muestra el *regret* con respecto al número de iteraciones de este juego, donde se observa la convergencia del *regret* en cada una de ellas.

Tabla 4.4: 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 |

Tabla 4.5: Resultados Experimentales del juego ficha vs. dominó.

|                                     | A                     | B                     | C                     |
|-------------------------------------|-----------------------|-----------------------|-----------------------|
| Ganancia esperada $u(\sigma)$       | 0,333                 | 0,334                 | 0,334                 |
| Explotabilidad $\varepsilon_\sigma$ | 0,010                 | 0,007                 | 0,004                 |
| Tiempo $T$                          | 319,179               | 11,275                | 0,237                 |
| Iteraciones $I$                     | 108.319.272,4         | 75.250,2              | 84.318,5              |
| $T/I$                               | $2,95 \times 10^{-6}$ | $1,50 \times 10^{-4}$ | $2,81 \times 10^{-6}$ |

### 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 debe ser asignado a un único campo, pero cualquier número de soldados puede ser colocado en cualquier campo, incluyendo 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 [16].

Formalmente el juego puede ser descrito de la siguiente manera. Cada jugador debe elegir  $N$  números enteros, digamos  $(a_1, a_2, \dots, a_N)$  y  $(b_1, b_2, \dots, b_N)$ , para el jugador 1 y 2 respectivamente, tales que  $a_1 + a_2 + \dots + a_N = S$  y  $b_1 + b_2 + \dots + b_N = S$ , 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. La ganancia del jugador 1 viene dada por:

$$|\{1 \leq i \leq N : a_i > b_i\}| - |\{1 \leq i \leq N : a_i < b_i\}|. \quad (4.17)$$

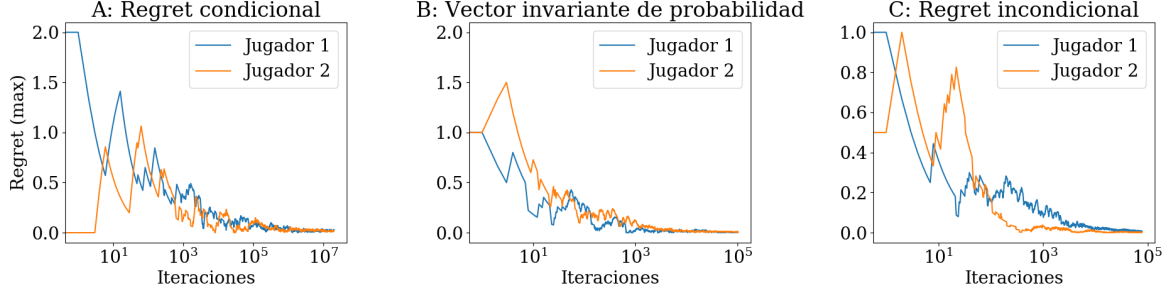


Figura 4.5: Gráficas del *regret* con respecto al número de iteraciones del juego ficha vs. dominó.

Tabla 4.6: Resultados Experimentales del juego coronel Blotto.

|                                     | A                     | B                     | C                     |
|-------------------------------------|-----------------------|-----------------------|-----------------------|
| Ganancia esperada $u(\sigma)$       | 0,000219              | 0,000150              | 0,000024              |
| Explotabilidad $\varepsilon_\sigma$ | 0,010                 | 0,010                 | 0,009                 |
| Tiempo $T$                          | 875,533               | 70,453                | 0,166                 |
| Iteraciones $I$                     | 190.222.305,3         | 58.794,4              | 48.613,5              |
| $T/I$                               | $4,60 \times 10^{-6}$ | $1,20 \times 10^{-3}$ | $3,41 \times 10^{-6}$ |

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 un juego con  $S$  soldados y  $N$  es una matriz cuadrada de tamaño  $\binom{N+S-1}{S-1}$ .

En este juego es necesario generar la matriz de pagos dependiendo de los parámetros. Para esto se creó un programa que dado el número de campos de batalla ( $N$ ) y el número de soldados ( $S$ ), se generan 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  $S = 5$ .

En este juego no se conoce un equilibrio de Nash teóricamente, para valores arbitrarios de  $N$  y  $S$ . Sin embargo, debido a que la matriz de pagos es simétrica, el valor del juego debe ser igual a 0. La Tabla 4.6 muestra los resultados experimentales y la Figura 4.6 muestra las gráficas del *regret* con respecto al número de iteraciones para cada uno de los procedimientos; note la convergencia en cada uno de los casos.

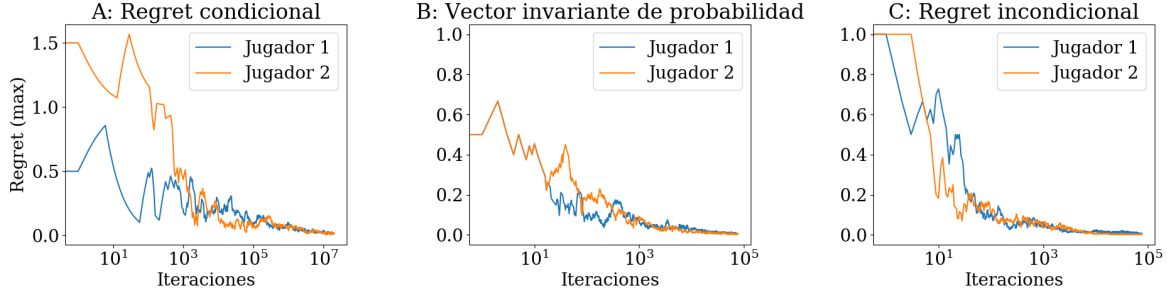


Figura 4.6: Gráficas del *regret* con respecto al número de iteraciones del juego Coronel Blotto.

### 4.3. Análisis de Experimentos

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

#### 4.3.1. 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. 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 en el 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 4.7 muestra un resumen de la complejidad en 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 matriz es sumamente grande, el segundo método sería el menos adecuado.

Tabla 4.7: Complejidad por iteración de cada uno de los procedimientos.

| 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             | $\mathcal{O}(N + M)$     | $\mathcal{O}(N + M)$     |

#### 4.3.2. Número de Iteraciones

Con respecto al número de iteraciones se nota, observando las Tablas 4.2, 4.3, 4.5 y 4.6, que el procedimiento A, *regret* incondicional es el que necesita muchas más iteraciones para converger. Por otra parte, en algunos casos esta métrica fue menor en el procedimiento B y en otros casos el mínimo se obtuvo con el 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 F); esos son casos donde se obtiene el equilibrio de Nash de forma exacta en pocas iteraciones.

#### 4.3.3. Tiempo Transcurrido

Observando el tiempo promedio de los procedimientos en las Tablas 4.2, 4.3, 4.5 y 4.6, 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 ralentiza 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 se puede garantizar que se encontrará alguna solución al juego.

# CAPÍTULO V

## COUNTERFACTUAL REGRET MINIMIZATION

El objetivo de este capítulo es presentar un algoritmo que permita encontrar un equilibrio de Nash en juegos en forma extensiva no determinista con información incompleta y probarlo empíricamente en distintos juegos. 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 minimización del *regret* 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. Descomposición del Regret

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

**Definición 5.1.** *El regret promedio general del jugador  $i$  a tiempo  $T$  es:*

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

$R_i^T$  representa el promedio de lo que el jugador  $i$  dejó de ganar al haber utilizado la estrategia  $\sigma_i^t$  en vez de la estrategia  $\sigma_i^*$  en cada repetición del juego, donde  $\sigma_i^*$  es la estrategia que maximiza, en promedio, la ganancia esperada del jugador  $i$  si éste utiliza una estrategia (mixta) constante.

Se denotará con  $\bar{\sigma}_i^T$  la estrategia promedio del jugador  $i$ , i.e, para cada conjunto de

información  $I \in \mathcal{I}_i$  y para cada acción  $a \in A(I)$  se define:

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

donde  $\sigma_i^t$  es la estrategia utilizada por el jugador  $i$  a tiempo  $t$  y  $\pi^{\sigma_i^t}(I) = \sum_{h \in I} \pi^{\sigma_i^t}$  y  $\pi^{\sigma_i^t}(h)$  es la probabilidad de alcanzar  $h$  si el jugador  $i$  utiliza  $\sigma_i^t$  y todos los demás jugadores (incluyendo el azar) juegan para alcanzar  $h$ .

Esta estrategia es el promedio ponderado de las probabilidades  $\sigma^t(I)(a)$  con respecto a que tan probable es alcanzar  $I$  dado  $\sigma_i^t$ . La relación entre el *regret* promedio general y el concepto de solución se muestra en el siguiente teorema.

**Teorema 5.2.** *En un juego de dos jugadores de suma cero si el regret promedio general a tiempo  $T$  es menor que  $\varepsilon$  entonces  $\bar{\sigma}^T = (\bar{\sigma}_1^T, \bar{\sigma}_2^T)$  es un  $2\varepsilon$ -equilibrio de Nash.*

Luego, un algoritmo que lleve el *regret* promedio general a cero conducirá a un equilibrio de Nash. La idea fundamental consiste en descomponer el *regret* promedio general en un conjunto de términos aditivos de *regret* que puedan ser minimizados de forma independientemente [7]. En particular, es necesario introducir un par de conceptos nuevos, la *utilidad contrafactual* y el *regret contrafactual inmediato*. Estos conceptos son de gran importancia porque son las definiciones análogas a la utilidad y el *regret* en los algoritmos de *Regret Matching* presentados en el Capítulo IV.

**Definición 5.3.** *Sea  $\sigma$  un perfil estratégico e  $I \in \mathcal{I}_i$  un conjunto de información del jugador  $i$ , la **utilidad contrafactual** del par  $(\sigma, I)$  es la ganancia esperada dado que el conjunto  $I$  es alcanzado y todos los jugadores juegan con la estrategia  $\sigma$  con excepción del jugador  $i$  que juega para alcanzar  $I$ . Formalmente:*

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

donde  $Z$  denota el conjunto de historias terminales,  $\pi^{\sigma-i}(h)$  denota la probabilidad de alcanzar  $h$  dado que todos los jugadores utilizan  $\sigma$  excepto el jugador  $i$  que juega para alcanzar  $h$ , i.e.  $\pi^{\sigma-i}(h) = \frac{\pi^\sigma(h)}{\pi^{\sigma_i}(h)}$  y  $\pi^\sigma(h, h')$  denota la probabilidad de ir de la historia  $h$  a la historia  $h'$  dado el perfil estratégico  $\sigma$ , es decir,  $\pi^\sigma(h, h') = \frac{\pi^\sigma(h')}{\pi^\sigma(h)}$  si  $h \sqsubset h'$  y  $\pi^\sigma(h, h') = 0$  en caso contrario.

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

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

$$R_{i,imm}^T(I) = \max_{a \in A(I)} \frac{1}{T} \sum_{t=1}^T \pi^{\sigma^t}_{-i}(I) \left[ u_i(\sigma^t|_{I \rightarrow a}, I) - u_i(\sigma^t, I) \right]. \quad (5.4)$$

Intuitivamente, el *regret* contrafactual inmediato es el arrepentimiento del jugador  $i$  en su decisión sobre 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  $R_{i,imm}^{T,+}(I) = \max\{R_{i,imm}^T(I), 0\}$ . Luego, se tiene el siguiente resultado:

**Teorema 5.5** ([7]).

$$R_i^T \leq \sum_{I \in \mathcal{I}_i} R_{i,imm}^{T,+}(I). \quad (5.5)$$

Debido a que minimizar el *regret* contrafactual inmediato conduce a minimizar el *regret* promedio general es posible enfocarse en minimizar los primeros para obtener un equilibrio de Nash.

## 5.2. Regret Minimization

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

**Definición 5.6.** *El regret promedio de no haber elegido la acción  $a \in A$  hasta tiempo  $T$ , y en su lugar elegir la acción  $a'$  con probabilidad  $p^t(a')$  para  $t = 1, 2, \dots, T$ , se define como:*

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

Sea  $R^{t,+}(a) = \max\{R^t(a), 0\}$ , el algoritmo de *Regret Minimization* consiste en utilizar



a tiempo  $t$  una distribución de probabilidad  $p^t$ , definida por:

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

**Teorema 5.7** ([7]). Si  $|u| = \max_{t \in \{1,2,\dots,T\}} \max_{a,a' \in A} [u^t(a) - u^t(a')]$ , entonces el regret del algoritmo de *Regret Minimization* está acotado por:

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

### 5.3. Counterfactual Regret Minimization (CFR)

El algoritmo CFR es una aplicación del algoritmo *Regret Minimization* de forma independiente a cada conjunto de información. En particular, se mantiene para cada conjunto de información  $I \in \mathcal{I}_i$  y cada acción  $a \in A(I)$ , la medida de regret:

$$R_i^T(I, a) = \frac{1}{T} \sum_{t=1}^T \pi^{\sigma^t-i}(I) \left[ u_i(\sigma^t|_{I \rightarrow a}, I) - u_i(\sigma^t, I) \right] \quad (5.9)$$

Se define  $R_i^{T,+}(I, a) = \max\{R_i^T(I, a), 0\}$ , luego la estrategia  $\sigma_i^{T+1}$  elegida por el jugador  $i$  a tiempo  $T + 1$  es:

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

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. Como cota de convergencia, se tiene el siguiente resultado:

**Teorema 5.8** ([7]). Si el jugador  $i$  selecciona las acciones de acuerdo a la Ecuación 5.10, entonces  $R_{i,imm}^T(I) \leq \Delta_{u,i} \sqrt{|A_i|/T}$ , donde  $|A_i| = \max\{|A(h)| : P(h) = i\}$  es el máximo número de acciones que el jugador  $i$  tiene disponibles en una historia dada y  $\Delta_{u,i} =$

$\max_{z \in Z} u_i(z) - \min_{z \in Z} u_i(z)$  es el rango de las utilidades del jugador  $i$ . Luego:

$$R_i^T \leq \sum_{I \in \mathcal{I}_i} R_{i,imm}^{T,+}(I) \leq \Delta_{u,i} |\mathcal{I}_i| \sqrt{\frac{|A_i|}{T}}. \quad (5.11)$$

#### 5.4. Monte Carlo Counterfactual Regret Minimization (MCCFR)

En la versión presentada del algoritmo CFR es necesario recorrer el árbol completo en cada iteración, algo que puede ser muy ineficiente cuando el árbol es muy grande. Dicha versión se conoce como *vanilla* CFR. En [17] se describe una familia general de algoritmos CFR (basados en muestreo) denominada Monte Carlo Counterfactual Regret Minimization (MCCFR) que evitan recorrer el árbol de forma completa 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. Sea  $\mathcal{Q} = \{Q_1, \dots, Q_r\}$  un conjunto de subconjuntos de  $Z$  (el conjunto de nodos terminales en el árbol) 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  $Q_j$  en la iteración actual donde  $\sum_{j=1}^r q_j = 1$ , y sea  $q(z) = \sum_{j|z \in Q_j} q_j$  la probabilidad de considerar el nodo terminal  $z$  para la iteración actual. La utilidad contrafactual muestreada, cuando se actualiza el bloque  $j$  se define como

$$\tilde{u}_i(\sigma, I|j) = \sum_{h \in I, z \in Q_j} \frac{\pi^{\sigma-i}(h) \pi^\sigma(h, z) u_i(z)}{q(z) \pi^{\sigma-i}(I)}. \quad (5.12)$$

Lo interesante es que la esperanza de la utilidad contrafactual muestreada coincide con la utilidad contrafactual, como se muestra a continuación.

**Teorema 5.9** ([17]).  $E_{j \sim q_j} [\tilde{u}_i(\sigma, I|j)] = u_i(\sigma, I)$ .

Si se elige un único bloque  $Q_1 = Z$ ,  $\mathcal{Q} = \{Q_1\}$ , que contiene todas las historias terminales y para el cual  $q_1 = 1$ , la utilidad contrafactual muestreada se convierte en la utilidad contrafactual, y se obtiene el algoritmo original que se denomina *vanilla* CFR. Por otra parte, si los bloques son separados eligiendo una única acción en cada nodo de azar (siguiendo la distribución de probabilidad correspondiente al nodo) se obtiene un algoritmo que se denomina *chance-sampled* CFR.

### 5.5. Detalles de Implementación y Ejecución

Los algoritmos y la representación de los juegos fueron implementados en el lenguaje C++, utilizando la librería estándar y una librería adicional denominada *Boost* [18] para obtener funciones de hash de los diferentes tipos de datos utilizados. Para la representación de los juegos se utilizó una clase abstracta llamada *Game* que contiene las funciones virtuales necesarias para recorrer el árbol del juego de forma **implícita** (como obtener las acciones de una historia dada y la utilidad de un nodo terminal, entre otras). Estas funciones son implementadas en las clases derivadas según las reglas de cada juego.

Los juegos implementados fueron: *One Card Poker* (OCP), dudo, un juego de dados, y una versión del juego de dominó para 2 personas. Se crearon varias instancias por cada juego de acuerdo a los parámetros de inicialización que reciben cada uno de ellos. Para la resolución de cada instancia se utilizó el algoritmo ***chance-sampled CFR***, el cual se describe con detalle en el Algoritmo 2 (Apéndice E).

