Universitat Autònoma de Barcelona

Facultat d'Economia i Empresa UAB

Trabajo de Final de Grado

Facultat de Economia y Empresa

TÍTULO: *Desafiando el modelo de Bertrand: cómo los algoritmos pueden generar colusión tácita en mercados oligopólicos*

AUTOR/A: *Genís Sánchez García*

TUTOR/A: *Pablo Lopez Argüello*

GRADO: DOBLE GRADO EN ADMINISTRACIÓN Y DIRECCIÓN DE EMPRESAS Y DERECHO

FECHA: *30/05/2023*

RESUM

Actualment, els algoritmes i la Intel·ligència Artificial estan adquirint molta importància en el context social gràcies a la seva aplicació en diversos àmbits que milloren substancialment l'eficiència i la productivitat de múltiples sectors econòmics, ja que són capaços d'ajudar o fins i tot de prendre decisions encertades de forma autònoma gràcies al seu gran poder de processament d'informació i l'anàlisi continu de decisions històriques. Un dels àmbits en què es pot aplicar aquest tipus de tecnologies és en l'establiment de preus mitjançant la monitorització dels moviments de les empreses competidores amb la finalitat d'establir aquell preu que maximitzi els beneficis. Si bé la utilització d'algoritmes o Intel·ligències Artificials és una estratègia legítima en quant a productivitat empresarial, també és cert que existeix literatura en aquest àmbit que afirma que els algoritmes fixadors de preus poden, o bé ajudar l'ésser humà a col·ludir, o bé propiciar de manera directa una restricció de la competència en aras de la maximització de beneficis en aprendre de forma autònoma (i sense ser una decisió de disseny) a col·ludir tàcitament sense que hi hagi acord o comunicació expressa entre algoritmes. El present text pretén desenvolupar tots els conceptes teòrics implícits en la col·lusió tàcita algorítmica, amb la finalitat de comprendre com un algoritme fixador de preus pot ser capaç d'aprendre a restringir la competència per compte propi, finalitzant amb l'assoliment d'un experiment econòmic computacional on es dissenya mitjançant Python un algoritme capaç de simular un duopoli de Bertrand en fixar simultàniament els preus establerts, emmagatzemant totes aquelles decisions preses per l'algoritme amb la finalitat de determinar si l'algoritme ha estat capaç d'aprendre a col·ludir tàcitament de forma automàtica i autònoma.

PARAULES CLAU: Algoritme, Intel·ligència Artificial, Python, Bertrand, col·lusió, col·lusió tàcita, paral·lelisme conscient, Markov, aprenentatge automàtic, aprenentatge per reforç, Q-learning.

RESUMEN

En la actualidad, los algoritmos e Inteligencias Artificiales están adquiriendo mucha importancia en el contexto social gracias a su aplicación en múltiples ámbitos que mejoran sustancialmente la eficiencia y productividad de múltiples sectores económicos, debido a que son capaces de ayudar o incluso de tomar decisiones acertadas de forma autónoma gracias a su gran poder de procesamiento de información y el análisis continuo de decisiones históricas. Uno de los ámbitos en los que se puede aplicar este tipo de tecnologías es en la fijación de precios mediante la monitorización de movimientos de las empresas competidores con el fin de establecer aquel precio maximizador de beneficios. Si bien la utilización de algoritmos o Inteligencias Artificiales es una estrategia legítima en cuanto a productividad empresarial, también es cierto que existe literatura en este ámbito que afirma que los algoritmos fijadores de precios pueden, o bien ayudar al ser humano a coludir, o bien a propiciar de manera directa una restricción de la competencia en aras de la maximización de beneficios al aprender de forma autónoma (y sin ser una decisión de diseño) a coludir tácitamente sin mediar acuerdo o comunicación expresa entre algoritmos. El presente texto pretende desarrollar todos los conceptos teóricos implícitos en la colusión tácita algorítmica, con el fin de comprender como un algoritmo fijador de precios puede ser capaz de aprender a restringir la competencia por su propia cuenta, finalizando en la consecución de un experimento económico computacional donde se diseña mediante Python un algoritmo capaz de simular un duopolio de Bertrand al fijar de forma simultánea los precios establecidos, almacenando todas aquellas decisiones tomadas por el algoritmo con el fin de determinar si el algoritmo ha sido capaz de aprender a coludir tácitamente de forma automática y autónoma.

