Problemas de Equilibrio de Nash Generalizado con Inconsciencia en las Restricciones

Vegetti Sara

Director: Barrea Andrés Alberto

Trabajo especial de
Licenciatura en matemática

Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba
Argentina

Problemas de Equilibrio de Nash Generalizado con Inconsciencia en las Restricciones

Vegetti Sara

Resumen

La Teoría de Juegos es una disciplina de la que escuchamos hablar cada vez más en distintas áreas. Esto se debe a que modela situaciones interactivas (llamadas juegos) entre dos o más participantes (llamados jugadores) que deben tomar una decisión que afectará a los involucrados. El presente trabajo tiene como objetivo estudiar dos conceptos recientes de este área de estudio, el de Equilibrio de Nash Generalizado y el de inconsciencia en juegos estáticos. El primer concepto modela las situaciones en las que el conjunto del que cada jugador puede elegir sus estrategias depende de las elecciones de los demás. El segundo hace referencia a modelos que reflejan la situación en la cual los jugadores no tienen un conocimiento acabado del juego donde participan, dando lugar a juegos subjetivos. El principal objetivo de este trabajo es proponer nociones de equilibrio para esta última situación mencionada, así como métodos numéricos para resolver la misma.

Palabras Claves:

- Juegos estratégicos.
- Equilibrio de Nash.
- Equilibrio de Nash generalizado.
- Inconsciencia.
- Jerarquía de creencias.

Clasificación (Math. Subject Classification)
91A10 - Noncooperative games
91B50 - General equilibrium theory

Problemas de Equilibrio de Nash Generalizado con Inconsciencia en las Restricciones

Vegetti Sara

Abstract

Game Theory is a discipline that we hear more and more about in different areas. It models interactive situations (called games) between two or more agents (called players) who has to make a decision that will affect those involved. This survey aims to study two recent concepts in this discipline, Generalized Nash Equilibrium and unawareness in static games. Generalized Nash Equilibrium problems models situations in which each player's strategy space depends on the other agents' choices. While unawareness refers to models that represent the situation in which players don't have full knowledge of the game, leading to the emergence of subjective games. The main goal of this paper is to propose notions of equilibrium in games with unawareness, as well as numerical methods to solve them.

Key words:

- Strategic Games.
- Nash Equilibrium.
- Generalized Nash Equilibrium.
- Unawareness.
- Belief hierarchy.

Classification (Math. Subject Classification)
91A10 - Noncooperative games
91B50 - General equilibrium theory

Índice general

Capítulo 1

Introducción

En cada momento, quizás sin darnos cuenta, estamos involucrados en situaciones que nos hacen interactuar con los demás y en estas ocasiones estamos forzados a tomar decisiones que los involucran. Por ejemplo, cuando se conduce un coche en una calle urbana transitada; cuando se puja en una subasta; cuando un productor decide el precio al que venderá sus productos o la cantidad que producirá; cuando una empresa y un sindicato deben decidir los salarios del año próximo de los empleados; etc.

En estas situaciones necesitamos poder decidir racionalmente, buscando la mejor opción que se adecúe o mejor satisfaga nuestros intereses personales, lo que más adelante llamaremos "optimizar nuestra utilidad". Sin embargo, no nos encontramos solos en estas situaciones, sino que estamos constantemente en relación con otros entes (empresas competidoras por ejemplo) o personas, que también tienen como objetivo hallar la mejor solución que satisfaga sus propios intereses y cuyas decisiones pueden afectar nuestro resultado, así como las nuestras pueden afectar el de ellos. Por esto es que para encontrar una solución a estos problemas de decisión, necesitamos de algún criterio que nos indique cuál es la opción óptima, teniendo en cuenta todos los factores involucrados.

Para definir dicho criterio primero se requiere poder modelar la situación de manera sistemática, para poder generar algún algoritmo que nos facilite la toma de decisiones. Es por esto que nace la *Teoría de Juegos*, teoría matemática que se encarga de modelar este tipo de situaciones llamadas de "interdependencia estratégica" (cuando las acciones de un jugador influyen tanto en los resultados propios como en los de los demás agentes con los que está interactuando).

La ventaja que tiene la teoría de juegos es que toma prestada su terminología de los juegos de salón (como por ejemplo el ajedrez), puesto que a la situación interactiva la llama *juego*, los agentes involucrados son llamados *jugadores*, las posibles decisiones que pueden tomar los participantes se conocen como *estrategias*. Incluso una colección de situaciones interactivas es llamada *juego de salón*.

Una vez que tenemos el modelo, debemos poder resolverlo, entonces nos preguntamos, ¿cuál es la mejor solución cuando se tienen tantos factores a considerar? Aquí es donde el famoso ganador del premio Nobel de Economía de 1994, John F. Nash, hace su aparición, brindándonos uno de los conceptos de solución de la Teoría de Juegos más importantes, el llamado *Equilibrio de Nash*. Este equilibrio es un perfil de estrategias (concepto que definiremos en el próximo capítulo), en el cual todos los jugadores obtienen la máxima utilidad que pueden obtener, tal que si alguno decide cambiar su estrategia unilateralmente, su utilidad disminuirá.

Este concepto es perfecto para brindar una solución a situaciones en las cuales las estrategias de cada jugador no dependen de las estrategias de los demás, como por ejemplo en el famoso *Dilema del Prisionero*, donde dos prisioneros son interrogados y cada uno tiene que decidir si confesarse culpable del delito, permanecer en silencio o delatar a su compañero. En ese caso cada jugador tiene un conjunto de opciones que no depende del conjunto de opciones del otro. Sin embargo, en la vida real existen situaciones en las cuales las opciones de una persona dependen de las opciones de otra, como por ejemplo cuando estamos transitando por una calle urbana donde nuestras posibles elecciones de movimiento dependen de los movimientos de los demás para evitar choques. Para este último tipo de situaciones, Gerard Debreu introduce el concepto de *Equilibrio de Nash Generalizado*, que es una generalización del concepto anteriormente mencionado, para juegos en los cuales el espacio de estrategias de un jugador (concepto que definiremos en el capítulo siguiente), depende de las estrategias de los demás.

Los conceptos de solución mencionados hasta el momento nos permiten hallar soluciones de juegos con información completa, donde se conocen todos los jugadores, los conjuntos de estrategias y las utilidades que cada resultado le brinda a cada uno. Además se asume que todos los individuos participantes del juego conocen todos estos datos. Pero en la vida real las situaciones interactivas no son tan perfectas. Por ejemplo, cuando estamos manejando, conocemos a los otros autos que vemos a nuestro alrededor, notamos las direcciones en las que les es factible moverse y sabemos que existen movimientos por los que pueden optar que los llevarán a producir o evitar un accidente; sin embargo no sabemos qué está ocurriendo dentro de los otros vehículos (lo que puede generar otras estrategias para los demás que nosotros desconocemos), o si estamos decidiendo qué rumbo tomar en una esquina y algún auto ha conducido en sentido contrario encontrándonos por sorpresa cuando decidamos avanzar (lo que puede generar que el conjunto de jugadores sea más amplio de lo que pensábamos), entre otros. Aquí es donde se introduce un nuevo concepto que acerca los modelos a la realidad, la *inconsciencia*, que supone la ignorancia sobre alguno (o todos) los elementos que constituyen el juego.

Esto último nos motiva a estudiar ciertos modelos y conceptos de solución para juegos con inconsciencia que se han desarrollado recientemente. Poniendo el foco en juegos estáticos (concepto que desarrollaremos más adelante), en este trabajo estudiaremos los *Problemas de Equilibrio de Nash Generalizado para juegos con inconsciencia en las restricciones* y resultados para probar la existencia de dichos equilibrios.

Este trabajo estará estructurado de la siguiente manera: en el capítulo 2 estudiaremos todos los conceptos teóricos básicos para el desarrollo, modelado y resolución de juegos. De esta manera introduciremos brevemente la teoría de decisión, luego definiremos los juegos estratégicos junto con la definición de equilibrio de Nash para los mismos y continuaremos con la introducción de los problemas de equilibrio de Nash generalizado. Cuando modelamos situaciones de la vida real y definimos conceptos de solución para las mismas, la intención detrás de esto es intentar resolver la situación en la realidad también; para esto no siempre se pueden tomar los problemas como son planteados en la teoría, sino que debemos realizar una pequeña reformulación para poder llevarlos a un formato que algoritmos ya desarrollados comprendan. Por esto es que en este capítulo presentaremos también una reformulación para poder asegurar la existencia de la solución buscada.

En el capítulo 3 desarrollaremos la teoría de juegos con inconsciencia, junto con ejemplos propiamente desarrollados para la mejor comprensión de estos nuevos conceptos. En el capítulo 4 presentaremos métodos numéricos para resolver este tipo de problemas. Aquí, haremos un breve repaso de la teoría básica de Optimización; brindaremos una generalización de las condiciones KKT (concepto que desarrollaremos en ese capítulo) a la Teoría de Juegos y finalizaremos con una aplicación de todo lo desarrollado en el trabajo.

En el capítulo 5 se podrán encontrar las conclusiones de este trabajo y luego un apéndice donde podrán consultarse los códigos utilizados para la resolución de los problemas presentados en el capítulo de métodos numéricos.

Antes de internarnos en el trabajo en sí queremos presentar un breve repaso histórico a modo informativo. Intentaremos aquí resumir un poco el desarrollo de la Teoría de Juegos actual, desde sus orígenes oficiales hasta el descubrimiento de su concepto más importante, el equilibrio de Nash.

1.1. Algo de Historia

Cuando las matemáticas son aplicadas en otras ciencias, es importante que las situaciones de la vida real se modelicen a través de objetos matemáticos cuyo estudio teórico facilite la resolución del problema. En este sentido, la teoría de juegos nace para modelar situaciones interactivas entre dos o más entes o personas, pues estudia la interdependencia estratégica. Este último hecho mencionado sucede cuando los distintos agentes de un juego interactúan entre ellos y las acciones de uno influyen, además de en su propio resultado, en los resultados que los demás agentes involucrados obtienen. En este sentido, se puede ver cómo la necesidad práctica del ser humano ha sido la causa necesaria para el surgimiento y posterior desarrollo de la *teoría de juegos*.

Desde sus orígenes, la teoría de juegos ha consistido en la modelización y el análisis de situaciones de conflicto y cooperación entre agentes racionales e inteligentes, reduciendo la toma de decisiones a un comportamiento puramente racional basado en cuáles serían las estrategias más beneficiosas o menos perjudiciales para afrontar la interacción. Robert J. Aumann, un matemático israelí, galardonado con el premio Nobel de Economía en 2005 por su trabajo en el análisis de la teoría de juego, en una entrevista con Sergiu Hart[1] dijo que “la teoría de juegos es la toma de decisiones óptima en presencia de otros con diferentes objetivos”.

[1] Center for the Study of Rationality, Department of Economics, and Department of Mathematics, The Hebrew University of Jerusalem, Feldman Building, Givat Ram Campus, 91904 Jerusalem, Israel.

Desde el comienzo, la teoría de juegos se concibió como una herramienta que permite alcanzar una mejor comprensión del comportamiento existente en los fenómenos sociales. Esta utilidad práctica ha hecho que no sólo se desarrolle su estudio puramente teórico, sino que se ha vuelto esencial su aplicación en casi todos los ámbitos de las ciencias sociales en lo relativo a la toma de decisiones.

Se considera que la teoría de juegos tal y como la conocemos en la actualidad comienza con la publicación de la obra *Theory of Games and Economic Behavoir* [1] escrita por John Von Neumann y Oskar Morgenstern en 1944. Sin embargo, tal como menciona [2], existen trabajos importantes que, sin ser parte de la teoría de juegos, sí pueden considerarse precursores de ella. En [2] se puede encontrar un breve e interesante repaso por dichos trabajos.

Uno de los primeros aportes fue el de Ernst Friedrich Ferdinand Zermelo, en un artículo de 1913 que establecía dos teoremas sobre el juego de ajedrez. Si bien este juego se utiliza como ejemplo, también permite analizar cualquier juego con dos jugadores sin movimientos de oportunidad y con intereses completamente enfrentados (lo que hoy se conoce como juego no cooperativo de suma cero). Este artículo fue complementado por dos trabajos realizados por dos matemáticos húngaros, Dénes König en [3] y László Kalmár que entre 1928 y 1929 publicó una generalización de los modelos de los anteriores.

En esa misma década, entre 1921 y 1927, Félix E. J. E. Borel realizó una serie de cinco artículos en los que establece los fundamentos de la teoría de juegos psicológicos. En ellos estudia los juegos estratégicos dando la primera formulación matemática moderna de una estrategia mixta y estudiando la búsqueda de la solución minimax para juegos simétricos para dos jugadores con intereses completamente opuestos, demostrándolo para jugadores con tres y cinco estrategias. Además introduce la idea de estrategia pura.

En 1928 Von Newmann en [4] dio una demostración del teorema minimax para estos juegos independientemente del número de estrategias (que ha de ser una cantidad finita) y asumiendo que los intereses son completamente opuestos. En dicho trabajo aparece la definición formal de estrategia utilizada actualmente y se introduce la forma extensiva de un juego como un árbol lógico enraizado.

Posteriormente se sucedieron una serie de trabajos en los cuales se demuestra, desde distintas perspectivas, el teorema anteriormente mencionado. Llegando así a lo que se acepta como la obra clave y referencia básica de la teoría de juegos, el libro *Theory of Games and Economic Behavoir* [1] de Von Newmann y Morgenstern. Esta obra fue el primer tratamiento riguroso y exhaustivo de los conceptos de juego, estrategia y resolución del mismo, así como sobre la forma de representar las preferencias de los jugadores. Todo desde una perspectiva puramente económica y con el objetivo de modelizar el comportamiento social en este área mediante la teoría de juegos.

La obra de Von Newmann y Morgenstern causó tal impacto entre los matemáticos y economistas de su época que conllevó a que la teoría de juegos comenzase a considerarse como una disciplina científica. En la década de 1950 junto con el comienzo de la de 1960, empiezan a desarrollarse numerosos artículos teóricos sobre ella y sus aplicaciones a distintos problemas. Precisamente es en la década de

1950 en la que se da un hecho clave de la teoría de juegos y que sustenta toda la investigación posterior sobre los juegos no cooperativos, la presentación de la tesis doctoral del matemático estadounidense John F. Nash Jr.

En este trabajo titulado *Non-cooperative games* [5] se establecen las bases generales de este tipo de juegos, se introduce el concepto de punto de equilibrio, o equilibrio de Nash como se conoce actualmente, y se prueba su existencia. Tras introducir este concepto, Nash junto con John Patterson Mayberry y Martin Shubic, publicaron un artículo en 1953 en el que demostraron que el equilibro de Nash generalizaba el equilibrio clásico en un duopolio de Cournot.

Desde finales de la década de 1950 y durante toda la década de 1960 el desarrollo de la teoría de juegos fue emparejado con aplicaciones de estos a distintos campos de conocimientos y a la resolución de problemas que surgen en el campo real. Sin embargo, el reconocimiento mundial y público de la gran relevancia de la teoría de juegos no llego hasta 1994, fecha en que J. C. Harsanyi, R. J. R. Selten y J. F. Nash se hicieron acreedores del premio Nobel de Economía.

Capítulo 2

Teoría de Juegos

La *teoría de juegos* es la teoría matemática que modela situaciones de decisión interactivas, es decir, con dos o más agentes participantes. Estas situaciones se caracterizan por contar con un conjunto de agentes que deben tomar una decisión sobre un conjunto de posibles acciones; un conjunto de resultados que serán función de las decisiones tomadas por todos los participantes; y donde cada agente tiene sus propias preferencias sobre los posibles resultados.

Ésta tiene su base en la *teoría de decisión*, que es la teoría matemática que modela situaciones de decisión unipersonales, es decir, aquellas en las que una única persona debe tomar una decisión. Esta teoría analiza cómo la persona elige la acción que la conduce al mejor resultado posible, dadas sus preferencias.

La teoría de juegos clásica, que es la que se pretende introducir aquí, es una teoría normativa ideal en el sentido que determina, para cada juego particular, cuán racionales deben ser los jugadores. Entendiendo por jugador racional a aquel que sabe lo que quiere, tiene el único objetivo de obtener lo que quiere y es capaz de identificar las mejores estrategias que lo lleven a cumplirlo.

Las situaciones interactivas que estudiaremos principalmente en este trabajo son los denominados *juegos estratégicos*, presentados en [6]. Éstos modelan situaciones interactivas entre múltiples participantes, considerándolos racionales, y son considerados juegos estáticos pues se asume que cada participante debe tomar su decisión de forma simultánea e independiente de los demás. Los juegos estratégicos son modelos *no cooperativos*, es decir que asumen que todas las posibilidades de cooperación han sido incluidas como movimientos formales en el juego (como por ejemplo en las funciones de utilidad y en los conjuntos factibles de estrategias de cada jugador), por lo que sólo lidia con estrategias y funciones de utilidad. Por esto asumiremos la no interacción entre los jugadores al momento de escoger una estrategia, pero contemplando la posibilidad de que, previo al comienzo del juego, los participantes se comuniquen entre ellos estableciendo acuerdos no vinculantes. Estos acuerdos pueden incluir, por ejemplo, el establecimiento de las reglas del juego, de los conjuntos de estrategias, etc., pero no pueden hacerse acuerdos que condicionen la elección de la estrategia a ningún jugador. En este trabajo no abordaremos temas como la generosidad, la solidaridad, las afinidades personales, etc., que son elementos a considerar en los llamados *juegos cooperativos*.

El concepto de solución para juegos estratégicos más importante es el equilibrio de Nash, introducido por John F. Nash en 1951, el cual es simplemente un perfil de estrategias desde el cual ningún agente se beneficia cambiando unilateralmente su estrategia.

En la primera mitad de este capítulo desarrollaremos los conceptos mencionados arriba. Mientras que en la segunda mitad estudiaremos los problemas de equilibrio de Nash generalizado (GNEP por sus siglas en inglés), un concepto introducido por Gerard Debreu en [7] y que en las últimas décadas ha ganado importancia para gran cantidad de aplicaciones prácticas, por ejemplo en economía, ciencias de la computación, ingenierías, etc.

Las reformulaciones de los problemas de equilibrio de Nash generalizado han resultado ser muy útiles a la hora de probar resultados teóricos, como por ejemplo, existencia de solución. Más aún, a menudo son la clave para diseñar algoritmos que resuelvan numéricamente esta clase de problemas. Entonces al final de este capítulo estudiaremos una reformulación de GNEP's basada en la función de Nikaido-Isoda y resultados que nos aseguren la existencia de, al menos, una solución.

2.1. Teoría de decisión

La ***teoría de decisión*** es la teoría matemática que modela la situación en la que una única persona debe tomar una decisión, que analiza cómo ésta elige aquella acción que le conduce al mejor resultado, dadas sus preferencias de entre un conjunto de acciones posibles.

En esta rama se consideran problemas de decisión en los cuales el agente debe decidir entre posibles *alternativas* de un conjunto X y sobre el cual este agente tiene *preferencias*.

Definición 2.1.1 *Una* ***preferencia débil*** *sobre el conjunto X, denotada por $\succeq$, es una relación binaria sobre X completa y transitiva. Entonces para cada par $a, b \in X$, decimos que $a \succeq b$ si el jugador prefiere a a sobre b o es indiferente ante a y b, y lo leemos como "a es preferido a b".*

Así, podemos definir un problema de decisión de la siguiente manera:

Definición 2.1.2 *Un* ***problema de decisión*** *es un par $(X, \succeq)$ donde X es un conjunto de alternativas y $\succeq$ es una preferencia débil sobre X.*

Trabajar con relaciones de preferencia así definidas suele ser muy complicado, por lo que comúnmente se trabaja con representaciones de estas relaciones. Una forma de representar una relación de preferencias es utilizar una función, llamada de utilidad.

Definición 2.1.3 *Dado un problema de decisión $(X, \succeq)$, una* ***función de utilidad*** *que representa a $\succeq$ es una función $u : X \longrightarrow \mathbb{R}$ que satisface que para cada par $a, b \in X$, $a \succeq b$ si y sólo si $u(a) \geq u(b)$.*

Una pregunta que puede surgir naturalmente es: ¿bajo qué condiciones la preferencia débil de un problema de decisión puede ser representada por una función de utilidad? Una respuesta parcial a esta pregunta nos la presenta el siguiente resultado:

Proposición 2.1.4 *Sea X un conjunto numerable y $(X, \succeq)$ un problema de decisión, entonces existe una función de utilidad u que representa a $\succeq$.*

Para obtener condiciones necesarias y suficientes para la existencia de una única función de utilidad que represente la relación binaria de un problema de decisión, presentaremos los siguientes conceptos y resultados:

Definición 2.1.5 *Sea $(X, \succeq)$ un problema de decisión, definimos la **preferencia estricta**, $\succ$, de la siguiente manera: para cada par $a, b \in X$, $a \succ b$ si y sólo si $b \not\succeq a$.*

Definición 2.1.6 *Sea $(X, \succeq)$ un problema de decisión, un conjunto $Y \subset X$ se dice un **orden denso** en X si, para cada par $a_1, a_2 \in X$ tales que $a_1 \succ a_2$, existe un $b \in Y$ tal que $a_1 \succeq b \succeq a_2$.*

Definición 2.1.7 *Sean $(X, \succeq)$ un problema de decisión y $a_1, a_2 \in X$ tales que $a_2 \succ a_1$, entonces (a_1, a_2) es una **brecha** si, para cada $b \in X$, ocurre alguna de las siguientes: $b \succeq a_2$ o $a_1 \succeq b$. Si (a_1, a_2) es una brecha, entonces a_1 y a_2 se llaman **extremos** de la brecha. Además definimos X^* como el conjunto de extremos de las brechas de X.*

Lema 2.1.8 *Sea $(X, \succeq)$ un problema de decisión y asumamos que $\succeq$ es antisimétrica. Entonces:*

- *Si existe un subconjunto numerable $Y \subset X$ que es un orden denso en X, entonces X^* es numerable.*
- *Si existe una función de utilidad que representa a $\succeq$, entonces X^* es numerable.*

Teorema 2.1.9 *Sea $(X, \succeq)$ un problema de decisión y asumimos $\succeq$ antisimétrica. Entonces $\succeq$ puede ser representada por una función de utilidad si y sólo si existe un subconjunto numerable $Y \subset X$ que es un orden denso en X.*

2.2. Juegos estratégicos

Un juego estratégico, también conocido como juego en forma normal, es un objeto matemático simple, caracterizado por el conjunto de estrategias de cada jugador, junto con las respectivas funciones de utilidad o costo[1]. Además se considera que todos los jugadores son *racionales* y que para este tipo de jugador, no

[1]En la literatura se puede encontrar utilidad o costo indistintamente usado. La diferencia radica en que los agentes siempre intentan maximizar su función de utilidad, pero minimizar su función de costo.

hay límites en la complejidad de los cálculos que puede realizar o en la sofisticación de las estrategias que utilice para alcanzar su objetivo[2].

De ahora en más denotaremos el conjunto de jugadores de un juego por $\mathcal{N} := \{1, \ldots, n\}$.