Para evaluar la convergencia de los algoritmos y la estrategia obtenida se utilizaron las métricas de *regret* y explotabilidad, respectivamente. Para calcular la explotabilidad en estos juegos se implementó el algoritmo propuesto en [13], denominado *Generalized Expectimax Best Response* (GEBR) (Algoritmos 3, 4, 5 y 6 del Apéndice E).

Cabe destacar que todos los juegos fueron representados mediante árboles con la raíz como único nodo de azar. Algunos juegos tienen esta representación de forma natural (como el Kuhn Póker o el dudo) y 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 implementación de la clase *Game* y de los algoritmos se realizaron suponiendo a la raíz como único nodo de azar en el juego.

Las ejecuciones de los algoritmos se realizaron en *Amazon Web Services* (AWS), utilizando el servicio *Amazon Elastic Compute Cloud* (Amazon EC2), instanciando máquinas virtuales con las siguientes características: procesador Intel Broadwell E5-2686v4, 8CPUs y 32Gb de memoria RAM. Para cada instancia se iteró sobre el árbol durante 10 horas aproximadamente, una vez finalizado el tiempo se calculó la explotabilidad con la estrategia obtenida. En este trabajo, se considera que una instancia de un juego está resuelta si la explotabilidad es no mayor que el 1 % de la mínima utilidad positiva posible.

Por cada juego se presenta una tabla que resume los resultados. Estas tablas contienen el número de nodos del árbol ( $N$ ), el número de conjuntos de información con más de una acción posible ( $I$ ), el valor del juego usando la estrategia obtenida ( $u(\sigma)$ ), la explotabilidad de la estrategia ( $\varepsilon_\sigma$ ), el número de iteraciones realizadas durante el tiempo de entrenamiento y la última columna indica si el juego fue resuelto o no. También se muestra la gráfica del *regret* con respecto al número de iteraciones sobre algunas de las instancias de cada juego. Las gráficas completas pueden ser encontradas en el Apéndice G.

## One-Card Poker

*One-Card Poker*, abreviado OCP( $N$ ), es la versión generalizada del juego Kuhn Póker, explicado en el Capítulo 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 Póker. 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.

La Tabla 5.1 muestra un resumen de los resultados de las instancias del juego OCP. Se observa que se lograron resolver todas los casos propuestos. La Gráfica 5.1 muestra el *regret* con respecto al número de iteraciones en el juego OCP(200). Se observa que al principio el regret aumenta debido a que el regret se inicializa en 0 y empieza a aumentar a medida que se descubren los conjuntos de información. Luego, se observa como converge a 0 a medida que transcurren las iteraciones.

Tabla 5.1: Resultados del algoritmo CFR en el juego OCP( $N$ ) con diferentes números de cartas  $N$ .

| Juego     | $N$         | $I$    | Iteraciones   | $u(\sigma)$ | $\varepsilon_\sigma$ (%) | Resuelto |
|-----------|-------------|--------|---------------|-------------|--------------------------|----------|
| OCP(3)    | 55          | 12     | 1.181.763.638 | -0,056      | 0,0098                   | ✓        |
| OCP(12)   | 1.189       | 48     | 1.147.919.240 | -0,062      | 0,0032                   | ✓        |
| OCP(50)   | 22.051      | 200    | 1.145.291.974 | -0,058      | 0,0099                   | ✓        |
| OCP(200)  | 358.201     | 800    | 1.128.993.847 | -0,056      | 0,0078                   | ✓        |
| OCP(1000) | 8.991.001   | 4.000  | 1.087.573.694 | -0,056      | 0,0098                   | ✓        |
| OCP(5000) | 224.955.001 | 20.000 | 1.038.367.354 | -0,056      | 0,0241                   | ✓        |

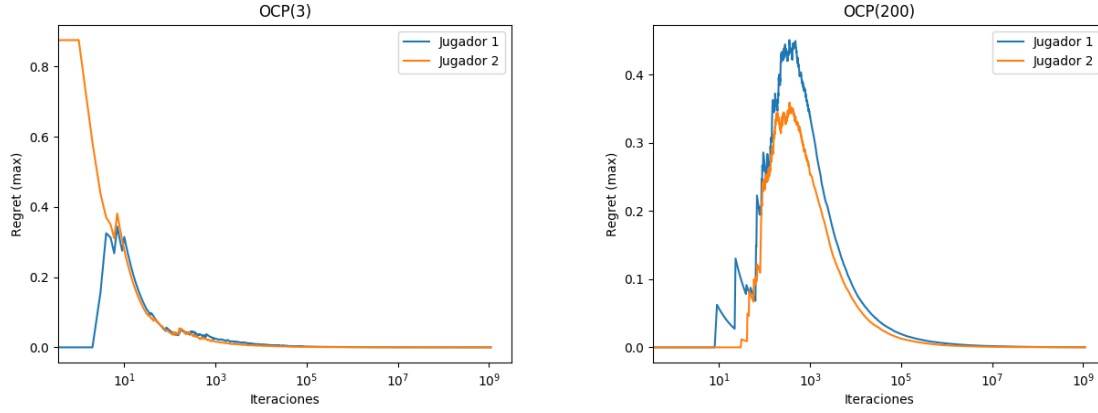


Figura 5.1: Gráficas del *regret* con respecto al número de iteraciones de los juegos *One Card Poker* (200) y *One Card Poker* (3)

## 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. La Figura 5.2 muestra una foto de este juego.

Una apuesta consiste en decir 2 números  $(x, y)$ , lo cual 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



Figura 5.2: Juego dudo. Los vasos se utilizan para lanzar los dados y evitar que los oponentes vean el resultado.

perdió la ronda anterior. Un jugador pierde cuando se queda sin dados, el ganador es el último jugador con al menos un dado restante.

En este trabajo se considerará este juego para 2 jugadores únicamente, además se parametriza según el número de caras de los dados y la cantidad de dados por jugador. De esta forma,  $\text{Dudo}(K, D_1, D_2)$  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  $\text{Dudo}(K, D_1 - 1, D_2)$  y  $\text{Dudo}(K, D_1, D_2 - 1)$  se precaculan y se utilizan como utilidad para las hojas del árbol  $\text{Dudo}(K, D_1, D_2)$ . 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 diferentes posibles, ya que cada resultado puede ser representado con una tupla  $(a_1, a_2, \dots, a_k)$ , donde  $a_j$  representa el número de dados con la cara  $j$ , por lo que  $\sum_j a_j = D_i$  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 secuencia más alta 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 tanto, existen  $2^{K(D_1+D_2)}$  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  $\binom{D_i+K-1}{K-1} 2^{K(D_1+D_2)}$ , incluyendo los conjuntos de información con una única acción posible. Para excluir estos conjuntos de información, se deben excluir las secuencias donde la última apuesta es la

máxima apuesta posible, pues el siguiente jugador sólo podría decir “dudo”. La cantidad de estas secuencias es igual a  $2^{K(D_1+D_2)-1}$ . Luego, el número de conjuntos de información con más de una acción posible es igual a  $\binom{D_i+K-1}{K-1}2^{K(D_1+D_2)-1}$  dado que ambos jugadores tienen el mismo número de dados.

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. El número total de nodos (incluyendo nodos terminales y no terminales) es igual a  $\binom{D_1+K-1}{K-1}\binom{D_2+K-1}{K-1}(2^{K(D_1+D_2)+1} - 1) + 1$ .

La Tabla 5.2 muestra el resumen de los resultados de las instancias del juego dudo. En este juego no se alcanzó la cota deseada para la explotabilidad para las instancias Dudo(4, 2, 2), Dudo(5, 2, 2), Dudo(6, 1, 2) y Dudo(6, 2, 1), siendo la instancia Dudo(5, 2, 2) la que posee la estrategia más explotable con más del 15 % de la ganancia posible. Esto debido al bajo número de iteraciones realizadas durante el entrenamiento (menos de 4.000) por el gran tamaño del árbol. La Gráfica 5.3 corresponde al juego Domino(5, 2, 2). Esta tiene un comportamiento similar a la gráfica anterior pero se observa que la convergencia está inconclusa, por lo que la estrategia final encontrada para Dudo(5, 2, 2) tiene alta explotabilidad.

Tabla 5.2: Resultados del algoritmo CFR en el juego dudo.

| Juego         | $N$         | $I$       | Iteraciones | $u(\sigma)$ | $\varepsilon_\sigma$ (%) | Resuelto |
|---------------|-------------|-----------|-------------|-------------|--------------------------|----------|
| Dudo(3, 1, 1) | 1.144       | 96        | 77.243.464  | -0,111      | 0,0098                   | ✓        |
| Dudo(3, 1, 2) | 18.415      | 1.152     | 10.050.143  | -0,465      | 0,0211                   | ✓        |
| Dudo(3, 2, 1) | 18.415      | 1.152     | 9.903.467   | 0,506       | 0,0111                   | ✓        |
| Dudo(3, 2, 2) | 294.877     | 12.288    | 1.137.993   | 0,0054      | 0,2887                   | ✓        |
| Dudo(4, 1, 1) | 8.177       | 512       | 18.697.532  | -0,125      | 0,0259                   | ✓        |
| Dudo(4, 1, 2) | 327.641     | 14.366    | 1.215.600   | -0,508      | 0,0971                   | ✓        |
| Dudo(4, 2, 1) | 327.641     | 14.366    | 1.213.799   | 0,552       | 0,3701                   | ✓        |
| Dudo(4, 2, 2) | 13.107.101  | 327.680   | 63.109      | 0,0069      | 2,1132                   | ✗        |
| Dudo(5, 1, 1) | 51.176      | 2.560     | 4.521.208   | -0,120      | 0,1186                   | ✓        |
| Dudo(5, 1, 2) | 4.915.126   | 163.840   | 151.235     | -0,565      | 0,6197                   | ✓        |
| Dudo(5, 2, 1) | 4.915.126   | 163.840   | 143.698     | 0,581       | 0,0122                   | ✓        |
| Dudo(5, 2, 2) | 471.858.976 | 7.864.320 | 3.826       | 0,836       | 15,1963                  | ✗        |
| Dudo(6, 1, 1) | 294.877     | 12.288    | 1.067.782   | -0,111      | 0,0975                   | ✓        |
| Dudo(6, 1, 2) | 66.060.163  | 1.769.472 | 17.702      | -0,593      | 4,5781                   | ✗        |
| Dudo(6, 2, 1) | 66.060.163  | 1.769.472 | 17.221      | 0,592       | 3,9594                   | ✗        |

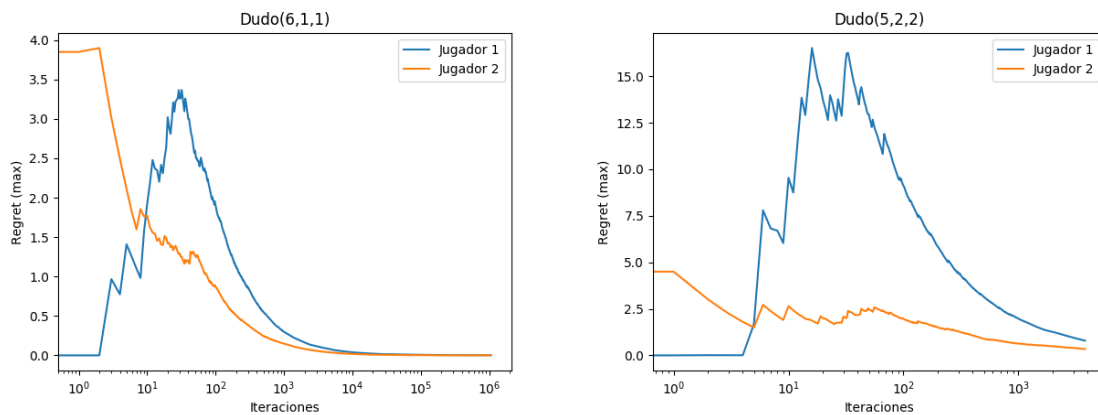


Figura 5.3: Gráficas del *regret* con respecto al número de iteraciones de los juego Dudo (6, 1, 1) y Dudo (5, 2, 2).

## Dominó para Dos Jugadores

Al inicio del juego cada jugador toma una cantidad específica de piezas de forma aleatoria y las piezas restantes se dejan sin descubrir para ser usadas en turnos posteriores. Al igual que 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. 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 (sólo se puede tomar una pieza o pasar si no se puede realizar una jugada con la mano actual).

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).

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  $\text{Domino}(M, N)$  refiere a 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 estado dependen tanto de la mano del jugador como de las piezas en la mesa. En el Kuhn Póker siempre 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 estos parámetros fueron determinados recorriendo el árbol del juego mediante búsqueda en profundidad.

La Tabla 5.3 muestra el resumen de los resultados del juego dominó. En este caso no fue posible resolver la instancia Domino(3, 4) con respecto a los juegos de dominó. La Gráfica 5.4 muestra el *regret* con respecto al número de iteraciones del juego Domino(3, 3). Al igual que en las gráficas anteriores, se observa como el regret crece al principio para luego converger a 0.

Tabla 5.3: Resultados del algoritmo CFR en el juego dominó.

| Juego        | $N$           | $I$     | Iteraciones | $u(\sigma)$ | $\varepsilon_\sigma$ (%) | Resuelto |
|--------------|---------------|---------|-------------|-------------|--------------------------|----------|
| Domino(2, 2) | 7.321         | 102     | 585.191.577 | 2,4000      | 0,0000                   | ✓        |
| Domino(3, 2) | 46.534.657    | 88.947  | 436.905.363 | 2,8767      | 0,0292                   | ✓        |
| Domino(3, 3) | 246.760.993   | 107.854 | 79.751.183  | 2,1539      | 0,3599                   | ✓        |
| Domino(3, 4) | 1.547.645.185 | 104.050 | 11.984.261  | 3,2034      | 1,3630                   | ✗        |

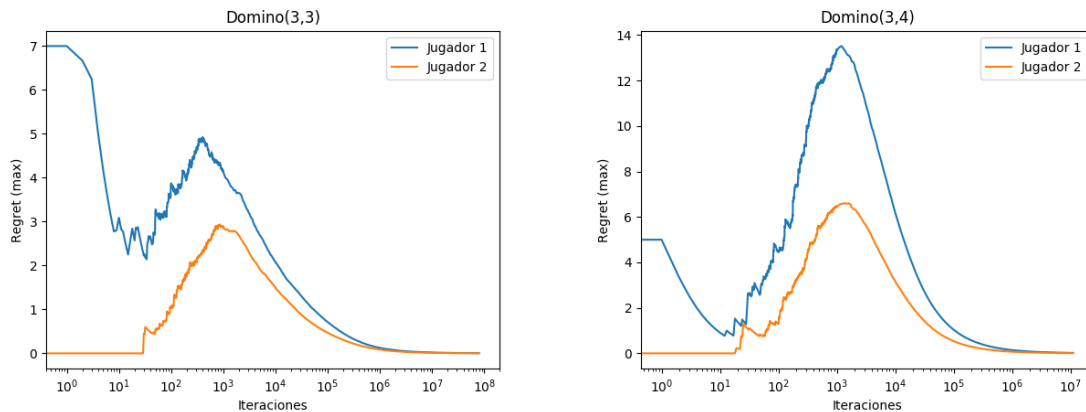


Figura 5.4: Gráficas del *regret* con respecto al número de iteraciones de los juegos Dominó (3, 3) y Dominó (3, 4).

## CONCLUSIONES Y RECOMENDACIONES

En esta investigación se estudiaron los modelos para representar juegos no deterministas con información incompleta, siendo la forma normal el modelo utilizado para representar juegos de una única acción y la forma extensiva el modelo utilizado para representar juegos secuenciales. Además, se estudiaron diferentes conceptos de solución y se utilizó el equilibrio de Nash, como el concepto de solución principal en el caso de juegos de dos jugadores con suma cero, que fueron los considerados en esta investigación.

Para encontrar equilibrios de Nash en juegos en forma normal se utilizaron procedimientos de *Regret Matching* y se evaluaron las estrategias mediante la explotabilidad. Los procedimientos fueron probados en 4 juegos diferentes: piedra, papel o tijera, *matching pennies*, ficha vs. dominó y una instancia del juego coronel Blotto. En todos los juegos se encontraron aproximaciones al equilibrio de Nash con una explotabilidad no mayor que 0,010 (lo que representa el 1 % de la mínima ganancia positiva en todos los casos) por lo que se consideran resueltos.

Para encontrar equilibrios de Nash en juegos en forma extensiva se utilizó el algoritmo *chance-sampled CFR*. Los juegos estudiados presentan *perfect recall*, condición que garantiza la convergencia en el algoritmo en juegos de dos jugadores de suma cero. Este algoritmo fue probado en tres clases de juegos: *One Card Poker* (OCP), dudo (un juego de dados) y una versión del juego de dominó para dos personas.

El juego OCP fue parametrizado por el número de cartas iniciales, representándose con  $OCP(N)$ . En este juego todas las instancias probadas (usando entre 3 y 5.000 cartas) se consideraron resueltas.

El juego dudo, representado con  $Dudo(K, D_1, D_2)$  fue parametrizado por la cantidad de caras de los dados ( $K$ ) y el número de dados de cada jugador ( $D_1, D_2$ ). Fue posible resolver todas las instancias con dados de hasta 4 caras y 2 dados por jugador, i.e.,  $K \leq 4$  y  $D_1, D_2 \leq 2$ . Para dados con 5 caras no fue posible resolver la instancia  $Dudo(5, 2, 2)$ , donde la explotabilidad fue mayor que 15 %, pero fueron resueltas las instancias más pequeñas. Por otra parte, no fue posible resolver ninguna de las instancias propuestas para  $K = 6$ .