PALABRAS CLAVE: Algoritmo, Inteligencia Artificial, Python, Bertrand, colusión, colusión tácita, paralelismo consciente, Markov, aprendizaje automático, aprendizaje por refuerzo, Q-learning

ABSTRACT

Currently, algorithms and Artificial Intelligence are gaining significant importance in the social context due to their application in multiple fields that substantially improve the efficiency and productivity of various economic sectors. They are capable of assisting or even making accurate decisions autonomously, thanks to their powerful data processing and continuous analysis of historical decisions. An area where these technologies can be applied is in price setting through monitoring the movements of competing companies, aiming to establish the profit-maximizing price. While the use of algorithms or Artificial Intelligence is a legitimate strategy for business productivity, literature in this field suggests that price-setting algorithms can either assist humans in colluding or directly facilitate competition restriction for profit maximization by autonomously learning to collude tacitly, without any explicit agreement or communication between algorithms. This text aims to develop all the implicit theoretical concepts in algorithmic tacit collusion to understand how a price-setting algorithm can learn to restrict competition on its own. It concludes with the achievement of a computational economic experiment where a Python designed algorithm is capable of simulating a Bertrand duopoly by simultaneously setting prices and storing all the decisions made by the algorithm to determine if it has learned to collude tacitly automatically and autonomously.

KEYWORDS: Algorithm, Artificial Intelligence, Python, Bertrand, collusion, tacit collusion, conscious parallelism, Markov, machine learning, reinforcement learning, Q-learning.

ÍNDICE

RESUMEN 1
ÍNDICE 3
ÍNDICE DE FIGURAS 4
1. Introducción 5
1.1. Contextualización 5
1.2. Expectativas 6
1.3. Estructura 6
2. Justificación 7
2.1. Pertenencia y actualidad del tema 7
2.2. Justificación de la elección del tema 8
2.3. Interés social y educativo 9
3. Objetivos del trabajo 9
MARCO TEÓRICO 10
4. La colusión tácita: Definición, prácticas colusorias y problemática legal 10
5. Condiciones de mercado que propician prácticas colusivas 12
5.1. Número de competidores 12
5.2. Asimetría de mercado 13
5.2.1. Asimetría de costes 13
5.2.2. Asimetría productiva 15
5.2.3. Asimetría de productos 20
5.3. Barreras de entrada y rotación competitiva 21
5.4. Transparencia 21
5.5. Interacción 22
6. Algoritmos informáticos: qué son y su capacidad para establecer precios y coludir tácitamente 23
6.1. La elusión legal algorítmica y la capacidad de coludir sin intención colusoria 25
6.2. Aprendizaje por Refuerzo: elementos, procesos de decisión de Markov, y política Epsilon-greedy. Q-Learning como mejor algoritmo de aprendizaje por refuerzo fijador de precios 28
7. Estudio particular: colusión tácita algorítmica y duopolio de Bertrand 32
MARCO EMPÍRICO 36
8. Metodología 36
9. Resultados e interpretación 37
9.1. Resultados obtenidos 38
a. Colusión de equilibrio cooperativo 38

a. Colusión de equilibrio dominante 39
9.2. Interpretación 40
9.3. Influencia del número de competidores al nivel colusorio de la toma de decisiones 41
10. Conclusiones 42
11. Futuras líneas de investigación 43
12. Bibliografía 44
13. Anexos 47
a. Código del algoritmo 47

ÍNDICE DE FIGURAS

Figura 1: Proceso de decisión de Markov. Figura extraída de Reinforcement Learning. An introduction (Sutton, 1998) 30
Figura 2: Árbol de decisión de Markov, probabilidades de transción entre estados. Figura de elaboración propia 30
Figura 3: Gráfico de beneficio nulo del modelo simulado de Bertrand. Elaboración propia mediante Python. 34
Figura 4: Gráfico de ejemplo sobre beneficio monopolístico máximo de modelo de Bertrand simulado. Elaboración propia mediante Python. 34
Figura 5: Evolución de beneficios en colusión cooperativa. Elaboración propia mediante Python 38
Figura 6: Evolución de precios en colusión cooperativa. Elaboración propia mediante Python 38
Figura 7: Evolución de colusoria cooperativa. Elaboración propia mediante Python 39
Figura 8: Evolución de beneficios en colusión dominante. Elaboración propia mediante Python 39
Figura 9: Evolución de precios en colusión dominante. Elaboración propia mediante Python .. 39
Figura 10: Evolución colusoria dominante. Elaboración propia mediante Python 40
Figura 11: Evolución de beneficios en colusión con n=3. Elaboración propia mediante Python 41
Figura 12: Evolución de beneficios en colusión con n=4. Elaboración propia mediante Python 41
Figura 13: Evolución colusoria con n=3 Elaboración propia mediante Python 41
Figura 14: Evolución colusoria con n=4. Elaboración propia mediante Python 41
Figura 15: Evolución de precios en colusión con n=3. Elaboración propia mediante Python 42
Figura 16: Evolución de precios en colusión con n=4. Elaboración propia mediante Python 42