Definición 2.2.1 *Un **juego estratégico** con un conjunto $\mathcal{N}$ de jugadores es un par $G := (X, u)$ cuyos elementos son:*

- *Un conjunto factible o espacio de estrategias: Para cada $i \in \mathcal{N}$, X_i es un conjunto no vacío llamado de **estrategias** del jugador i y $X := \prod_{i=1}^{n} X_i$ es el conjunto de **perfiles de estrategias**.*
- *Funciones de utilidad: Para cada $i \in \mathcal{N}$, $u_i : X \to \mathbb{R}$ es la **función de utilidad** del jugador i y $u := \prod_{i=1}^{n} u_i$, donde u_i asigna a cada perfil de estrategias $x \in X$, la utilidad que el jugador i obtiene al ser jugado x.*

Notemos que, si bien cada jugador toma su decisión de manera independiente, su función de utilidad depende de las decisiones de los otros participantes. Al finalizar el juego cada jugador recibe un *pago* que depende de cual haya sido el resultado del juego. El significado de dicho pago es la utilidad, es decir, es la valoración que el jugador hace de las consecuencias de alcanzar un cierto resultado, no necesariamente es una remuneración económica.

En el tipo de situación interactiva que modelan los juegos estratégicos están involucrados implícitamente los siguientes elementos:

- $\mathcal{R}$, un conjunto de posibles respuestas o resultados del juego, que se traducen en las distintas combinaciones de las utilidades que cada jugador obtiene.
- Una función $f : X \to \mathcal{R}$ que asigna a cada perfil $x \in X$, su correspondiente resultado en $\mathcal{R}$.
- $\{\succeq_i\}_{i \in \mathcal{N}}$, las preferencias de los jugadores sobre las posibles respuestas en $\mathcal{R}$. Éstas se asumen completas, transitivas y representables a través de las funciones de utilidad.

Para comprender mejor estos conceptos presentaremos un ejemplo conocido, el llamado **Oligopolio de Cournot** (Cournot 1838).

Ejemplo 2.2.2

En el oligopolio de Cournot el conjunto $\mathcal{N}$ de jugadores se corresponde con el conjunto de productores de cierto producto y puede modelarse a través de un juego estratégico $G = (X, u)$. En G, cada productor $i \in \mathcal{N}$ debe elegir una estrategia

[2] Esta última asunción es estándar en la teoría de juegos clásica y en la mayoría de los campos en los que se aplica, especialmente en economía.

$x_i \in X_i = [0, \infty)$ que representa la cantidad de unidades que producirá, considerando la función de utilidad $u_i(x) = \pi \left(\sum_{j \in \mathcal{N}} x_j \right) x_i - c_i(x_i)$ donde $c_i(x_i)$ denota el costo total que tiene que afrontar el jugador i al elegir la estrategia x_i. A su vez, se asume que los productores colocan sus productos en el mismo mercado, por lo que el precio de una unidad de producto depende de $\sum_{i \in \mathcal{N}} x_i$ y lo denotamos por $\pi \left(\sum_{i \in \mathcal{N}} x_i \right)$.

2.3. Equilibrio de Nash

El concepto más importante de solución para juegos estáticos es el equilibrio de Nash, introducido por John F. Nash en su tesis de doctorado, en 1951. En *Kohlberg* (1990), se dice que la idea más importante detrás del equilibrio de Nash es "una audaz simplificación y, en lugar de preguntarse cómo podría desarrollarse el proceso de deducción, se pregunta dónde podrían estar los *puntos muertos* del juego". De hecho, un equilibrio de Nash de un juego estratégico es simplemente un perfil de estrategias en el cual ningún jugador se beneficia desviándose unilateralmente de él, es decir, el concepto del equilibrio de Nash busca estos puntos muertos de las situaciones interactivas descriptas por los juegos estratégicos.

Dado un juego $G = (X, u)$, con conjunto de jugadores $\mathcal{N}$, y un perfil estratégico $x = (x_1, ..., x_n) \in X$, definimos $x_{-i} := (x_j)_{j \in \mathcal{N}, j \neq i}$. Sea $(\hat{x}_i, x_{-i})$ el perfil $(x_1, ..., x_{i-1}, \hat{x}_i, x_{i+1}, ..., x_n)$, podemos definir:

Definición 2.3.1 *Un* ***equilibrio de Nash*** *del juego G es un perfil de estrategias $x^* \in X$ tal que, para cada $i \in \mathcal{N}$ y cada $\hat{x}_i \in X_i$,*

$$u_i(x^*) \geq u_i(\hat{x}_i, x^*_{-i})$$

Notemos que la notación $(\hat{x}_i, x_{-i})$ no afecta el orden del perfil de estrategias, simplemente enfatiza al jugador del cual estamos hablando.

Además, denotando $X_{-i} := \prod_{j \neq i} X_j$, podemos definir:

Definición 2.3.2 *Sea $G = (X, u)$ un juego estratégico tal que, para cada $i \in \mathcal{N}$, existe un $m_i \in \mathbb{N}$ tal que X_i es un conjunto no vacío y compacto de $\mathbb{R}^{m_i}$ y u_i es continua. Definimos entonces para cada $x_{-i} \in X_{-i}$ el conjunto de* ***mejores respuestas*** *de i a x_{-i} como*

$$\begin{aligned} BR_i(x_{-i}) &:= \{x_i \in X_i \mid u_i(x_i, x_{-i}) = \max_{\tilde{x}_i \in X_i} u_i(\tilde{x}_i, x_{-i})\} \\ &= \{x_i \mid \textit{para cada } \tilde{x}_i \in X_i, \ u_i(x_i, x_{-i}) \geq u_i(\tilde{x}_i, x_{-i})\} \end{aligned}$$

Así, $BR_i(x_{-i})$ es el conjunto de estrategias del agente i tales que, dado que los demás agentes jugaron las estrategias en x_{-i}, le convienen jugar a i para aumentar la utilidad percibida, i.e., el conjunto de las mejores respuestas que puede dar i frente a los demás jugando x_{-i}.

Denotaremos así al conjunto de mejores respuestas para $x \in X$ por

$$BR(x) := \prod_{i \in \mathcal{N}} BR_i(x_{-i})$$

Para entender mejor este concepto, analicemos el equilibrio de Nash en el oligopolio de Cournot.

Ejemplo 2.3.3

Retomaremos el ejemplo 2.2.2 teniendo en cuenta lo siguiente:

- Analizaremos un *duopolio de Cournot*, es decir, $n = 2$.
- Para cada $i \in \{1, 2\}$, $c_i(x_i) = cx_i$, con $c > 0$.
- Sea d fijo con $d > c$. La función de costo viene dada por:

$$\pi(x_1 + x_2) = \begin{cases} d - (x_1 + x_2) & x_1 + x_2 < d \\ \\ 0 & c.c. \end{cases}$$

Para cada $i = 1, 2$ y cada $x \in X$, las funciones de utilidad asociadas a este juego son:

$$u_i(x) = \begin{cases} x_i(d - x_1 - x_2 - c) & x_1 + x_2 < d \\ \\ -x_i c & c.c. \end{cases}$$

Por definición, un equilibrio de Nash de este juego (a veces llamado *equilibrio de Cournot*) es un par $(x_1^*, x_2^*) \in X_1 \times X_2$ tal que, para cada $\hat{x}_1 \in X_1$ y cada $\hat{x}_2 \in X_2$, $u_1(x_1^*, x_2^*) \geq u_1(\hat{x}_1, x_2^*)$ y $u_2(x_1^*, x_2^*) \geq u_2(x_1^*, \hat{x}_2)$. En este caso, el conjunto de mejores respuestas para cada $i \in \mathcal{N}$ y cada $x_{-i} \in X_{-i}$ es

$$BR_i(x_{-i}) := \{\hat{x}_i \mid \forall\ \overline{x}_i \in X_i,\ u_i(\hat{x}_i, x_{-i}) \geq u_i(\overline{x}_i, x_{-i})\}$$

Sea, para cada $i = 1, 2$ y cada $x \in X$, $f_i(x) := x_i(d - x_1 - x_2 - c)$. Luego, aplicando herramientas del análisis matemático para hallar solución de un sistema cuadrático, llegamos a que la única solución de este sistema es

$$x^* = (x_1^*, x_2^*) = \left(\frac{d - x_2 - c}{2}, \frac{d - x_1 - c}{2}\right)$$

Los cálculos correspondientes pueden encontrarse en [6].

Notemos que, por definición, un equilibrio de Nash es un perfil x tal que $x_i \in BR_i(x_{-i})$ $\forall i \in \mathcal{N}$ y $\forall x_{-i} \in X_{-i}$. Entonces, podemos encontrarnos con juegos que tengan un único equilibrio de Nash, más de un equilibrio o juegos que no tengan ninguno. Por esta razón presentaremos un resultado conocido como el *Teorema* *de Nash*[3] que nos asegura la existencia de, al menos, un equilibrio. Para esto enunciaremos a continuación algunos conceptos y motivaciones previas.

Un primer resultado que nos da condiciones suficientes para la existencia de, al menos, un equilibrio de Nash es el siguiente:

Teorema 2.3.4 *Sea $G = (X, u)$ un juego estratégico tal que, para cada $i \in \mathcal{N}$,*

1. *X_i es un subconjunto no vacío, convexo y compacto de $\mathbb{R}^{m_i}$.*

2. *u_i es continua.*

3. *Para cada x_{-i}, $u_i(\cdot, x_{-i})$ es cuasi-cóncava en X_i.*

Entonces, el juego G tiene, por lo menos, un equilibrio de Nash.

Además, podemos definir:

Definición 2.3.5 *Sea $G = (X, u)$ un juego estático, decimos que G es un* ***juego finito*** *si, para cada $i \in \mathcal{N}$, $|X_i| < \infty$.*

Notemos que como los conjuntos de estrategias de los juegos finitos no son conjuntos convexos, entonces no podemos aplicar el teorema 2.3.4 en estos casos. Sin embargo existe un "truco teórico" que nos permite extender los juegos finitos y garantizar la existencia de equilibrios de Nash en dicha extensión. Este "truco" consiste en aumentar las posibilidades de estrategias de los jugadores y permitirles elegir no sólo las estrategias de las que disponen originalmente, llamadas ***estrategias puras***, sino también las loterías sobre sus conjuntos (finitos) de estrategias puras. Esta extensión del juego original se llama ***extensión mixta*** y las estrategias de los agentes en dicha extensión son llamadas ***estrategias mixtas***. Aunque nos referimos a ellas como "truco teórico", las estrategias mixtas son naturales en muchas situaciones prácticas. A continuación daremos la definición formal de estos conceptos.

Definición 2.3.6 *Sea $G = (X, u)$ un juego finito. La* ***extensión mixta*** *de G es el juego estratégico $E(G) := (S, \overline{u})$, cuyos elementos son los siguientes:*

- *Un conjunto de estrategias mixtas: para cada $i \in \mathcal{N}$, $S_i := \Delta X_i$ y $S = \prod_{i \in \mathcal{N}} S_i$. Para cada $s \in S$ y cada $x \in X$, definimos $s(x) := s_1(x_1) \cdot \dots \cdot s_n(x_n)$.*

- *Funciones de utilidad: para cada $s \in S$, $\overline{u}_i(s) := \sum_{x \in X} u_i(x)s(x)$ y $\overline{u} := \prod_{i=1}^{N} \overline{u}_i$.*

[3] En este trabajo no se profundizará en la demostración de dicho resultado, sino que nos limitaremos a enunciarlo, sin embargo el lector puede encontrar la demostración en la sección 2.2 de [6].

Donde ΔX_i es el conjunto de distribuciones de probabilidad sobre X_i, para cada $i \in \mathcal{N}$ y, dado que $s_i(x_i)$ es la probabilidad del jugador i de elegir la estrategia x_i, $s(x)$ es la probabilidad de que el perfil x sea jugado.

La extensión mixta de un juego finito sólo tiene sentido si los jugadores tienen preferencias sobre el conjunto de loterías en $\mathcal{R}$. Además, $E(G)$ es, de hecho, una extensión de G en el sentido que para cada jugador $i \in \mathcal{N}$, cada elemento de X_i puede identificarse unívocamente con un elemento de S_i. En este sentido, podemos escribir que $X_i \subset S_i$. Incluso, las funciones de utilidad en $E(G)$ son realmente extensiones de las funciones de utilidad en G.

Notemos ahora que la extensión mixta de un juego finito satisface las condiciones del teorema 2.3.4, por lo que podemos afirmar que la extensión de un juego finito siempre posee, al menos, un equilibrio de Nash. Esto fue lo demostrado por J. F. Nash en su trabajo original, expresado formalmente como:

Teorema 2.3.7 ***(Teorema de Nash)*** *Sea $G = (X, u)$ un juego estratégico finito. Entonces la extensión mixta de G, $E(G)$, tiene, al menos, un equilibrio de Nash.*

2.4. Problemas de Equilibrio de Nash Generalizado

En 1952, Gerard Debreu publicó el artículo [7] en el que introdujo los problemas de equilibrios de Nash generalizado y en el cual fue acuñado el término *equilibrio social*. En este trabajo, Debreu nos presenta estos problemas como *economía abstracta* y es en el campo de la economía en el que más energía se ha invertido para estudiar este tipo de situaciones. Más aún, éstos se han convertido en un campo de investigación importante desde los 90's, debido a que surgen de manera natural de los problemas de equilibrio de Nash clásico[4] si los jugadores comparten algún recurso (como los sistemas de comunicación, de transporte, las redes eléctricas, etc.) o comparten limitaciones (como el caso de la acumulación de tóxicos en un río que estudiaremos en este trabajo). En esta sección introduciremos breve y formalmente el concepto de Problemas de Equilibrio de Nash Generalizado, basándonos en [8] y [9], así como una reformulación del mismo y la notación necesaria para desarrollar estas teorías.

Considerando un juego de n jugadores, $\mathcal{N} = \{1, \ldots, n\}$, donde cada jugador i controla su vector de estrategias $x^i = (x_1^i, \ldots, x_{n_i}^i)^T \in \mathbb{R}^{n_i}$ donde n_i es la cantidad de variables de decisión involucradas en el problema, obtenemos el vector de estrategias $x = (x^1, \ldots, x^n)^T \in \mathbb{R}^N$, donde $N = \sum\limits_{i=1}^{n} n_i$. Además, sabemos que cada jugador i intentará optimizar su función de utilidad
$u_i : \mathbb{R}^N \longrightarrow \mathbb{R}$.

A diferencia de los problemas de equilibrio de Nash, en estos problemas de equilibrios generalizados, las estrategias de un jugador pueden depender de las estrategias de los rivales. Esto es, el conjunto factible del jugador i es un conjunto

[4]Los definidos en la sección 2.3 del presente trabajo.

$X_i(x_{-i}) \subset \mathbb{R}^{n_i}$. En muchas aplicaciones, este espacio suele estar definido por restricciones de desigualdad, es decir, para cada $i = 1, \ldots, n$, $\exists\ g^i : \mathbb{R}^N \longrightarrow \mathbb{R}^{m_i}$ tal que

$$X_i(x_{-i}) = \{x_i \in \mathbb{R}^{n_i} \mid g^i(x_i, x_{-i}) \leq 0\}$$

Luego, para cada $x \in \mathbb{R}^N$ definimos

$$X(x) = \prod_{i=1}^{n} X_i(x_{-i}) = \{y \in \mathbb{R}^N \mid y_i \in X_i(x_{-i}) \ \ \forall i = 1, \ldots, n\}$$

Definición 2.4.1 *Un* ***problema de equilibrio de Nash generalizado*** *es el problema de encontrar* $x^* \in X(x^*)$ *tal que,* $\forall i = 1, \ldots, n$

$$u_i(x_i^*, x_{-i}^*) \geq u_i(x_i, x_{-i}^*)$$

$\forall x_i \in X(x_{-i}^*)$. *Tal vector* x^* *es llamado* ***equilibrio de Nash generalizado****.*

Denotaremos por **SOL(GNEP)** el conjunto de soluciones de un GNEP y por **SOL(NEP)** el conjunto de soluciones de un problema de equilibrio de Nash. Notemos que si, para todo $i = 1, \ldots, n$, el conjunto factible del jugador i no depende de las estrategias de los demás, el problema generalizado se reduce a un problema de equilibrio de Nash clásico. Por lo tanto podemos ver que
SOL(NEP) $\subset$ **SOL(GNEP)**.

2.5. Reformulación de problemas de equilibrio de Nash generalizado

En esta sección desarrollaremos una reformulación basada en la función de Nikaido-Isoda, presentada en [8] y [9], que utilizaremos más adelante para corroborar la existencia o no de solución de los casos prácticos estudiados en el capítulo 4 de este trabajo.

Para esta reformulación asumiremos que las funciones de utilidad u_i son, al menos, continuas $\forall i = 1, \ldots, n$.

Definición 2.5.1 *Sea el parámetro* $\alpha \geq 0$ *cualquiera, definimos la función*
$\Psi_\alpha : \mathbb{R}^N \times \mathbb{R}^N \longrightarrow \mathbb{R}$ *por*

$$\Psi_\alpha(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} [u_i(x_i, x_{-i}) - u_i(y_i, x_{-i})] - \frac{\alpha}{2} ||\boldsymbol{x} - \boldsymbol{y}||^2$$

Tomando $\alpha = 0$ obtenemos la ***función de Nikaido-Isoda*** clásica.

Notemos que el valor máximo que esta función puede alcanzar es siempre no negativo. Además, si x^* es un equilibrio de Nash del problema, $\forall \mathbf{y} \in X(x^*)$, $\Psi(\mathbf{x}^*, \mathbf{y})$ es no positiva. Entonces podríamos concluir que, cuando la función de Nikaido-Isoda no toma valores significativamente positivos, nos estamos acercando a un equilibrio de Nash. Con esta conclusión se podría fijar una condición de corte para un *algoritmo de relajación*. Según expresa [9], muchos estudios han demostrado que esta clase de algoritmos pueden ser usados para hallar soluciones de juegos no cooperativos con funciones de utilidad no lineales y restricciones acopladas, como se hace en dicho trabajo.

Con el objetivo de presentar los resultados de convergencia de [9], debemos presentar primero una serie de definiciones y resultados.

Definición 2.5.2 *Un elemento $x^* \in X(x^*)$ se dice un **equilibrio de Nash normalizado** si*

$$\max_{\boldsymbol{y}\in X(x^*)} \Psi(\boldsymbol{x}^*, \boldsymbol{y}) = 0$$

Lema 2.5.3 *Un equilibrio de Nash normalizado es también un equilibrio de Nash.*

Se desprende automáticamente del lema anterior que, encontrando un equilibrio de Nash normalizado del problema, estaremos hallando un equilibrio de Nash de nuestro problema.

Teniendo en cuenta que el objetivo de este desarrollo es obtener un algoritmo que nos resuelva el problema numéricamente, tenemos que tener en cuenta que en cada iteración del mismo queremos movernos en una dirección que optimice la iteración actual. Para ello definimos:

Definición 2.5.4 *La **función de respuesta óptima** en el punto $\boldsymbol{x}$ es*

$$Z(\boldsymbol{x}) = \arg\max_{\boldsymbol{y}\in X(Z(\boldsymbol{x}))} \Psi(\boldsymbol{x}, \boldsymbol{y}) \quad \forall \boldsymbol{x}, Z(\boldsymbol{x}) \in X(Z(\boldsymbol{x}))$$

Definición 2.5.5 *Sea X un subconjunto convexo de $\mathbb{R}^m$, una función continua $f : X \longrightarrow \mathbb{R}$ se dice **débilmente convexa en** X si para todo $\boldsymbol{x}, \boldsymbol{y} \in X$ y para todo $0 \leq \alpha \leq 1$ se satisface*

$$\alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}) \geq f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) + \alpha(1-\alpha) r(\boldsymbol{x}, \boldsymbol{y})$$

donde el residuo $r : X \times X \longrightarrow \mathbb{R}$ satisface que cuando $\mathbf{x} \longrightarrow \mathbf{z}$ y $\mathbf{y} \longrightarrow \mathbf{z}$,

$$\frac{r(\mathbf{x}, \mathbf{y})}{||\mathbf{x} - \mathbf{y}||} \longrightarrow 0$$

para todo $\mathbf{z} \in X$.

Definición 2.5.6 *Una función $f(\boldsymbol{x})$ se dice **débilmente cóncava en** X si $-f(\boldsymbol{x})$ es débilmente convexa en X.*

Definición 2.5.7 *Una función $\Psi : X \times X \longrightarrow \mathbb{R}$, donde X es un conjunto cerrado y convexo de $\mathbb{R}^m$, se dice que es **débilmente convexa-cóncava** si es débilmente convexa en X en la primera coordenada y débilmente cóncava en X en la segunda, i.e., si $\forall \alpha \in [0, 1]$, $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in X$ se cumple que*

$$\alpha \Psi(\boldsymbol{x}, \boldsymbol{z}) + (1-\alpha)\Psi(\boldsymbol{y}, \boldsymbol{z}) \geq \Psi(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}, \boldsymbol{z}) + \alpha(1-\alpha) r_z(\boldsymbol{x}, \boldsymbol{y})$$

$$\alpha \Psi(\boldsymbol{z}, \boldsymbol{x}) + (1-\alpha)\Psi(\boldsymbol{z}, \boldsymbol{y}) \leq \Psi(\boldsymbol{z}, \alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) + \alpha(1-\alpha) \mu_z(\boldsymbol{x}, \boldsymbol{y})$$

donde r_z y μ_z cumplen que, $\forall \mathbf{x}, \mathbf{y} \in X$,

$$\frac{r_z(\mathbf{x}, \mathbf{y})}{||\mathbf{x} - \mathbf{y}||} \longrightarrow 0 \qquad \frac{\mu_z(\mathbf{x}, \mathbf{y})}{||\mathbf{x} - \mathbf{y}||} \longrightarrow 0$$

para todo $\mathbf{z} \in X$, cuando $\mathbf{x} \longrightarrow \mathbf{z}$ e $\mathbf{y} \longrightarrow \mathbf{z}$.

Una vez definidos todos estos conceptos, queremos hallar un equilibrio de Nash para nuestro problema a través de un algoritmo iterativo y partiendo de una estimación inicial $\mathbf{x}^0$. Considerando la función $Z(\mathbf{x})$ single-valued, el algoritmo de relajación presentado en [9] está dado por la siguiente fórmula recursiva:

$$\mathbf{x}^{s+1} = (1 - \alpha_s)\mathbf{x}^s + \alpha_s Z(\mathbf{x})$$

$\forall s \in \mathbb{N}_0$[5] y donde $0 < \alpha_s \leq 1$. Así, el paso $s + 1$ se construye como un promedio pesado entre el punto óptimo $Z(\mathbf{x}^s)$ y el punto $\mathbf{x}^s$.

El siguiente teorema establece condiciones de convergencia del algoritmo de relajación de [9].

Teorema 2.5.8 *Existe un único equilibrio de Nash normalizado para el cual el algoritmo de relajación converge si se satisface que:*

1. *X es un subconjunto compacto y convexo de $\mathbb{R}^N$.*

2. *La función de Nikaido-Isoda $\Psi : X \times X \longrightarrow \mathbb{R}$ es una función débilmente convexa-cóncava y $\Psi(\boldsymbol{x}, \boldsymbol{x}) = 0,\ \forall \boldsymbol{x} \in X$.*

3. *La función de respuesta óptima $Z(\boldsymbol{x})$ es single-valued y continua en X.*

4. *El residuo $r_{\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{y})$ es uniformemente continuo en X con respecto a $\boldsymbol{z}$, para todo $\boldsymbol{x}, \boldsymbol{y} \in X$.*

5. *Los residuos satisfacen:*

$$r_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{y}) - \mu_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{y}) \geq \beta(||\boldsymbol{x} - \boldsymbol{y}||)$$

   para todo $\boldsymbol{x}, \boldsymbol{y} \in X$, donde β es una función estrictamente creciente y $\beta(0) = 0$.

6. *La sucesión de términos de relajación $\{\alpha_s\}$ satisface que:*

   a) $\alpha_s > 0$.

   b) $\sum_{s=0}^{\infty} \alpha_s = \infty$.

   c) $\alpha_s \longrightarrow 0$ *cuando* $s \longrightarrow \infty$.