El juego de dominó fue parametrizado por el mayor doble presente en el mazo  $M$  y la cantidad de fichas repartidas a cada jugador al inicio del juego  $N$ , representándose con  $\text{Domino}(M, N)$ . Se consideraron la instancia  $\text{Domino}(2, 2)$  y las instancias  $\text{Domino}(3, N)$  con  $N \leq 4$ . Se resolvieron todas las instancias propuestas con la excepción de la instancia  $\text{Domino}(3, 4)$  donde se obtuvo una explotabilidad de 1,363 %.

Futuras investigaciones pueden estar enfocadas en el juego de dominó en particular intentando resolver instancias más grandes, siendo la primera vez que se estudia este juego en particular. Para esto, se recomienda utilizar abstracciones (juegos más pequeños), que pueden ser obtenidos mediante la unión de varios conjuntos de información en uno solo. La idea es unir conjuntos de información similares o idénticos, tales que la información que se pierda no sea importante en cuanto a las estrategias [13, pp. 71-72]. Cabe destacar que estas abstracciones pueden o no tener *perfect recall*, y CFR no garantiza calcular una aproximación a un equilibrio de Nash en un juego con *imperfect recall*. Sin embargo, aún sin las garantías teóricas, esta técnica ha sido utilizada para calcular estrategias fuertes en versiones del juegos de póker que pueden incluso superar sus contrapartes con *perfect recall* [19].

Posibles abstracciones que pueden ser utilizadas en el juego de dominó consisten en considerar únicamente la secuencia final después de cada jugada y no el orden específico en que las fichas fueron colocadas; o simplemente considerar dicha secuencia como un conjunto, sabiendo cuales son los números en los extremos y la información sobre las fichas jugadas por cada jugador. Incluso se puede considerar una abstracción donde no se distinguen qué fichas fueron jugadas por cada jugador. La motivación para proponer estas abstracciones consiste en que las jugadas posibles dependen únicamente de las manos actuales, las fichas restantes y las fichas en los extremos. Sin embargo, se pierde información sobre la secuencia de las jugadas del oponente que puede ser relevante en ciertas circunstancias.

Finalmente, otro posible enfoque para este juego consiste en intentar resolver el juego de dominó para 4 jugadores con equipos de 2 personas, según las reglas clásicas venezolanas, considerando cada equipo como un sólo jugador con *imperfect recall* y observar si es posible calcular una aproximación a un equilibrio de Nash con el algoritmo CFR (o alguna variación del mismo).

## REFERENCIAS

- [1] Myerson, Roger B.: *Game Theory: Analysis of Conflict*. Harvard University Press, 1997.
- [2] Neumann, Jhon von: *Zur Theorie der Gesellschaftsspiele*. Matematische Annalen, 100:295–320, 1928. Traducción al inglés: Tucker, A. W. y Luce, R. D. *On the Theory of Games of Strategy*. Contributions to the Theory of Games, 4, pp. 13–42, 1959.
- [3] Nash, John Forbes: *Non-cooperative games*. Annals of mathematics, 51:286–295, 1954.
- [4] Osborne, Martin J. y Ariel Rubinstein: *A Course in Game Theory*. The MIT Press, Cambridge, Massachusetts, 1994.
- [5] González, Maximiliano y Isabella Otero: *Curso básico de Teoría de Juegos*. Libros de Textos. Ediciones IESA, primera edición, 2007.
- [6] Hart, Sergiu y Andreu Mas-Colell: *A simple adaptative procedure leading to correlated equilibrium*. Econometrica, 68(5):1127–1150, Septiembre 2000.
- [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] Neller, Todd W. y Marc Lanctot: *An Introduction to Counterfactual Regret Minimization*. Informe técnico, Gettysburg College, 2000.
- [9] 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.
- [10] 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.

- [11] 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.
- [12] Leyton-Brown, Kevin y Yoav Shoham: *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. Morgan & Claypool, 2008.
- [13] 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, 2013.
- [14] Chvátal, Vašek: *Linear Programming*. W. H. Freeman and Company, 1983.
- [15] Jacob, Benoît y Gaël Guennebaud: *Eigen*. Disponible en Internet: [http://eigen.tuxfamily.org/index.php?title=Main\\_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page), consultado el 13 de Octubre del 2018.
- [16] 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.
- [17] 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.
- [18] *Boost c++ libraries*. Disponible en Internet: <https://www.boost.org/>, consultado el 6 de Noviembre del 2019.
- [19] Waugh, Kevin, Martin Zinkevich, Michael Johanson, Morgan Kan, David Schnizlein y Michael Bowling: *A Practical Use of Imperfect Recall*. En *Proceedings of SARA 2009: The Eighth Symposium on Abstraction, Reformulation and Approximation*, 2009.
- [20] Blackwell, David: *An analog of the Minimax Theorem for Vector Payoffs*. *Pacific Journal of Mathematics*, 6(1), Noviembre 1956.
- [21] 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

### Capítulo I

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

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

*Demostración.* Partiendo de la Definición 1.6 se obtiene

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

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

□

**Lema A.1.** *Sea  $\sigma_i^*$  una estrategia mixta para el jugador  $i$  que es mejor respuesta a  $\sigma_{-i}$ , y sea  $x \in S_i$  una estrategia pura para el jugador  $i$ . Entonces, para toda estrategia pura  $y \in S_i$  diferente de  $x$ ,*

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

*Demostración.* Considere la estrategia mixta  $\sigma'_i$  definida por:

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

Utilizando el Teorema 1.7 y el hecho que  $\sigma_i^*$  es mejor respuesta a  $\sigma_{-i}$ :

$$u_i(\sigma_i^*, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i}) \quad (\text{A.6})$$

$$= \sum_{z \in S_i} \sigma_i'(z) \sum_{s_{-i}} u_i(z, s_{-i}) \sigma_{-i}(s_{-i}) \quad (\text{A.7})$$

$$= \sum_{z \neq x} \sigma_i^*(z) \sum_{s_{-i}} u_i(z, s_{-i}) \sigma_{-i}(s_{-i}) + \sigma_i^*(x) \sum_{s_{-i}} u_i(y, s_{-i}) \sigma_{-i}(s_{-i}). \quad (\text{A.8})$$

Por el Teorema 1.7,  $u_i(\sigma_i^*, \sigma_{-i}) = \sum_{z \in S_i} \sigma_i^*(z) \sum_{s_{-i}} u_i(z, s_{-i}) \sigma_{-i}(s_{-i})$ . Entonces,

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

□

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

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

Por el Lema A.1,

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

En particular, si  $x$  y  $x'$  son distintos, y ambos pertenecen al soporte de  $\sigma_i$ ,

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

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

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

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

**Teorema 1.13.** *Si  $\sigma$  es un equilibrio de Nash, entonces  $\sigma$  es un equilibrio correlacionado.*

*Demostración.* Sea  $\sigma$  un equilibrio de Nash, sean  $x, y \in S_i$  estrategias puras distintas cualesquiera para el jugador  $i$ , y sea  $\sigma'_i$  una estrategia mixta cualquiera para el jugador  $i$ . Por el Lema A.1,

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

Es decir,

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

Luego,  $\sigma$  es un equilibrio correlacionado.  $\square$

**Teorema 1.14.** *Sea  $\psi \in \Delta(S)$  un equilibrio correlacionado. Si  $\psi$  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  $\psi$  es un equilibrio de Nash.*

*Demostración.* Sea  $\psi = \prod_{i \in N} \sigma_i$  un equilibrio correlacionado en forma factorizada. Se debe mostrar que para cualquier jugador  $i$  y estrategia mixta  $\sigma'_i$  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  $\sigma$  es un equilibrio correlacionado,

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

Al sumar sobre  $x \in S_i$  se obtiene,

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

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

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

donde  $\sigma'$  denota la estrategia  $\sigma' = (\sigma'_i, \sigma_{-i})$ .



Al sumar (A.16) y (A.17), se obtiene que para cualquier  $y$  y  $x^*$  tal que  $\sigma_i(x^*) > 0$ :

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

Por otra parte, note que:

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

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

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

$$= 0. \quad (\text{A.22})$$

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

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

Como  $\sigma'_i$  es una estrategia cualquiera para el jugador  $i$ ,  $\sigma$  es un equilibrio de Nash.  $\square$

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

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

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

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

Sumando las ecuaciones anteriores y factorizando se obtiene:

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

Concluyendo que  $\alpha\sigma + (1 - \alpha)\sigma'$  es un equilibrio correlacionado.  $\square$

## Capítulo II

**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  $\sigma_i^m$  dada por:*

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

*es equivalente a la estrategia  $\sigma_i^b$ .*

*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  $\sigma_i$ . Además, la probabilidad de elegir una estrategia  $s_i$  bajo la estrategia  $\sigma_i^b$  es exactamente el lado derecho de la Ecuación 2.7, la cual, por definición es la probabilidad de elegir  $s_i$  bajo  $\sigma_i^m$ . 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  $\sigma_i$  (mixta o de comportamiento):

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

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

**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  $\sigma_i^m$ .*

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

$$\sigma_i^b(I_i)(a) = P[\text{se elija } a \text{ bajo } \sigma_i^m | I_i \text{ es alcanzable bajo } \sigma_i^m] \quad (\text{A.29})$$

$$= \frac{P[I_i \text{ sea alcanzable bajo } \sigma_i^m \text{ y se elija la opción } a \text{ bajo } \sigma_i^m]}{P[I_i \text{ es alcanzable bajo } \sigma_i^m]} \quad (\text{A.30})$$

$$= \frac{\pi^{\sigma_i^m}(I_i, a)}{\pi^{\sigma_i^m}(I_i)}. \quad (\text{A.31})$$

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  $a_1, a_2, \dots, a_k$  las acciones elegidas por el jugador  $i$  (en ese orden), y sean  $I_i^1, I_i^2, \dots, I_i^k$  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:

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

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

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

Entonces

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

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

### Capítulo III

**Teorema 3.1.** *Sea  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  un equilibrio de Nash de un juego de dos jugadores de suma cero, tal que  $u_1(\sigma) = u$ . Entonces  $u_i(\sigma^*) \leq u_i(\sigma_i^*, \sigma_{-i})$ , para cualquier estrategia  $\sigma_{-i}$ .*

*Demostración.* Como  $\sigma^*$  es un equilibrio de Nash,  $\sigma_{-i}^*$  es mejor respuesta a  $\sigma_i^*$  y por lo tanto, para cualquier estrategia  $\sigma_{-i}$  se obtiene que:

$$u_{-i}(\sigma^*) = u_{-i}(\sigma_i^*, \sigma_{-i}^*) \geq u_{-i}(\sigma_i^*, \sigma_{-i}) \quad (\text{A.35})$$

$$\implies -u_i(\sigma^*) \geq -u_i(\sigma_i^*, \sigma_{-i}) \quad (\text{A.36})$$

$$\implies u_i(\sigma^*) \leq u_i(\sigma_i^*, \sigma_{-i}) \quad (\text{A.37})$$

La inecuación A.36 se obtiene al estar en un juego para dos jugadores de suma cero y la inecuación A.37 se obtiene al multiplicar por menos y cambiar la orientación de la desigualdad.  $\square$

**Teorema 3.2.** Sean  $\sigma = (\sigma_1, \sigma_2)$  y  $\sigma' = (\sigma'_1, \sigma'_2)$  equilibrios de Nash en un juego de dos jugadores con suma cero. Entonces  $\sigma'' = (\sigma_1, \sigma'_2)$  y  $\sigma''' = (\sigma'_1, \sigma_2)$  son también equilibrios de Nash. Además,  $u_i(\sigma) = u_i(\sigma') = u_i(\sigma'') = u_i(\sigma''')$ , para  $i \in \{1, 2\}$ .

*Demostración.* Note que:

- $u_i(\sigma_i, \sigma_{-i}) \leq u_i(\sigma_i, \sigma'_{-i})$  ocurre porque  $\sigma$  es un equilibrio de Nash y el Teorema 3.1.
- $u_i(\sigma_i, \sigma'_{-i}) \leq u_i(\sigma'_i, \sigma'_{-i})$  ocurre porque  $\sigma'$  es un equilibrio de Nash y entonces  $\sigma'_i$  es mejor respuesta a  $\sigma'_{-i}$ .
- $u_i(\sigma'_i, \sigma'_{-i}) \leq u_i(\sigma'_i, \sigma_{-i})$  ocurre porque  $\sigma'$  es un equilibrio de Nash y el Teorema 3.1.
- $u_i(\sigma'_i, \sigma_{-i}) \leq u_i(\sigma_i, \sigma_{-i})$  ocurre porque  $\sigma$  es un equilibrio de Nash y entonces  $\sigma_i$  es mejor respuesta a  $\sigma_{-i}$ .

De lo anterior se obtiene que:

$$u_i(\sigma_i, \sigma_{-i}) \leq u_i(\sigma_i, \sigma'_{-i}) \leq u_i(\sigma'_i, \sigma'_{-i}) \leq u_i(\sigma'_i, \sigma_{-i}) \leq u_i(\sigma_i, \sigma_{-i}). \quad (\text{A.38})$$

Luego, todas las desigualdades en A.38 se cumplen como igualdad. Es decir  $u_i(\sigma) = u_i(\sigma') = u_i(\sigma'') = u_i(\sigma''')$ . Además  $u_i(\sigma_i, \sigma'_{-i}) = u_i(\sigma'_i, \sigma'_{-i}) \geq u_i(\sigma''_i, \sigma'_{-i})$  y  $u_i(\sigma'_i, \sigma_{-i}) = u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma''_i, \sigma_{-i})$ , para cualquier estrategia de  $\sigma''_i$  del jugador  $i$ , por lo tanto  $\sigma_i$  es mejor respuesta a  $\sigma'_{-i}$  y  $\sigma'_i$  es mejor respuesta a  $\sigma_{-i}$ , para  $i = 1, 2$  y por lo tanto  $(\sigma'')$  y  $\sigma'''$  también son equilibrios de Nash.  $\square$

## Capítulo IV

**Teorema 4.2.** Sea  $(s_t)_{t=1,2,\dots}$  una secuencia de juegos de  $\Gamma$ . 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:

$$D_i^t(j, k) = \frac{1}{t} \sum_{\substack{1 \leq \tau \leq t \\ s_i^\tau = j}} u_i(k, s_{-i}^\tau) - u_i(s^\tau) \quad (\text{A.39})$$

$$= \sum_{\substack{s \in S \\ s_i = j}} \frac{1}{t} |\{1 \leq \tau \leq t : s^\tau = s\}| [u_i(k, s_{-i}) - u_i(s)] \quad (\text{A.40})$$

$$= \sum_{\substack{s \in S \\ s_i = j}} z_t(s) [u_i(k, s_{-i}) - u_i(s)]. \quad (\text{A.41})$$

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

$$\sum_{s \in S: s_i = j} z_t(s) [u_i(k, s_{-i}) - u_i(s)] = D_i^t(j, k) \leq \varepsilon, \quad (\text{A.42})$$

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  $\varepsilon$ -equilibrio correlacionado. Por lo tanto, todos los *regrets* convergen a cero si y sólo si  $z_t$  converge al conjunto de equilibrio correlacionado.  $\square$

**Teorema 4.3.** Sea  $R_i^t(j, j) = 0$ . El vector  $q_t^i$ , definido en 4.5, cumple que:

$$q_t^i(j) \sum_{k \in S_i} R_t^i(j, k) = \sum_{k \in S_i} q_t^i(k) R_t^i(k, j). \quad (\text{A.43})$$

*Demostración.*

$$q_t^i(j) = \left[ \sum_{k \in S_i} q_t^i(k) \frac{1}{\mu} R_t^i(k, j) \right] + q_t^i(j) \left[ 1 - \sum_{k \in S_i} \frac{1}{\mu} R_t^i(j, k) \right] \quad (\text{A.44})$$

$$\implies \mu q_t^i(j) = \left[ \sum_{k \in S_i} q_t^i(k) R_t^i(k, j) \right] + q_t^i(j) \left[ \mu - \sum_{k \in S_i} R_t^i(j, k) \right] \quad (\text{A.45})$$

$$\implies \mu q_t^i(j) = \left[ \sum_{k \in S_i} q_t^i(k) R_t^i(k, j) \right] + \mu q_t^i(j) - q_t^i(j) \sum_{k \in S_i} R_t^i(j, k). \quad (\text{A.46})$$

Luego,

$$q_t^i(j) \sum_{k \in S_i} R_t^i(j, k) = \sum_{k \in S_i} q_t^i(k) R_t^i(k, j). \quad (\text{A.47})$$

$\square$

**Teorema 4.4.** 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 (4.6). 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 (Apéndice B) 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

$$[v(s_i, s_{-i})](j, k) = \begin{cases} u_i(k, s_{-i}) - u_i(j, s_{-i}) & \text{si } s_i = j \\ 0 & \text{en otro caso.} \end{cases} \quad (\text{A.48})$$