1. Introducción

1.1. Contextualización

La irrupción tecnológica de internet, de los *smartphones* y otras tecnologías de la información que facilitan su acceso, han transformado por completo la vida de las personas al "digitalizar" a la humanidad en múltiples aspectos, pues ha revolucionado la forma en que se realizan transacciones comerciales y ha permitido tanto la expansión de mercados y de nuevas oportunidades o modelos de negocio a escala global como el surgimiento de nuevas maneras de promocionar productos y tener visibilidad. Gracias a la computarización económica, las empresas tienen acceso a nuevas herramientas que permiten tomar decisiones más precisas, rápidas y adaptables a un entorno tan cambiante como el actual, sin embargo, a menudo dichas decisiones no son tomadas por el empresario si no por un algoritmo informático, gracia al desarrollo y la investigación de la inteligencia artificial y otros algoritmos de toma de decisiones.

Gracias al gran poder de procesamiento de información de las computadoras, los algoritmos son capaces de tomar decisiones mucho más precisas teniendo en cuenta una infinidad de variables en cuestión de segundos, y de tomar decisiones dinámicas en tiempo real. La automatización de decisiones puede aplicarse a multitud de contextos, incluyendo la fijación de precios. Siendo los precios una variable clave que influye directamente en la demanda, la competitividad y la rentabilidad de las empresas, la capacidad de establecer precios óptimos y adaptativos es fundamental para maximizar los beneficios y mantener una posición competitiva en el mercado, por lo que la utilización de algoritmos fijadores de precios es una opción a considerar por múltiples sectores económicos.

Sin embargo, algunos estudios sugieren que la utilización de algoritmos dotados de aprendizaje automático, es decir, capaces de aprender de forma automática y autónoma a partir de su propia experiencia, puede incentivar o ser el origen de restricciones automáticas de la competencia debido a que pueden llegar a la conclusión de que la mejor decisión de mercado para maximizar beneficios es establecer un precio colusorio con el que obtener beneficios supra competitivos. Así pues, se hace necesario el estudio de las implicaciones económicas que proliferan este tipo de algoritmos en los consumidores y usuarios al existir la posibilidad de experimentar un aumento gradual del nivel de precios.

La característica principal de los algoritmos de aprendizaje automático reside en su capacidad de ofrecer resultados inesperados o imprevistos. Este tipo de algoritmos aprenden a tomar decisiones mediante su experiencia, por lo que sus creadores no definen las decisiones especificas que debe tomar si no que le brindan la capacidad de decidir y de aprender de sus decisiones pasadas. Debido a ello, los propios diseñadores de este tipo de algoritmos no son capaces de predecir las decisiones que puede llegar a tomar un algoritmo, por lo que si un algoritmo fijador de precios establece precios colusorios, existe la posibilidad de que tal decisión no solo no haya sido diseñada intrínsecamente, si no que sea del todo involuntaria.

En gran parte de los ordenamientos jurídicos, las leyes antitrust definen la colusión como aquella restricción de competencia que los empresarios acuerdan para ganar beneficios supra competitivos. Dotan a la colusión de un carácter inherentemente humano, exigiendo la existencia de un acuerdo y de una intencionalidad para restringir la competencia.

Debido a que la colusión algorítmica puede ser tácita, involuntaria, impredecible, y en ocasiones, indetectable, es muy difícil de perseguir legalmente incluso cuando se demuestra su existencia, ya que en ocasiones la colusión tácita algorítmica no se subsume completamente en ningún precepto legal o no existe ningún responsable que deba responder legalmente. En este sentido, los

algoritmos de aprendizaje automático fijadores de precios se vuelven muy atractivos debido a su capacidad de eludir responsabilidad legal en el caso de que aprenda a coludir, y de la posibilidad de obtener beneficios por encima de los competitivos sin mediar acuerdo o intención.