[5] El conjunto de los números naturales más el 0 (cero).

Capítulo 3

Inconsciencia en Juegos Estratégicos

En la teoría de juegos presentada hasta el momento, siempre se ha asumido pleno conocimiento de la situación por parte de los agentes. Esto es, cada jugador conoce al resto de los participantes y conoce las estrategias y funciones de utilidad tanto ajenas como propias. Pero esto no siempre es así en general, si el lector alguna vez ha jugado un juego de cartas, sabrá que en esos casos no se sabe a ciencia cierta con qué cartas cuentan los rivales, simplemente se sabe que son elementos de un mazo conocido menos las que él mismo tiene en sus manos. Además existen casos en los que los jugadores ni siquiera cuentan con esa información. Por ejemplo, un productor puede conocer la información pública de sus competidores (como precios, mercado en el cual actúan), pero existe información privada (como costos, infraestructura disponible) que ocasiona que el productor no sea consciente en su totalidad de la situación.

Éstos y una amplia variedad de ejemplos motivan la inclusión de cierta incerteza e inconsciencia en el análisis de la toma de decisiones, dando lugar así a juegos con incerteza y con inconsciencia respectivamente. Los dos tipos de juegos se diferencian entre sí en que, bajo el concepto de incerteza los jugadores sí conocen el juego y pueden concebir todos los estados relevantes del mismo a los cuales les pueden asignar probabilidad de ocurrencia (como en el ejemplo de las cartas), pero bajo la inconsciencia los jugadores pueden no concebir siquiera todos los estados (como en el ejemplo del productor). En las situaciones estratégicas que modelan los juegos, el último caso puede ser muy complejo debido a la asimetría de la inconsciencia de los jugadores, pues no sólo se considera la posibilidad de que un agente no sea consciente de algo, sino también la posibilidad de que un jugador crea que otro agente no es consciente de algo, la posibilidad de que un jugador crea que otro agente cree que él no es consciente de algo, y así siguiendo, lo que hace que se consideren, además, jerarquías de percepciones. Incluir este último concepto es muy complejo, por lo que, por simplicidad, hemos decidido no abordar el análisis de jerarquías y creencias en este trabajo. Sin embargo consideramos importante mencionarlo para comprender mejor el concepto, y la complejidad, de los juegos con inconsciencia.

En el presente trabajo, como su título indica, estudiaremos los juegos con inconsciencia. En este sentido, en las últimas décadas se han desarrollado modelos y conceptos de solución para este tipo de juegos. En particular, Halpern y Rêgo

[10] han desarrollado el concepto de *equilibrio de Nash generalizado para juegos con inconsciencia* como una generalización del equilibrio de Nash presentado en la sección anterior. Este concepto será el que analizaremos en este capítulo, basándonos principalmente en el trabajo de Yasuo Sasaki en [11]. En ambos trabajos, un juego con inconsciencia consiste en una colección de juegos estándar que representan al "verdadero" juego y a las distintas percepciones del mismo que tengan los jugadores, junto con un conjunto de funciones que representan las creencias de los agentes sobre las versiones que cada uno percibe. Además, aunque el equilibrio de Nash generalizado ha sido definido en general para juegos con inconsciencia dinámicos (o en forma extensiva), tanto en [11] como en este trabajo el foco estará puesto en juegos estáticos.

Como se ha mencionado, un jugador puede no conocer cualquier elemento del juego, esto incluye otros jugadores, estrategias (propias o ajenas), funciones de utilidad (propias o ajenas), etc. Por simplicidad, en este trabajo no consideraremos la inconsciencia de agentes o la incerteza de un jugador sobre los niveles de inconsciencia de los demás, sino que sólo analizaremos la inconsciencia en las acciones, es decir que los participantes del juego conocerán a los demás jugadores y sus utilidades, pero podrían no conocer los conjuntos de estrategias tanto propios como ajenos.

Como vimos en capítulos anteriores, el conjunto factible de estrategias de un jugador es un conjunto de entre el cual cada agente escoge su estrategia tal que optimiza su función de utilidad. Más adelante veremos que esto se traduce a un problema de optimización en el cual el espacio de estrategias de un jugador se transformará en la región factible de un problema, la cual representa las *restricciones* del problema de optimización que cada jugador debe resolver, de aquí el nombre "inconsciencia en las restricciones".

3.1. Juegos estáticos con inconsciencia

Aquí definiremos los juegos estáticos con inconsciencia basados en [11], que es básicamente una restricción del modelo dinámico de Halpern y Rêgo en [10] a juegos estáticos.

Para definir un juego estático con inconsciencia necesitamos primero una colección de juegos estratégicos, $G^0, \ldots, G^r$, de entre los cuales G^0 es el llamado ***juego objetivo***, es decir el juego que percibe el modelador como el "verdadero" juego. Como asumimos inconsciencia, está la posibilidad de que los agentes perciban distintos juegos entonces necesitamos otros juegos estratégicos $G^1, \ldots, G^r$ que describan los puntos de vista de los agentes, llamados ***juegos subjetivos***.

Las percepciones y creencias de los agentes estarán descriptas por las llamadas ***correspondencias de conocimiento***. La correspondencia de conocimiento del jugador i es una función $f_i : \{G^0, \ldots, G^r\} \longrightarrow \{G^0, \ldots, G^r\}$ y se interpreta del siguiente modo: si $f_i(G^k) = G^l$ esto significa "en G^k, el agente i cree que se está jugando G^l". Un uso más útil de esto sería cuando $f_i(G^0) = G^k$, pues se interpreta como "el jugador i cree que el verdadero juego es G^k". Además, con composiciones de estas funciones podemos describir percepciones de mayor orden, por ejemplo

$f_j \circ f_i(G^k) = G^l$ que se interpreta como "en G^k el jugador i cree que el jugador j cree que se está jugando G^l", lo cual nos puede servir para representar $f_j \circ f_i(G^0) = G^l$, que significa "el jugador i cree que el agente j cree que G^l es el verdadero juego". De esta manera podemos ir describiendo todas las percepciones y cadenas de creencias de los participantes. También decimos que G^l es ***alcanzable desde*** G^k si $G^k = G^l$ o $\exists i, j, \ldots, p \in \mathcal{N}$ tal que $f_i \circ f_j \circ \cdots \circ f_p(G^k) = G^l$.

Para visualizar estos órdenes de percepciones descriptos, podemos representarlos gráficamente por medio de *árboles de percepciones*. Para cada uno tendremos los siguientes elementos respresentados en la Figura 3.1:

- H_0: la raíz del árbol que contiene al juego G^k del cual va a partir nuestra jerarquía de percepciones.
- H_1: primer nivel que contiene los juegos que cada jugador cree que está jugando en G^k. Para simplificar la notación dentro de los nodos del árbol llamaremos $G_i^k := f_i(G^k)$
- H_2: nivel que contiene los juegos que cada jugador cree que los demás creen que están jugando en G^k. Aquí también, $G_{i,j}^k := f_j \circ f_i(G^k)$

y así siguiendo.

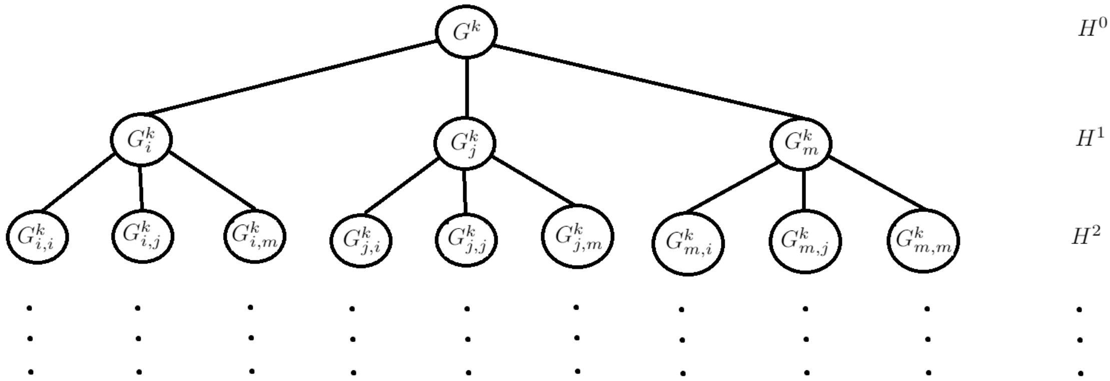


Figura 3.1: Árbol de percepciones

Introducidos estos conceptos, podemos definir formalmente lo siguiente:

Definición 3.1.1 *Un* ***juego estático con inconsciencia*** *es un par* $\Gamma^U = (\mathcal{G}, \mathcal{F})$ *donde:*

- $\mathcal{G} = \{G^0, \ldots, G^r\}$ *es un conjunto de juegos estáticos, donde, para cada* $k = 1, \ldots, r$, $G^k = (\mathcal{N}, A^k, u^k)$ *es un juego estratégico de los definidos en la sección anterior dados por:*
  - $\mathcal{N}$*, el conjunto finito de jugadores, es común para todo* $G^k \in \mathcal{G}$.
  - $A^k = \prod_{i \in \mathcal{N}} A_i^k$*, donde* A_i^k *es un conjunto finito de estrategias para el agente* i.

- $u^k = (u_i^k)_{i\in\mathcal{N}}$, *donde* $u_i^k : A^k \to \mathcal{R}^k$ *es la función de utilidad de* i *y* $\mathcal{R}^k$ *es el conjunto de posibles respuestas del juego* G^k.

- $\mathcal{F} = (f_i)_{i\in\mathcal{N}}$ *es una colección de correspondencias de conocimiento, donde* $f_i : \mathcal{G} \to \mathcal{G}$.

Definimos además, para cada $i \in \mathcal{N}$, $\mathcal{N}_{-i} = \mathcal{N} - \{i\}$, y a un elemento $a \in A^k$ lo llamamos ***perfil de acción***.

Utilizando la notación de esta definición, podemos definir un juego estático clásico como un caso particular de un juego con inconsciencia como: $G = (A, u)$ con conjunto de agentes $\mathcal{N}$, y estaría compuesto por $\mathcal{G} = \{G\}$ y $\mathcal{F} = (f_i)_{i\in\mathcal{N}}$, donde, para cada $i \in \mathcal{N}$, $f_i(G) = G$. Luego, para todo $i, j, \ldots, p \in \mathcal{N}$, $f_i(G) = G$, $f_j \circ f_i(G) = G$, $f_i \circ f_j \circ \cdots \circ f_p(G) = G$, se interpretan como "$i$ percibe que está jugando G", "i cree que j cree estar jugando G" y así siguiendo.

En un juego con inconsciencia los jugadores pueden no conocer todos los juegos en $\mathcal{G}$ o las correspondencias en $\mathcal{F}$, sin embargo, si $f_i(G^k) = G^l$ en algún punto del modelo, tiene sentido que i sea consciente de todos los juegos alcanzables desde G^l y todas las correspondencias involucradas. Esto podemos describirlo definiendo, para cada $i \in \mathcal{N}$, $\mathcal{V}_i(G^k) = (v_i^1(G^k), v_i^2(G^k), ..., v_i^m(G^k), ...)$, la ***jerarquía de percepciones de*** i ***en*** G^k, donde $v_i^1(G^k) = G^k$, $v_i^2(G^k) = (f_j(G^k))_{j\in\mathcal{N}_{-i}}, \ldots,$ $v_i^m(G^k) = (v_j^{m-1}(f_j(G^k)))_{j\in\mathcal{N}_{-i}}$, o sea, $v_i^1(G^k)$ es el juego que percibe i en G^k, $v_i^2(G^k)$ es el vector que contiene los juegos que cree i que perciben los demás jugadores en G^k, y así siguiendo.

Notemos que sólo tiene sentido considerar esta jerarquía de percepciones de i en G^k si existe $G^l \in \mathcal{G}$ tal que $f_i(G^l) = G^k$. Entonces, para cada $i \in \mathcal{N}$ definimos

$$\mathcal{G}_i := \{G^l \in \mathcal{G} \mid \exists k \text{ tal que } G^k \in \mathcal{G} \ \wedge \ f_i(G^k) = G^l\}$$

Luego, la jerarquía de percepciones de i tiene sentido sólo sobre los juegos en $\mathcal{G}_i$. Intuitivamente, $\mathcal{G}_i$ está formado por todos los juegos estáticos que i cree que son el verdadero juego en alguna parte del modelo.

Decimos que dos juegos son ***equivalentes*** si poseen los mismos elementos (conjuntos de estrategias y funciones de utilidad), a pesar de ser juegos distintos. También decimos que $\mathcal{V}_i(G^k)$ y $\mathcal{V}_i(G^l)$ son ***equivalentes*** si $\mathcal{V}_i(G^k)$ puede reformularse como $\mathcal{V}_i(G^l)$ reemplazando apropiadamente cada juego en $\mathcal{V}_i(G^k)$ por algún juego equivalente en $\mathcal{V}_i(G^l)$. Entonces, si $\mathcal{V}_i(G^k)$ y $\mathcal{V}_i(G^l)$ son equivalentes, esto significa que cada juego en $\mathcal{V}_i(G^k)$ es equivalente a cada juego en $\mathcal{V}_i(G^l)$ en cada orden de percepción del agente i.

En este trabajo asumiremos las siguientes condiciones respecto a la estructura de los juegos con inconsciencia.

- **C1**: Para cada $G^k, G^l \in \mathcal{G}$, si G^l es alcanzable desde G^k, para cada $i \in \mathcal{N}$, $A_i^l \subseteq A_i^k$.

- **C2**: Para cada $a \in A^0$, en cada $G^k \in \mathcal{G}$, si $a \in A^k$, entonces $u_i^k(a) = u_i^0(a), \ \forall i \in \mathcal{N}$.

- **C3**: Para cada $G^k \in \mathcal{G}$ y cada $i \in \mathcal{N}$, si $f_i(G^k) = G^l$, entonces $f_i(G^l) = G^l$.

- **C4**: Para cada $i \in \mathcal{N}$ y cada $G^k, G^l, \in \mathcal{G}$, si $\mathcal{V}_i(G^k)$ es equivalente a $\mathcal{V}_i(G^l)$, entonces $G^k = G^l$.
- **C5**: Todo $G^k \in \mathcal{G}$ es alcanzable desde G^0.

Notemos que las condiciones **C1** a **C3** son esenciales para dar sentido a la definición 3.1.1, mientras que las condiciones **C4** y **C5** son requerimientos técnicos para simplificar y hacer funcional el modelo. Esto es claro pues:

- **C1** establece que, si G^l es alcanzable desde G^k, entonces cada jugador i que percibe G^k, es consciente de todas las posibles acciones en G^l. Esta condición nos indica también que el jugador i no puede percibir que el jugador j percibe un juego que contenga estrategias que i desconozca, puesto que, de ser así, i estaría percibiendo otro juego y no el que percibe j, dejando así sin sentido la composición de correspondencias $f_j \circ f_i(G^k) = G^l$.
- **C2** nos dice que si, para algún juego $G^k \in \mathcal{G}$, un agente es consciente de una estrategia del juego objetivo, entonces la utilidad de los jugadores en dicha estrategia será la misma que en el juego objetivo. Esto nos indica que no estamos considerando la inconsciencia de los jugadores acerca de las utilidades de los demás o las propias, sólo inconsciencia de acciones, que se ve reflejada en la percepción de distintos juegos por parte de los agentes. Además nos brinda consistencia de la definición 3.1.1 pues si no se cumple **C2**, podemos encontrarnos ante la situación de que, jugar un perfil estratégico en G^0 nos brinda una respuesta distinta a jugar el mismo perfil en otro juego G^k, lo cual indicaría que, en G^k, los que perciben G^0 no son conscientes verdaderamente de todos los elementos del juego, contradiciendo así la definición de juegos estáticos.
- **C3** nos indica que si G^l es alcanzable desde G^k por algún mapeo, digamos $f_i \circ f_j \circ \cdots \circ f_p(G^k) = G^l$, entonces los jugadores $i, j, \ldots, p$ en G^l creerán que perciben completamente el juego que se está jugando. Esto es equivalente a decir que $f_i(G^k) = f_i \circ f_i(G^k)$, para cualquier $G^k \in \mathcal{G}_i$, esto se asume para evitar lidiar con situaciones en las cuales los agentes duden de sus propias percepciones, lo cual hace razonable definir el primer orden de percepciones de i, que es el juego que i percibe estar jugando.
- **C4** establece que, si la jerarquía de percepciones de i en G^k es equivalente a la jerarquía en G^l, entonces G^k y G^l son el mismo juego. Esto no permite que un jugador tenga jerarquías de percepciones equivalentes en dos juegos distintos, puesto que si esto ocurriera, entonces el equilibrio que estamos tratando de hallar dependería no sólo de la situación que se analiza, sino también de cómo se genera, lo cual añade complejidad al análisis y nos quita el carácter genérico de la definición.
- **C5** establece que todos los juegos subjetivos son alcanzables desde G^0. Esto nos permite preguntarnos "¿cuál juego creen los jugadores que es el verdadero?", además de darle sentido al análisis, puesto que el modelador pretende

analizar cómo se comportan los jugadores en el juego objetivo, a través de sus acciones en los juegos subjetivos percibidos.

En los juegos con inconsciencia, el modelador sabe que el "verdadero" juego que se está jugando es G^0 y, por las condiciones **C1**, **C2** y **C5**, tenemos que los juegos $G^1, \ldots, G^r$ "están contenidos" en G^0, lo que nos indica que, bajo estas condiciones, podemos asegurar que cualquier resultado de los juegos subjetivos se encuentra en el conjunto de posibles resultados de G^0.

3.2. Equilibrio de Nash generalizado en juegos con inconsciencia

En los juegos con inconsciencia nos interesa analizar cómo actúan los agentes en el juego objetivo, pero esto viene acompañado del análisis de cómo actúa cada uno a la luz de su propia percepción, cómo cree cada agente que los demás actuarán en pos de sus percepciones, y así siguiendo. Por esto, para definir una solución a un juego con inconsciencia necesitamos capturar no sólo el comportamiento de los agentes en el juego objetivo, sino también en los juegos subjetivos.

Teniendo esto en cuenta, para poder definir un equilibrio de Nash generalizado para juegos con inconsciencia debemos introducir algunos conceptos previos. En esta sección trabajaremos con estrategias mixtas para reflejar la generalidad de los conceptos, así que denotaremos por ΔA_i^k al conjunto de distribuciones de probabilidad en A_i^k. Sea $\Delta A^k := \prod_{i \in \mathcal{N}} \Delta A_i^k$, $\delta_i \in \Delta A_i^k$ y $\delta \in \Delta A^k$ son las acciones mixtas de i y un perfil de estrategias mixtas en G^k, respectivamente.

Definición 3.2.1 *Sea $\Gamma^U = (\mathcal{G}, \mathcal{F})$ un juego estratégico con inconsciencia, para cada $i \in \mathcal{N}$ y para cada $G^k \in \mathcal{G}_i$, $\sigma_{i,k} \in \Delta A_i^k$ es llamado* ***acción local*** *de i en G^k.*

Esta última definición debe interpretarse de la siguiente manera: $\sigma_{i,k}$ es la elección de i en G^k cuando, en algún punto del modelo, algún jugador percibe que i cree que G^k es el "verdadero" juego, por lo tanto, las acciones locales de un jugador son interpretadas como la creencia de algún agente acerca de su elección, y no exactamente la elección del jugador. La única excepción es el caso de las acciones locales usadas en el juego objetivo, esto es, las acciones locales de los jugadores en los juegos que ellos creen que son el "verdadero" juego. Este concepto no es necesario cuando tenemos pleno conocimiento del juego, puesto que las acciones locales serían las decisiones que efectivamente toma cada jugador en el juego. Ésto se puede ver más claro en el siguiente ejemplo.

Ejemplo 3.2.2

Si $f_i(G^0) = G^1$, entonces una acción local de i en G^1 es realmente su elección. Mientras que si $f_j \circ f_i(G^0) = G^2$, entonces una acción local de i en G^2 es lo que el jugador j cree que i elegirá, que puede diferir de la elección real de i.

Nota: Usaremos σ cuando queramos enfatizar que es una acción local y δ cuando nos queramos referir a acciones mixtas generales. Por conveniencia, cuando $\sigma_{i,k}$ sea una estrategia pura, asignándole probabilidad 1 a $a_i \in A_i^k$, simplemente denotaremos $\sigma_{i,k} = a_i$.

Definición 3.2.3 *Sea σ_i la combinación de las acciones locales del jugador i en todos los juegos en $\mathcal{G}_i$ y denotemos el conjunto de tales combinaciones por Σ_i. Entonces $\sigma_i \in \Sigma_i$ es llamado una* ***estrategia*** *del jugador i. Definiendo $\Sigma := \prod_{i \in \mathcal{N}} \Sigma_i$, $\sigma \in \Sigma$ es llamado* ***perfil de estrategias****.*

Usualmente, el modelador busca conocer el resultado que surge cuando los jugadores actúan de acuerdo a una estrategia, es decir, cuando se juega un perfil $\sigma \in \Sigma$. En vistas de representar esto definimos:

Definición 3.2.4 *Sea el perfil estratégico $\sigma \in \Sigma$, llamamos $(\delta_i)_{i\in\mathcal{N}} \in \Delta A^0$ su* ***resultado objetivo****, cuando $(\delta_i)_{i\in\mathcal{N}} \in \Delta A^0$ es tal que, para cada $i \in \mathcal{N}$, δ_i es equivalente a $\sigma_{i,k}$ si $f_i(G^0) = G^k$.*

Cuando hablamos de equivalencia de estrategias mixtas nos referimos a que, sean A_i^k y A_i^l los conjuntos de acciones para i en G^k y G^l respectivamente, si $\sigma_{i,k} \in A_i^k$ y $\sigma_{i,l} \in A_i^l$, se dice que $\sigma_{i,k}$ es ***equivalente*** a $\sigma_{i,l}$ si y sólo si tienen el mismo soporte y, además, las probabilidades en cada una son equivalentes. Lo denotaremos por $\sigma_{i,k} \equiv \sigma_{i,l}$.

Dado un perfil estratégico $\sigma \in \Sigma$, su resultado objetivo es un vector de resultados $(\delta_i)_{i\in\mathcal{N}} \in \Delta A^0$, tal que, cada vez que el jugador i cree que el verdadero juego es G^k, la elección del jugador i es la acción local $\sigma_{i,k}$. Intuitivamente, un resultado objetivo de un perfil estratégico se interpreta como el resultado realmente obtenido cuando la estrategia es jugada.

Además, dada la naturaleza de la inconsciencia debemos introducir la noción de ***utilidad esperada***. Cuando un jugador i juega $\sigma_{i,k}$, puede que un jugador j juegue $\sigma_{j,l} \neq \sigma_{j,k}$ si
$f_j(G^k) \neq G^k$, es decir, si j percibe un juego distinto a i en G^k. Generalmente, un jugador j juega $\sigma_{j,l}$ si $f_j(G^k) = G^l$ y, por lo visto en la sección anterior, en virtud de **C1**, $A_j^l \subseteq A_j^k$, por lo que al jugar $\sigma_{j,l}$, j está tomando una decisión equivalente a $\sigma_{j,k}$. Por lo tanto, la utilidad esperada por el jugador i en $G^k \in \mathcal{G}_i$ cuando se juega la estrategia $\sigma \in \Sigma$, denotada por $Eu_i^k(\sigma)$, se interpreta como la utilidad que el jugador i calcula que percibirá en G^k al jugar $\sigma_{i,k}$ dado que los demás jugadores $j \neq i$ juegan una acción equivalente a $\sigma_{j,k}$.