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

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

$$w_{\mathcal{C}}(\lambda) = \sup\{\lambda \cdot c : c \in \mathcal{C}\} = \sup\left\{\sum_{i \in L} \lambda_i c_i : c_i \leq 0\right\}. \quad (\text{A.49})$$

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,

$$w_{\mathcal{C}} = \begin{cases} 0 & \text{si } \lambda \in \mathbb{R}_+^L, \\ \infty & \text{en caso contrario.} \end{cases} \quad (\text{A.50})$$

Por otra parte, se tiene que:

$$\lambda \cdot v(q_{\lambda}, s_{-i}) = \sum_{(j,k) \in L} \lambda(j, k) \cdot [v(q_{\lambda}, s_{-i})](j, k) \quad (\text{A.51})$$

$$= \sum_{(j,k) \in L} \lambda(j, k) \left[ \sum_{s_i \in S_i} q_{\lambda}(s_i) v(s_i, s_{-i}) \right] (j, k) \quad (\text{A.52})$$

$$= \sum_{(j,k) \in L} \lambda(j, k) q_{\lambda}(j) [v(j, s_{-i})](j, k) \quad (\text{A.53})$$

$$= \sum_{(j,k) \in L} \lambda(j, k) q_{\lambda}(j) [u_i(k, s_{-i}) - u_i(j, s_{-i})] \quad (\text{A.54})$$

$$= \sum_{(j,k) \in L} \lambda(j, k) q_{\lambda}(j) u_i(k, s_{-i}) - \sum_{(j,k) \in L} \lambda(j, k) q_{\lambda}(j) u_i(j, s_{-i}) \quad (\text{A.55})$$

$$= \sum_{k \in S_i} u_i(k, s_{-i}) \sum_{j \in S_i} \lambda(j, k) q_{\lambda}(j) - \sum_{j \in S_i} q_{\lambda}(j) u_i(j, s_{-i}) \sum_{k \in S_i} \lambda(j, k) \quad (\text{A.56})$$

$$= \sum_{j \in S_i} u_i(j, s_{-i}) \sum_{k \in S_i} \lambda(k, j) q_{\lambda}(k) - \sum_{j \in S_i} q_{\lambda}(j) u_i(j, s_{-i}) \sum_{k \in S_i} \lambda(j, k) \quad (\text{A.57})$$

$$= \sum_{j \in S_i} u_i(j, s_{-i}) \left[ \sum_{k \in S_i} \lambda(k, j) q_{\lambda}(k) - q_{\lambda}(j) \sum_{k \in S_i} \lambda(j, k) \right]. \quad (\text{A.58})$$

Se define:

$$\alpha(j) = \sum_{k \in S_i} \lambda(k, j) q_\lambda(k) - q_\lambda(j) \sum_{k \in S_i} \lambda(j, k), \quad (\text{A.59})$$

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

$$\sum_{j \in S_i} u_i(j, s_{-i}) \alpha(j) \leq 0. \quad (\text{A.60})$$

Si se elige  $q_\lambda$  que cumpla

$$q_\lambda(j) \sum_{k \in S_i} \lambda(j, k) = \sum_{k \in S_i} \lambda(k, j) q_\lambda(k) \quad (\text{A.61})$$

para todo  $j \in S_i$ , entonces  $\alpha(j) = 0$  para  $j \in S_i$ , y la condición del Teorema B.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,

$$D_t[j, k] = \sum_{\tau=1}^t v(s_\tau)[j, k] = \sum_{1 \leq \tau \leq t, s_i^\tau = j} u_i(k, s_{-i}^\tau) - u_i(j, s_{-i}^\tau) = D_i^t(j, k), \quad (\text{A.62})$$

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

$$\lambda(D_t) = (R_t^i(j, k))_{(j, k) \in L}. \quad (\text{A.63})$$

Luego, usar una estrategia que cumpla

$$q_\lambda(j) \sum_{k \in S_i} \lambda(j, k) = \sum_{k \in S_i} q_\lambda(k) \lambda(k, j) \quad (\text{A.64})$$

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

$$p_{t+1}^i(j) \sum_{k \in S_i} R_t^i(j, k) = \sum_{k \in S_i} R_t^i(k, j) p_{t+1}^i(k). \quad (\text{A.65})$$

Aplicando el Teorema B.4 se tiene que al usar dicha estrategia,  $D_t$  alcanza a  $\mathbb{R}^-$  que

es equivalente a que  $R_i^t(j, k) \rightarrow 0$  para todo  $j, k \in S_i$ .  $\square$

**Teorema 4.6.** *El procedimiento adaptativo definido en (4.10) 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 B.4 de la siguiente manera:

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

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

$$w_{\mathcal{C}} = \begin{cases} 0 & \text{si } \lambda \in \mathbb{R}_+^L, \\ \infty & \text{en caso contrario.} \end{cases} \quad (\text{A.66})$$

Por otra parte,

$$\lambda \cdot v(q_{\lambda}, s_{-i}) = \sum_{k \in L} \lambda(k) \cdot [v(q_{\lambda}, s_{-i})](k) \quad (\text{A.67})$$

$$= \sum_{k \in S_i} \lambda(k) \cdot \sum_{j \in S_i} q_{\lambda}(j) [v(j, s_{-i})](k) \quad (\text{A.68})$$

$$= \sum_{k \in S_i} \lambda(k) \cdot \sum_{j \in S_i} q_{\lambda}(j) [u_i(k, s_{-i}) - u_i(j, s_{-i})] \quad (\text{A.69})$$

$$= \sum_{\substack{k \in S_i \\ j \in S_i}} \lambda(k) q_{\lambda}(j) [u_i(k, s_{-i}) - u_i(j, s_{-i})] \quad (\text{A.70})$$

$$= \sum_{\substack{k \in S_i \\ j \in S_i}} \lambda(k) q_{\lambda}(j) u_i(k, s_{-i}) - \sum_{\substack{k \in S_i \\ j \in S_i}} \lambda(k) q_{\lambda}(j) u_i(j, s_{-i}) \quad (\text{A.71})$$

$$= \sum_{\substack{j \in S_i \\ k \in S_i}} u_i(j, s_{-i}) \lambda(j) q_{\lambda}(k) - \sum_{\substack{j \in S_i \\ j \in S_i}} u_i(j, s_{-i}) \lambda(k) q_{\lambda}(j) \quad (\text{A.72})$$

$$= \sum_{\substack{j \in S_i \\ k \in S_i}} u_i(j, s_{-i}) [\lambda(j) q_{\lambda}(k) - \lambda(k) q_{\lambda}(j)] \quad (\text{A.73})$$

$$= \sum_{j \in S_i} u_i(j, s_{-i}) \left[ \lambda(j) \sum_{k \in S_i} q_{\lambda}(k) - q_{\lambda}(j) \sum_{k \in S_i} \lambda(k) \right] \quad (\text{A.74})$$

$$= \sum_{j \in S_i} u_i(j, s_{-i}) \left[ \lambda(j) - q_{\lambda}(j) \sum_{k \in S_i} \lambda(k) \right]. \quad (\text{A.75})$$



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

$$\alpha(j) = \lambda(j) - q_\lambda(j) \sum_{k \in S_i} \lambda(k), \quad (\text{A.76})$$

se obtiene  $\lambda \cdot v(q_\lambda, s_{-i}) = \sum_{j \in S_i} u_i(j, s_{-i}) \alpha(j)$ . 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 B.4 en forma de igualdad. Además, para  $D_t = \frac{1}{t} \sum_{\tau=1}^t v(s_\tau)$ , se tiene

$$D_t[k] = \sum_{\tau=1}^t v(s_\tau)[k] = \sum_{\tau \leq t} [u_i(k, s_{-i}^\tau) - u_i(s_\tau)] = D_i^t(k). \quad (\text{A.77})$$

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

$$q_{\lambda(D_t)} = \frac{[\lambda(D_t)](j)}{\sum_{k \in S_i} [\lambda(D_t)](k)} = \frac{R_i^t(j)}{\sum_{k \in S_i} R_i^t(k)}. \quad (\text{A.78})$$

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  $R_i^t(j) \rightarrow 0$  para todo  $j \in S_i$ .  $\square$

**Teorema 4.7.** *Sea  $\Gamma$  un juego de dos jugadores de suma cero y sea  $(s^t)_{t=1,2,\dots,T}$  una secuencia de juegos de  $\Gamma$ , tales que, para todo  $s_i \in S_i$ , para todo  $i \in 1, 2$ :*

$$\frac{1}{T} \sum_{t=1}^T u_i(s_i, s_{-i}^t) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \leq \varepsilon \quad (\text{A.79})$$

para algún  $\varepsilon > 0$ . Sea  $\bar{\sigma}^T = (\bar{\sigma}_1^T, \bar{\sigma}_2^T)$ , donde:

$$\bar{\sigma}_i^T(s_i) = \frac{|\{t \leq T : s_i^t = s_i\}|}{T}, \quad (\text{A.80})$$

es decir,  $\bar{\sigma}^T$  es la distribución empírica de probabilidad, note que  $|\{t \leq T : s_i^t = s_i\}|$  es igual al número de veces que se eligió  $s_i$  hasta el tiempo  $T$ . Entonces,  $\bar{\sigma}^T$  es un  $2\varepsilon$ -equilibrio de Nash.

*Demostración.* Se denotará con  $\#(s_i)$  el número de veces que se ha elegido  $s_i$  a tiempo  $T$ , i.e.,  $\#(s_i) = |\{t \leq T : s_i^t = s_i\}|$ . Por hipótesis del teorema, se tiene que:

$$\frac{1}{T} \sum_{t=1}^T u_i(s_i, s_{-i}^t) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \leq \varepsilon. \quad (\text{A.81})$$

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

$$\frac{1}{T} \sum_{s_{-i} \in S_{-i}} \#(s_{-i}) u_i(s_i, s_{-i}) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \leq \varepsilon \quad (\text{A.82})$$

$$\implies \sum_{s_{-i} \in S_{-i}} \bar{\sigma}_{-i}^T(s_{-i}) u_i(s_i, s_{-i}) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \leq \varepsilon. \quad (\text{A.83})$$

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

$$\sum_{s_i \in S_i} \sigma_i(s_i) \left[ \sum_{s_{-i} \in S_{-i}} \bar{\sigma}_{-i}^T(s_{-i}) u_i(s_i, s_{-i}) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \right] \leq \sum_{s_i \in S_i} \sigma_i(s_i) \varepsilon \quad (\text{A.84})$$

$$\implies \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma_i(s_i) \bar{\sigma}_{-i}^T(s_{-i}) u_i(s_i, s_{-i}) - \sum_{s_i \in S_i} \sigma_i(s_i) u_i(s^t) \leq \varepsilon \quad (\text{A.85})$$

$$\implies u_i(\sigma_i, \bar{\sigma}_{-i}^T) - \frac{1}{T} \sum_{t=1}^T u_i(s^t) \leq \varepsilon. \quad (\text{A.86})$$

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

$$u_1(\sigma_1, \bar{\sigma}_2^T) - \frac{1}{T} \sum_{t=1}^T u_1(s^t) \leq \varepsilon \quad (\text{A.87})$$

$$u_2(\bar{\sigma}_1^T, \sigma_2) - \frac{1}{T} \sum_{t=1}^T u_2(s^t) \leq \varepsilon. \quad (\text{A.88})$$

Además, como  $\Gamma$  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:

$$u_2(\bar{\sigma}_1^T, \sigma_2) - \frac{1}{T} \sum_{t=1}^T u_2(s^t) = -u_1(\bar{\sigma}_1^T, \sigma_2) - \frac{1}{T} \sum_{t=1}^T -u_1(s^t) \leq \varepsilon. \quad (\text{A.89})$$

En particular, si  $\sigma_2 = \bar{\sigma}_2^T$  entonces:

$$-u_1(\bar{\sigma}_1^T, \bar{\sigma}_2^T) + \frac{1}{T} \sum_{t=1}^T u_1(s^t) \leq \varepsilon. \quad (\text{A.90})$$

Al sumar las desigualdades A.87 y A.90 se obtiene que:

$$u_1(\sigma_1, \bar{\sigma}_2^T) - u_1(\bar{\sigma}_1^T, \bar{\sigma}_2^T) \leq 2\varepsilon \quad (\text{A.91})$$

$$\implies u_1(\bar{\sigma}^T) + 2\varepsilon \geq u_1(\sigma_1, \bar{\sigma}_2^T). \quad (\text{A.92})$$

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  $2\varepsilon$ -equilibrio de Nash.  $\square$

**Teorema 4.8.** *En un procedimiento adaptativo de Regret Matching, si el regret condicional converge a 0, entonces el procedimiento es universalmente consistente.*

*Demostración.* Se demostrará, como en el teorema anterior, que el *regret* incondicional tiende a 0. De la Ecuación 4.2 se tiene que:

$$D_i^t(j, k) = \frac{1}{t} \sum_{\substack{1 \leq \tau \leq t \\ s_i^\tau = j}} u_i(k, s_{-i}^\tau) - u_i(s^\tau) \quad (\text{A.93})$$

si se suman los  $D_i^t(j, k)$  sobre  $j \in S_i$ , se obtiene:

$$\sum_{j \in S_i} D_i^t(j, k) = \sum_{j \in S_i} \left( \frac{1}{t} \sum_{\substack{1 \leq \tau \leq t \\ s_i^\tau = j}} u_i(k, s_{-i}^\tau) - u_i(s^\tau) \right) = \frac{1}{t} \sum_{1 \leq \tau \leq t} u_i(k, s_{-i}^\tau) - u_i(s^\tau). \quad (\text{A.94})$$

De la Ecuación 4.9 se obtiene el último miembro de la igualdad es igual a  $D_i^t(k)$ . Luego:

$$\sum_{j \in S_i} D_i^t(j, k) = D_i^t(k). \quad (\text{A.95})$$

Por otra parte, utilizando la desigualdad triangular en  $\mathbb{R}$ , se tiene que

$$\sum_{j \in S_i} \max \{0, D_i^t(j, k)\} \geq \max \left\{ 0, \sum_{j \in S_i} D_i^t(j, k) \right\}, \quad (\text{A.96})$$

al sustituir por las definiciones de  $R_i^t(j, k)$  y  $R_i^t(k)$  se concluye que:

$$\sum_{j \in S_i} R_i^t(j, k) \geq R_i^t(k). \quad (\text{A.97})$$

Luego como  $R_i^t(j, k) \rightarrow 0$  cuando  $t \rightarrow \infty$  para todo  $j, k \in S_i$ , entonces  $\sum_{j \in S_i} R_i^t(j, k) \rightarrow$

0 cuando  $t \rightarrow \infty$  y por lo tanto  $R_i^t(k) \rightarrow 0$  cuando  $t \rightarrow \infty$ . Luego si el *regret* condicional converge a 0, entonces el *regret* incondicional también converge a 0 y por lo tanto el procedimiento es universalmente consistente.  $\square$

## Capítulo V

**Teorema 5.2.** *En un juego de 2 jugadores de suma cero si el regret promedio general a tiempo  $T$  es menor que  $\varepsilon$  entonces  $\sigma^{-T}$  es un  $2\varepsilon$ -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  $h_1 \sqsubset h_2 \sqsubset h_3 \sqsubset \dots \sqsubset h_m \sqsubset z$  todos los prefijos de  $z$  correspondientes al jugador  $i$ , es decir  $P(h_k) = i \ \forall k : 1 \leq k \leq m$  y sean  $a_1, a_2, \dots, a_m$  las acciones correspondientes en  $z$  en cada historia respectiva. Luego:

$$\pi^{\bar{\sigma}_i^T}(z) = \prod_{k=1}^m \bar{\sigma}_i^T(I(h_k))(a_k) \quad (\text{A.98})$$

$$= \prod_{k=1}^m \frac{\sum_{t=1}^T \pi^{\sigma_i^t}(I(h_k)) \sigma_i^t(I(h_k))(a_k)}{\sum_{t=1}^T \pi^{\sigma_i^t}(I(h_k))} \quad (\text{A.99})$$

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:

$$\pi^{\bar{\sigma}_i^T}(z) = \frac{\sum_{t=1}^T \pi^{\sigma_i^t}(I_m) \sigma_i^t(I_m)(a_m)}{\sum_{t=1}^T \pi^{\sigma_i^t}(I_1)} \quad (\text{A.100})$$

$$= \frac{\sum_{t=1}^T \pi^{\sigma_i^t}(z)}{\sum_{t=1}^T 1} \quad (\text{A.101})$$

$$= \frac{1}{T} \sum_{t=1}^T \pi^{\sigma_i^t}(z). \quad (\text{A.102})$$

Además, se tiene que, para cualquier jugador  $i$  y cualquier estrategia de  $\sigma_i$ :

$$\frac{1}{T} \sum_{t=1}^T u_i(\sigma'_i, \sigma_{-i}^t) = \frac{1}{T} \sum_{t=1}^T \left( \sum_{z \in Z} \pi^{\sigma'_i}(z) \pi^{\sigma_{-i}^t}(z) \pi^c(z) \right) \quad (\text{A.103})$$

$$= \sum_{z \in Z} u_i(z) \pi^{\sigma'_i}(z) \pi^c(z) \left( \frac{1}{T} \sum_{t=1}^T \pi^{\sigma_{-i}^t}(z) \right) \quad (\text{A.104})$$

$$= \sum_{z \in Z} u_i(z) \pi^{\sigma'_i}(z) \pi^{\bar{\sigma}_i^T}(z) \pi^c(z) = u_i(\sigma'_i, \bar{\sigma}_{-i}^T). \quad (\text{A.105})$$