1.2. Expectativas

Con el objetivo de explorar la capacidad de los algoritmos de fijación de precios para restringir la competencia, este estudio se enfocará en desarrollar un algoritmo informático capaz de simular un duopolio de Bertrand mediante técnicas de aprendizaje por refuerzo. El algoritmo será responsable de establecer precios para ambas empresas y supervisar las decisiones tomadas a lo largo del tiempo, con el propósito de demostrar cómo un algoritmo puede aprender a coludir de forma autónoma, sin existir una intención colusoria, comunicación explícita o acuerdo entre las empresas.

En este sentido, gran parte del desarrollo algorítmico en este trabajo se basará en investigaciones previas realizadas por Calvano et al. (2020) en su estudio "Artificial Intelligence, Algorithmic Pricing, and Collusion", y por Nicolas Lepore (2021) en su trabajo "AI Pricing Collusion: Multi-Agent Reinforcement Learning Algorithms in Bertrand Competition". Aunque ambos trabajos han obtenido resultados que difieren entre sí, se espera que el presente estudio, desarrollado completamente en Python, logre resultados más parejos a los obtenidos por Lepore, debido a que utiliza el mismo sistema.

Mediante la implementación de este algoritmo de simulación y aprovechando las técnicas de aprendizaje por refuerzo en inteligencia artificial, se busca explorar cómo los algoritmos de fijación de precios pueden influir en la competencia en un entorno de duopolio. La demostración empírica permitirá comprender como la capacidad para coludir de los mencionados algoritmos no está exenta de la necesidad de encontrarse en mercados con características óptimas para la colusión y que, si bien la colusión algorítmica es posible, es al mismo tiempo improbable, motivo por el cual se hace tan difícil de detectar.

Asimismo, este estudio tiene como objetivo contribuir al conocimiento y la comprensión de la interacción entre la inteligencia artificial y la economía, destacando la importancia de regular y supervisar el uso de algoritmos de fijación de precios para prevenir prácticas anticompetitivas en el mercado. Por lo tanto, se espera que los resultados obtenidos tengan implicaciones tanto en el ámbito académico como en el sector empresarial y regulatorio, proporcionando una base para la toma de decisiones informadas y la implementación de políticas adecuadas.

1.3. Estructura

El presente trabajo consta dividido en dos apartados principales, el marco teórico donde se exponen todas las implicaciones teóricas relativas a la colusión tácita algorítmica, y el marco empírico donde se desarrollan los resultados obtenidos por el algoritmo al fijar precios de manera simultánea en una simulación computacional de un duopolio de Bertrand.

Respecto el marco teórico, consta dividido en cuatro apartados principales. En el primero, se define el concepto de colusión, las diferencias entre la colusión explícita y tácita, su relación con el duopolio de Bertrand, sus similitudes y diferencias con el paralelismo consciente, y el carácter inherentemente humano que le confieren las diferentes legislaciones anticompetitivas al concepto de colusión. En el segundo apartado, se tratan las condiciones de mercado idóneas para que la

colusión tácita prolifere y sus respectivas modelizaciones económicas, incluyendo la modelización propia de un caso teórico particular ausente en la literatura económica existente al respecto. En el tercer apartado se define el concepto de algoritmo, las ventajas de su utilización en detrimento de personal humano, la capacidad de coludir tácitamente al establecer precios, los diferentes tipos de colusión tácita algorítmica que son capaces de generar, cuál es el mejor tipo de algoritmo para cada tipo de colusión, finalizando en una detallada introducción a los diferentes tipos de aprendizaje automático, con especial atención al aprendizaje por refuerzo y sus implicaciones teóricas para el algoritmo desarrollado. Por último, en el cuarto apartado se enumeran diferentes aspectos teóricos implicados en el algoritmo de Q-learning desarrollado, las diferentes asunciones establecidas y sus respectivas limitaciones.

Por otro lado, la parte empírica consta de tres apartados principales. En el primero de ellos, se describe la metodología utilizada para el desarrollo del algoritmo, incluyendo los motivos principales del lenguaje de programación utilizado, y como se generarán los datos para mostrar los resultados deseados. En el segundo apartado se muestran los resultados obtenidos y su interpretación y, finalmente, en el tercer apartado se realizan las conclusiones obtenidas en el presente estudio y las futuras líneas de investigación.