Dado un perfil estratégico $\sigma \in \Sigma$, denotamos por $\sigma_{-(i,k)}$ a todas las acciones locales que componen σ, salvo $\sigma_{i,k}$. Entonces el concepto de equilibrio está definido como:

Definición 3.2.5 *En un juego estático con inconsciencia, $\Gamma^U = (\mathcal{G}, \mathcal{F})$, $\sigma^* \in \Sigma$ es un* ***equilibrio de Nash generalizado*** *si y sólo si, para cada $i \in \mathcal{N}$, cada $G^k \in \mathcal{G}_i$ y cada $\sigma_{i,k} \in \Delta A_i^k$,*

$$Eu_i^k(\sigma^*) \geq Eu_i^k(\sigma_{i,k}, \sigma^*_{-(i,k)})$$

Al igual que en la definición 2.3.1, la idea del equilibrio de Nash generalizado para juegos con inconsciencia es un perfil estratégico tal que, si algún jugador se desvía unilateralmente de él, no puede incrementar su utilidad. En este sentido, la última definición es una generalización del equilibrio de Nash tradicional, para juegos con inconsciencia. La diferencia entre ambos conceptos radica en que el primero está calculado usando el pleno conocimiento con el que se cuenta, mientras que el último está definido basado en acciones locales, las cuales pueden ser interpretadas como creencias, más que como acciones reales. Con esto podemos decir que la primera definición es un equilibrio de estrategias, mientras que la segunda es más bien un equilibrio de creencias, pues cada agente escoge la mejor respuesta a su *creencia* en todas las circunstancias.

Recordemos que cada equilibrio de Nash clásico es un perfil a tal que $\forall i \in \mathcal{N}$, $a_i \in BR_i(a_{-i})$. Para esta generalización podemos sacar una conclusión análoga, definiendo

$$EBR_i^k(\sigma_{-(i,k)}) := \{\hat{\sigma}_{i,k} \mid Eu_i^k(\hat{\sigma}_{i,k}, \sigma_{-(i,k)}) \geq Eu_i^k(\tilde{\sigma}_{i,k}, \sigma_{-(i,k)}), \text{para cada } \tilde{\sigma_{i,k}}\}$$

el conjunto de mejores respuestas del jugador i dado que los demás juegan (acciones equivalentes a) $\sigma_{-(i,k)}$. Entonces si σ^* es un equilibrio de Nash generalizado, $\forall i \in \mathcal{N}$, $\forall k$ tal que $G^k \in \mathcal{G}_i$ y $\forall \sigma_{-(i,k)}$, $\sigma^* \in EBR_i^k(\sigma^*_{-(i,k)})$; es decir que las acciones locales de cada jugador i serán las mejores respuestas que i puede ofrecer dadas las elecciones que cree que los demás jugadores realizarán.

3.3. Ejemplos

Con el objetivo de ilustrar la inconsciencia en juegos estratégicos, presentaremos dos ejemplos simples. Queremos aquí que el lector note cómo dependen los equilibrios de Nash generalizados de las jerarquías de percepción de cada jugador. Veremos cómo las creencias de cada agente pueden afectar la solución y generar una desviación de los equilibrios de Nash del juego objetivo que el modelador percibe, provocando resultados inesperados. De esta manera pretendemos hacer explícita la diferencia entre un juego con información completa y un juego con inconsciencia.

3.3.1. Ejemplo 1

Consideraremos un juego de dos jugadores, Alice y Bob, denotados por A y B respectivamente, representado en la Tabla (1.1), donde las estrategias de Alice en el juego objetivo están representadas en las filas, mientras que las de Bob, en las columnas. Éste será el juego que el modelador perciba como "verdadero", es decir, el juego objetivo G^0. Asumiremos aquí que Alice ve este juego, mientras que cree que Bob no conoce una de sus acciones, a_3, y por lo tanto, cree que percibe el juego, al que llamaremos G^2, representado en la Tabla (1.2). Alice cree que G^2 es de conocimiento común.

Por otro lado, Bob es consciente de a_3 y ve el juego G^0. Más aún, él cree que Alice cree que él cree que G^2 es de conocimiento común.

	b_1	b_2	b_3
a_1	0, 2	3, 3	0, 2
a_2	2, 2	2, 1	2, 1
a_3	1, 0	4, 0	0, 1

(1.1)

	b_1	b_2	b_3
a_1	0, 2	3, 3	0, 2
a_2	2, 2	2, 1	2, 1

(1.2)

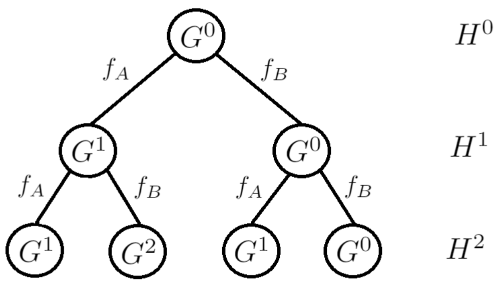
En este trabajo, modelaremos la situación con el juego estático con inconsciencia $\Gamma^U = (\mathcal{G}, \mathcal{F})$, con $\mathcal{G} = \{G^0, G^1, G^2\}$ y $\mathcal{F} = \{f_A, f_B\}$, donde $G^1 \equiv G^0$. Además, dadas las creencias de los jugadores, tenemos que $\mathcal{G}_A = \{G^1, G^2\}$ y $\mathcal{G}_B = \{G^0, G^2\}$. $f_A : \mathcal{G}_A \to \mathcal{G}_A$, $f_B : \mathcal{G}_B \to \mathcal{G}_B$ se comportan de la siguiente manera:

Para el árbol con raíz $H^0 = G^0$:

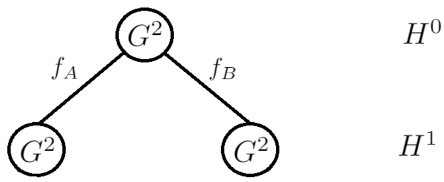
- El nivel H^1 contiene el juego que cada jugador cree estar jugando.
  - Alice cree que el verdadero juego es G^1, entonces $f_A(G^0) = G^1$.
  - Bob cree que el verdadero juego es G^0, entonces $f_B(G^0) = G^0$.
- El nivel H^2 contiene los juegos que cada jugador cree que el otro está jugando.
  - Alice cree que Bob cree que están jugando G^2, entonces $f_B \circ f_A(G^0) = G^2$.
  - Bob cree que Alice cree estar jugando G^1, entonces $f_A \circ f_B(G^0) = G^1$.
  - En virtud de la condición **C3**, las siguientes composiciones son: $f_A \circ f_A(G^0) = G^1$ y $f_B \circ f_B(G^0) = G^0$.

En $H^0 = G^2$, ambos jugadores creen que este juego es de conocimiento común, por lo que $f_A(G^2) = G^2$, $f_B(G^2) = G^2$, $f_A \circ f_A(G^2) = G^2$, $f_B \circ f_B(G^2) = G^2$, $f_A \circ f_B(G^2) = G^2$, $f_B \circ f_A(G^2) = G^2$, y así siguiendo.

Entonces, por lo visto en la Sección 2.2, los árboles que representan las jerarquías de percepción de cada jugador son:



(1.1)



(1.2)

Sólo hemos graficado estos niveles de los árboles por simplicidad, pues ya representamos todas las distintas creencias de los jugadores, los demás niveles sólo serán copias de los que están allí representados.

Busquemos entonces los equilibrios de Nash generalizados de Γ^U. Para esto, primero debemos determinar las acciones locales. En este caso trabajaremos con estrategias puras, es decir que $\sigma_{i,k} \in A_i^k$ es una acción pura que asigna probabilidad 1 a $a_i \in A_i^k$ (siguiendo la notación de las secciones anteriores), por lo que en nuestro caso tenemos que las acciones locales de Alice en el juego G^1 son $\{a_1, a_2, a_3\}$ y en G^2 son $\{a_1, a_2\}$ y las acciones locales de Bob son $\{b_1, b_2, b_3\}$ en G^0 y G^2.

Ahora, cada juego estático G^k podemos analizarlo por separado y encontrar así sus equilibrios de Nash. En virtud de esto tenemos que, G^0 y G^1 son claramente juegos con un único equilibrio de Nash, (a_2, b_1), mientras que G^2 tiene dos equilibrios de Nash, (a_2, b_1) y (a_1, b_2). Esperaríamos entonces que los equilibrios de Nash generalizados se encuentren entre estos equilibrios.

Sabemos que estamos buscando un perfil de estrategias tal que la utilidad esperada no aumente si nos desviamos unilateralmente de él, entonces debemos analizar las utilidades que cada participante cree que obtendrá con cada estrategia que juegue en cada juego y siguiendo su jerarquía de percepciones. Aquí realizaremos un procedimiento parecido a la inducción hacia atrás realizada en juegos iterados[1], pero sobre las jerarquías de percepción: primero analizaremos los equilibrios de Nash de G^2 y en virtud de las creencias de los jugadores iremos "hacia atrás" en los juegos estáticos (o "hacia arriba" en los árboles) para analizar las estrategias que utilizarán en los demás juegos. Esto quedará más claro una vez que hayamos encontrado los equilibrios generalizados.

Primero analizaremos las elecciones en G^2. Como dijimos antes, este juego tiene dos equilibrios de Nash, pero (a_2, b_1) es un equilibrio de Nash compartido con G^0 y G^1, por lo que, al momento de considerar las estrategias jugadas en G^2, siempre consideraremos esta, puesto que los participantes no se desviarán unilateralmente de esta decisión, en la "inducción hacia atrás".

Ahora, para analizar las elecciones en G^0 y G^1, tendremos que tener en cuenta las percepciones y creencias de cada jugador. Como Alice cree que Bob sólo ve G^2, entonces considera que Bob tiene dos opciones para elegir, b_1 o b_2, puesto que son las acciones locales de B en los equilibrios encontrados.

Si Bob elige b_1, entonces a Alice, que ve G^1, le convendrá escoger a_2, puesto que es la acción que más utilidades le genera dado que Bob eligió b_1. Así tenemos que $Eu_A^2(a_2, b_1) = 2$, que es mayor o igual a $Eu_A^2(a_1, b_1) = 0$ y además $Eu_A^1(a_2, b_1) = 2 \geq Eu_A^1(a_1, b_1), Eu_A^1(a_3, b_1)$. Para Bob tenemos que $Eu_B^2(a_2, b_1) = 2 \geq Eu_B^2(a_2, b_2) = 1 = Eu_B^2(a_2, b_3)$, al igual que en G^0.

Sean así, $\sigma_A^{*,1} = (\sigma_{A,1}^{*,1}, \sigma_{A,2}^{*,1}) = (a_2, a_2)$ y $\sigma_B^{*,1} = (\sigma_{B,0}^{*,1}, \sigma_{B,2}^{*,1}) = (b_1, b_1)$. Por definición, $\sigma_1^* = (\sigma_A^{*,1}, \sigma_B^{*,1})$ es un equilibrio de Nash generalizado de Γ^U, con equilibrio de Nash objetivo (a_2, b_1) en G^0.

Ahora, si Bob no juega b_1, entonces Alice espera que juegue b_2. Entonces como Alice ve G^1 y cree que Bob jugará b_2, entonces elegirá a_3, para maximizar su ganancia (siguiendo el razonamiento anterior). Como Bob ve G^0 y cree que Alice

[1] También llamados juegos en forma extensiva o dinámicos.

cree que ve G^2, asume que Alice elegirá a_3, en virtud de que cree que él escogerá b_2, entonces la mejor respuesta de Bob ante esto es escoger b_3, que le proporciona mayor utilidad dado que Alice elige a_3. En este caso, Alice espera conseguir una utilidad de 4, que es la mayor utilidad que puede conseguir según sus creencias, en ambos juegos, G^1 y G^2; así mismo, Bob espera conseguir una utilidad de 1, que es mayor que una ganancia de 0, que conseguiría eligiendo $b_i \neq b_3$.

Sean $\sigma_A^{*,2} = (\sigma_{A,1}^{*,2}, \sigma_{A,2}^{*,2}) = (a_3, a_2)$ y $\sigma_B^{*,2} = (\sigma_{B,0}^{*,2}, \sigma_{B,2}^{*,2}) = (b_3, b_1)$. Por definición, $\sigma_2^* = (\sigma_A^{*,2}, \sigma_B^{*,2})$ es un equilibrio de Nash generalizado de Γ^U, cuyo equilibrio de Nash objetivo es (a_3, b_3) en G^0.

Notemos que el conjunto de equilibrios de Nash del "verdadero" juego está contenido en el conjunto de equilibrios de Nash generalizados objetivo del juego con inconsciencia. En este ejemplo se pudo observar claramente que el nuevo concepto introducido en la definición 3.2.5 no es un equilibrio de Nash clásico, puesto que si Alice desviaba su elección y escogía la estrategia a_1, entonces hubiera obtenido una utilidad de 1, mayor a la que consigue jugando a_3, dada la elección del jugador B, pero condicionada por sus creencias sobre la inconsciencia de Bob, finalmente escoge a_3. Ésta es una clara diferencia con los equilibrios de Nash de la definición 2.3.1.

3.3.2. Ejemplo 2

Consideremos nuevamente la situación planteada en el **Ejemplo 1**, con los mismos jugadores, las mismas creencias y las mismas jerarquías de percepción. Consideremos que el juego objetivo es el representado en la Tabla (2.1), que $G^0 \equiv G^1$ y G^2 el juego representado en la Tabla (2.2).

	b_1	b_2	b_3
a_1	0, 2	3, 3	0, 2
a_2	2, 2	2, 1	2, 1
a_3	1, 0	4, 0	0, 1

(2.1)

	b_1	b_2	b_3
a_2	2, 2	2, 1	2, 1
a_3	1, 0	4, 0	0, 1

(2.2)

Luego, el juego que modela esta situación es $\Gamma^{U'} = (\mathcal{G}', \mathcal{F}')$, donde $\mathcal{G}' = \{G^0, G^1, G^2\}$ y $\mathcal{F}'$ análoga a $\mathcal{F}$ del Ejemplo 1.

En este caso tenemos que los juegos G^0 y G^1 son los mismos que el ejemplo anterior, mientras que el juego G^2 es un juego estático cuyo único equilibrio de Nash es (a_2, b_1). Entonces, tanto en G^0, como en G^1 y G^2, el perfil estratégico que satisface la definición 3.2.5 es (a_2, b_1). Es decir que el equilibrio generalizado de $\Gamma^{U'}$ es $\sigma^* = (\sigma_A^*, \sigma_B^*)$, donde $\sigma_A^* = (a_2, a_2)$ y $\sigma_B^* = (b_1, b_1)$, cuyo resultado objetivo es (a_2, b_1).

Por lo tanto, en este ejemplo, el conjunto de equilibrios de Nash del juego estático G^0 coincide con el conjunto de equilibrios de Nash generalizados objetivo de $\Gamma^{U'}$.

Capítulo 4

Métodos numéricos

En el fondo, en un juego estático, el problema que cada jugador debe resolver se reduce a minimizar una función $f(\mathbf{x})$ con $\mathbf{x}$ en un conjunto factible X, que está definido por ciertas condiciones de la forma $g(\mathbf{x}) \leq 0$. Éstos son los típicos elementos de un problema de optimización. Entonces daremos una breve introducción a conceptos básicos de optimización donde definiremos dos tipos de problemas clásicos de este área, junto con requerimientos necesarios y suficientes para que las soluciones sean óptimas, las llamadas condiciones Karush-Kuhn-Tucker (KKT). Luego, generalizaremos estos conceptos a problemas de equilibrio de Nash generalizado y estudiaremos una reformulación del problema de optimización con restricciones para resolver numéricamente el sistema de ecuaciones resultante de plantear las mencionadas condiciones.

Por último, presentaremos y resolveremos un juego conocido como *River Basin Pollution Game*, sacado de [12] y [9]. En ambos trabajos se lo resuelve utilizando dos métodos distintos desarrollados en los mismos, en este trabajo también utilizaremos un algoritmo propio para resolverlo. Luego, presentaremos 4 reversiones de dicho juego y trataremos de resolverlas numéricamente a través de los métodos presentados en este capítulo. Esto es, para cada versión desarrollaremos la reformulación del problema correspondiente y presentaremos los resultados numéricos obtenidos junto con las conclusiones correspondientes.

4.1. Preliminares de Optimización

Aquí introduciremos brevemente conceptos básicos de optimización y la notación correspondiente, para una mejor comprensión de los métodos utilizados más adelante en este capítulo.

Definición 4.1.1 *Dada una función $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, al problema de hallar un punto $x^* \in \mathbb{R}^n$ tal que $f(x^*) \leq f(x)$ $\forall x \in \mathbb{R}^n$ se lo conoce como* ***problema de optimización sin restricciones*** *y se lo denota por:*

$$\begin{array}{ll} minimizar & f(x) \\ s.a & x \in \mathbb{R}^n \end{array}$$

donde "s. a" es la abreviación de "sujeto a".

De manera análoga, sea $\Omega \subset \mathbb{R}^n$, definimos:

Definición 4.1.2 *Un* ***problema de optimización con restricciones*** *es el problema de encontrar un punto $x^* \in \Omega$ tal que $f(x^*) \leq f(x)\ \forall x \in \Omega$ y se lo denota por:*

$$\begin{array}{cc} minimizar & f(x) \\ s.a & x \in \Omega \end{array}$$

En estos problemas, a Ω se lo conoce como ***conjunto factible*** y para todo $x \in \mathbb{R}^n$, los puntos que cumplen $x \in \Omega$ se llaman ***puntos factibles*** del problema.

Definición 4.1.3 *Para ambos tipos de problemas, la función f se conoce como* ***función objetivo*** *y el mencionado punto x^*, como* ***mínimo global****. Si, en cambio, existe un número real $\varepsilon > 0$ tal que el punto x^* satisface que*
$f(x^*) \leq f(x)\ \forall x \in \mathbb{R}^n$ tal que $||x - x^*|| < \varepsilon$ para el problema sin restricciones, y $f(x^*) \leq f(x)\ \forall x \in \Omega$ tal que $||x - x^*|| < \varepsilon$ para el problema con restricciones, se dice que x^* es un ***mínimo local****. En este caso, $||\cdot||$ denota la norma euclídea en $\mathbb{R}^n$.*

En los problemas con restricciones en los que éstas son de la forma $g(x) \leq 0$, donde
$g : \mathbb{R}^n \longrightarrow \mathbb{R}^p$, o sea que $g_i(x) \leq 0\ \forall i = 1, \ldots, p$, diremos que g_i es una ***restricción activa en x*** si $g_i(x) = 0$ y diremos que está ***inactiva en x*** si $g_i(x) < 0$. En pos de esto, definiendo $I(x) = \{i \in \{1, \ldots, p\} | g_i(x) = 0\}$, diremos que $\hat{x}$ es un ***punto regular*** del problema

$$\begin{array}{cc} minimizar & f(x) \\ s.a & g_i(x) = 0, \forall x \in \Omega \ e\ i \in I(x) \end{array}$$

si $I(x) = \{i \in \{1, \ldots, p\} | g_i(\hat{x}) = 0\}$ y $\{\nabla g_1(\hat{x}), \ldots, \nabla g_p(\hat{x})\}$ es un conjunto linealmente independiente (L.I.).

En este capítulo analizaremos un problema particular, por lo que sólo introduciremos las condiciones KKT para asegurar que nuestra solución es óptima. Estas condiciones son relaciones entre las derivadas de la función objetivo y las derivadas de las funciones que definen las restricciones, por lo que el primer requisito para poder aplicarlas es que todas las funciones involucradas en nuestro problema sean, al menos, C^1.

Consideramos entonces el siguiente problema de optimización:

$$\begin{array}{cc} minimizar & f(x) \\ s.a & g(x) \leq 0 \end{array} \tag{4.1}$$

donde $g : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ y tanto f como g son, al menos, C^1.

Teorema 4.1.4 *Si x^* es un minimizador local regular del problema 4.1, entonces existen únicos $\mu^i \in \mathbb{R}\ \forall i \in I(x^*)$ tales que:*

1. $\nabla f(x^*) + \sum_{i \in I(x^*)} \mu^i \nabla g_i(x^*) = 0.$

2. $\mu^i g_i(x^*) = 0 \ \forall i \in I(x^*).$

3. $\mu^i \geq 0 \ \forall i \in I(x^*).$

Aquí, ∇f denota el gradiente de f y a los coeficientes $\mu^1, \ldots, \mu^p$ se los llama *multiplicadores de Lagrange*. A las condiciones 1 - 3 del teorema se las llama *condiciones de optimalidad KKT*, o *condiciones necesarias de optimalidad de primer orden*.

Además podemos definir:

Definición 4.1.5 *Dado el problema de minimización 4.1 y llamando μ al vector que contiene a los multiplicadores de Lagrange del mismo, llamamos* ***Lagrangiano*** *del problema a*

$$L(x, \mu) := f(x) + \sum_{i \in I(x)} \mu^i g_i(x)$$

4.2. Condiciones KKT para Equilibrios de Nash Generalizados

Para resolver los problemas de equilibrio de Nash generalizado, en este trabajo utilizaremos una versión más general de las condiciones KKT vistas en la sección de optimización, extraída de [13].

Recordemos el problema de equilibrio de Nash generalizado para n jugadores, es decir $\mathcal{N} = \{1, \ldots, n\}$, donde el jugador i controla el vector $x^i \in \mathcal{R}^{n_i}$ y quiere resolver el siguiente problema de optimización:

$$\begin{array}{cc} minimizar & u_i(x_i, x_{-i}) \\ s.a & g^i(x_i, x_{-i}) \leq 0 \end{array} \tag{4.2}$$

donde $u_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ y $g^i : \mathbb{R}^N \longrightarrow \mathbb{R}^m$, con $N = n_1 + \cdots + n_n$ denotando el total de variables y $m = m_1 + \cdots + m_n$ el total de restricciones del problema.

Sea $\mathbf{x}^*$ una solución del problema 4.2, es sabido que entonces las siguientes condiciones KKT se satisfacen para cada jugador $i = 1, \ldots, n$:

$$\nabla_{x_i} u_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) + \sum_{j=1}^{m_i} \mu_j^i \nabla_{x_i} g_j^i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) = 0$$

$$\mu_j^i \geq 0,$$
$$g_j^i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \leq 0,$$
$$\mu_j^i g_j^i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) = 0,$$

$\forall j = 1, \ldots, m_i$, donde $\mu^i = (\mu_1^i, \ldots, \mu_{n_i}^i)^T$ es el vector de multiplicadores de Lagrange del jugador i. Además, como estamos ante un problema convexo (las funciones $u_i(\cdot, x_{-i})$ y $g_j^i(\cdot, x_{-i})$ para todo $i = 1, \ldots, n$ y $j = 1, \ldots, m_i$) , sabemos que si un perfil estratégico $\mathbf{x}^*$ satisface dichas condiciones para todo jugador $i = 1, \ldots, n$ junto con un vector $\mu = (\mu^1, \ldots, \mu^n)^T$, entonces es una solución del problema generalizado. Por lo tanto, lo más natural es tratar de resolver el problema resolviendo simultáneamente los sistemas KKT de los jugadores involucrados.

Para cada $i = 1, \ldots, n$, sea $L^i(\mathbf{x}, \mu^i) := u_i(x_i, x_{-i}) + \sum_{j-1}^{m_i} \mu_j^i g_j^i(x_i, x_{-i})$ el Lagrangiano para el jugador i. Para simplificar la notación introduciremos el siguiente concepto.