Por otra parte, como  $R_2^T \leq \varepsilon$ , para todo  $\sigma'_2 \in B_2$  se tiene que:

$$\frac{1}{T} \sum_{t=1}^T [u_2(\sigma_1^t, \sigma'_2) - u_2(\sigma^t)] \leq \varepsilon \quad (\text{A.106})$$

$$\Rightarrow \frac{1}{T} \sum_{t=1}^T u_2(\sigma^t) + \varepsilon \geq \frac{1}{T} \sum_{t=1}^T u_2(\sigma_1^t, \sigma'_2) \quad (\text{A.107})$$

$$\Rightarrow \frac{1}{T} \sum_{t=1}^T u_2(\sigma^t) + \varepsilon \geq u_2(\bar{\sigma}_1^T, \sigma'_2). \quad (\text{A.108})$$

Como se cumple para cualquier estrategia  $\sigma'_2$ , se cumple para  $\sigma_2^t$  para  $t = 1, 2, \dots, T$ , obteniendo:

$$\frac{1}{T} \sum_{t=1}^T u_2(\sigma^t) + \varepsilon \geq \frac{1}{T} \sum_{t=1}^T u_2(\bar{\sigma}_1^T, \sigma_2^t) \quad (\text{A.109})$$

$$= \frac{1}{T} \sum_{t=1}^T \sum_{z \in Z} u_2(z) \pi^{\bar{\sigma}_1^T} \pi^{\sigma_2^t}(z) \pi^c(z) \quad (\text{A.110})$$

$$= \sum_{z \in Z} u_2(z) \pi^{\bar{\sigma}_1^T} \pi^c(z) \left( \frac{1}{T} \sum_{t=1}^T \pi^{\sigma_2^t}(z) \right) \quad (\text{A.111})$$

$$= \sum_{z \in Z} u_2(z) \pi^{\bar{\sigma}_1^T} \pi^{\bar{\sigma}_2^T} \pi^c(z) \quad (\text{A.112})$$

$$= u_2(\bar{\sigma}^T). \quad (\text{A.113})$$

Como  $\Gamma$  es un juego de suma cero, se tiene que  $u_2(\sigma) = -u_1(\sigma)$  para cualquier estrategia  $\sigma$ , luego:

$$\frac{1}{T} \sum_{t=1}^T u_2(\sigma^t) + \varepsilon \geq u_2(\bar{\sigma}^T) \quad (\text{A.114})$$

$$\Rightarrow \frac{1}{T} \sum_{t=1}^T -u_1(\sigma^t) + \varepsilon \geq -u_1(\bar{\sigma}^T) \quad (\text{A.115})$$

$$\Rightarrow u_1(\bar{\sigma}^T) + \varepsilon \geq \frac{1}{T} \sum_{t=1}^T u_1(\sigma^t) \quad (\text{A.116})$$

$$\Rightarrow u_1(\bar{\sigma}^T) + 2\varepsilon \geq \frac{1}{T} \sum_{t=1}^T u_1(\sigma^t) + \varepsilon \quad (\text{A.117})$$

Por otra parte, como  $R_i^t \leq \varepsilon$  se tiene que, para cualquier  $\sigma'_1 \in B_1$ :

$$\frac{1}{T} \sum_{t=1}^T u_1(\sigma^t) + \varepsilon \geq \frac{1}{T} \sum_{t=1}^T u_1(\sigma'_1, \sigma_2^t) = u_1(\sigma'_1, \bar{\sigma}_2^T). \quad (\text{A.118})$$

Luego, se obtiene que:

$$u_1(\bar{\sigma}^T) + 2\varepsilon \geq \frac{1}{T} \sum_{t=1}^T u_1(\sigma^t) + \varepsilon \geq u_1(\sigma'_1, \bar{\sigma}_2^T) \quad (\text{A.119})$$

$$\implies u_1(\bar{\sigma}^T) + 2\varepsilon \geq u_1(\sigma'_1, \bar{\sigma}_2^T). \quad (\text{A.120})$$

Análogamente, se demuestra que  $u_2(\bar{\sigma}^T) + 2\varepsilon \geq u_2(\bar{\sigma}_1^T, \sigma'_2)$  concluyendo que  $\bar{\sigma}^T$  es un  $2\varepsilon$ -equilibrio de Nash.  $\square$

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

*Demostración.*

$$E_{j \sim q_j}[\tilde{u}_i(\sigma, I|j)] = \sum_j q_j u_i(\sigma, I) \quad (\text{A.121})$$

$$= \sum_j q_j \frac{\sum_{h \in I, z \in Q_j} \pi^{\sigma-i}(h) \pi^\sigma(h, z) u_i(z)}{q(z) \pi^{\sigma-i}(I)} \quad (\text{A.122})$$

$$= \sum_j \sum_{\substack{h \in I \\ z \in Q_j}} \frac{q_j \pi^{\sigma-i}(h) \pi^\sigma(h, z) u_i(z)}{q(z) \pi^{\sigma-i}(I)} \quad (\text{A.123})$$

$$= \sum_{\substack{h \in I \\ z \in Z}} \sum_{j|z \in Q_j} \frac{q_j \pi^{\sigma-i}(h) \pi^\sigma(h, z) u_i(z)}{q(z) \pi^{\sigma-i}(I)} \quad (\text{A.124})$$

$$= \sum_{\substack{h \in I \\ z \in Z}} \left( \frac{\sum_{j|z \in Q_j} q_j}{q(z)} \right) \frac{\pi^{\sigma-i}(h) \pi^\sigma(h, z) u_i(z)}{\pi^{\sigma-i}(I)} \quad (\text{A.125})$$

$$= \sum_{\substack{h \in I \\ z \in Z}} \frac{\pi^{\sigma-i}(h) \pi^\sigma(h, z) u_i(z)}{\pi^{\sigma-i}(I)} = u_i(\sigma, I) \quad (\text{A.126})$$

La ecuación A.123 se obtiene de la definición de  $\tilde{u}_i(\sigma, I|j)$ . A.124 y A.125 se obtienen al reordenar las sumatorias y considerando que la unión de los bloques generan a  $Z$ . La ecuación A.126 es la definición de utilidad contrafactual.  $\square$

## APÉNDICE B

### TEOREMA DE APROXIMACIÓN DE BLACKWELL

Los procedimientos que calculan equilibrios correlacionados se basan en el método de aproximación de Blackwell [20]. En este apéndice se muestra el teorema como es presentado en [6].

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

$$D_t = \frac{1}{t} \sum_{\tau=1}^t v(s_\tau) = \frac{1}{t} \sum_{\tau=1}^t v(s_i^\tau, s_{-i}^\tau) \quad (\text{B.1})$$

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 B.1), un conjunto alcanzable (Definición B.2), y de función de soporte (Definición B.3).

**Definición B.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

$$\text{dist}(x, A) = \min\{\|x - a\| : a \in A\} \quad (\text{B.2})$$

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

**Definición B.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.  $\text{dist}(D_t, \mathcal{C}) \rightarrow 0$  (a.s.) sin importar la elección del oponente  $-i$ .

**Definición B.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

$$w_{\mathcal{C}}(\lambda) = \sup\{\lambda \cdot c : c \in \mathcal{C}\} \quad (\text{B.3})$$

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

Dado un conjunto convexo y cerrado  $\mathcal{C}$  se denotará con  $F(x)$  el punto (único) más cercano a  $x$  de  $\mathcal{C}$ , 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 B.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  $s_{-i} \in S_{-i}$ :

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

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  $\text{dist}(D_t, \mathcal{C}) \rightarrow 0$  (a.s.) cuando  $t \rightarrow \infty$ : en el tiempo  $t+1$ , jugar  $q_{\lambda(D_t)}$  si  $D_t \notin \mathcal{C}$ , y jugar arbitrariamente si  $D_t \in \mathcal{C}$ .



# APÉNDICE C

## ESTRATEGIAS MINIMAX Y MAXIMIN

Una estrategia *minimax* del jugador  $i$ , consiste en minimizar la ganancia de la mejor respuesta del jugador  $-i$ . Es decir, el jugador  $i$  juega para “castigar” al jugador  $-i$ , sin tomar en cuenta su propia ganancia. Por otra parte en una estrategia *maximin*, el jugador busca maximizar su ganancia, suponiendo que su oponente juega para perjudicarlo.

**Definición C.1** ([12, p. 15–16]). *El conjunto de estrategias minimax para el jugador  $i$  en contra del jugador  $-i$  es*

$$\{\sigma_i : \max_{\sigma_{-i}} u_{-i}(\sigma_i, \sigma_{-i}) = \arg\min_{\sigma'_i} \max_{\sigma_{-i}} u_{-i}(\sigma'_i, \sigma_{-i})\}, \quad (\text{C.1})$$

*y el valor minimax del jugador  $-i$  es  $\min_{\sigma_i} \max_{\sigma_{-i}} u_{-i}(\sigma_i, \sigma_{-i})$ . El conjunto de estrategias maximin para el jugador  $i$  en contra del jugador  $-i$  es*

$$\{\sigma_i : \min_{\sigma_{-i}} u_i(\sigma_i, \sigma_{-i}) = \arg\max_{\sigma'_i} \min_{\sigma_{-i}} u_i(\sigma'_i, \sigma_{-i})\}, \quad (\text{C.2})$$

*y el valor maximin del jugador  $i$  es  $\max_{\sigma_i} \min_{\sigma_{-i}} u_i(\sigma_i, \sigma_{-i})$ .*

Como la estrategia *minimax* o *maximin* de un jugador no depende de la estrategia del oponente, se pueden definir perfiles estratégicos *minimax* y *maximin*. Un perfil estratégico mixto  $\sigma = (\sigma_1, \sigma_2)$  es un perfil estratégico *minimax* (*maximin*) si  $\sigma_1$  es un estrategia minimax (resp. maximin) para el jugador 1 y  $\sigma_2$  es una estrategia *minimax* (resp. maximin) para el jugador 2.

**Ejemplo C.2.** *Considere el juego en forma normal de 2 jugadores mostrado en la Tabla C.1 donde  $S_1 = S_2 = \{1, 2\}$ .*

Calculemos estrategias *minimax* y *maximin* para el primer jugador. Las estrategias

Tabla C.1: Tabla de pagos del juego del Ejemplo C.2.

|   | 1      | 2      |
|---|--------|--------|
| 1 | 2, 4   | -1, -2 |
| 2 | -1, -1 | 2, 2   |

*minimax* del primer jugador vienen expresadas por:

$$\operatorname{argmín}_{(\beta_1, \beta_2) \in \Delta_2} \max_{(\theta_1, \theta_2) \in \Delta_2} \theta_1(4\beta_1 - \beta_2) + \theta_2(-2\beta_1 + 2\beta_2). \quad (\text{C.3})$$

Note que si  $4\beta_1 - \beta_2 = x$  y  $-2\beta_1 + 2\beta_2 = y$ , entonces  $\max_{(\theta_1, \theta_2) \in \Delta_2} \theta_1(4\beta_1 - \beta_2) + \theta_2(-2\beta_1 + 2\beta_2) = \theta_1 x + \theta_2 y = \max\{x, y\}$ , el cual se alcanza cuando el coeficiente del valor mayor es igual a 1 y el coeficiente del menor valor es igual a 0. Luego, la solución es obtenida cuando  $x = y$ , pues en este caso la ganancia del segundo jugador no depende de las elecciones de  $\theta_1$  y  $\theta_2$ .

Esto último ocurre cuando  $4\beta_1 - \beta_2 = -2\beta_1 + 2\beta_2$ , lo que implica que  $(\beta_1, \beta_2) = (\frac{1}{3}, \frac{2}{3})$ . El valor *minimax* del segundo jugador es entonces  $\frac{2}{3}$ . En este caso, el primer jugador elige su estrategia considerando, únicamente, la ganancia del oponente, sin tomar en cuenta su propia ganancia.

Por otra parte, las estrategias *maximin* se corresponden con

$$\operatorname{argmáx}_{(\beta_1, \beta_2) \in \Delta_2} \min_{(\theta_1, \theta_2) \in \Delta_2} \theta_1(2\beta_1 - \beta_2) + \theta_2(-\beta_1 + 2\beta_2). \quad (\text{C.4})$$

La estrategia *maximin* es alcanzada cuando la ganancia esperada del primer jugador no depende de la elección de  $\theta_1$  y  $\theta_2$ , es decir cuando  $2\beta_1 - \beta_2 = -\beta_1 + 2\beta_2$ , lo que ocurre si y sólo si  $(\beta_1, \beta_2) = (\frac{1}{2}, \frac{1}{2})$ . El valor *maximin* del primer jugador es  $\frac{1}{2}$ .

**Teorema C.3** ([2]). *Para cualquier juego finito para dos jugadores de suma cero, y para cualquier equilibrio de Nash del juego, cada jugador tiene una ganancia esperada cuyo valor es igual al valor minimax y valor maximin de dicho jugador.*

El Teorema C.3 muestra que las estrategias *minimax*, *maximin* y los equilibrios de Nash coinciden en los juegos de dos jugadores con suma cero.

# APÉNDICE D

## FORMA NORMAL Y PROGRAMACIÓN LINEAL

En el Capítulo IV se afirma que para todo juego de dos jugadores de suma cero en forma normal, encontrar el equilibrio de Nash es equivalente a un problema de programación lineal. En esta sección se detalla cómo obtener el programa de programación lineal a un juego dado en forma normal [14, pp. 228-233].

Todo juego en forma normal puede ser descrito por una tabla  $N$  dimensional, donde  $N$  es el número de jugadores. En particular un juego de dos jugadores puede ser representado por una matriz, y además, si el juego es de suma 0, basta con definir la utilidad del primer jugador. Luego, este tipo de juegos pueden ser representados con una matriz  $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{m \times n}$ . Luego, el elemento  $a_{ij}$  de la matriz  $A$  es la utilidad obtenida por el primer jugador si éste utiliza la  $i$ -ésima estrategia y su oponente utiliza la  $j$ -ésima estrategia.

Si el jugador 1 juega con una estrategia  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  y el jugador 2 con una estrategia  $\mathbf{y} = (y_1, y_2, \dots, y_n)$ , entonces la ganancia esperada viene dada por

$$u_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j, \quad (\text{D.1})$$

lo cual se puede escribir matricialmente como  $\mathbf{x}^\top \mathbf{A} \mathbf{y}$ .

Del Teorema C.3 se sabe que el valor *maximin* y *minimax* es igual al valor del juego, al igual que los perfiles estratégicos *maximin* y *minimax* coinciden con equilibrios de Nash. Luego, un equilibrio de Nash  $(x^*, y^*)$  cumple que:

$$x^* \in \arg\max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^\top \mathbf{A} \mathbf{y} \quad (\text{D.2})$$

$$y^* \in \arg\min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{y}. \quad (\text{D.3})$$

Ya que la ecuación D.2 es la definición de estrategia *maximin* para el primer jugador y

la ecuación D.3 es la definición *minimax* para el segundo jugador. La primera observación importante proviene del Teorema 1.9, ya que como corolario se obtiene que, para cualquier estrategia, siempre existe una mejor respuesta cuyo soporte tiene un único elemento. De esto último se obtiene que, dado  $\mathbf{x} \in \mathbb{R}^m$  e  $\mathbf{y} \in \mathbb{R}^n$ :

$$\min_{\mathbf{y}} \mathbf{x}^T \mathbf{A} \mathbf{y} = \min_j \sum_{i=1}^n a_{ij} x_i \quad (\text{D.4})$$

$$\max_{\mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{y} = \max_i \sum_{j=1}^m a_{ij} y_j \quad (\text{D.5})$$

Luego, los problemas que se quieren resolver son equivalentes a:

$$\max_x \min_j \sum_{i=1}^n a_{ij} x_i \quad (\text{D.6})$$

$$\min_y \max_i \sum_{j=1}^m a_{ij} y_j \quad (\text{D.7})$$

Sujetos a  $x_i, y_j \geq 0$  y a  $\sum_{i=1}^m x_i = 1$  y  $\sum_{j=1}^n y_j = 1$ . La observación clave consiste en ver la equivalencia de los problemas D.6 y D.7 con los problemas de programación lineal D.8 y D.9, respectivamente [14, p. 232].

$$\max \quad z \quad (\text{D.8})$$

sujeto a

$$z - \sum_{i=1}^m a_{ij} x_i \leq 0 \quad (j = 1, 2, \dots, n)$$

$$\sum_{i=1}^m x_i = 1$$

$$x_i \geq 0 \quad (i = 1, 2, \dots, m)$$

$$\min \quad w \quad (\text{D.9})$$

sujeto a

$$w - \sum_{j=1}^n a_{ij} y_j \geq 0 \quad (i = 1, 2, \dots, m)$$

$$\sum_{j=1}^n y_j = 1$$

$$y_j \geq 0 \quad (j = 1, 2, \dots, n).$$

Para ver la equivalencia, note que cualquier solución óptima  $z^*, x_1^*, x_2^*, \dots, x_m^*$  de D.8 satisface al menos una restricción con el signo de igualdad y por lo tanto  $z^* = \min_j \sum_i a_{ij} x_i^*$ . De forma similar, cualquier solución óptima  $w^*, y_1^*, y_2^*, \dots, y_n^*$  de D.9 satisface al menos una restricción con el signo de igualdad y por lo tanto  $w^* = \min_i \sum_j a_{ij} y_j^*$ . Por último, note que D.8 y D.9 son problemas duales entre sí, lo cual confirma que  $z^* = w^*$ .