2. Justificación

2.1. Pertenencia y actualidad del tema

Si bien el concepto de inteligencia artificial no es un término completamente nuevo, actualmente se está experimentado una irrupción sin precedentes de inteligencias artificiales que se estan introduciendo como clave angular en el sistema productivo de multitud de empresas incluso en sectores donde no se preveía posible (al menos, a corto plazo) su introducción, como el sector servicios, gracias a los grandes avances en el conocimiento, diseño y desarrollo de inteligencias artificiales de procesamiento de lenguaje natural (PNL). Sus implicaciones en términos de competitividad son muy significativas ya que permiten una mejora de la productividad y una reducción de costes muy considerable al ser capaces de sustituir gran parte del personal humano y de ser capaces de obtener mejores resultados en un sinfín de ámbitos diferentes.

El interés empresarial por el uso de inteligencias artificiales o algoritmos para mejorar su competitividad está llegando a tal punto que su introducción en determinados sectores económicos se está volviendo un requisito indispensable ya no para obtener una ventaja competitiva, si no para mantener el nivel competitivo conseguido y no quedarse atrás respecto los competidores. Debido a ello, si bien la utilización de algoritmos fijadores de precios quizá estaba restringida en empresas grandes, de alto nivel o pertenecientes al sector tecnológico, la proliferación de inteligencias artificiales de PNL permite facilitar mucho el desarrollo de los mencionados algoritmos por lo que es lógico que en un futuro no muy lejano las mencionadas tecnologías se introduzcan de forma generalizada en la mayoría de sectores económicos al ser la utilización de algoritmos un hecho terminante para la obtención de beneficios competitivos y rendimientos positivos.

Teniendo en cuenta los posibles efectos negativos de la fijación de precios mediante algoritmos de aprendizaje automático, la proliferación e introducción generalizada de dichas tecnologías plantea interrogantes significativos en términos de competencia y colusión y su consecuente impacto en el bienestar de los consumidores y en la eficiencia de los mercados. La capacidad de estos algoritmos para aprender y adaptarse dinámicamente, sin la necesidad de una comunicación

explícita o acuerdos entre las empresas, plantea desafíos importantes desde una perspectiva regulatoria y antimonopolio.

La pertinencia de este tema radica en la necesidad de comprender y evaluar el impacto y los desafíos que plantean los algoritmos de fijación de precios en el entorno empresarial actual, comprendiendo su potencial para restringir la competencia y su impacto en el bienestar económico y social. En un contexto en el que las decisiones de precios están cada vez más automatizadas y las empresas dependen de algoritmos inteligentes para competir en el mercado, es esencial examinar las implicaciones económicas, sociales y legales de estas prácticas. En este sentido, el estudio permitirá analizar a fondo las implicaciones y ofrecer conocimientos valiosos sobre la toma de decisiones automatizadas y la posible formulación de futuras políticas regulatorias.

La actualidad del tema escogido destaca por la creciente preocupación en los círculos académicos y gubernamentales sobre las posibles consecuencias negativas de la utilización de algoritmos en los sistemas productivos a costa del perjuicio al consumidor y a la sociedad en su conjunto. La comprensión de cómo los algoritmos aprenden y toman decisiones en entornos de competencia es esencial para desarrollar políticas y regulaciones adecuadas que promuevan una competencia justa y eficiente en la economía digital.

2.2. Justificación de la elección del tema

Ante una introducción tan acelerada de la inteligencia artificial y de otros algoritmos similares en multitud de sectores económicos, será fundamental en un futuro próximo que las autoridades de la competencia conozcan y comprendan como los algoritmos informáticos son capaces de generar colusión tácita cuando fijan precios. Ante esta necesidad, la realización de la presente investigación se justifica por su relevancia práctica, teórica y metodológica para comprender como aflora la colusión tácita algorítmica cuando no se define explícitamente la toma de decisiones colusorias.

Desde un punto de vista práctico, para comprender el funcionamiento de la colusión tácita algorítmica, se propone el desarrollo e implementación de un algoritmo capaz de simular un mercado en el que un número determinado de agentes económicos interactúen competitivamente maximizando beneficios, con el fin de monitorizar sus acciones y evaluar el nivel colusorio de sus decisiones.