Definición 4.2.1 *Sean* $\boldsymbol{F}(\boldsymbol{x}, \mu) := (\nabla_{x_i} L^i(\boldsymbol{x}, \mu^i))_{i=1}^n$ *y* $\boldsymbol{g}(\boldsymbol{x}) := (g^i(\boldsymbol{x}))_{i=1}^n$, *el* ***sistema KKT concatenado*** *puede ser reescrito como:*

$$\boldsymbol{F}(\boldsymbol{x}, \mu) = 0, \quad \mu \geq 0, \quad \boldsymbol{g}(\boldsymbol{x}) \leq 0, \quad \mu^T \boldsymbol{g}(\boldsymbol{x}) = 0$$

Para resolver este problema utilizaremos el método desarrollado en [13], el cual es un acercamiento muy utilizado que consiste en reducir nuestro sistema de ecuaciones mediante la inclusión de una ***función complementaria***, transformando así nuestro problema en uno de optimización sin restricciones.

Sea $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}$ una función tal que $\phi(a, b) = 0$ si y sólo si $a \geq 0$ y $b \geq 0$. Entonces el sistema KKT concatenado puede reescribirse como:

$$\mathbf{F}(\mathbf{x}, \mu) = 0, \quad \Phi(\mathbf{x}, \mu) = 0$$

donde:

$$\mathbf{\Phi}(\mathbf{x}, \mu) := \begin{pmatrix} \phi(\mu_1^1, -g_1^1(\mathbf{x})) \\ \vdots \\ \phi(\mu_{m_1}^1, -g_{m_1}^1(\mathbf{x})) \\ \phi(\mu_1^2, -g_1^2(\mathbf{x})) \\ \vdots \\ \phi(\mu_{m_n}^n, -g_{m_n}^n(\mathbf{x})) \end{pmatrix} \in \mathbb{R}^m$$

Existen muchas posibles funciones complementarias ϕ y en este trabajo usaremos una de las más destacadas, la función de Fischer-Burmeister, que está dada por:

$$\phi(a, b) := \sqrt{a^2 + b^2} - (a + b)$$

Una vez reformulado el sistema KKT concatenado como un nuevo sistema de ecuaciones, podemos resolverlo hallando un mínimo (global) de la siguiente ***función de mérito***:

$$\mathbf{\Theta}(\mathbf{x}, \mu) := \frac{1}{2} \left\| \begin{pmatrix} \mathbf{F}(\mathbf{x}, \mu) \\ \mathbf{\Phi}(\mathbf{x}, \mu) \end{pmatrix} \right\|^2$$

La ventaja de utilizar la función de Fischer-Burmeister es que nos brinda una función de mérito continuamente diferenciable. Por lo tanto podemos usar paquetes de optimización estándar para intentar hallar la solución numérica del problema sin restricciones

$$\begin{array}{cc} \textit{minimizar} & \mathbf{\Theta}(\mathbf{x}, \mu) \\ \textit{s.a} & \mathbf{x} \in \mathbb{R}^n, \mu \in \mathbb{R}^m \end{array}$$

Para intentar resolver el sistema KKT concatenado, primero debemos asegurarnos que se satisfagan dos condiciones:

- Una condición que garantice que los puntos estacionarios de $\boldsymbol{\Theta}$ son soluciones globales del problema de equilibrio de Nash generalizado.
- Una condición bajo la cual $\boldsymbol{\Theta}$ sea coercitiva.

Introduciremos ahora notación y definiciones que nos simplificarán el enunciado de dos resultados que nos definirán estas condiciones. Tenemos así:

- $\mathbf{E}(\mathbf{x}) := \begin{bmatrix} \nabla_{x_1} g^1(\mathbf{x}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \nabla_{x_n} g^n(\mathbf{x}) \end{bmatrix}$
- $\mathbf{J_x} f$ será la matriz Jacobiana respecto de $\mathbf{x}$ de la función f.
- $\mathbf{D}_\mu(\mathbf{x}, \mu) := diag(a^1(\mathbf{x}, \mu^1), \ldots, a^n(\mathbf{x}, \mu^n)) \in \mathbb{R}^{N \times N}$
- $\mathbf{D}_g(\mathbf{x}, \mu) := diag(b^1(\mathbf{x}, \mu^1), \ldots, b^n(\mathbf{x}, \mu^n)) \in \mathbb{R}^{N \times N}$

Donde $a^i(\mathbf{x}, \mu^i)$ y $b^i(\mathbf{x}, \mu^i)$ son vectores cuyas coordenadas están dadas por:

$$(a_j^i(\mathbf{x}, \mu_j^i), b_j^i(\mathbf{x}, \mu_j^i)) \begin{cases} = \frac{(\mu_j^i, -g_j^i(\mathbf{x}))}{\sqrt{(\mu_j^i)^2 + (g_j^i(\mathbf{x}))^2}} - (1,1) & si \quad (\mu_j^i, -g_j^i(\mathbf{x})) \neq (0,0) \\ \\ \in \mathbf{cl}(\mathbf{B}(0,1)) - (1,1) & si \quad (\mu_j^i, -g_j^i(\mathbf{x})) = (0,0) \end{cases}$$

$\forall i = 1, \ldots, n$ y $\forall j = 1, \ldots, n_i$, donde $\mathbf{cl}(\mathbf{B}(0{,}1))$ es la clausura de la bola cerrada de centro 0 y radio 1.

Así, [13] nos muestra que el gradiente de Θ viene dado por:

$$\nabla\Theta(\mathbf{x}, \mu) = \begin{bmatrix} \mathbf{J_x F}(\mathbf{x}, \mu) & \mathbf{E}(\mathbf{x}) \\ -\mathbf{D}_g(\mathbf{x}, \mu)\mathbf{J_x} g(\mathbf{x}) & \mathbf{D}_\mu(\mathbf{x}, \mu) \end{bmatrix} \begin{bmatrix} \mathbf{F}(\mathbf{x}, \mu) \\ \Phi(\mathbf{x}, \mu) \end{bmatrix}$$

Definición 4.2.2 *Una matiz cuadrada A se dice **P-matriz** si cada menor principal es positivo. Una **P_0-matriz** es la clausura de la clase de las P-matriz, es decir, el conjunto de matrices cuadradas tales que sus menores principales son no negativos. Los **menores principales** de una matriz cuadrada $A \in \mathbb{C}^{n \times n}$ son los determinantes de las submatrices cuadradas $B \in \mathbb{C}^{m \times m}$ de A, para todo $m = 1, \ldots, n$.*

El siguiente resultado, demostrado en [13], nos brinda condiciones suficientes para que un punto estacionario de u sea solución del problema generalizado.

Teorema 4.2.3 *Sea $(\overline{\boldsymbol{x}}, \overline{\mu}) \in \mathbb{R}^N \times \mathbb{R}^m$ un punto estacionario de Θ y supongamos que $\boldsymbol{J_x F}(\overline{\boldsymbol{x}}, \overline{\mu})$ es no singular y*

$$\boldsymbol{M}(\overline{\boldsymbol{x}}, \overline{\mu}) := \boldsymbol{J_x} g(\overline{\boldsymbol{x}}) \boldsymbol{J_x F}(\overline{\boldsymbol{x}}, \overline{\mu})^{-1} \boldsymbol{E}(\overline{\boldsymbol{x}})$$

es una P_0-matriz. Entonces $\overline{\boldsymbol{x}}$ es una solución del problema de equilibrio de Nash generalizado.

Nota: En [9] se asegura que este resultado es simple de verificar cuando el problema de optimización tiene una función objetivo cuadrática y restricciones lineales (como ocurre en los problemas analizados al final del capítulo), pues en esos casos, $M(\mathbf{x}, \mu)$ es independiente de $(\mathbf{x}, \mu)$.

Para la coercitividad, consideraremos un problema de equilibrio de Nash generalizado más general, pero tal que los que nosotros estamos estudiando son un caso particular de este, definido a través del siguiente problema de optimización:

$$\begin{array}{cc} \textit{minimizar} & u_i(x_i, x_{-i}) \\ s.a & g^i(x_i, x_{-i}) \leq 0, \;\; h^i(x_i) \leq 0 \end{array} \tag{4.3}$$

$\forall i = 1, \ldots, n$, donde las funciones g^i son las mismas de antes, $h^i_j : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}$ y las asumimos convexas en x_i. Entonces definimos el conjunto

$$X_0 := \{x \in \mathbb{R}^n \mid h^i(x_i) \leq 0 \; \forall i = 1, \ldots, n\}$$

que resulta ser cerrado y convexo.

Proposición 4.2.4 *Supongamos que $h^i_j : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}$ es convexa para todo $i = 1, \ldots, n$ y todo $j = 1, \ldots, n_i$. Supongamos que el conjunto X_0 es no vacío y acotado. Más aún, sea $\{(x^k, \mu^k)\}$ cualquier sucesión tal que $u(x^k, \mu^k) \leq u(x^0, \mu^0)$, $\forall k \in \mathbb{N}$. Entonces la sucesión $\{x^k\}$ es acotada.*

La demostración de esta proposición también podemos encontrarla en [13]. Este resultado nos asegura que cualquier método de descenso razonable tendrá un punto de acumulación, que se sabe, es un punto estacionario de u y, por lo tanto, una solución al problema 4.3.

Si bien el primer resultado es fácil de probar, la reformulación del problema por la función de Nikaido-Isoda presentada en el capítulo 3 nos asegura de forma más directa la existencia de solución. Por eso dicha reformulación es la que utilizaremos para probar el siguiente resultado:

Proposición 4.2.5 *Los juegos con inconsciencia presentados en la sección 4.3 de este capítulo tienen solución.*

Demostración:
Todas las versiones presentadas comparten, junto con el juego original o base, la función objetivo u_i, pero varían sus restricciones. Sin embargo, estas variaciones comparten una característica, que los conjuntos factibles X resultantes son de la forma $X = \{ax \leq b, \text{ con } a, b \in \mathbb{R}\}$. Entonces nuestro problema se reduce a demostrar que las hipótesis del teorema 2.5.8 se satisfacen para el siguiente problema:

$$\begin{array}{cc} \textit{minimizar} & u_i(x_i, x_{-i}) \\ s.a & a_1 x_1 \leq b_1, \;\; a_2 x_2 \leq b_2, \;\; a_3 x_3 \leq b_3 \end{array}$$

con a_i y $b_i \in \mathbb{R}$, $a_i \neq 0$, para todo $i = 1, 2, 3$.

Así nos queda $X = [0, \frac{b_1}{a_1}] \times [0, \frac{b_2}{a_2}] \times [0, \frac{b_3}{a_3}]$, que es un subconjunto convexo y cerrado de $\mathbb{R}^3$. Por otro lado, la función de Nikaido-Isoda en este caso nos queda:

$$\begin{aligned}\Psi(\mathbf{x},\mathbf{y}) =&(d_2+c_{21})(x_1^2-y_1^2)+[d_2(x_2+x_3)+c_{11}-d_1](x_1-y_1)\\&+(d_2+c_{22})(x_2^2-y_2^2)+[d_2(x_1+x_3)+c_{12}-d_1](x_2-y_2)\\&+(d_2+c_{23})(x_3^2-y_3^2)+[d_2(x_1+x_2)+c_{13}-d_1](x_3-y_3)\end{aligned}$$

Llamando $A_i = d_2 + c_{2i}$ y $B_i = d_2 \sum_{j \neq i} x_j + c_{1i} - d_1$, con $i = 1, 2, 3$, obtenemos:

$$\Psi(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{3} A_i(x_i^2 - y_i^2) + \sum_{i=1}^{3} B_i(x_i - y_i)$$

Es claro entonces que Ψ es débilmente convexa en X en la primera coordenada y débilmente cóncava en X, en la segunda. Por lo tanto, la función de Nikaido-Isoda de nuestro problema es débilmente convexa-cóncava.

Por otro lado, notemos que las funciones $Z(\mathbf{x})$ y $r_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ representan las mismas fórmulas que para el juego base, debido a que las fórmulas sólo dependen de la función objetivo. Además, la continuidad de $Z(\mathbf{x})$ depende principalmente de la estructura de la fórmula de Ψ. Entonces, como en [9] se ha demostrado que las hipótesis del teorema 2.5.8 que involucran sólo a estas funciones valen para el juego original, también valen en este caso.

Ahora, por [9] sabemos que:

$$r_{\mathbf{y}}(\mathbf{x},\mathbf{y}) = \frac{1}{2}\langle A(\mathbf{x},\mathbf{x})(\mathbf{x}-\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle + \mathbf{o}_1(||\mathbf{x}-\mathbf{y}||^2)$$

$$\mu_{\mathbf{y}}(\mathbf{x},\mathbf{y}) = \frac{1}{2}\langle B(\mathbf{x},\mathbf{x})(\mathbf{x}-\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle + \mathbf{o}_2(||\mathbf{x}-\mathbf{y}||^2)$$

donde $A(\mathbf{x},\mathbf{x}) = \Psi_{\mathbf{xx}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}\,=\,\mathbf{x}}$ y $B(\mathbf{x},\mathbf{x}) = \Psi_{\mathbf{yy}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}\,=\,\mathbf{x}}$

Así tenemos:

$$A(\mathbf{x},\mathbf{x}) = \begin{bmatrix} 2(d_2+c_{21}) & 2d_2 & 2d_2 \\ 2d_2 & 2(d_2+c_{22}) & 2d_2 \\ 2d_2 & 2d_2 & 2(d_2+c_{23}) \end{bmatrix}$$

$$B(\mathbf{x},\mathbf{x}) = \begin{bmatrix} -2(d_2+c_{21}) & 0 & 0 \\ 0 & -2(d_2+c_{22}) & 0 \\ 0 & 0 & -2(d_2+c_{23}) \end{bmatrix}$$

Entonces es claro que se satisfacen las condiciones 4 y 5 del teorema 2.5.8 en X también.

Por último, la condición 6 hace referencia al paso óptimo usado en el algoritmo que nos presenta el trabajo citado. Como nosotros no utilizaremos el algoritmo en este trabajo y las fórmulas de todas las funciones involucradas en la elección del paso son las mismas aquí que en el juego original, adoptaremos el paso usado en [9] y para el cual sabemos que se cumple la condición 6.

Por lo tanto podemos decir que el problema general planteado tiene solución. Como los juegos analizados en la siguiente sección son casos particulares de éste, entonces dichos problemas tienen solución.

□

4.3. Aplicación numérica

En este capítulo analizaremos un ejemplo sacado de [12] y estudiado también en [9]. Esta situación es un problema de equilibrio de Nash generalizado clásico, sin inconsciencia. En el presente trabajo utilizaremos el planteo de la situación y reversionaremos las restricciones para transformarlo en un problema de nuestro interés, tomando como juego objetivo el juego original presentado en los trabajos citados.

La situación que se plantea en todos los casos es la de tres empresas dedicadas a la producción de un bien determinado, digamos producción de celulosa de papel. Consideraremos que todas colocan sus productos en el mismo mercado, por lo tanto $\mathcal{N} = \{1, 2, 3\}$. Las tres compañías tienen sus fábricas situadas en la ribera de un río, al cual echan los residuos de sus procesos de fabricación. Para monitorear los niveles de concentración de tóxicos en el agua, la autoridad local dispone dos estaciones de monitoreo a lo largo del río y establece que los niveles de contaminación totales no pueden superar un cierto límite K_l en la estación l, con $l = 1, 2$. Consideraremos por simplicidad que estas tres empresas son la única fuente de contaminación del río.

Ahora, cada empresa i deberá decidir el nivel de producción x_i que minimice su función de costo u_i que viene dada por:

$$u_i(x) = (d_2(x_1 + x_2 + x_3) + c_{1i} + c_{2i}x_i - d_1)x_i$$

Al momento de minimizar esta función objetivo, en el juego original, se consideran las restricciones ambientales impuestas por el límite permitido en cada estación de monitoreo:

$$q_l(x) = \sum_{i=1}^{3} h_{il}e_ix_i \leq K_l, \quad l = 1, 2$$

Como las restricciones nos definen el conjunto factible de estrategias para cada jugador, con las definiciones de q_1 y q_2 queda más que claro que se trata de un problema de equilibrio de Nash generalizado. Las constantes involucradas en las fórmulas anteriores vienen dadas por la tabla 4.1, tomada de [12]:

Jugador i	c_{1i}	c_{2i}	e_i	h_{i1}	h_{i2}	d_1	d_2	K_1	K_2
1	0.10	0.01	0.50	6.5	4.583	3.0	0.01	100	100
2	0.12	0.05	0.25	5.0	6.250				
3	0.15	0.01	0.75	5.5	3.750				

Tabla 4.1: Constantes

Si bien este juego fue resuelto en más de un trabajo, nosotros aquí hemos optado por utilizar un procedimiento propio para resolverlo y decidimos utilizar la función "*least squares*" del paquete "*scipy.optimize*" en *Python 3.10.6* para la resolución computacional, por lo que al principio del análisis plantearemos nuevamente y resolveremos con nuestros métodos el juego objetivo. Utilizaremos los resultados de los trabajos citados para corroborar que nuestro método funcione correctamente.

Una vez resuelto esto, analizaremos 4 versiones del problema presentado. En una primera instancia contemplaremos la situación en la cual las tres empresas desconocen las restricciones ambientales. Naturalmente, si las tres empresas fabrican una cantidad infinita de productos no lograrán vender todo e incurrirán en pérdidas innecesarias. Por esta razón se plantean límites razonables de producción, M_1, M_2 y M_3, respectivamente, dependiendo de la infraestructura de cada una. Como no tienen el monopolio del mercado, deben contemplar la producción de la competencia en sus cálculos y en el planteo de sus límites, así como su propia infraestructura. Sin embargo, los recursos de cada empresa no son de conocimiento común, sólo consideraremos de público conocimiento la relación entre las capacidades productivas de cada una. Basados en los resultados del artículo [9], estas relaciones son las siguientes: La empresa 1 tiene 8 veces la capacidad productiva de la empresa 3 y la empresa 2, 6 veces la de la empresa 3.

En una segunda versión del problema, asumiremos que la empresa 1 ve el juego base y las otras dos perciben lo mismo que en la versión anterior. Entonces en este caso, la empresa 1 será consciente de la existencia de las estaciones de control, en cambio los otros dos jugadores considerarán como sus restricciones, los límites M_2 y M_3 definidos anteriormente. En esta versión no sólo analizaremos a qué solución llegamos, sino que también queremos saber si el conocimiento de la empresa 1 se transmite a las demás. En caso de ser así esto último, nos preguntaremos si su tamaño respecto a las demás tiene alguna influencia en este comportamiento o no.

Luego plantearemos una tercera versión en la cual dos empresas conocen las restricciones ambientales y una no. Aquí estudiaremos dos casos: uno en el que la empresa 3 será la que no conozca la existencia de las estaciones y otro en el que la empresa 1 sea la que desconoce esta información. En esta versión, al igual que en la anterior, queremos analizar si la información de las empresas que ven un juego equivalente al juego objetivo, se transmite a la que no lo es o no. En caso de cumplirse esto último, queremos ver también si esto se debe al tamaño de las empresas o a la cantidad de empresas conscientes respecto de la cantidad de empresas que no lo son.

Por último tomaremos nuevamente la primera versión en la cual las empresas no conocen las restricciones ambientales y se plantean los límites razonables M_1, M_2 y M_3. A diferencia de la primera versión, agregaremos un cuarto jugador, el estado, que participará en el juego como un ente regulador activo, es decir, si las empresas superan los límites de contaminación permitidos, aplicará una multa, aumentando así los costos de los productores y reduciendo sus ganancias. Como consideramos que los jugadores son seres razonables, al momento de recibir la multa, la naturaleza del juego cambiará, presentando así un ejemplo de que la línea entre los tipos de juego es muy delgada.

En todas las versiones, por simplicidad en los cálculos, asumiremos que cada jugador cree que los demás están viendo el mismo juego; que los demás creen que él está viendo el mismo juego; y así siguiendo. Por lo tanto en todas las definiciones de los juegos a continuación, $\mathcal{F} = (id, id, id)$, con $id : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ la función identidad dada por $id(\mathbf{y}) = \mathbf{y}$.

En el análisis de los juegos con inconsciencia, a continuación, esperamos corroborar la veracidad o falsedad de los siguientes enunciados:

- Las soluciones de los juegos con inconsciencia de las versiones presentadas no son solución del juego base.
- Las soluciones de las versiones 1 y 2 no satisfacen las restricciones ambientales.
- Las soluciones de los dos casos analizados en la versión 3 satisfacen las restricciones ambientales.

El código de los cálculos realizados en cada iteración de las versiones con inconsciencia puede encontrarse en el apéndice A, en el apartado de *Soluciones*.

Versión 0: Juego base

En este caso, para cada agente $i = 1, 2, 3$ el problema a resolver es:

$$\begin{array}{cc} minimizar & u_i(x_i, x_{-i}) \\ s.a & g^i(x_i, x_{-i}) \leq 0 \end{array}$$

donde $u_i(x_i, x_{-i})$ es la función de costo del jugador i mencionada en la introducción y $g^i(x_i, x_{-i}) = (g_1^i(x_i, x_{-i}), g_2^i(x_i, x_{-i}))^T$, con g_1^i y g_2^i dadas por

$$g_1^i(x_i, x_{-i}) = q_1(x_i, x_{-i}) - K_1 \quad y \quad g_2^i(x_i, x_{-i}) = q_2(x_i, x_{-i}) - K_2$$

Así las condiciones KKT para el jugador i serán:

$$\begin{cases} \nabla_{x_i} u_i(x_i, x_{-i}) + \sum_{j=1}^{2} \mu_j^i \nabla_{x_i} g_j^i(x_i, x_{-i}) = 0 \\ \mu_j^i \geq 0 \\ g_j^i(x_i, x_{-i}) \leq 0 \\ \mu_j^i g_j^i(x_i, x_{-i}) = 0 \end{cases}$$

$\forall j = 1, 2$, donde $\mu^i = (\mu_1^i, \mu_2^i)^T$ es el vector de multiplicadores de Lagrange para el jugador i.