Con respecto a los juego en forma extensiva, si bien se podrían representar en forma normal para resolverlos con programación lineal, esto no es práctico, pues la forma normal es exponencialmente más grande que la forma extensiva. En [21] se propone un modelo para calcular un equilibrio de Nash mediante programación lineal con una complejidad polinómica respecto al número de nodos del árbol, esto fue el estado del arte hasta que se desarrolló el algoritmo *Counterfactual Regret Minimization* (CFR).

A continuación, se presentan los problemas de programación lineal asociados a los juegos *matching pennies*, piedra, papel o tijera y ficha vs. dominó; presentados en el Capítulo IV, con una solución primal y dual.

#### ■ *Matching Pennies*

$$\text{máx } z \tag{D.10}$$

sujeto a

$$\begin{aligned} x_1 + x_2 &= 1 \\ z - x_1 + x_2 &\leq 0 \\ z + x_1 - x_2 &\leq 0 \\ x_1, \quad x_2 &\geq 0. \end{aligned}$$

La única solución es:

$$(z^*, x_1^*, x_2^*) = (w^*, y_1^*, y_2^*) = \left(0, \frac{1}{2}, \frac{1}{2}\right). \tag{D.11}$$

#### ■ *Piedra, papel o tijera*

$$\text{máx } z \tag{D.12}$$

sujeto a

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ z + x_2 - x_3 &\leq 0 \\ z - x_1 + x_3 &\leq 0 \end{aligned}$$

$$\begin{aligned} z + x_1 - x_2 &\leq 0 \\ x_1, \quad x_2, \quad x_3 &\geq 0. \end{aligned}$$

La única solución es:

$$(z^*, x_1^*, x_2^*, x_4^*, x_5^*, x_6^*, x_7^*) = \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0, \frac{1}{3} \right), \quad (\text{D.13})$$

$$(w^*, y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*) = \left( \frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3}, 0, \frac{1}{3}, 0 \right). \quad (\text{D.14})$$

### ■ Ficha vs. dominó

$$\begin{aligned} &\text{máx } z \\ &\text{sujeto a} \end{aligned} \quad (\text{D.15})$$

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 &= 1 \\ z + x_1 - x_2 - x_3 - x_4 + x_5 - x_6 - x_7 &\leq 0 \\ z + x_1 - x_2 + x_3 - x_4 - x_5 + x_6 - x_7 &\leq 0 \\ z - x_1 - x_2 + x_3 - x_4 - x_5 - x_6 + x_7 &\leq 0 \\ z - x_1 + x_2 - x_3 - x_4 + x_5 - x_6 - x_7 &\leq 0 \\ z - x_1 + x_2 - x_3 + x_4 - x_5 + x_6 - x_7 &\leq 0 \\ z - x_1 - x_2 - x_3 + x_4 - x_5 - x_6 + x_7 &\leq 0 \\ x_1, \quad x_2, \quad x_3, \quad x_4, \quad x_5, \quad x_6, \quad x_7, &\geq 0. \end{aligned}$$

Una de las soluciones es:

$$(z^*, x_1^*, x_2^*, x_4^*, x_5^*, x_6^*, x_7^*) = \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0, \frac{1}{3} \right), \quad (\text{D.16})$$

$$(w^*, y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*) = \left( \frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3}, 0, \frac{1}{3}, 0 \right). \quad (\text{D.17})$$

# APÉNDICE E

## ALGORITMOS

### Chance-sampled Counterfactual Regret Minimization (CFR)

En esta sección se presenta el algoritmo *chance-sampled* como es descrito en [8]. Cada iteración se realiza mediante búsqueda en profundidad, seleccionando una única acción en un nodo de azar de acuerdo a su distribución de probabilidad correspondiente. En cada conjunto de información visitado en una iteración de entrenamiento, una estrategia mixta es calculada de acuerdo a la Ecuación 5.10 (Algoritmo 2). La estrategia promedio  $\sigma^T$ , en cada conjunto de información  $I$ , aproxima a un equilibrio de Nash cuando  $T \rightarrow \infty$  (Algoritmo 1).

---

**Algoritmo 1** Entrenamiento de *chance-sampled* CFR

---

```
1: Inicializar tablas acumulativas de regret:  $\forall I, r_I[a] \leftarrow 0$ .
2: Inicializar tablas acumulativas de estrategias:  $\forall I, s_I[a] \leftarrow 0$ .
3: Inicializar perfil inicial:  $\sigma^1(I, a) \leftarrow 1/|A(I)|$ 
4:
5: function SOLVE
6:   for  $t = 1, 2, \dots, T$  do
7:     for  $i \in \{1, 2\}$  do
8:       CFR( $\emptyset, i, t, 1, 1$ )
9:     end for
10:  end for
11: end function
12: Calcular estrategia promedio  $\bar{\sigma}^T$  de las estrategias  $\sigma^1, \sigma^2, \dots, \sigma^T$ .
```

---

---

**Algoritmo 2** *Counterfactual Regret Minimization (CFR) con chance-sampled*


---

```

1: function CFR( $h, i, t, \pi_1, \pi_2$ )
2:   if  $h$  es terminal then
3:     return  $u_i(h)$ 
4:   else if  $h$  es un nodo de azar then
5:     Seleccionar una acción  $a \sim f_c(h)$ 
6:     return CFR( $ha, i, t, \pi_1, \pi_2$ )
7:   end if
8:   Sea  $I$  el conjunto de información que contiene a  $h$ .
9:    $v_\sigma \leftarrow 0$ 
10:   $v_{\sigma_{I \rightarrow [a]}} \leftarrow 0$  para todo  $a \in A(I)$ 
11:  for  $a \in A(I)$  do
12:    if  $P(h) = 1$  then
13:       $v_{\sigma_{I \rightarrow [a]}} \leftarrow \text{CFR}(ha, i, t, \sigma^t(I, a) \cdot \pi_1, \pi_2)$ 
14:    else if  $P(h) = 2$  then
15:       $v_{\sigma_{I \rightarrow [a]}} \leftarrow \text{CFR}(ha, i, t, \pi_1, \sigma^t(I, a) \cdot \pi_2)$ 
16:    end if
17:     $v_\sigma \leftarrow v_\sigma + \sigma^t(I, a) \cdot v_{\sigma_{I \rightarrow [a]}}$ 
18:  end for
19:  if  $P(h) = i$  then
20:    for  $a \in A(I)$  do
21:       $r_I[a] \leftarrow r_I[a] + \pi_{-i} \cdot (v_{\sigma_{I \rightarrow [a]}} - v_\sigma)$ 
22:       $s_I[a] \leftarrow s_I[a] + \pi_i \cdot \sigma^t(I, a)$ 
23:    end for
24:     $\sigma^{t+1}(I) \leftarrow$  estrategia calculada con la Ecuación 5.10 y la tabla de regret  $r_I$ 
25:  end if
26: end function

```

---



## Generilized Expectimax Best Response

En esta sección se presenta el algoritmo *Generilized Expectimax Best Response* (GEBR) (Algoritmos 4, 5 y 6) utilizado para obtener la explotabilidad de una estrategia (Algoritmo 3).

En al algoritmo GEBR (Algoritmo 4) se tiene un jugador  $i$  para el cual se calculará la mejor respuesta  $\sigma_i^*$  ante una estrategia fija  $\sigma_{-i}$  del jugador  $-i$ . Este algoritmo tiene 3 partes, primero se recorre el árbol del juego mediante búsqueda por profundidad, para determinar las profundidades de los conjuntos de información por cada uno de los jugadores, esto es el Algoritmo 5. Estas listas se ordenan de forma decreciente.

La segunda parte del algoritmo GEBR consiste en recorrer el árbol varias veces, una vez por cada profundidad diferente, de mayor a menor, como se presenta en el Algoritmo 6. En el recorrido a profundidad  $d$ , se calculan los valores  $t_I[a]$  y  $b_I[a]$  para todos los conjuntos de información  $I$  a una profundidad  $d$  y toda acción  $a \in A(I)$ .

Estos arreglos permiten calcular la utilidad contrafactual (Definición 5.3) en los conjuntos de información  $I$ . En efecto, note que  $t_I[a] = u_i((\sigma_i^*|_{I \rightarrow a}, \sigma_{-i}), I) \cdot \pi^{\sigma_{-i}}(I)$  y  $b_I[a] = \pi^{\sigma_{-i}}(I)$ . Observe que  $\sigma_i^*$  se obtiene al tomar alguna acción  $a \in \arg\max_{a \in A(I)} \frac{t[a]}{b[a]}$  (se utiliza una estrategia pura como mejor respuesta por lo visto en el Capítulo III). Luego, durante el recorrido hecho para la profundidad  $d$ , ya se conoce el valor  $\sigma_i^*(I')$  para todos los  $I'$  a una profundidad  $d' > d$ , por lo que es posible calcular  $u_i((\sigma_i^*|_{I \rightarrow a}, \sigma_{-i}), I)$ .

La última parte del algoritmo consiste en calcular el valor esperado  $u_i(\sigma_i^*, \sigma_{-i})$ , esto se puede obtener al utilizar el Algoritmo 6 con  $d = -1$ . Finalmente, la explotabilidad de la estrategia  $\sigma$  se obtiene al sumar las ganancias esperadas de  $u_1(\sigma_1, \sigma_2^*)$  y  $u_2(\sigma_1^*, \sigma_2)$  (Algoritmo 3).

La complejidad de este algoritmo es  $\mathcal{O}(\mathcal{ND})$  donde  $N$  es el número de nodos del árbol y  $D$  es su profundidad. Debido a la alta complejidad asintótica, se utilizó este algoritmo únicamente para calcular la explotabilidad de la estrategia final.

---

**Algoritmo 3** Explotabilidad
 

---

```

1: Inicializar el conjunto de profundidades del jugador  $i$ 
2: Inicializar las tablas de los valores esperados:  $\forall I, t_I[a] \leftarrow 0$ 
3: Inicializar las tablas de las probabilidades de alcance:  $\forall I, b_I[a] \leftarrow 0$ 
4: Inicializar  $\sigma$  con la estrategia para la cual se desea calcular la explotabilidad
5:
6: function EXPLOTABILITY
7:   return GEBR(1) + GEBR(2)
8: end function

```

---



---

**Algoritmo 4** Generilized Expectimax Best Response (GEBR)
 

---

```

1: function GEBR( $i$ )
2:   GEBR-Pass1( $\emptyset, i, 0$ )
3:   Ordenar las profundidades en orden decreciente
4:   for  $d$  en el conjunto de profundidades del jugador  $i$  do
5:     GEBR-Pass2( $\emptyset, i, d, 0, 1$ )
6:   end for
7:   return GEBR-Pass2( $\emptyset, i, -1, 0, 1$ )
8: end function

```

---



---

**Algoritmo 5** Generilized Expectimax Best Response (GEBR): primer recorrido
 

---

```

1: function GEBR-PASS1( $h, i, d$ )
2:   if  $h$  es un nodo terminal then
3:     return
4:   end if
5:   if  $h$  no es un nodo de azar then
6:     Agregar  $d$  al conjunto de profundidades del jugador  $i$ 
7:   end if
8:   for  $a \in A(h)$  do
9:     GEBR-Pass1( $ha, i, d + 1$ )
10:  end for
11: end function

```

---

---

**Algoritmo 6** Generalized Expectimax Best Response (GEBR): segundos recorridos
 

---

```

1: function GEBR-Pass2( $h, i, d, l, \pi_{-i}$ )
2:   if  $h$  es un nodo terminal then
3:     return  $u_i(h)$ 
4:   else if  $h$  es un nodo de azar then
5:     return  $\sum_{a \in A(h)} f_c(a|h) \cdot \text{GEBR-Pass2}(ha, i, d, l+1, \pi_{-i} \cdot f_c(a|h))$ 
6:   end if
7:   Sea  $I$  el conjunto de información que contiene a  $h$ 
8:    $v \leftarrow 0$ 
9:   if  $P(I) = i$  and  $l > d$  then
10:     $a \leftarrow \operatorname{argm\acute{a}x}_{a \in A(I)} \frac{t[a]}{b[a]}$ 
11:    return  $\text{GEBR-Pass2}(ha, i, d, l+1, \pi_{-i})$ 
12:  end if
13:  for  $a \in A(I)$  do
14:     $\pi'_{-i} \leftarrow \pi_{-i}$ 
15:    if  $P(I) = -i$  then
16:       $\pi'_{-i} \leftarrow \pi_{-i} \cdot \sigma(I, a)$ 
17:    end if
18:     $v' \leftarrow \text{GEBR-Pass2}(ha, i, d, l+1, \pi'_{-i})$ 
19:    if  $P(I) = -i$  then
20:       $v \leftarrow v + \sigma(I, a) \cdot v'$ 
21:    else if  $P(I) = i$  and  $l = d$  then
22:       $t_I[a] \leftarrow t_I[a] + v' \cdot \pi_{-i}$ 
23:       $b_I[a] \leftarrow b_I[a] + \pi_{-i}$ 
24:    end if
25:  end for
26:  return  $v$ 
27: end function

```

---

# APÉNDICE F

## REGRET MATCHING

En este apéndice se presentan las tablas detalladas para los juegos en forma normal descritos en el capítulo IV. 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 (EN). 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  $\varepsilon_\sigma$  (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.

### 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 estrategia. 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 F.1. Por lo que todas las estrategias obtenidas son un  $\varepsilon$ -equilibrio de Nash, con  $\varepsilon < 0,008$ .

La Tabla F.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 3.892.550,4; obteniendo un promedio de  $2,64 \times 10^{-6}$  segundos por iteración. Con el procedimiento B, que

Tabla F.1: Estrategias obtenidas en el juego *matching pennies*.

|                      | EN            | A             | B             | C             |
|----------------------|---------------|---------------|---------------|---------------|
| $\sigma_1$           | (0,500 0,500) | (0,500 0,500) | (0,500 0,500) | (0,500 0,500) |
| $\sigma_2$           | (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) |
| $\varepsilon_\sigma$ | 0             | 0,006         | 0,006         | 0,008         |

utiliza un vector invariante de probabilidad, se obtuvo un tiempo, número de iteraciones y tiempo por iteración promedios de 3,777 segundos, 25.616,6 iteraciones y  $3,03 \times 10^{-5}$  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 16.260,5; obteniendo un promedio de  $2,58 \times 10^{-6}$  segundos por iteración.

Tabla F.2: Resultados experimentales del juego *matching pennies*.

| A      |             |                       | B     |          |                       | C     |          |                       |
|--------|-------------|-----------------------|-------|----------|-----------------------|-------|----------|-----------------------|
| $T$    | $I$         | $T/I$                 | $T$   | $I$      | $T/I$                 | $T$   | $I$      | $T/I$                 |
| 7,663  | 3.068.341   | $2,50 \times 10^{-6}$ | 0,985 | 32.510   | $3,03 \times 10^{-5}$ | 0,002 | 955      | $2,53 \times 10^{-6}$ |
| 9,650  | 3.857.071   | $2,50 \times 10^{-6}$ | 1,748 | 56.946   | $3,07 \times 10^{-5}$ | 0,064 | 24.968   | $2,55 \times 10^{-6}$ |
| 23,313 | 8.950.013   | $2,60 \times 10^{-6}$ | 0,552 | 18.401   | $3,00 \times 10^{-5}$ | 0,061 | 23.854   | $2,57 \times 10^{-6}$ |
| 11,757 | 4.240.611   | $2,77 \times 10^{-6}$ | 0,309 | 10.197   | $3,03 \times 10^{-5}$ | 0,025 | 9.724    | $2,57 \times 10^{-6}$ |
| 2,377  | 877.335     | $2,71 \times 10^{-6}$ | 0,747 | 24.892   | $3,00 \times 10^{-5}$ | 0,011 | 4.188    | $2,59 \times 10^{-6}$ |
| 5,062  | 1.818.992   | $2,78 \times 10^{-6}$ | 0,848 | 28.142   | $3,01 \times 10^{-5}$ | 0,025 | 9.666    | $2,60 \times 10^{-6}$ |
| 4,281  | 1.557.496   | $2,75 \times 10^{-6}$ | 0,132 | 4.405    | $3,01 \times 10^{-5}$ | 0,045 | 16.951   | $2,64 \times 10^{-6}$ |
| 22,110 | 8.230.100   | $2,69 \times 10^{-6}$ | 1,307 | 43.116   | $3,03 \times 10^{-5}$ | 0,021 | 8.155    | $2,64 \times 10^{-6}$ |
| 3,691  | 1.432.846   | $2,58 \times 10^{-6}$ | 0,639 | 21.311   | $3,00 \times 10^{-5}$ | 0,093 | 35.270   | $2,64 \times 10^{-6}$ |
| 12,853 | 4.892.699   | $2,63 \times 10^{-6}$ | 0,500 | 16.246   | $3,08 \times 10^{-5}$ | 0,076 | 28.874   | $2,64 \times 10^{-6}$ |
| 10,276 | 3.892.550,4 | $2,64 \times 10^{-6}$ | 0,777 | 25.616,6 | $3,03 \times 10^{-5}$ | 0,042 | 16.260,5 | $2,58 \times 10^{-6}$ |

## Piedra, Papel o Tijera

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 F.3. No todas corresponden al equilibrio de Nash exacto, sin embargo, cada una de ellas es un  $\varepsilon$ -equilibrio de Nash con  $\varepsilon < 0,01$ .