Para ello, ha sido necesario construir un marco teórico sólido en torno al objeto de desarrollar el mencionado algoritmo, mediante un análisis exhaustivo de la literatura económica sobre la colusión tanto en su modalidad explícita, tácita, ordinaria o algorítmica. Gracias a dicha investigación, como se puede observar a continuación, se han suplido ciertas lagunas de conocimiento necesarias para confeccionar la simulación de mercado, aportando conocimiento a la comunidad científica y permitiendo una comprensión más precisa del fenómeno objeto del presente estudio, mediante la modelización de un caso particular respecto la determinación de cuotas de mercado en tratos colusorios cuando cada uno de los respectivos agentes económicos son capaces de satisfacer por completo y de forma individual, la totalidad de la demanda de mercado.

Asimismo, el presente trabajo desafía el modelo oligopólico de Bertrand al desarrollar un algoritmo fijador de precios bajo, que contraviene sus disposiciones al presentar decisiones colusorias y no alcanzar un nivel de precios cercano al equilibrio de Nash conforme aumentan el número de iteraciones.

Metodológicamente, los resultados obtenidos en el marco empírico difieren en ciertos aspectos con el resto de estudios existentes relacionados, debido a que se realizan gráficos nuevos o similares pero con información adicional o ampliada, como la posibilidad de observar la evolución de los precios establecidos junto con los periodos de convergencia o el momento en el que se obtiene la tasa de aprendizaje mínima.

2.3. Interés social y educativo

Desde un punto de vista social, la fijación de precios algorítmica tiene implicaciones directas en la vida cotidiana de los consumidores y en el funcionamiento de los mercados. Comprender cómo los algoritmos afectan la competencia y a los precios es fundamental para garantizar un entorno empresarial justo y equitativo. Al centrar el estudio en cómo los algoritmos son capaces de generar colusión tácita, el presente trabajo brinda conocimientos que pueden ser utilizados para la detección y prevención de prácticas anticompetitivas y ayudar a diseñar políticas públicas que promuevan la competencia y protejan a los consumidores, por lo que también contribuye a promover la transparencia y la confianza en el sistema económico beneficiando a la sociedad en su conjunto.

En términos educativos, el presente estudio puede utilizarse como breve introducción a la economía computacional o a la programación de algoritmos de aprendizaje automático, pues el algoritmo propuesto se ha diseñado con conocimientos básicos de programación. Además, la intersección entre la inteligencia artificial y la economía brinda la oportunidad de explorar nuevas perspectivas y enfoques metodológicos en la investigación académica.

3. Objetivos del trabajo

El **objetivo principal** del presente estudio consiste en analizar y comprender cómo los algoritmos fijadores de precios son capaces de aprender a coludir tácitamente de forma completamente autónoma (sin intervención humana) y sin mediar ningún tipo de comunicación.

Dicho objetivo conlleva aparejados otros objetivos específicos cuya consecución es primordial, esencial y necesaria para alcanzar el objetivo general. Dichos objetivos son:

- Investigar y analizar la literatura científica existente sobre la colusión tácita y explícita perpetuada y originada tanto por personas como por algoritmos informáticos.

- Investigar y determinar las condiciones óptimas de mercado que facilitan o permiten aparecer contextos de colusión tácita

- Analizar las diferentes modalidades de colusión tácita algorítmica y evaluar la idoneidad para su generación en distintas clases de algoritmos.

- Desarrollar un algoritmo capaz de simular un mercado donde varios agentes económicos compitan interactuando en el entorno generado computacionalmente, y evaluar el nivel colusorio de las decisiones tomadas por cada empresa.

- Determinar la tipología algorítmica idónea para la elaboración del algoritmo de simulación

- Comparar los resultados obtenidos con el algoritmo desarrollado con los hallados en otras investigaciones previas, identificar similitudes y diferencias, y determinar el éxito del algoritmo desarrollado.

MARCO TEÓRICO

4. La colusión tácita: Definición, prácticas colusorias y problemática legal

En términos económicos, se entiende por colusión el acuerdo que realizan firmas o empresas independientes con las que determinar conjuntamente estructuras de precios y niveles productivos, con el fin último de maximizar ganancias (Stigler, 1964) y de "*coordinar y/o monitorear sus acciones para alcanzar beneficios por encima de los niveles competitivos*" (Garrod & Olczak, 2018), pudiendo alcanzar precios cercanos a los umbrales monopolísticos (Motta, 2004).