Denotando ahora $L^{i,0}(\mathbf{x}, \mu^i) := u_i(x_i, x_{-i}) + \sum_{j=1}^{2} \mu_j^i g_j^i(x_i, x_{-i})$, definimos $\mathbf{F}^0(\mathbf{x}, \mu) := (\nabla_{x_1} L^{1,0}(\mathbf{x}, \mu^1), \nabla_{x_2} L^{2,0}(\mathbf{x}, \mu^2), \nabla_{x_3} L^{3,0}(\mathbf{x}, \mu^3))^T$ y podemos escribir el sistema KKT concatenado como:

$$\begin{cases} \mathbf{F}^0(\mathbf{x}, \mu) = 0 \\ \mu \geq 0 \\ \mathbf{g}(\mathbf{x}) \leq 0 \\ \mu^T \mathbf{g}(\mathbf{x}) = 0 \end{cases}$$

Luego, para cada par (i, j), con $i = 1, 2, 3$ y $j = 1, 2$, calculando la función complementaria evaluada en $(\mu_j^i, -g_j^i(\mathbf{x}))$ obtenemos:

$$\begin{aligned} \phi^0(\mu_1^1, -g_1^1(\mathbf{x})) &= \sqrt{(\mu_1^1)^2 + (-g_1^1(\mathbf{x}))^2} - (\mu_1^1 - g_1^1(\mathbf{x})) \\ &= \sqrt{(\mu_1^1)^2 + (K_1 - q_1(\mathbf{x}))^2} - \mu_1^1 - K_1 + q_1(\mathbf{x}) \end{aligned}$$

$$\begin{aligned} \phi^0(\mu_2^1, -g_2^1(\mathbf{x})) &= \sqrt{(\mu_2^1)^2 + (-g_2^1(\mathbf{x}))^2} - (\mu_2^1 - g_2^1(\mathbf{x})) \\ &= \sqrt{(\mu_2^1)^2 + (K_2 - q_2(\mathbf{x}))^2} - \mu_2^1 - K_2 + q_2(\mathbf{x}) \\ \phi^0(\mu_1^2, -g_1^2(\mathbf{x})) &= \sqrt{(\mu_1^2)^2 + (-g_1^2(\mathbf{x}))^2} - (\mu_1^2 - g_1^2(\mathbf{x})) \\ &= \sqrt{(\mu_1^2)^2 + (K_1 - q_1(\mathbf{x}))^2} - \mu_1^2 - K_1 + q_1(\mathbf{x}) \\ \phi^0(\mu_2^2, -g_2^2(\mathbf{x})) &= \sqrt{(\mu_2^2)^2 + (-g_2^2(\mathbf{x}))^2} - (\mu_2^2 - g_2^2(\mathbf{x})) \\ &= \sqrt{(\mu_2^2)^2 + (K_2 - q_2(\mathbf{x}))^2} - \mu_2^2 - K_2 + q_2(\mathbf{x}) \\ \phi^0(\mu_1^3, -g_1^3(\mathbf{x})) &= \sqrt{(\mu_1^3)^2 + (-g_1^3(\mathbf{x}))^2} - (\mu_1^3 - g_1^3(\mathbf{x})) \\ &= \sqrt{(\mu_1^3)^2 + (K_1 - q_1(\mathbf{x}))^2} - \mu_1^3 - K_1 + q_1(\mathbf{x}) \\ \phi^0(\mu_2^3, -g_2^3(\mathbf{x})) &= \sqrt{(\mu_2^3)^2 + (-g_2^3(\mathbf{x}))^2} - (\mu_2^3 - g_2^3(\mathbf{x})) \\ &= \sqrt{(\mu_2^3)^2 + (K_2 - q_2(\mathbf{x}))^2} - \mu_2^3 - K_2 + q_2(\mathbf{x}) \end{aligned}$$

Quedando así la función complementaria para el caso base:

$$\mathbf{\Phi}^0(\mathbf{x}, \mu) := \begin{pmatrix} \phi^0(\mu_1^1, -g_1^1(\mathbf{x})) \\ \phi^0(\mu_2^1, -g_2^1(\mathbf{x})) \\ \phi^0(\mu_1^2, -g_1^2(\mathbf{x})) \\ \phi^0(\mu_2^2, -g_2^2(\mathbf{x})) \\ \phi^0(\mu_1^3, -g_1^3(\mathbf{x})) \\ \phi^0(\mu_2^3, -g_2^3(\mathbf{x})) \end{pmatrix} \in \mathbb{R}^6$$

Así obtenemos que la función de mérito para este problema es:

$$\mathbf{\Theta}^0(\mathbf{x}, \mu) := \frac{1}{2} \left\| \begin{pmatrix} \mathbf{F}^0(\mathbf{x}, \mu) \\ \Phi^0(\mathbf{x}, \mu) \end{pmatrix} \right\|^2$$

Resolviendo esta formulación con el código presentado en el anexo A, obtenemos que la solución a nuestro problema es $\mathbf{x} = (21{,}119; 16{,}068; 2{,}733)^T$.

Por último, la solución dada en [9] es $\mathbf{x} = (21{,}14; 16{,}03; 2{,}728)^T$, por lo que el error relativo de nuestra aproximación es $E_r = 0{,}0016$, mucho menor a media unidad. Podemos entonces decir que nuestro algoritmo nos brinda una buena solución del juego.

Versión 1: Todos inconscientes

En este caso consideraremos que todos los jugadores desconocen las restricciones ambientales. Asumiremos que las empresas tienen la infraestructura necesaria para producir la cantidad que se propongan, pero debido a que no es conveniente que produzcan al máximo de su capacidad, se plantearán límites razonables de producción. Estos límites serán cotas superiores para los niveles de producción de cada uno y estarán motivados por la capacidad productiva de cada uno, más la consideración de la competencia en el mercado que le quitará algunos clientes, es decir, $x_i \leq M_i$, $\forall i = 1, 2, 3$, donde los M_i no necesariamente son todos iguales.

Además mencionamos antes que la relación entre las capacidades productivas de las empresas es de conocimiento común, por lo que, una vez planteado el límite de producción para sí mismo, cada jugador utilizará esta relación para establecer el límite de producción para los demás. Entonces las restricciones consideradas por cada agente son las siguientes:

Jugador 1:	Jugador 2:	Jugador 3:
$x_1 \leq M_1$	$x_1 \leq \frac{4}{3}M_2$	$x_1 \leq 8M_3$
$x_2 \leq \frac{3}{4}M_1$	$x_2 \leq M_2$	$x_2 \leq 6M_3$
$x_3 \leq \frac{1}{8}M_1$	$x_3 \leq \frac{1}{6}M_2$	$x_3 \leq M_3$

También asumiremos que cada jugador cree que los demás están jugando el mismo juego. De esta manera, para cada $i = 1, 2, 3$ el problema a resolver es:

$$\begin{array}{cc} minimizar & u_i(x_i, x_{-i}) \\ s.a & g^i(x_i, x_{-i}) \leq 0 \end{array}$$

donde $g^i(x_i, x_{-i}) = (g^i_1(x_i, x_{-i}), g^i_2(x_i, x_{-i}), g^i_3(x_i, x_{-i}))^T$ y éstas están dadas por:

$$g^1(x_1, x_2, x_3) = (x_1 - M_1, x_2 - \frac{3}{4}M_1, x_3 - \frac{1}{8}M_1)^T$$

$$g^2(x_1, x_2, x_3) = (x_1 - \frac{4}{3}M_2, x_2 - M_2, x_3 - \frac{1}{6}M_2)^T$$

$$g^3(x_1, x_2, x_3) = (x_1 - 8M_3, x_2 - 6M_3, x_3 - M_3)^T$$

Definimos así los juegos G^0 como el juego base enunciado en la versión 0 y $G^1 = (A^1, u^1)$, $G^2 = (A^2, u^2)$, $G^3 = (A^3, u^3)$, donde para cada $j = 1, 2, 3$:

- $A^j = \{x \in \mathbb{R}^3 \mid g^j(x) \leq 0\}$.

- $u^j = (u_1, u_2, u_3)$.

Entonces, formalmente el juego definido en este caso es $\Gamma^1 = (\mathcal{G}^1, \mathcal{F})$, donde $\mathcal{G}^1 = \{G^0, G^1, G^2, G^3\}$.

Así, las condiciones KKT para el jugador i serán:

$$\begin{cases} \nabla_{x_i} u_i(x_i, x_{-i}) + \sum_{j=1}^{3} \mu_j^i \nabla_{x_i} g_j^i(x_i, x_{-i}) = 0 \\ \mu_j^i \geq 0 \\ g_j^i(x_i, x_{-i}) \leq 0 \\ \mu_j^i g_j^i(x_i, x_{-i}) = 0 \end{cases}$$

$\forall j = 1, 2, 3$.

Denotando $L^{i,1}(\mathbf{x}, \mu^i) := u_i(x_i, x_{-i}) + \sum_{j=1}^{3} \mu_j^i g_j^i(x_i, x_{-i})$ y definiendo $\mathbf{F}^1(\mathbf{x}, \mu) := (\nabla_{x_1} L^{1,1}(\mathbf{x}, \mu^1), \nabla_{x_2} L^{2,1}(\mathbf{x}, \mu^2), \nabla_{x_3} L^{3,1}(\mathbf{x}, \mu^3))^T$ el sistema de condiciones KKT concatenadas queda:

$$\begin{cases} \mathbf{F}^1(\mathbf{x}, \mu) = 0 \\ \mu \geq 0 \\ \mathbf{g}(\mathbf{x}) \leq 0 \\ \mu^T \mathbf{g}(\mathbf{x}) = 0 \end{cases}$$

Entonces, para cada par (i, j), con $i, j = 1, 2, 3$, calculando la función complementaria evaluada en $(\mu_j^i, -g_j^i(\mathbf{x}))$ obtenemos:

$$\begin{aligned} \phi^1(\mu_1^1, -g_1^1(\mathbf{x})) &= \sqrt{(\mu_1^1)^2 + (-g_1^1(\mathbf{x}))^2} - (\mu_1^1 - g_1^1(\mathbf{x})) \\ &= \sqrt{(\mu_1^1)^2 + (M_1 - x_1)^2} - \mu_1^1 - M_1 + x_1 \end{aligned}$$

$$\begin{aligned} \phi^1(\mu_2^1, -g_2^1(\mathbf{x})) &= \sqrt{(\mu_2^1)^2 + (-g_2^1(\mathbf{x}))^2} - (\mu_2^1 - g_2^1(\mathbf{x})) \\ &= \sqrt{(\mu_2^1)^2 + (\frac{3}{4}M_1 - x_2)^2} - \mu_2^1 - \frac{3}{4}M_1 + x_2 \\ \phi^1(\mu_3^1, -g_3^1(\mathbf{x})) &= \sqrt{(\mu_3^1)^2 + (-g_3^1(\mathbf{x}))^2} - (\mu_3^1 - g_3^1(\mathbf{x})) \\ &= \sqrt{(\mu_3^1)^2 + (\frac{1}{8}M_1 - x_3)^2} - \mu_3^1 - \frac{1}{8}M_1 + x_3 \\ \phi^1(\mu_1^2, -g_1^2(\mathbf{x})) &= \sqrt{(\mu_1^2)^2 + (-g_1^2(\mathbf{x}))^2} - (\mu_1^2 - g_1^2(\mathbf{x})) \\ &= \sqrt{(\mu_1^2)^2 + (\frac{4}{3}M_2 - x_1)^2} - \mu_1^2 - \frac{4}{3}M_2 + x_1 \end{aligned}$$

$$\begin{aligned} \phi^1(\mu_2^2, -g_2^2(\mathbf{x})) &= \sqrt{(\mu_2^2)^2 + (-g_2^2(\mathbf{x}))^2} - (\mu_2^2 - g_2^2(\mathbf{x})) \\ &= \sqrt{(\mu_2^2)^2 + (M_2 - x_2)^2} - \mu_2^2 - M_2 + x_2 \end{aligned}$$

$$\begin{aligned}
\phi^1(\mu_3^2, -g_3^2(\mathbf{x})) &= \sqrt{(\mu_3^2)^2 + (-g_3^2(\mathbf{x}))^2} - (\mu_3^2 - g_3^2(\mathbf{x})) \\
&= \sqrt{(\mu_3^2)^2 + (\frac{1}{6}M_2 - x_3)^2} - \mu_3^2 - \frac{1}{6}M_2 + x_3 \\
\phi^1(\mu_1^3, -g_1^3(\mathbf{x})) &= \sqrt{(\mu_1^3)^2 + (-g_1^3(\mathbf{x}))^2} - (\mu_1^3 - g_1^3(\mathbf{x})) \\
&= \sqrt{(\mu_1^3)^2 + (8M_3 - x_1)^2} - \mu_1^3 - 8M_3 + x_1 \\
\phi^1(\mu_2^3, -g_2^3(\mathbf{x})) &= \sqrt{(\mu_2^3)^2 + (-g_2^3(\mathbf{x}))^2} - (\mu_2^3 - g_2^3(\mathbf{x})) \\
&= \sqrt{(\mu_2^3)^2 + (6M_3 - x_2)^2} - \mu_2^3 - 6M_3 + x_2 \\
\phi^1(\mu_3^3, -g_3^3(\mathbf{x})) &= \sqrt{(\mu_3^3)^2 + (-g_3^3(\mathbf{x}))^2} - (\mu_3^3 - g_3^3(\mathbf{x})) \\
&= \sqrt{(\mu_3^3)^2 + (M_3 - x_3)^2} - \mu_3^3 - M_3 + x_3
\end{aligned}$$

Luego, la función complementaria de este problema es un elemento de $\mathbb{R}^9$ dado por:

$$\mathbf{\Phi}^1(\mathbf{x}, \mu) := \begin{pmatrix} \phi^1(\mu_1^1, -g_1^1(\mathbf{x})) \\ \phi^1(\mu_2^1, -g_2^1(\mathbf{x})) \\ \phi^1(\mu_3^1, -g_3^1(\mathbf{x})) \\ \phi^1(\mu_1^2, -g_1^2(\mathbf{x})) \\ \phi^1(\mu_2^2, -g_2^2(\mathbf{x})) \\ \phi^1(\mu_3^2, -g_3^2(\mathbf{x})) \\ \phi^1(\mu_1^3, -g_1^3(\mathbf{x})) \\ \phi^1(\mu_2^3, -g_2^3(\mathbf{x})) \\ \phi^1(\mu_3^3, -g_3^3(\mathbf{x})) \end{pmatrix}$$

Quedando así la función de mérito de este problema:

$$\mathbf{\Theta}^1(\mathbf{x}, \mu) := \frac{1}{2} \left\| \begin{pmatrix} \mathbf{F}^1(\mathbf{x}, \mu) \\ \mathbf{\Phi}^1(\mathbf{x}, \mu) \end{pmatrix} \right\|^2$$

En este caso, tomaremos inicialmente los límites $M_1 = 30, M_2 = 22$ y $M_3 = 4$, que sabemos que son cotas superiores de los resultados del juego objetivo. Además nuestro algoritmo nos exige una aproximación inicial de $(x_1, x_2, x_3)^T$ y de $(\mu^1, \mu^2, \mu^3)^T$, que tomaremos como $(20, 16, 2)^T$ y $(0, 0, 0)^T$ respectivamente. Resolviendo esto obtenemos los siguientes resultados:

JUGADOR 1:	$x^{*,1} = (30; 21{,}20; 3{,}6)^T$
JUGADOR 2:	$x^{*,2} = (29{,}26; 21{,}27; 3{,}52)^T$
JUGADOR 3:	$x^{*,3} = (32; 21; 4)^T$

Esto representa los siguientes costos para las empresas: 61.55 para la empresa 1, 27.14 para la empresa 2 y 8.95 para la empresa 3.

Pero realmente se jugará el perfil $x^R = (x_1^R, x_2^R, x_3^R)^T = (30, 22, 4)^T$. Con los siguientes costos para las empresas: 61.41 para la empresa 1, 26.88 para la empresa 2 y 9.03 para la empresa 3.

Una vez jugado este perfil, los jugadores se encuentran con que los demás no han jugado lo esperado. Entonces deben decidir cómo actuar para la segunda ronda del juego, en base a la nueva información que ahora poseen. Como el costo para las dos primeras empresas es menor del esperado, entonces creyendo que pueden aumentar su producción sin incurrir en costos extras, recalcularán su equilibrio. En cambio la tercera empresa debe recalcular su equilibrio porque ha incurrido en un costo mayor al esperado.

Como no hay restricciones en la complejidad de los cálculos para elegir una estrategia, cada productor i modificará su elección del nivel de producción x_i como se explica a continuación.

Definimos $f_i : \mathbb{R} \longrightarrow \mathbb{R}$ por

$$f_i(z) = u_i(x^R) - u_i(x^{N,i})$$

donde $x^{N,i} = (z, (x^R)_{-i})$. Esta función nos brinda la diferencia de costos entre el afrontado al jugar x^R y $x^{N,i}$. Entonces, en cada iteración del juego, los agentes buscarán el $x^{N,i}$ que minimice la función f.

Sea m_i el mínimo de f_i, definimos $M_{i,nuevo} = m_i$. Como bajar la producción obviamente llevará a minimizar los costos, pero también a la reducción de ganancias, consideraremos que los productores modifican sus límites sólo si $M_{i,nuevo} \geq M_i$. Si el límite cambia, las empresas vuelven a calcular su equilibrio replanteando el sistema KKT concatenado, teniendo en cuenta en cada paso y para cada $i = 1, 2, 3$ que el participante i cree que los demás agentes se comportarán según lo esperado. Esto es, si según el jugador i algún jugador $j \neq i$ debe bajar su producción para aumentar su ganancia, entonces lo hará. Esto tiene sentido pues al aumentar la producción x_i, debido a la función de utilidad considerada, los costos para los demás participantes aumentarán si no disminuyen su producción. Así, para poder recalcular el equilibrio, primero deben actualizar los límites M_i, obteniendo así:

$$M_1 = 66, \quad M_2 = 22, \quad M_3 = 59$$

Recalculando entonces los equilibrios se obtiene:

$$x^{*,1} = (66, 21{,}27, 4)^T$$
$$x^{*,2} = (30, 21{,}27, 4)^T$$
$$x^{*,3} = (30, 21{,}27, 53{,}68)^T$$

Entonces el resultado de la segunda iteración es $x^R = (66; 21{,}27; 53{,}68)^T$.

Haciendo un análisis de costo, al aumentar las producciones es claro que los costos serán mayores, sin embargo los reales son bajos en comparación con los esperados, por lo que las empresas no tendrán esto en cuenta al momento de decidir y aumentarán su producción. Sin embargo, nuevamente los competidores jugaron estrategias inesperadas, por lo que se deberán recalcular los equilibrios. Obteniendo así:

$$x^{*,1} = (66; 21; 54)^T$$

$$x^{*,2} = (66; 21{,}27; 54)^T$$

$$x^{*,3} = (66; 21; 53{,}68)^T$$

La empresa 3 ha obtenido que debe bajar su producción y establecimos que eso no es una actitud aceptable, entonces el resultado de la tercera iteración es $x^R = (66; 21{,}27; 54)^T$.

En esta última iteración, las tres empresas han llegado al perfil efectivamente jugado. Si bien la producción real de la segunda empresa no es estrictamente la misma esperada por sus competidores, contemplamos una tolerancia de $\pm 0{,}5$ unidades debido a que al mercado se sacan unidades enteras, no fraccionadas. Entonces ningún jugador recalculará sus equilibrios, ya que sus creencias no fueron refutadas.

Llegamos así al resultado de este juego, $\mathbf{x} = (66; 21{,}27; 54)^T$. Claramente este no es el resultado del juego base. Más aún, $q_1(\mathbf{x}) = 462{,}53 > K_1$ y $q_2(\mathbf{x}) = 335{,}46 > K_2$. Entonces podemos concluir que, si ningún jugador es consciente de la existencia de las estaciones de control y no existe algún agente externo que induzca de alguna manera esta información, no se logrará llegar a una solución del juego original.

Versión 2: Uno consciente

En esta versión consideraremos que la empresa 1 conoce las restricciones ambientales, por lo que ve un juego equivalente al juego base, es decir, $G^1 \equiv G^0$. Por otro lado, asumimos que los jugadores 2 y 3 desconocen dichas restricciones y ven los juegos G^2 y G^3 definidos en la sección anterior. Supondremos también que cada jugador cree que los demás ven el mismo juego, que cree que ellos creen que él percibe el mismo juego, y así siguiendo.

Puesto que dos jugadores no conocen el juego original, se jugarán perfiles de estrategias inesperados. Ante esta situación asumiremos que los jugadores 2 y 3 siguen el mismo curso de acción que en la versión 1 para actualizar sus límites M_2 y M_3 y recalcular sus equilibrios. En cambio, el jugador 1, que está totalmente convencido de que ve el verdadero juego, no modifica su elección, incluso si sus competidores escogen estrategias muy distintas a las esperadas.

Las funciones g^i que definen las restricciones para el jugador i deben estar definidas de $\mathbb{R}^3$ en $\mathbb{R}$. Como sabemos, las restricciones del juego objetivo vienen dadas por dos fórmulas, por lo que definiremos $g^1(\mathbf{x}) = (g^1_1(\mathbf{x}), g^1_2(\mathbf{x}), g^1_3(\mathbf{x}))^T$, donde $g^1_1(\mathbf{x}) = q_1(\mathbf{x}) - K_1$, $g^1_2(\mathbf{x}) = q_2(\mathbf{x}) - K_2$ y $g^1_3(\mathbf{x}) = 0$. Mientras que para los jugadores 2 y 3, las funciones g^2 y g^3 serán las mismas que en la versión anterior.

Formalmente, el juego con inconsciencia que modela esta situación es $\Gamma^2 = (\mathcal{G}^2, \mathcal{F})$, donde $\mathcal{G}^2 = \{G^0, G^1, G^2, G^3\}$ con $G^i = (A^i, u^i)$ dado por
$A^i = \{x \in \mathbb{R}^3 \mid g^i \leq 0\}$ y $u^i = (u_1, u_2, u_3)$.

A continuación desarrollaremos en detalle las cuentas para el jugador 1, pues son las que cambian con respecto a la situación anterior.

Las condiciones KKT para el jugador 1 quedan:

$$\begin{cases} \nabla_{x_1} u_1(x_1, x_{-1}) + \sum_{j=1}^{3} \mu_j^1 \nabla_{x_1} g_j^1(x_1, x_{-1}) = 0 \\ \mu_j^1 \geq 0 \\ g_j^1(x_1, x_{-1}) \leq 0 \\ \mu_j^1 g_j^1(x_1, x_{-1}) = 0 \end{cases}$$

$\forall j = 1, 2, 3.$

Como la tercera función coordenada de g^1 es constantemente cero, no importa el valor que pongamos a μ_3^1, en particular podemos tomar $\mu_3^1 \geq 0$ y las funciones complementarias para este jugador resultan:

$$\phi^2(\mu_1^1, -g_1^1(\mathbf{x})) = \phi^0(\mu_1^1, -g_1^1(\mathbf{x})) = \sqrt{(\mu_1^1)^2 + (K_1 - q_1(\mathbf{x}))^2} - \mu_1^1 - K_1 + q_1(\mathbf{x})$$

$$\phi^2(\mu_2^1, -g_2^1(\mathbf{x})) = \phi^0(\mu_2^1, -g_2^1(\mathbf{x})) = \sqrt{(\mu_2^1)^2 + (K_2 - q_2(\mathbf{x}))^2} - \mu_2^1 - K_2 + q_2(\mathbf{x})$$

$$\phi^2(\mu_3^1, -g_3^1(\mathbf{x})) = |\mu_3^1| - \mu_3^1 = \mu_3^1 - \mu_3^1 = 0$$

Luego, definiendo $L^{1,2}(\mathbf{x}, \mu^1) = L^{1,0}(\mathbf{x}, \mu^1)$, y para cada $i = 2, 3$ y $j = 1, 2, 3$, $L^{i,2}(\mathbf{x}, \mu^i) = L^{i,1}(\mathbf{x}, \mu^i)$, $\phi^2(\mu_j^i, -g_j^i(\mathbf{x})) = \phi^1(\mu_j^i, -g_j^i(\mathbf{x}))$, las funciones complementaria y de mérito de este problema son:

$$\mathbf{\Phi}^2(\mathbf{x}, \mu) := \begin{pmatrix} \phi^2(\mu_1^1, -g_1^1(\mathbf{x})) \\ \phi^2(\mu_2^1, -g_2^1(\mathbf{x})) \\ 0 \\ \phi^2(\mu_1^2, -g_1^2(\mathbf{x})) \\ \phi^2(\mu_2^2, -g_2^2(\mathbf{x})) \\ \phi^2(\mu_3^2, -g_3^2(\mathbf{x})) \\ \phi^2(\mu_1^3, -g_1^3(\mathbf{x})) \\ \phi^2(\mu_2^3, -g_2^3(\mathbf{x})) \\ \phi^2(\mu_3^3, -g_3^3(\mathbf{x})) \end{pmatrix} \in \mathbb{R}^9$$

$$\mathbf{\Theta}^2(\mathbf{x}, \mu) := \frac{1}{2} \left\| \begin{pmatrix} \mathbf{F}^2(\mathbf{x}, \mu) \\ \mathbf{\Phi}^2(\mathbf{x}, \mu) \end{pmatrix} \right\|^2$$

Inicialmente tomaremos $M_2 = 22,\ M_3 = 4,\ x_{inicial} = (20; 16; 2)^T$ y $\mu_{inicial} = (0; 0; 0)^T$. Resolviendo este problema, obtenemos los siguientes resultados:

Jugador 1: $x^{*,1} = (21{,}119; 16{,}068; 2{,}733)^T$
Jugador 2: $x^{*,2} = (29{,}26; 21{,}27; 3{,}52)^T$
Jugador 3: $x^{*,3} = (32; 21; 4)^T$

Entonces realmente se jugará el perfil $x^R = (x_1^R, x_2^R, x_3^R)^T = (21{,}119; 21{,}27; 4)^T$.