La Tabla F.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 4.519.054,1,

Tabla F.3: Estrategias obtenidas del juego piedra, papel o tijera.

| Estrategias |            |                     | $v_1/v_2$ | $\varepsilon_\sigma$ |
|-------------|------------|---------------------|-----------|----------------------|
| EN          | $\sigma_1$ | (0,333 0,333 0,333) | 0,000     | 0,000                |
|             | $\sigma_2$ | (0,333 0,333 0,333) | 0,000     |                      |
| A           | $\sigma_1$ | (0,332 0,335 0,332) | 0,003     | 0,006                |
|             | $\sigma_2$ | (0,331 0,334 0,335) | 0,003     |                      |
| B           | $\sigma_1$ | (0,330 0,334 0,336) | 0,006     | 0,010                |
|             | $\sigma_2$ | (0,329 0,335 0,337) | 0,004     |                      |
| C           | $\sigma_1$ | (0,333 0,337 0,330) | 0,005     | 0,009                |
|             | $\sigma_2$ | (0,336 0,330 0,335) | 0,004     |                      |

obteniendo un promedio de  $2,7 \times 10^{-6}$  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, 6.601,3 iteraciones y  $5,23 \times 10^{-5}$  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 19.321,1, obteniendo un promedio de  $2,54 \times 10^{-6}$  segundos por iteración.

Tabla F.4: Resultados experimentales del juego piedra, papel o tijera.

| A      |             |                       | B     |         |                       | C     |          |                       |
|--------|-------------|-----------------------|-------|---------|-----------------------|-------|----------|-----------------------|
| $T$    | $I$         | $T/I$                 | $T$   | $I$     | $T/I$                 | $T$   | $I$      | $T/I$                 |
| 25,715 | 9.107.389   | $2,82 \times 10^{-6}$ | 0,724 | 13.750  | $5,26 \times 10^{-5}$ | 0,034 | 12.967   | $2,64 \times 10^{-6}$ |
| 29,494 | 10.951.479  | $2,69 \times 10^{-6}$ | 0,692 | 13.257  | $5,22 \times 10^{-5}$ | 0,041 | 16.096   | $2,57 \times 10^{-6}$ |
| 7,015  | 2.641.656   | $2,66 \times 10^{-6}$ | 0,000 | 6       | $4,36 \times 10^{-5}$ | 0,063 | 24.423   | $2,56 \times 10^{-6}$ |
| 4,610  | 1.748.365   | $2,64 \times 10^{-6}$ | 0,849 | 16.255  | $5,22 \times 10^{-5}$ | 0,048 | 18.613   | $2,56 \times 10^{-6}$ |
| 8,051  | 3.033.028   | $2,65 \times 10^{-6}$ | 0,000 | 3       | $4,28 \times 10^{-5}$ | 0,082 | 32.222   | $2,55 \times 10^{-6}$ |
| 9,870  | 3.717.278   | $2,66 \times 10^{-6}$ | 0,000 | 3       | $4,28 \times 10^{-5}$ | 0,084 | 33.042   | $2,54 \times 10^{-6}$ |
| 2,749  | 1.037.895   | $2,65 \times 10^{-6}$ | 0,000 | 3       | $4,06 \times 10^{-5}$ | 0,049 | 19.316   | $2,55 \times 10^{-6}$ |
| 11,971 | 4.517.546   | $2,65 \times 10^{-6}$ | 0,556 | 10.644  | $5,23 \times 10^{-5}$ | 0,024 | 9.601    | $2,54 \times 10^{-6}$ |
| 14,974 | 5.606.070   | $2,67 \times 10^{-6}$ | 0,000 | 3       | $3,74 \times 10^{-5}$ | 0,014 | 5.621    | $2,55 \times 10^{-6}$ |
| 7,532  | 2.829.835   | $2,66 \times 10^{-6}$ | 0,631 | 12.089  | $5,22 \times 10^{-5}$ | 0,054 | 21.310   | $2,55 \times 10^{-6}$ |
| 12,198 | 4.519.054,1 | $2,70 \times 10^{-6}$ | 0,345 | 6.601,3 | $5,23 \times 10^{-5}$ | 0,049 | 19.321,1 | $2,54 \times 10^{-6}$ |

## 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

la Tabla F.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.

Tabla F.5: Estrategias obtenidas del juego ficha vs dominó.

|    |            | Estrategias                                 |  |  |  |  |  | $v_1/v_2$ | $\varepsilon_\sigma$ |
|----|------------|---|--|--|--|--|--|-----------|----------------------|
| EN | $\sigma_1$ | (0,333 0,333 0,000 0,000 0,000 0,000 0,333) |  |  |  |  |  | 0,333     | 0,000                |
|    | $\sigma_2$ | (0,333 0,000 0,333 0,000 0,333 0,000)       |  |  |  |  |  | -0,333    |                      |
| A  | $\sigma_1$ | (0,136 0,137 0,116 0,118 0,198 0,081 0,214) |  |  |  |  |  | 0,338     | 0,010                |
|    | $\sigma_2$ | (0,165 0,171 0,163 0,166 0,166 0,169)       |  |  |  |  |  | -0,328    |                      |
| B  | $\sigma_1$ | (0,121 0,118 0,135 0,137 0,214 0,078 0,198) |  |  |  |  |  | 0,335     | 0,007                |
|    | $\sigma_2$ | (0,157 0,178 0,156 0,177 0,157 0,175)       |  |  |  |  |  | -0,331    |                      |
| C  | $\sigma_1$ | (0,128 0,128 0,129 0,134 0,208 0,073 0,202) |  |  |  |  |  | 0,334     | 0,004                |
|    | $\sigma_2$ | (0,169 0,165 0,168 0,164 0,169 0,165)       |  |  |  |  |  | -0,330    |                      |

La Tabla F.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 108.319.272,4, obteniendo un promedio de  $2,95 \times 10^{-6}$  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, 75.250,2 iteraciones y  $1,5 \times 10^{-4}$  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 84.318,5, obteniendo un promedio de  $2,81 \times 10^{-6}$  segundos por iteración.

## 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 F.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 0,010.

Tabla F.6: Resultados experimentales del juego ficha vs. dominó.

| A       |               |                       | B      |          |                       | C     |          |                       |
|---------|---------------|-----------------------|--------|----------|-----------------------|-------|----------|-----------------------|
| $T$     | $I$           | $T/I$                 | $T$    | $I$      | $T/I$                 | $T$   | $I$      | $T/I$                 |
| 669,839 | 215.859.538   | $3,10 \times 10^{-6}$ | 4,458  | 29.721   | $1,50 \times 10^{-4}$ | 0,188 | 66.700   | $2,81 \times 10^{-6}$ |
| 309,685 | 117.568.373   | $2,63 \times 10^{-6}$ | 9,019  | 60.333   | $1,49 \times 10^{-4}$ | 0,260 | 92.401   | $2,82 \times 10^{-6}$ |
| 399,170 | 152.612.646   | $2,62 \times 10^{-6}$ | 3,646  | 24.338   | $1,50 \times 10^{-4}$ | 0,212 | 75.674   | $2,81 \times 10^{-6}$ |
| 131,570 | 38.097.125    | $3,45 \times 10^{-6}$ | 12,996 | 86.898   | $1,50 \times 10^{-4}$ | 0,145 | 51.776   | $2,80 \times 10^{-6}$ |
| 263,482 | 96.741.015    | $2,72 \times 10^{-6}$ | 4,516  | 30.170   | $1,50 \times 10^{-4}$ | 0,134 | 47.862   | $2,80 \times 10^{-6}$ |
| 203,854 | 77.156.602    | $2,64 \times 10^{-6}$ | 15,420 | 103.021  | $1,50 \times 10^{-4}$ | 0,385 | 136.950  | $2,81 \times 10^{-6}$ |
| 201,267 | 76.467.409    | $2,63 \times 10^{-6}$ | 17,399 | 115.935  | $1,50 \times 10^{-4}$ | 0,351 | 124.882  | $2,81 \times 10^{-6}$ |
| 316,007 | 97.849.871    | $3,23 \times 10^{-6}$ | 17,266 | 115.056  | $1,50 \times 10^{-4}$ | 0,203 | 72.315   | $2,81 \times 10^{-6}$ |
| 383,736 | 110.341.861   | $3,48 \times 10^{-6}$ | 12,805 | 85.532   | $1,50 \times 10^{-4}$ | 0,271 | 96.438   | $2,81 \times 10^{-6}$ |
| 313,177 | 100.498.284   | $3,12 \times 10^{-6}$ | 15,227 | 101.498  | $1,50 \times 10^{-4}$ | 0,220 | 78.187   | $2,81 \times 10^{-6}$ |
| 319,179 | 108.319.272,4 | $2,95 \times 10^{-6}$ | 11,275 | 75.250,2 | $1,50 \times 10^{-4}$ | 0,237 | 84.318,5 | $2,81 \times 10^{-6}$ |

Tabla F.7: Estrategias obtenidas del juego coronel Blotto.

| 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)                             |
| $(v_1 \ v_2) = (0,002 \ 0,008)$   |
| $\varepsilon_\sigma = 0,010$  |
| Procedimiento B   |
| (0 0,001 0,134 0,093 0 0 0,001 0,085 0,001 0,106 0 0,110 0,002 0 0,117 0,150 0,065 0,132 0,001 0,001 0)         |
| (0 0,002 0,101 0,173 0,001 0 0,001 0,073 0,001 0,065 0,001 0,163 0,002 0,001 0,109 0,068 0,109 0,130 0,001 0 0) |
| $(v_1 \ v_2) = (0,006 \ 0,004)$   |
| $\varepsilon_\sigma = 0,010$  |
| 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)                                 |
| $(v_1 \ v_2) = (0,004 \ 0,005)$   |
| $\varepsilon_\sigma = 0,009$  |

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

El procedimiento A, regret condicional, tuvo una duración promedio de 875,533 segundos, con un número promedio de iteraciones de 190.222.305,3, obteniendo un promedio de  $4,60 \times 10^{-6}$  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 70,453 segundos, 58.394,4 iteraciones y  $1,2 \times 10^{-3}$  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 48.613,5, obteniendo un promedio de  $3,41 \times 10^{-6}$  segundos por iteración.



Tabla F.8: Resultados experimentales del juego coronel Blotto.

| A         |               |                        | B      |          |                        | C     |          |                        |
|-----------|---------------|------------------------|--------|----------|------------------------|-------|----------|------------------------|
| $T$       | $I$           | $T/I$                  | $T$    | $I$      | $T/I$                  | $T$   | $I$      | $T/I$                  |
| 940,377   | 197.127.165   | $4,77 \times 10^{-06}$ | 84,017 | 70.075   | $1,20 \times 10^{-03}$ | 0,047 | 13.559   | $3,50 \times 10^{-06}$ |
| 532,020   | 109.697.363   | $4,85 \times 10^{-06}$ | 95,841 | 79.849   | $1,20 \times 10^{-03}$ | 0,192 | 56.383   | $3,41 \times 10^{-06}$ |
| 396,583   | 82.924.728    | $4,78 \times 10^{-06}$ | 46,773 | 39.008   | $1,20 \times 10^{-03}$ | 0,046 | 13.664   | $3,39 \times 10^{-06}$ |
| 362,203   | 80.521.418    | $4,50 \times 10^{-06}$ | 53,621 | 44.774   | $1,20 \times 10^{-03}$ | 0,162 | 47.742   | $3,40 \times 10^{-06}$ |
| 967,890   | 207.963.652   | $4,65 \times 10^{-06}$ | 83,351 | 69.475   | $1,20 \times 10^{-03}$ | 0,090 | 26.547   | $3,40 \times 10^{-06}$ |
| 1.016,540 | 245.737.655   | $4,14 \times 10^{-06}$ | 63,440 | 52.865   | $1,20 \times 10^{-03}$ | 0,118 | 34.715   | $3,41 \times 10^{-06}$ |
| 553,971   | 112.170.109   | $4,94 \times 10^{-06}$ | 67,121 | 56.101   | $1,20 \times 10^{-03}$ | 0,261 | 76.657   | $3,40 \times 10^{-06}$ |
| 966,339   | 204.832.370   | $4,72 \times 10^{-06}$ | 99,069 | 82.762   | $1,20 \times 10^{-03}$ | 0,358 | 105.149  | $3,40 \times 10^{-06}$ |
| 1.787,020 | 384.044.065   | $4,65 \times 10^{-06}$ | 58,764 | 49.171   | $1,20 \times 10^{-03}$ | 0,121 | 35.434   | $3,42 \times 10^{-06}$ |
| 1.232,380 | 277.204.528   | $4,45 \times 10^{-06}$ | 52,534 | 43.864   | $1,20 \times 10^{-03}$ | 0,260 | 76.285   | $3,41 \times 10^{-06}$ |
| 875,533   | 190.222.305,3 | $4,60 \times 10^{-06}$ | 70,453 | 58.794,4 | $1,20 \times 10^{-03}$ | 0,166 | 48.613,5 | $3,41 \times 10^{-06}$ |

# APÉNDICE G

## CONTERFACTUAL REGRET MINIMIZATION

### Gráficas del juego One Card Poker

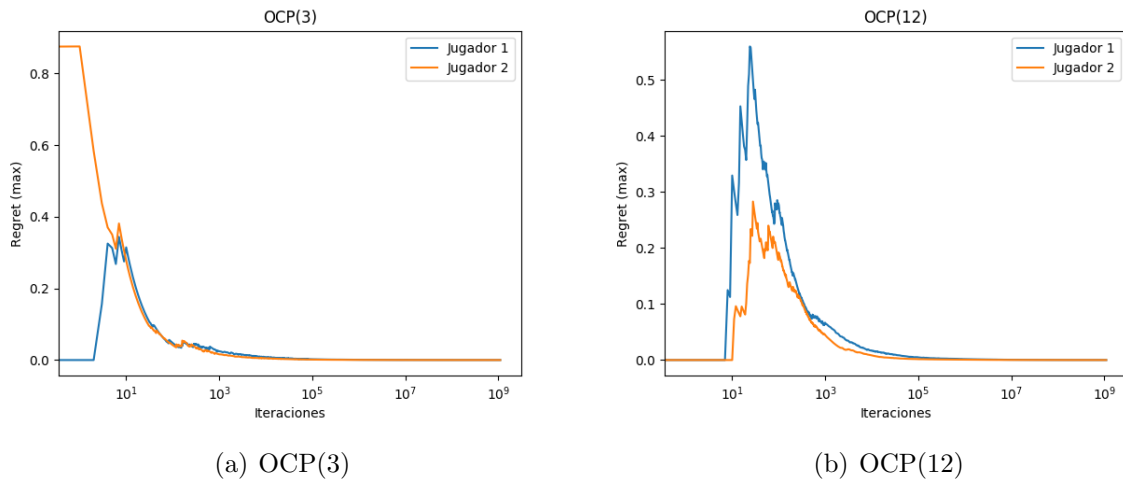
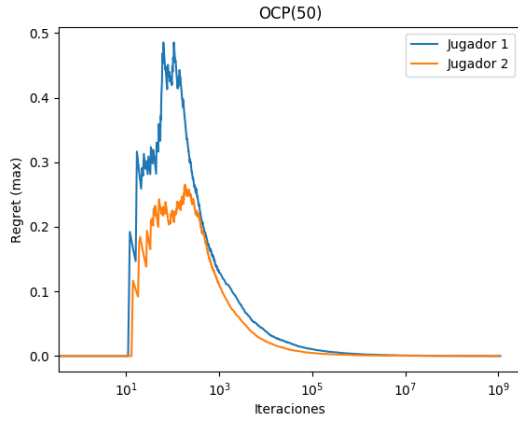
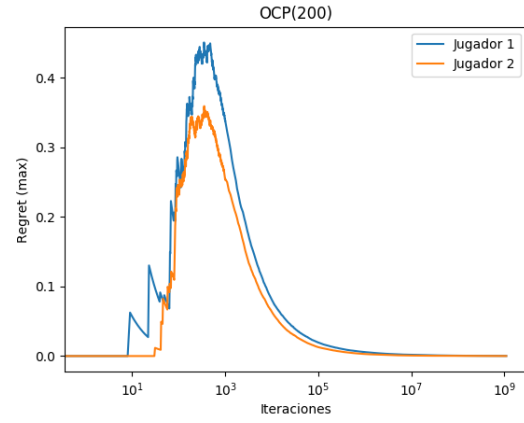