Como ninguno de los jugadores juega lo que el otro espera de ellos, se recalcularán nuevamente los equilibrios, respetando los supuestos mencionados anteriormente. Procediendo iterativamente como en la versión 1, llegamos a que el resultado de este juego es el perfil

$$\mathbf{x} = (21{,}119; 21{,}27; 53{,}68)^T$$

Nuevamente nos encontramos con que la solución de esta situación no es el resultado del juego base. Más aún, $q_1(\mathbf{x}) = 316{,}66 > K_1$ y $q_2(\mathbf{x}) = 232{,}61 > K_2$.

Notemos sin embargo, que esta vez se redujo la concentración de tóxicos en el río. Entonces cabe preguntarnos si el hecho de que más empresas sean conscientes de las restricciones ambientales nos conducirá a satisfacer dichas cotas o no. Esto motiva el planteo del siguiente juego.

Versión 3: Dos conscientes

En esta versión consideraremos que dos empresas conocen las restricciones ambientales y una no. Entonces, en términos de cuentas, dos empresas obtendrán los resultados de la misma forma que en el juego base y la otra, de la misma forma que lo hacía en la versión 1. Por lo tanto no presentaremos aquí los detalles de las cuentas, sino que nos enfocaremos en cambio, en analizar los resultados obtenidos intentando responder preguntas como ¿se propaga el conocimiento del juego base de los jugadores conscientes a los demás?¿Tiene alguna influencia en esta propagación el tamaño de las empresas conscientes?¿Tiene alguna influencia en este resultado la cantidad de empresas conscientes respecto de la cantidad de las que no lo son?

Para responder algunos interrogantes consideraremos dos casos. Primero supondremos que la empresa inconsciente es la más pequeña y en una segunda instancia, asumiremos que la empresa más grande es la que no conoce el juego base.

Empresa 3 inconsciente Supongamos que la empresa 3 no conoce la existencia de las estaciones de control. Sabemos de las versiones anteriores que en la segunda iteración cuando el jugador 3 aumenta considerablemente su producción. En esta iteración, si este jugador cambia su estrategia de 4 a 54 unidades, entonces los costos reales estarán muy por encima de los calculados (costo esperado 57,64 y costo real 75,40). Entonces, al jugador 3 no le conviene moverse de su estrategia original, por lo que siempre escogerá $x_3^{*;3} = 4$.

Entonces obtenemos la siguiente solución al problema: $\mathbf{x}^* = (21{,}119; 16{,}068; 4)^T$. Claramente $\mathbf{x}^*$ no es solución del juego original, sin embargo, $q_1(\mathbf{x}^*) = 105{,}22 > K_1$ y $q_2(\mathbf{x}^*) = 84{,}75 < K_2$.

Si bien los niveles de concentración de tóxicos en el río no cumplen con las normas de la primera estación de control, el valor es más cercano al límite permitido que en las versiones anteriores. Además se ha propagado suficiente información como para alcanzar niveles de contaminación permitidos en la segunda estación. Entonces podemos decir que hay información que ha logrado propagarse.

Empresa 1 inconsciente Si suponemos ahora que el jugador 1 desconoce las restricciones ambientales, obtenemos que la solución del problema es $\mathbf{x}^* = (30; 16{,}068; 2{,}733)^T$. Con este resultado obtenemos que $q_1(\mathbf{x}^*) = 128{,}86 > K_1$ y $q_2(\mathbf{x}^*) = 101{,}54 > K_2$. En este caso no sólo no obtuvimos una solución del juego objetivo, sino que además el resultado no satisface las restricciones ambientales.

En el apéndice A, en el apartado de soluciones, podemos ver que este resultado fue obtenido pues, cuando el jugador 1 actualiza su límite M_1 e intenta aumentar su producción, el algoritmo nos dice que la empresa 3 debería producir una cantidad negativa de producto. Debido a que el obtenido no es un perfil factible, la empresa más grande no modifica nunca su elección.

Versión 4: El estado se involucra

En este caso asumiremos que nuestros jugadores ven la versión 1 en la cual no conocen las restricciones ambientales, pero agregaremos un jugador 0, el estado, que actuará como agente regulador. En este caso el estado ve el juego base y los demás no. Asumiremos que las empresas se comportan como hemos descripto en la primera versión y el jugador cero interviene de la siguiente manera: si los niveles de producción de los agentes satisfacen las restricciones $q_l \leq K_l$, para $l = 1, 2$, entonces no se involucra, pero si los niveles de producción superan los límites permitidos de contaminación, entonces el estado cobra una multa a las tres empresas.

En este caso, una vez que se cobra la primera multa, la acción más razonable por parte de las empresas sería averiguar en concepto de qué. Allí se les notificará acerca de estas restricciones ambientales y el juego pasará a ser un juego con incerteza en lugar de con inconsciencia. Puesto que los agentes serán conscientes de todos los estados posibles y asignarán una probabilidad de ocurrencia a la activación (o no) de la multa.

Este trabajo se enfoca en estudiar juegos con inconsciencia, por lo que no indagaremos más en el desarrollo de este ejemplo, pero queremos dejar asentado esto para notar que la línea que separa estos dos tipos de juegos es muy delgada.

Por lo tanto hemos logrado probar el siguiente resultado:

Proposición 4.3.1 *Las soluciones de los juegos con inconsciencia presentados en esta sección no son soluciones del juego objetivo. Más aún, estas soluciones no satisfacen las restricciones ambientales.*

Capítulo 5

Conclusiones

En este trabajo hemos estudiado problemas de equilibrio de Nash generalizado con inconsciencia, que nos han permitido modelar situaciones de la vida real más complejas que las modelables a través de problemas generalizados tradicionales. Esto ha agregado información a dichos problemas pues ahora se pueden tener en cuenta los distintos niveles de información incompleta que pueden manejar los agentes al momento de enfrentarse a un problema de decisión.

En el estudio de estos conceptos hemos descubierto que actualmente hay una amplia variedad de métodos para resolver esta clase de problemas, mencionados aquí y en los trabajos citados. Aquí mismo incluso hemos presentado dos maneras distintas (pero utilizando en ambas la reformulación por la función de Nikaido-Isoda) de probar la existencia del equilibrio deseado. También hemos desarrollado dos reformulaciones para problemas de equilibrio de Nash generalizado que nos permiten resolverlos de forma numérica.

Además el método mostrado que utiliza una versión generalizada de las condiciones KKT, nos permite transformar un problema de equilibrio de Nash generalizado en un problema simple de optimización sin restricciones. Para atacar esta clase de problemas, existen gran variedad de algoritmos ya optimizados en distintos lenguajes de programación. Podemos concluir así que, si bien unos métodos de resolución son más simples de utilizar que otros dependiendo de las características de nuestro problema, si vemos que existe solución tenemos disponibles formas de hallar una solución.

En el estudio de los ejemplos presentados a lo largo del trabajo hemos podido corroborar también que hay situaciones más fáciles de modelar a través de estos nuevos conceptos, que otras. En los ejemplos del capítulo 3 resultó muy simple modelar una situación con inconsciencia genérica, pero cuando intentamos plantear y resolver las distintas versiones del problema de contaminación del río, nos encontramos con que debíamos agregar cada vez más supuestos para poder plantear el modelo. Por esto es que no hemos tenido en cuenta las jerarquías de creencias en esta última parte. Un posible trabajo a futuro podría ser agregar esta noción a nuestro análisis y ver cómo se modifica el resultado, en especial, en la tercera reversión del problema.

Cada vez que intentamos resolver una situación interactiva con inconsciencia, estamos tratando de hallar un equilibrio de Nash (o un equilibrio generalizado dependiendo del caso), independientemente del nombre que el resultado tome al finalizar el juego. Cuando no contamos con toda la información, la solución del juego no es llamada propiamente "equilibrio de Nash", pero los jugadores siempre intentan escoger la estrategia que maximice sus utilidades tal que si se desvían unilateralmente del perfil de equilibrio, éstas disminuyen. Entonces es apropiado decir que la información adicional que insertan estos juegos con inconsciencia en el modelo no está en los cálculos propiamente dichos, sino en el análisis de las decisiones que los jugadores toman entre cada iteración del juego. Pudiéndose concluir que, si bien subjetivamente resolvemos problemas análogos, el hecho mencionado es la razón por la cual este tipo de situaciones no puede ser modelable a través de problemas de equilibrio de Nash clásicos y su generalización (con información completa).

Esto último queda claro si observamos que una constante, en casi todos los ejemplos, es que los resultados de los juegos subjetivos, no han coincidido con las soluciones del juego base. Sin embargo, las soluciones obtenidas en la tercera versión del ejemplo del río nos llevan a pensar que, bajo ciertas condiciones, podríamos llegar a obtener un equilibrio de Nash generalizado en un juego con inconsciencia. Lo cual motiva el planteo de los siguientes interrogantes: si tenemos n empresas, ¿cuántos jugadores deben conocer el juego base para que el conocimiento se propague? El valor de n, ¿influye en la convergencia del algoritmo? ¿Cuál es el mínimo de empresas que deben conocer las restricciones para que en una cantidad finita de pasos lleguemos a un equilibrio que cumple las condiciones del juego original?

Capítulo 6

Apéndices

6.1. Apéndice A

6.1.1. Funciones generales

Las siguientes son las funciones utilizadas para computar los costos y concentración de contaminación en el río, presentadas en el planteo del *River Basin Pollution Game*:

```python
import numpy as np
from settings import c1, c2, d1, d2, u1, u2, e


# Funciones del problema original

# Funcion de costo
def funcion_de_costo(x: np.array, player: int):
    sumando1 = x.sum()
    sumando1 = sumando1*d2
    sumando2 = c2[player] * x[player]
    factor1 = sumando1 + c1[player] + sumando2 - d1
    res = factor1 * x[player]

    return res

# Restricciones ambientales
def q1(x):
    termino1 = u1[0] * e[0] * x[0]
    termino2 = u1[1] * e[1] * x[1]
    termino3 = u1[2] * e[2] * x[2]

    return (termino1 + termino2 + termino3)

def q2(x):
    termino1 = u2[0] * e[0] * x[0]
    termino2 = u2[1] * e[1] * x[1]
    termino3 = u2[2] * e[2] * x[2]

    return (termino1 + termino2 + termino3)

```

```
# Funciones para la reformulacci n del problema

def funcion_complementaria(a, b):
    minuendo = a**2 + b**2
    minuendo = np.sqrt(minuendo)

    sustraendo = a + b

    return minuendo - sustraendo
```

6.1.2. Versiones

El código a continuación muestra las clases utilizadas para resolver cada versión planteada en el capítulo 5.

```
from scipy.optimize import least_squares


class Version0():

    def __init__(self) -> None:
        pass

    # Restricciones ambientales
    def q_1(self, x):
        suma = 0
        for i in range(3):
            suma = suma + (u1[i]*e[i]*x[i])
        return suma

    def q_2(self, x):
        suma = 0
        for i in range(3):
            suma = suma + (u2[i]*e[i]*x[i])
        return suma

    # Restricciones para los jugadores
    def g_1(self, x):
        return self.q_1(x) - K1

    def g_2(self, x):
        return self.q_2(x) - K2

    def F(self, x, multiplicadores):
        f1 = funcion_de_costo_der(x, 0) + (multiplicadores[0]*u1[0] + multiplicadores[1]*u2[0]) * e[0]
        f2 = funcion_de_costo_der(x, 1) + (multiplicadores[2]*u1[1] + multiplicadores[3]*u2[1]) * e[1]
        f3 = funcion_de_costo_der(x, 2) + (multiplicadores[4]*u1[2] + multiplicadores[5]*u2[2]) * e[2]

        return [f1, f2, f3]

```

```python
    def Phi(self, x, multiplicadores):
        phi_1 = funcion_complementaria(multiplicadores[0], -(self.g_1(x)))
        phi_2 = funcion_complementaria(multiplicadores[1], -(self.g_2(x)))
        phi_3 = funcion_complementaria(multiplicadores[2], -(self.g_1(x)))
        phi_4 = funcion_complementaria(multiplicadores[3], -(self.g_2(x)))
        phi_5 = funcion_complementaria(multiplicadores[4], -(self.g_1(x)))
        phi_6 = funcion_complementaria(multiplicadores[5], -(self.g_2(x)))

        return [phi_1, phi_2, phi_3, phi_4, phi_5, phi_6]

    def fun(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5], x[6], x[7], x[8]]
        X1 = self.F(x_vect, multiplicadores)
        X2 = self.Phi(x_vect,multiplicadores)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2], X2[3], X2[4], X2[5]]
        return X

    def solucion(self, x0):
        sol = least_squares(self.fun, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)

        return sol
```

```python
class Version1():

    def __init__(self, M1, M2, M3):
        self.M1 = M1
        self.M2 = M2
        self.M3 = M3

    # Restricciones para el Jugador 1
    def g1_1(self, x):
        return x - self.M1

    def g1_2(self, x):
        return x - (0.75 * self.M1)

    def g1_3(self, x):
        return x - (0.12 * self.M1)

    # Restricciones para el Jugador 2
    def g2_1(self, x):
        return x - (1.33 * self.M2)

    def g2_2(self, x):
        return x - self.M2

```

```python
    def g2_3(self, x):
        return x - (0.16 * self.M2)

    # Restricciones para el Jugador 3
    def g3_1(self, x):
        return x - (8 * self.M3)

    def g3_2(self, x):
        return x - (6 * self.M3)

    def g3_3(self, x):
        return x - self.M3

    # Funciones para resolver elproblema
    def Phi1(self, x, multiplicadores, g1, g2, g3):
        phi_1 = funcion_complementaria(multiplicadores[0], -g1(x[0]))
        phi_2 = funcion_complementaria(multiplicadores[1], -g2(x[1]))
        phi_3 = funcion_complementaria(multiplicadores[2], -g3(x[2]))
        return [phi_1, phi_2, phi_3]

    def F1(self, x, multiplicadores):
        f1 = funcion_de_costo_der(x, 0) + multiplicadores[0]
        f2 = funcion_de_costo_der(x, 1) + multiplicadores[1]
        f3 = funcion_de_costo_der(x, 2) + multiplicadores[2]
        return [f1,f2,f3]

    def fun1(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F1(x_vect, multiplicadores)
        X2 = self.Phi1(x_vect, multiplicadores, self.g1_1, self.g1_2, self.g1_3)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def fun2(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F1(x_vect, multiplicadores)
        X2 = self.Phi1(x_vect,multiplicadores, self.g2_1, self.g2_2, self.g2_3)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def fun3(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F1(x_vect, multiplicadores)
        X2 = self.Phi1(x_vect,multiplicadores, self.g3_1, self.g3_2, self.g3_3)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X
```

```

    def solucion(self, x0, player):
        if player==1:
            sol = least_squares(self.fun1, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)
        elif player == 2:
            sol = least_squares(self.fun2, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)
        elif player == 3:
            sol = least_squares(self.fun3, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)
        return sol
```

```
class Version2():

    def __init__(self, M2, M3) -> None:
        self.M2 = M2
        self.M3 = M3

    # Restricciones ambientales
    def q_1(self, x):
        suma = 0
        for i in range(3):
            suma = suma + (u1[i]*e[i]*x[i])
        return suma

    def q_2(self, x):
        suma = 0
        for i in range(3):
            suma = suma + (u2[i]*e[i]*x[i])
        return suma

    # Restricciones para el Jugador 1
    def g1_1(self, x):
        return self.q_1(x) - K1

    def g1_2(self, x):
        return self.q_2(x) - K2

    def g1_3(self, x):
        return 0

    # Restricciones para el Jugador 2
    def g2_1(self, x):
        return x - (1.33 * self.M2)

    def g2_2(self, x):
        return x - self.M2

    def g2_3(self, x):
        return x - (0.16 * self.M2)

    # Restricciones para el Jugador 3
    def g3_1(self, x):
        return x - (8 * self.M3)
```

```

    def g3_2(self, x):
        return x - (6 * self.M3)

    def g3_3(self, x):
        return x - self.M3

    # Funciones para resolver elproblema
    def Phi2_1(self, x, multiplicadores):
        phi_1 = funcion_complementaria(multiplicadores[0], -(self.g1_1(x)))
        phi_2 = funcion_complementaria(multiplicadores[1], -(self.g1_2(x)))
        phi_3 = funcion_complementaria(multiplicadores[2], -(self.g1_3(x)))

        return [phi_1, phi_2, phi_3]

    def Phi2_2(self, x, multiplicadores, g1, g2, g3):
        phi_1 = funcion_complementaria(multiplicadores[0], -g1(x[0]))
        phi_2 = funcion_complementaria(multiplicadores[1], -g2(x[1]))
        phi_3 = funcion_complementaria(multiplicadores[2], -g3(x[2]))
        return [phi_1, phi_2, phi_3]

    def F2(self, x, multiplicadores):
        f1 = funcion_de_costo_der(x, 0) + multiplicadores[0]
        f2 = funcion_de_costo_der(x, 1) + multiplicadores[1]
        f3 = funcion_de_costo_der(x, 2) + multiplicadores[2]
        return [f1,f2,f3]

    def fun1(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F2(x_vect, multiplicadores)
        X2 = self.Phi2_1(x_vect, multiplicadores)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def fun2(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F2(x_vect, multiplicadores)
        X2 = self.Phi2_2(x_vect,multiplicadores, self.g2_1, self.g2_2, self.g2_3)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def fun3(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F2(x_vect, multiplicadores)
        X2 = self.Phi2_2(x_vect,multiplicadores, self.g3_1, self.
```

```
g3_2, self.g3_3)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def solucion(self, x0, player):
        if player==1:
            sol = least_squares(self.fun1, x0=x0, method='lm',
loss='linear', ftol=1e-15, xtol=1e-15)
        elif player == 2:
            sol = least_squares(self.fun2, x0=x0, method='lm',
loss='linear', ftol=1e-15, xtol=1e-15)
        elif player == 3:
            sol = least_squares(self.fun3, x0=x0, method='lm',
loss='linear', ftol=1e-15, xtol=1e-15)
        return sol
```

```
class Version3Inconsciente():

    def __init__(self, M3) -> None:
        self.M3 = M3

    # Restricciones ambientales
    def q_1(self, x):
        suma = 0
        for i in range(3):
            suma = suma + (u1[i]*e[i]*x[i])
        return suma

    def q_2(self, x):
        suma = 0
        for i in range(3):
            suma = suma + (u2[i]*e[i]*x[i])
        return suma

    # Restricciones para Jugador 1 y Jugador 2
    def gi_1(self, x):
        return self.q_1(x) - K1

    def gi_2(self, x):
        return self.q_2(x) - K2

    def gi_3(self, x):
        return 0

    # Restricciones para el Jugador 3
    def g3_1(self, x):
        return x - (8 * self.M3)

    def g3_2(self, x):
        return x - (6 * self.M3)

    def g3_3(self, x):
        return x - self.M3

    # Funciones para resolver elproblema
```

```
    def Phi3_1(self, x, multiplicadores):
        phi_1 = funcion_complementaria(multiplicadores[0], -(self.gi_1(x)))
        phi_2 = funcion_complementaria(multiplicadores[1], -(self.gi_2(x)))
        phi_3 = funcion_complementaria(multiplicadores[2], -(self.gi_3(x)))

        return [phi_1, phi_2, phi_3]

    def Phi3_2(self, x, multiplicadores, g1, g2, g3):
        phi_1 = funcion_complementaria(multiplicadores[0], -g1(x[0]))
        phi_2 = funcion_complementaria(multiplicadores[1], -g2(x[1]))
        phi_3 = funcion_complementaria(multiplicadores[2], -g3(x[2]))
        return [phi_1, phi_2, phi_3]

    def F3(self, x, multiplicadores):
        f1 = funcion_de_costo_der(x, 0) + multiplicadores[0]
        f2 = funcion_de_costo_der(x, 1) + multiplicadores[1]
        f3 = funcion_de_costo_der(x, 2) + multiplicadores[2]
        return [f1,f2,f3]

    def fun(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F3(x_vect, multiplicadores)
        X2 = self.Phi3_1(x_vect, multiplicadores)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def fun3(self, x):
        x_vect = [x[0], x[1], x[2]]
        multiplicadores = [x[3], x[4], x[5]]
        X1 = self.F3(x_vect, multiplicadores)
        X2 = self.Phi3_2(x_vect,multiplicadores, self.g3_1, self.g3_2, self.g3_3)
        X = [X1[0], X1[1], X1[2], X2[0], X2[1], X2[2]]
        return X

    def solucion(self, x0, player):
        if player==1:
            sol = least_squares(self.fun, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)
        elif player == 2:
            sol = least_squares(self.fun, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)
        elif player == 3:
            sol = least_squares(self.fun3, x0=x0, method='lm', loss='linear', ftol=1e-15, xtol=1e-15)
        return sol
```