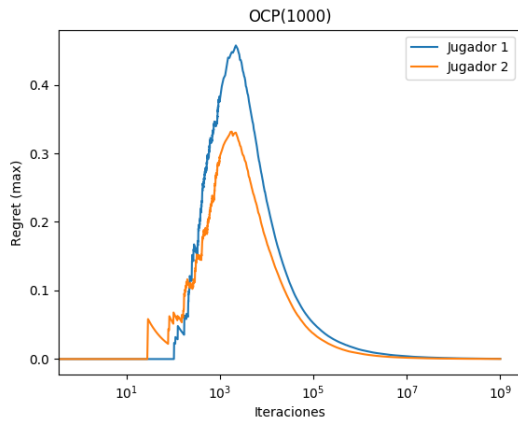
Figura G.1: Gráficas del regret con respecto al número de iteraciones de los juegos OCP(3) y OCP(12)



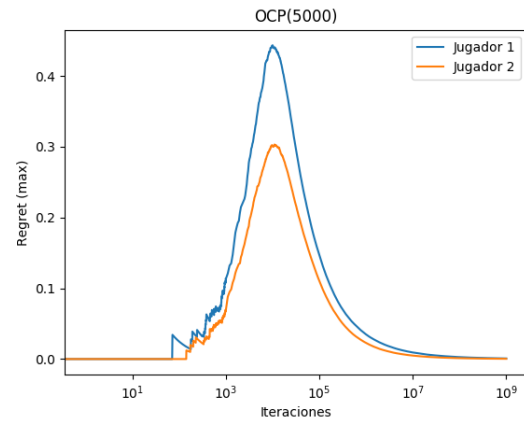
(a) OCP(50)



(b) OCP(200)



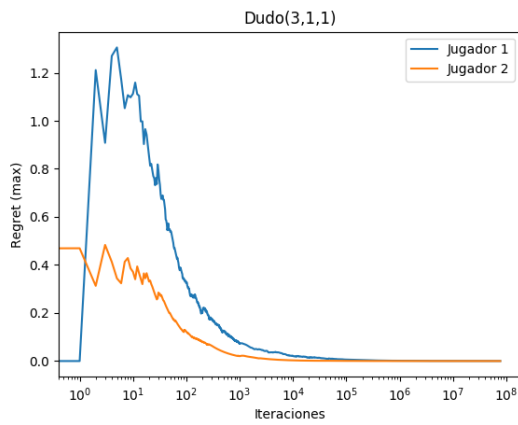
(c) OCP(1000)



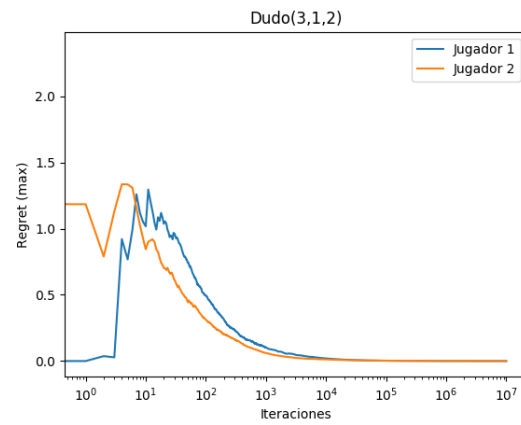
(d) OCP(5000)

Figura G.2: Gráficas del regret con respecto al número de iteraciones de los juegos OCP(50), OCP(200), OCP(1000) y OCP(5000)

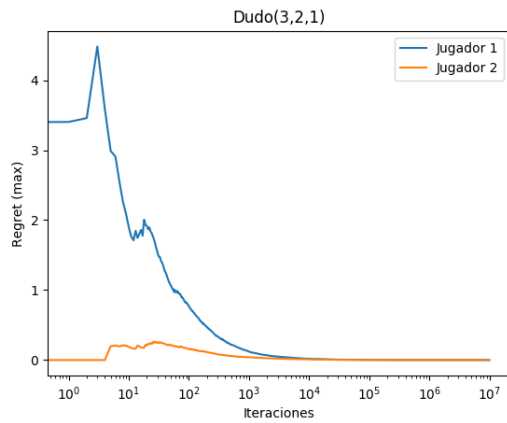
## Gráficas del juego Dudo



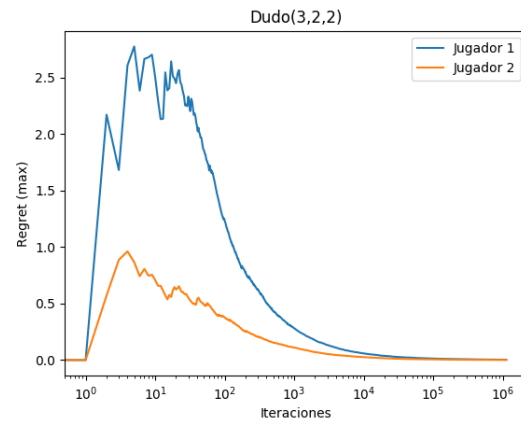
(a) Dudo(3,1,1)



(b) Dudo(3,1,2)

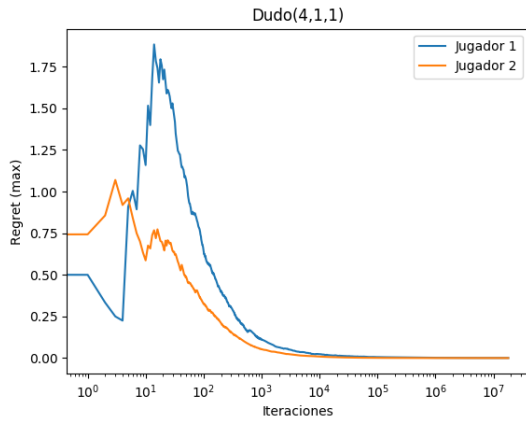


(c) Dudo(3,2,1)

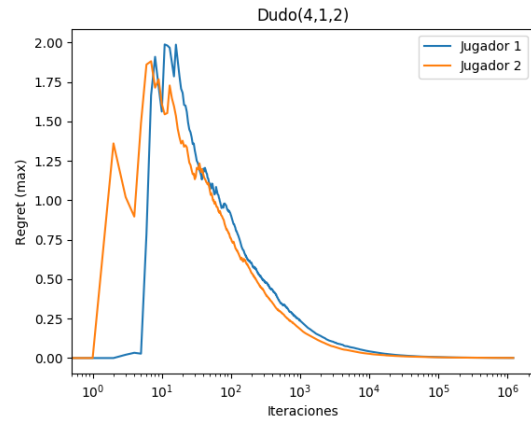


(d) Dudo(3,2,2)

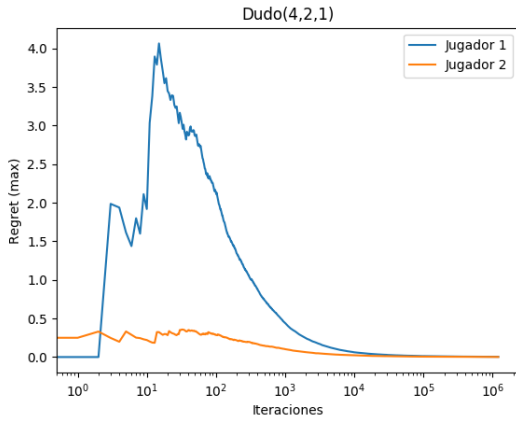
Figura G.3: Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(3,1,1), Dudo(3,1,2), Dudo(3,2,1) y Dudo(3,2,2)



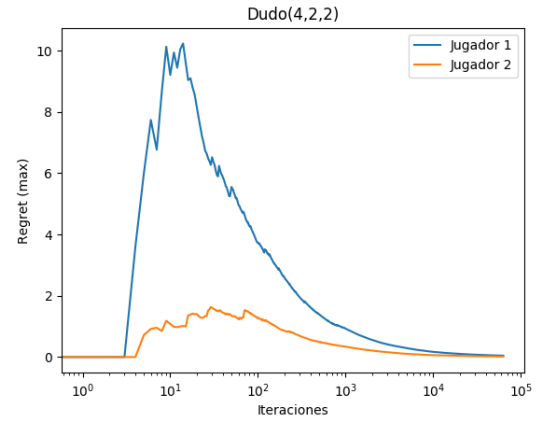
(a) Dudo(4,1,1)



(b) Dudo(4,1,2)

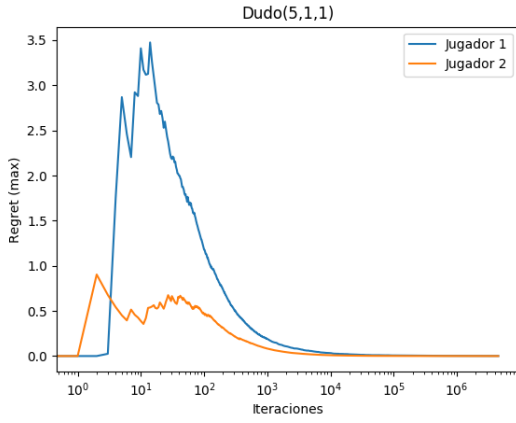


(c) Dudo(4,2,1)

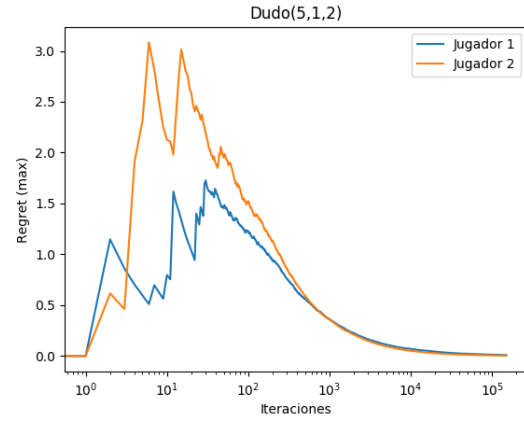


(d) Dudo(4,2,2)

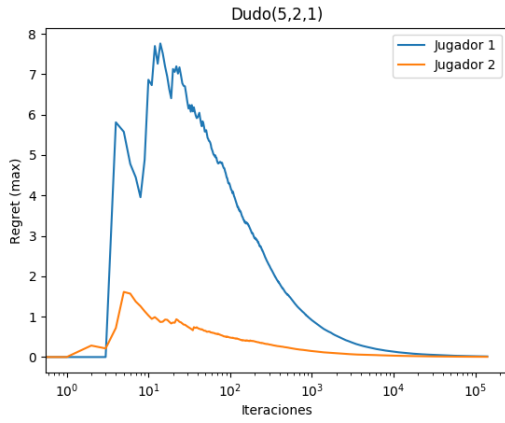
Figura G.4: Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(4,1,1), Dudo(4,1,2), Dudo(4,2,1) y Dudo(4,2,2)



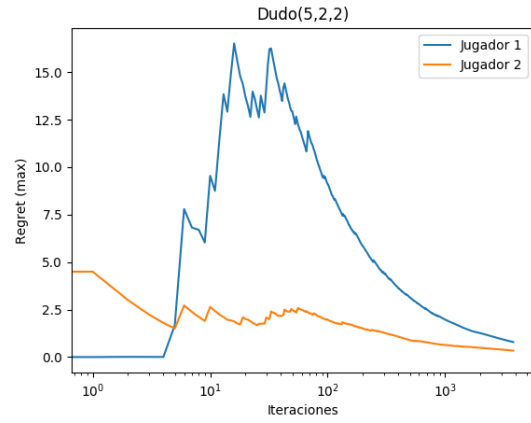
(a) Dudo(5,1,1)



(b) Dudo(5,1,2)

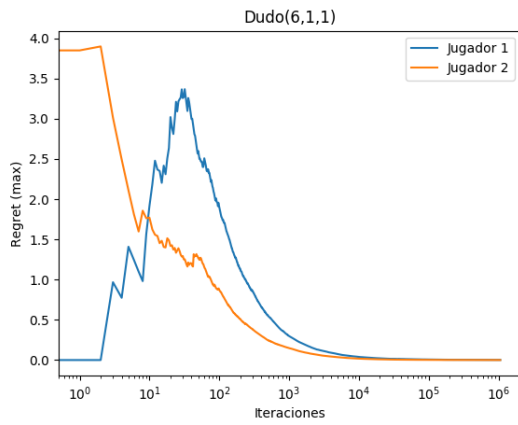


(c) Dudo(5,2,1)

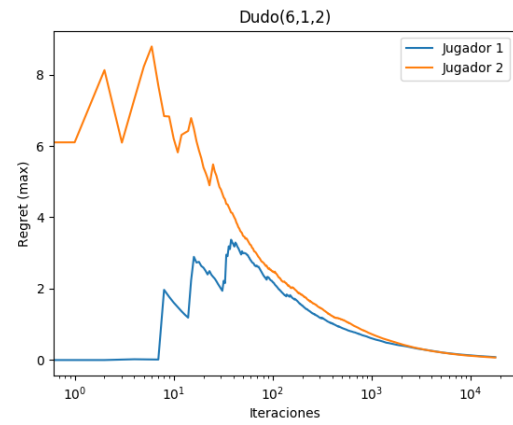


(d) Dudo(5,2,2)

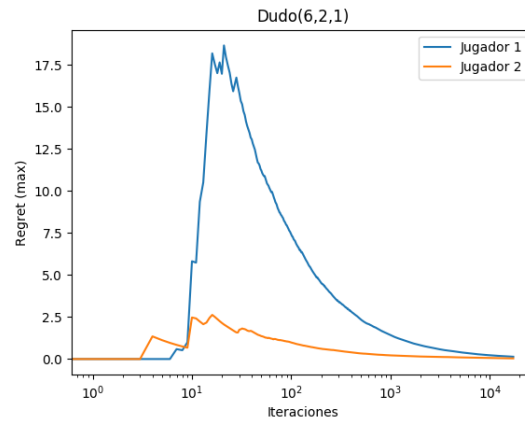
Figura G.5: Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(5,1,1), Dudo(5,1,2), Dudo(5,2,1) y Dudo(5,2,2)



(a) Dudo(6,1,1)



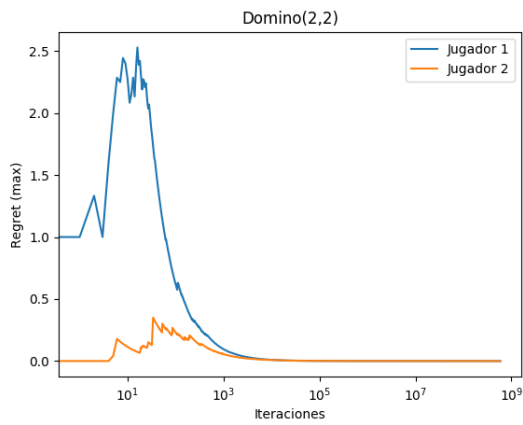
(b) Dudo(6,1,2)



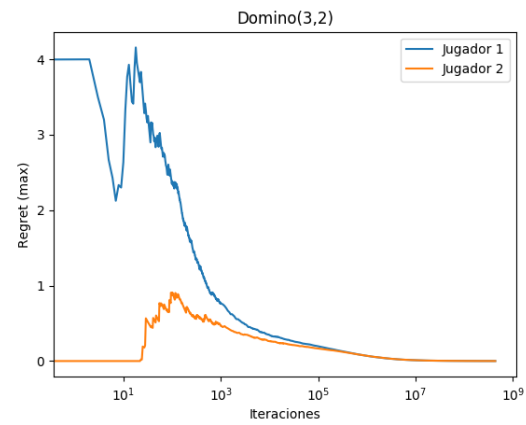
(c) Dudo(6,2,1)

Figura G.6: Gráficas del regret con respecto al número de iteraciones de los juegos Dudo(6,1,1), Dudo(6,1,2) y Dudo(6,2,1)

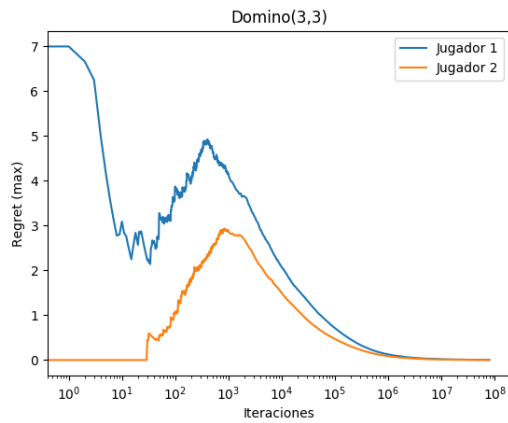
## Gráficas del juego Dominó



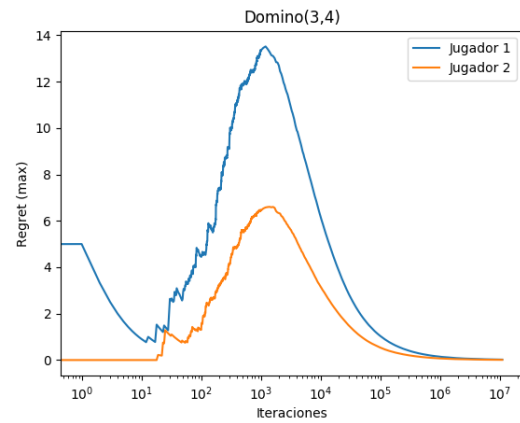
(a) Domino(2,2)



(b) Domino(3,2)



(c) Domino(3,3)



(d) Domino(3,4)

Figura G.7: Gráficas del regret con respecto al número de iteraciones de los juegos Domino(2,2), Domino(3,2), Domino(3,3) y Domino(3,4)