6.1.3. Soluciones

Solución de la versión 1:

```python
import numpy as np

from versiones import Version1
from nuevo_limite import f_i
from funcionesgenerales import funcion_de_costo, q1, q2

# SOLUCIONES DE LA VERSION 1

# Primera iteracion
M1 = 30
M2 = 22
M3 = 4
max_iter = 100
tol = 1e-4
vers1 = Version1(M1, M2, M3)

x1_inicial = np.array([20, 16, 2, 0, 0, 0])
x2_inicial = np.array([20, 16, 2, 0, 0, 0])
x3_inicial = np.array([20, 16, 2, 0, 0, 0])

x1_sol = vers1.solucion(x1_inicial, 1).x
x1_mult = np.array([x1_sol[3], x1_sol[4], x1_sol[5]])
x1_sol = np.array([x1_sol[0], x1_sol[1], x1_sol[2]])

x2_sol = vers1.solucion(x2_inicial, 2).x
x2_mult = np.array([x2_sol[3], x2_sol[4], x2_sol[5]])
x2_sol = np.array([x2_sol[0], x2_sol[1], x2_sol[2]])

x3_sol = vers1.solucion(x3_inicial, 3).x
x3_mult = np.array([x3_sol[3], x3_sol[4], x3_sol[5]])
x3_sol = np.array([x3_sol[0], x3_sol[1], x3_sol[2]])

print(x1_sol, x2_sol, x3_sol)

x_nash_real = np.array([x1_sol[0], x2_sol[1], x3_sol[2]])

print(x_nash_real)

print('Costo esperado por 1: ', -funcion_de_costo(x1_sol, 0), '
    Costo real para 1: ', -funcion_de_costo(x_nash_real, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_sol, 1), '
    Costo real para 2: ', -funcion_de_costo(x_nash_real, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_sol, 2), '
    Costo real para 3: ', -funcion_de_costo(x_nash_real, 2))

# Segunda iteracion

M1_nuevo = f_i(x2_sol[1], x3_sol[2], jugador=0)
M2_nuevo = f_i(x1_sol[0], x3_sol[2], jugador=1)
M3_nuevo = f_i(x1_sol[0], x2_sol[1], jugador=2)

print(M1_nuevo, M1, M2_nuevo, M2, M3_nuevo, M3)
```

```

M1_nuevo = int(round(M1_nuevo, 0))
M2_nuevo = int(round(M2_nuevo, 0)) + 1
M3_nuevo = int(round(M3_nuevo, 0)) + 1

print(M1_nuevo, M2_nuevo, M3_nuevo)

vers_1_1 = Version1(M1_nuevo, M2, M3)
x1 = vers_1_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x1_mult[0], x1_mult[1], x1_mult[2]], 1).x
x1_nuevo = np.array([x1[0], x2_sol[1], x3_sol[2]])

vers_1_1 = Version1(M1, M2_nuevo, M3)
x2 = vers_1_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x2_mult[0], x2_mult[1], x2_mult[2]], 2).x
x2_nuevo = np.array([x1_sol[0], x2[1], x3_sol[2]])

vers_1_1 = Version1(M1, M2, M3_nuevo)
x3 = vers_1_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x3_mult[0], x3_mult[1], x3_mult[2]], 3).x
x3_nuevo = np.array([x1_sol[0], x2_sol[1], x3[2]])

print(x1_nuevo, x2_nuevo, x3_nuevo)

x_nash_real_nuevo = np.array([x1_nuevo[0], x2_nuevo[1], x3_nuevo[2]])

print(x_nash_real_nuevo)

print('Costo esperado por 1: ', -funcion_de_costo(x1_nuevo, 0), 'Costo real para 1: ', -funcion_de_costo(x_nash_real_nuevo, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_nuevo, 1), 'Costo real para 2: ', -funcion_de_costo(x_nash_real_nuevo, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_nuevo, 2), 'Costo real para 3: ', -funcion_de_costo(x_nash_real_nuevo, 2))


# Tercera iteracion

M1 = M1_nuevo
M2 = M2_nuevo
M3 = M3_nuevo
x1_sol = np.array([int(round(x1_nuevo[0], 0)), int(round(x1_nuevo[1], 0)), int(round(x1_nuevo[2], 0))])
x2_sol = np.array([int(round(x2_nuevo[0], 0)), int(round(x2_nuevo[1], 0)), int(round(x2_nuevo[2], 0))])
x3_sol = np.array([int(round(x3_nuevo[0], 0)), int(round(x3_nuevo[1], 0)), int(round(x3_nuevo[2], 0))])

M1_nuevo = f_i(x2_sol[1], x3_sol[2], jugador=0)
M2_nuevo = f_i(x1_sol[0], x3_sol[2], jugador=1)
M3_nuevo = f_i(x1_sol[0], x2_sol[1], jugador=2)

M1_nuevo = int(round(M1_nuevo, 0))
M2_nuevo = int(round(M2_nuevo, 0))
```

```python
M3_nuevo = int(round(M3_nuevo, 0))

print(M1_nuevo, M1, M2_nuevo, M2, M3_nuevo, M3)

# Los valores de las constantes han disminuido, por lo que no las
    actualizaremos

M1_nuevo = M1
M2_nuevo = M2
M3_nuevo = M3

vers_1_1 = Version1(M1_nuevo, M2, M3)
x1 = vers_1_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x1_mult
    [0], x1_mult[1], x1_mult[2]], 1).x
x1_nuevo = np.array([x1[0], x2_sol[1], x3_sol[2]])

vers_1_1 = Version1(M1, M2_nuevo, M3)
x2 = vers_1_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x2_mult
    [0], x2_mult[1], x2_mult[2]], 2).x
x2_nuevo = np.array([x1_sol[0], x2[1], x3_sol[2]])

vers_1_1 = Version1(M1, M2, M3_nuevo)
x3 = vers_1_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x3_mult
    [0], x3_mult[1], x3_mult[2]], 3).x
x3_nuevo = np.array([x1_sol[0], x2_sol[1], x3[2]])

print(x1_nuevo, x2_nuevo, x3_nuevo)

x_nash_real_nuevo = np.array([x1_nuevo[0], x2_nuevo[1], x3_nuevo
    [2]])

print(x_nash_real_nuevo)

print(q1(x_nash_real_nuevo))

print(q2(x_nash_real_nuevo))
```

Solución de la versión 2:

```python
import numpy as np

from versiones import Version1, Version2, Version3Inconsciente,
    Version1Inconsciente
from nuevo_limite import f_i
from funcionesgenerales import funcion_de_costo, q1, q2

# SOLUCIONES VERSION 2

# Primera iteracion

M2 = 22
M3 = 4
max_iter = 100
tol = 1e-4
vers2 = Version2(M2, M3)
```

```

# El primer jugador ve el juego original, definimos su vector solucion y no lo modificamos
x1_sol = np.array([21.119, 16.068, 2.733])

x2_inicial = np.array([20, 16, 2, 0, 0, 0])
x3_inicial = np.array([20, 16, 2, 0, 0, 0])

x2_sol = vers2.solucion(x2_inicial, 2).x
x2_mult = np.array([x2_sol[3], x2_sol[4], x2_sol[5]])
x2_sol = np.array([x2_sol[0], x2_sol[1], x2_sol[2]])

x3_sol = vers2.solucion(x3_inicial, 3).x
x3_mult = np.array([x3_sol[3], x3_sol[4], x3_sol[5]])
x3_sol = np.array([x3_sol[0], x3_sol[1], x3_sol[2]])

print(x1_sol, x2_sol, x3_sol)

x_nash_real = np.array([x1_sol[0], x2_sol[1], x3_sol[2]])

print(x_nash_real)

print('Costo esperado por 1: ', -funcion_de_costo(x1_sol, 0), ' Costo real para 1: ', -funcion_de_costo(x_nash_real, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_sol, 1), ' Costo real para 2: ', -funcion_de_costo(x_nash_real, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_sol, 2), ' Costo real para 3: ', -funcion_de_costo(x_nash_real, 2))


# Segunda iteracion

M2_nuevo = f_i(x_nash_real[0], x_nash_real[2], jugador=1)
M3_nuevo = f_i(x_nash_real[0], x_nash_real[1], jugador=2)

print(M2_nuevo, M3_nuevo)

M2_nuevo = 22
M3_nuevo = 61

vers_2_1 = Version2(M2_nuevo, M3)
x2 = vers_2_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x2_mult[0], x2_mult[1], x2_mult[2]], 2).x
x2_nuevo = np.array([x1_sol[0], x2[1], x3_sol[2]])

vers_2_1 = Version2(M2, M3_nuevo)
x3 = vers_2_1.solucion([x1_sol[0], x2_sol[1], x3_sol[2], x3_mult[0], x3_mult[1], x3_mult[2]], 3).x
x3_nuevo = np.array([x1_sol[0], x2_sol[1], x3[2]])

x1_nuevo = x1_sol

print(x1_nuevo, x2_nuevo, x3_nuevo)

x_nash_real_nuevo = np.array([x1_nuevo[0], x2_nuevo[1], x3_nuevo
```

```
    [2]])

print(x_nash_real_nuevo)

print('Costo esperado por 1: ', -funcion_de_costo(x1_nuevo, 0), '
    Costo real para 1: ', -funcion_de_costo(x_nash_real_nuevo, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_nuevo, 1), '
    Costo real para 2: ', -funcion_de_costo(x_nash_real_nuevo, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_nuevo, 2), '
    Costo real para 3: ', -funcion_de_costo(x_nash_real_nuevo, 2))


#Tercera iteracion

M2 = M2_nuevo
x2_sol = x_nash_real_nuevo

M2_nuevo = f_i(x_nash_real_nuevo[0], x_nash_real_nuevo[2], jugador
    =1)

print(M2, M2_nuevo)

# M2 disminuye y bajar la produccion no era una actitud aceptable,
     por lo que no actualizamos la constante

vers_2_1 = Version2(M2, M3)
x2 = vers_2_1.solucion([x_nash_real_nuevo[0], x_nash_real_nuevo
    [1], x_nash_real_nuevo[2], x2_mult[0], x2_mult[1], x2_mult[2]],
     3).x

x1_nuevo = x1_sol
x2_nuevo = x2_sol

print(x1_nuevo, x2_nuevo, x3_nuevo)

x_nash_real_nuevo = np.array([x1_nuevo[0], x2_nuevo[1], x3_nuevo
    [2]])

print(x_nash_real_nuevo)

print('Costo esperado por 1: ', -funcion_de_costo(x1_nuevo, 0), '
    Costo real para 1: ', -funcion_de_costo(x_nash_real_nuevo, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_nuevo, 1), '
    Costo real para 2: ', -funcion_de_costo(x_nash_real_nuevo, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_nuevo, 2), '
    Costo real para 3: ', -funcion_de_costo(x_nash_real_nuevo, 2))

# Veamos si cumplen las restricciones ambientales

print(q1(x_nash_real_nuevo))
print(q2(x_nash_real_nuevo))
```

Solución de la versión 3, con la empresa 3 inconsciente:

```
import numpy as np
```

```
from versiones import Version3Inconsciente
from nuevo_limite import f_i
from funcionesgenerales import funcion_de_costo, q1, q2

# Primera iteracion

M3 = 4
max_iter = 100
tol = 1e-4
vers3 = Version3Inconsciente(M3)

x1_sol = np.array([21.119, 16.068, 2.733])
x2_sol = np.array([21.119, 16.068, 2.733])

x3_inicial = np.array([20, 16, 2, 0, 0, 0])

x3_sol = vers3.solucion(x3_inicial, 3).x
x3_mult = np.array([x3_sol[3], x3_sol[4], x3_sol[5]])
x3_sol = np.array([x3_sol[0], x3_sol[1], x3_sol[2]])

print(x1_sol, x2_sol, x3_sol)

x_nash_real = np.array([x1_sol[0], x2_sol[1], x3_sol[2]])

print(x_nash_real)

print('Costo esperado por 1: ', -funcion_de_costo(x1_sol, 0), ' Costo real para 1: ', -funcion_de_costo(x_nash_real, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_sol, 1), ' Costo real para 2: ', -funcion_de_costo(x_nash_real, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_sol, 2), ' Costo real para 3: ', -funcion_de_costo(x_nash_real, 2))

# Segunda iteracion

M3_nuevo = f_i(x_nash_real[0], x_nash_real[1], jugador=2)

print(M3_nuevo)

M3_nuevo = int(round(M3_nuevo, 0))

vers_3_1 = Version3Inconsciente(M3_nuevo)
x3 = vers_3_1.solucion([x_nash_real[0], x_nash_real[1], x_nash_real[2], x3_mult[0], x3_mult[1], x3_mult[2]], 3).x
x3_nuevo = np.array([x3[0], x3[1], x3[2]])

x1_nuevo = x1_sol
x2_nuevo = x2_sol

print(x1_nuevo, x2_nuevo, x3_nuevo)

x_nash_real_nuevo = np.array([x1_nuevo[0], x2_nuevo[1], x3_nuevo[2]])

```

```
print(x_nash_real_nuevo)

print('Costo esperado por 1: ', -funcion_de_costo(x1_nuevo, 0), '
    Costo real para 1: ', -funcion_de_costo(x_nash_real_nuevo, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_nuevo, 1), '
    Costo real para 2: ', -funcion_de_costo(x_nash_real_nuevo, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_nuevo, 2), '
    Costo real para 3: ', -funcion_de_costo(x_nash_real_nuevo, 2))

# Tercera iteracion

M3_nuevo = f_i(x_nash_real_nuevo[0], x_nash_real_nuevo[1], jugador
    =2)

print(M3_nuevo)
M3_nuevo = int(round(M3_nuevo, 0))

x3_sol = x3_nuevo

vers_3_1 = Version3Inconsciente(M3_nuevo)
x3 = vers_3_1.solucion([x_nash_real_nuevo[0], x_nash_real_nuevo
    [1], x_nash_real_nuevo[2], x3_mult[0], x3_mult[1], x3_mult[2]],
     3).x
x3_nuevo = np.array([x3[0], x3[1], x3[2]])

x1_nuevo = x1_sol
x2_nuevo = x2_sol

print(x1_nuevo, x2_nuevo, x3_nuevo)

x_nash_real_nuevo = np.array([x1_nuevo[0], x2_nuevo[1], x3_nuevo
    [2]])

print(x_nash_real_nuevo)

print('Costo esperado por 1: ', -funcion_de_costo(x1_nuevo, 0), '
    Costo real para 1: ', -funcion_de_costo(x_nash_real_nuevo, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_nuevo, 1), '
    Costo real para 2: ', -funcion_de_costo(x_nash_real_nuevo, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_nuevo, 2), '
    Costo real para 3: ', -funcion_de_costo(x_nash_real_nuevo, 2))

x_nash = np.array([21.119, 16.068, 4.])

# Veamos si cumplen las condiciones ambientales

print(q1(x_nash))

print(q2(x_nash))
```

Solución de la versión 3, con la empresa 1 inconsciente:

```
import numpy as np

from versiones import Version1Inconsciente
```

```python
from nuevo_limite import f_i
from funcionesgenerales import funcion_de_costo, q1, q2

# Primera iteracion

M1 = 30
max_iter = 100
tol = 1e-4
vers1_inc = Version1Inconsciente(M1)

x3_sol = np.array([21.119, 16.068, 2.733])
x2_sol = np.array([21.119, 16.068, 2.733])

x1_inicial = np.array([20, 16, 2, 0, 0, 0])

x1_sol = vers1_inc.solucion(x1_inicial, 1).x
x1_mult = np.array([x1_sol[3], x1_sol[4], x1_sol[5]])
x1_sol = np.array([x1_sol[0], x1_sol[1], x1_sol[2]])

print(x1_sol, x2_sol, x3_sol)

x_nash_real = np.array([x1_sol[0], x2_sol[1], x3_sol[2]])

print(x_nash_real)

print('Costo esperado por 1: ', -funcion_de_costo(x1_sol, 0), '
    Costo real para 1: ', -funcion_de_costo(x_nash_real, 0))
print('Costo esperado por 2: ', -funcion_de_costo(x2_sol, 1), '
    Costo real para 2: ', -funcion_de_costo(x_nash_real, 1))
print('Costo esperado por 3: ', -funcion_de_costo(x3_sol, 2), '
    Costo real para 3: ', -funcion_de_costo(x_nash_real, 2))


# Segunda iteraccion

M1_nuevo = f_i(x_nash_real[1], x_nash_real[2], jugador=0)

print(M1_nuevo)

M1_nuevo = int(round(M1_nuevo, 0))

vers_1_inc_1 = Version1Inconsciente(M1_nuevo)
x1 = vers_1_inc_1.solucion([x_nash_real[0], x_nash_real[1],
    x_nash_real[2], x1_mult[0], x1_mult[1], x1_mult[2]], 3).x
x1_nuevo = np.array([x1[0], x1[1], x1[2]])

x3_nuevo = x3_sol
x2_nuevo = x2_sol

print(x1_nuevo, x2_nuevo, x3_nuevo)

# El nuevo equilibrio implica que la empresa 3 produzca una
    cantidad negativa, por lo que no actualizaremos el equilibrio

x_nash_final = np.array([30., 16.068, 2.733])
```

```

# Veamos si cumplen las restricciones ambientales

print(q1(x_nash_final))

print(q2(x_nash_final))
```

6.2. Apéndice B

6.2.1. Glosario

Conceptos

- **Agente, jugador o participante**: persona involucrada en una situación interactiva (juego) y que debe tomar una decisión.

- **Función $\mathbf{C^k}$**: Para $k \in \mathbb{N}$, las funciones C^k son funciones tales que existen sus derivadas parciales hasta el orden k y las derivadas parciales de orden k son también continuas.

- **Juego de suma cero**: juego no cooperativo que modela una situación en la que la ganancia o la pérdida de un jugador se equilibra con exactitud con las pérdidas o ganancias de los otros participantes.

- **Juego en forma extensiva**: Situación interactiva donde las decisiones se toman en distintos momentos en el tiempo, no en simultáneo como en los juegos estáticos. Por ejemplo juegos de mesa donde cada participante debe esperar su turno para jugar. El lector puede encontrar información al respecto en el capítulo 3 de [8].

- **Juego o problema generalizado**: En el contexto de este trabajo, un juego o un problema generalizado se utilizan como sinónimos de un problema de equilibrio de Nash generalizado.

- **Minimax**: algoritmo recursivo para *minimizar* la pérdida *máxima* esperada en juegos con adversarios y con información perfecta (o completa). Puede resumirse en "cómo elegir el mejor movimiento para uno mismo, suponiendo que el contrincante escogerá el peor para uno".

- **Soporte de una estrategia mixta**: dado un juego estratégico con inconsciencia
$\Gamma^U = (\mathcal{G}, \mathcal{F})$ con conjunto común de $\mathcal{N}$ jugadores, y para cada $i \in \mathcal{N}$ y cada $k \in |\mathcal{G}|$ son probabilidades, se entiende por soporte el conjunto de acciones para el cual la probabilidad de ocurrencia es no nula.

Notación

Aquí se presenta una guía de la notación más utilizada a lo largo del trabajo, en orden de aparición:

- X, X_i: Conjunto de estrategias y estrategias para el jugador i respectivamente, para juegos sin inconsciencia.
- x, x_i: Elementos de los conjuntos X y X_i respectivamente.
- u, u_i: Función de utilidad o costo y utilidad o costo para el jugador i, respectivamente.
- $\mathcal{N}$: Conjunto de jugadores de un juego.
- n: Cantidad de jugadores de un juego estratégico.
- N: Cantidad total de variables de decisión en un juego estratégico.
- x_{-i}: $(x_j)_{j\in\mathcal{N}, j\neq i}$.
- (x_i, x_{-i}): Perfil de estrategias de un juego.
- x^*: Punto de equilibrio de un juego.
- $|Y|$: Cardinal del conjunto Y.
- G: Juego estratégico.
- $E(G)$: Extensión mixta de un juego estratégico.
- $X(x)$ y $X_i(x_{-i})$: Conjunto de estrategias de un problema de equilibrio de Nash generalizado.
- Γ^U: Juego estratégico con inconsciencia.
- $\mathcal{G}$: Conjunto de juegos en un juego estratégico con inconsciencia.
- $\mathcal{F}$: Conjunto de correspondencias de conocimiento en un juego estético con inconsciencia.
- $\mathcal{G}_i$: Conjunto de juegos estratégicos que percibe el jugador i en algún punto del juego con inconsciencia.
- G^k: Juegos estratégicos en $\mathcal{G}$.
- A^k, A_i^k: Conjunto de acciones en un juego subjetivo G^k.
- u^k y u_i^k: Funciones de utilidad general y para cada jugador i en los juegos en $\mathcal{G}$.
- $\sigma_{i,k}$ o $\delta_{i,k}$: Acción local del agente i en el juego G^k.

- Σ y Σ_i: Conjunto de estrategias y estrategias del jugador i en un juego estático con inconsciencia.
- ΔY: Conjunto de distribuciones de probabilidad sobre el conjunto Y.
- Eu_i^k: Utilidad esperada por el jugador i en el juego G^k.
- μ: Vector de multiplicadores de Lagrange de un problema de optimización.
- μ_i: Multiplicadores de Lagrange del jugador i en la resolución numérica de problemas con inconsciencia.
- $\nabla_z f$: Gradiente de la función f respecto de la variable z.
- g: Restricciones de un problema de optimización.
- g^i: Restricciones del problema de optimización que debe resolver el jugador i.
- $L(x, \mu)$: Lagrangiano del problema de optimización 4.2.
- ϕ: Función de Fischer-Burmeister.
- $\mathbf{\Phi}$: Función complementaria en la reformulación de un problema de equilibrio de Nash generalizado.
- $\mathbf{\Theta}$: Función de mérito en la reformulación de un problema generalizado.
- $|| \cdot ||$: Norma euclídea del espacio $\mathbb{R}^n$ correspondiente.

Agradecimientos

Quiero agradecer en primer lugar a mi familia por brindarme todo su apoyo tanto económico como emocional y psicológico. Por siempre priorizar mi estudio y por siempre animarme a continuar cuando creía que ya no podía más. Por seguir siempre a mi lado a pesar de mis locas idas y vueltas. Quiero agradecer especialmente a mis padres por alentarme a buscar y perseguir aquello que me hiciera feliz. También a mis hermanas con las cuales conviví toda mi carrera, por darme mi lugar, por ayudarme a levantarme cuando las cosas no iban de acuerdo al plan y por festejar conmigo todos mis logros.

A mis amigos de Rafaela, por seguir conmigo todo este tiempo sabiendo que la carrera a veces me mantenía alejada de todos. A los amigos que me regaló la facultad por acompañarme en todo el camino, por compartir tantas clases, tardes de estudio y mates. Un agradecimiento especial a Rocío que se tomó mi trabajo final como si fuera propio y me acompañó desde el primer momento, ayudándome a enviar el correo para tener al director que yo había elegido. Y otra mención especial a mi primer grupo de estudio, que incluso en pandemia y siendo de diversas carreras, me ayudaron siempre que pudieron, ayudándome en temas que no entendía o simplemente estudiando a mi lado.

A mi director, Andrés, por su guía y compromiso. Por el tiempo que dedicó a reunirse conmigo, recolectar información, salvarme de crisis cuando las cuentas no tenían sentido. También por preocuparse además por mi futuro y ampliar mis horizontes académicos.

A la FAMAF, por brindarme un espacio donde instruirme profesionalmente. A los docentes que me acompañaron toda la carrera, en especial a aquellos que se preocuparon y alegraron conmigo en cada paso, a aquellos que me motivaron a descubrir lo que me apasionaba en las matemáticas y alcanzarlo. A los alumnos con los que me fui cruzando y la gente que fui conociendo a lo largo de este trayecto, no sólo de matemática, sino de todas las disciplinas estudiadas en la facultad, por brindarme no sólo un espacio físico para que yo pudiera sentarme a escribir este trabajo, sino también por compartir ratos de esparcimiento conmigo tan necesarios.

Por útlimo, un agradecimiento a Rosalía, por incentivarme a salir de la comodidad de mi ciudad y a seguir lo que me hiciera feliz sin importar lo que los demás pensaran, ayudándome así a emprender este viaje que aún no termina.

Bibliografía

[1] J. V. Neumann and O. Morgenstern, "Theory of games and economic behavoir," 1944.

[2] Ángel F. Tenorio Villalón and A. M. M. Caraballo, "Un paseo por la historia de la teoría de juegos," 2015.

[3] D. König, "Über eine schlussweise aus dem endluchen ins unendliche," pp. 2–3, 121–130, 1927.

[4] J. V. Neumann, "Zur theorie der gesellschaftsspiele," pp. 295–320, 1928.

[5] J. F. Nash, "Non-cooperative games," pp. 286–95, 1951.

[6] J. González-Díaz, I. García-Jurado, and M. G. Fiestras-Janeiro, "An introductory course on mathematical game theory," 2010.

[7] D. Gerard, "A social equilibrium existence theorem," vol. 38, pp. 886–893, 1952.

[8] A. Fischer, M. Herrich, and K. Schönefeld, "Generalized nash equilibrium problems - recent advances and challenges," 2014.

[9] J. B. Krawczyk and S. Uryasev, "Relaxation algorithms to find nash equilibria with economic applications," 1999.

[10] J. Y. Halpern and L. C. Rêgo, "Extensive games with possibly unaware players," vol. 70, pp. 42–58, 2014.

[11] Y. Sasaki, "Generalized nash equilibrium with stable belief hierarchies in static games with unawareness," 2016.

[12] A. von Heusinger, "Numerical methods for the solution of the generalized nash equilibrium problem," 2009.

[13] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, "On the solution of the kkt conditions of generalized nash equilibrium problems," 2011.

Los abajo firmantes, miembros del Tribunal de evaluación de tesis, damos fe que el presente ejemplar impreso se corresponde con el aprobado por este Tribunal.