Si bien mediante esta definición todos los acuerdos de semejante naturaleza se consideran prácticas anticompetitivas, no toda colusión resulta perseguible legalmente, haciendo deseable para algunas compañías alcanzar una situación de mercado colusoria semejante con la que maximizar sus ganancias y eludir las consecuencias legales. Es por ello que resulta necesario abordar este concepto, a modo introductorio, desde una perspectiva jurídica para discernir entre las diferencias entre una tipología u otra.

Según Ezrachi y Stucke (2015) en *Artificial Intelligence & Collusion: When Computers Inhibit Competition,* para la mayoría de las legislaciones las prácticas colusivas deben conllevar algún tipo de acuerdo en el que haya mediado alguna clase de comunicación directa entre las partes revistiendo dicho trato de un carácter inherentemente humano. Sin embargo, según Harrington (2012) en *A Theory of Tacit Collusion*, una situación colusoria desde la teoría económica, puede esgrimirse tanto con comunicación explícita como con falta absoluta de ella, pues no se centra en cómo se llega a dicha situación si no a sus consecuencias económicas, que resultan ser identicas. La ventaja de coludir sin comunicación o trato es la posibilidad de evadir en ciertas situaciones las consecuencias legales que comportan dichas prácticas anticompetitivas. La colusión sin comunicación ni acuerdo expreso, es lo que se conoce comúnmente como colusión tácita.

Sin perjuicio de parecer una definición contradictoria, la colusión tácita se da cuando se alcanza una situación típicamente coordinada o colusoria, sin mediar comunicación directa ni acuerdo entre las diferentes partes. Generalmente, la colusión tácita es legal a ciertos niveles, cuya ilegalidad radica en la posibilidad de demostrar cierta intención colusoria. *"Uno de los riesgos de los algoritmos es que ellos expanden el área gris existente entre la colusión explicita ilegal y la colusión tácita legal"* (OECD, 2017).

No debemos confundir la intención colusiva con la consciencia colusiva. El paralelismo consciente, según Harrington (2012), es una forma de colusión tácita donde las compañías de cierto sector oligopólico, comprenden de forma independiente (sin mediar acuerdo) que de forma generalizada los intereses de cada una de las empresas convergen en mantener un nivel de precios elevado o incluso por encima del nivel competitivo. El paralelismo consciente suele darse ante la simetría de información de mercado, donde las decisiones que toman cada una de las diferentes empresas se dan por razones competitivamente objetivas, donde una decisión contraria comportaría actuar a niveles infra competitivos. Es por ello que, aunque se dé una situación colusiva desde la perspectiva económica, no puede perseguirse legalmente al no poder obligar a las empresas a no competir.

Un claro ejemplo de paralelismo consciente se da en aquellos sectores oligopólicos con muy pocos vendedores, donde se emula el conocido modelo oligopólico de Bertrand. A modo de ejemplificar e introducir el mencionado modelo, es necesario definir brevemente el concepto que más se usa como solución en teoría de juegos en juegos estratégicos: el equilibrio de Nash.

Dado un juego estratégico con un número *n* de jugadores, el equilibrio de Nash se define como la situación donde cada uno de los jugadores elige cierta estrategia optima al conocer las estrategias del resto de jugadores, por lo que el equilibrio de Nash es aquel conjunto de estrategias donde

ninguno de los jugadores tiene incentivos para cambiar su estrategia puesto que cualquier desvío no produce resultados rentables dadas las acciones de los otros jugadores.

Volviendo al modelo oligopólico de Bertrand, imaginemos un mercado con muy pocas empresas, donde todas las funciones de costes son idénticas e iguales a sus costes marginales. Ante dicha competencia, el modelo de Bertrand advierte que el equilibrio de Nash se encuentra allí donde los beneficios son 0, es decir, allí donde los ingresos son iguales a los costes marginales. Esto se debe a que un competidor comprende que, ante costes iguales, si vende por un precio inferior al del resto de competidores, se quedaría con la totalidad de la cuota de mercado. Esta sería la estrategia óptima. Por otro lado, como el resto de competidores también son conscientes de esta posibilidad, hacen lo mismo. Teniendo todas las empresas (jugadores) la misma estrategia, cada una de ellas sigue dicha estrategia bajando los precios con el fin de ganar cuota de mercado y maximizar beneficios, sabiendo que las consecuencias de desviarse de dicha estrategia comportan perder una significante cuota de mercado pudiendo generar pérdidas. Dada esta situación, se produce un descenso generalizado de los precios que finaliza allí donde los beneficios son 0 (Gibbons, 2011), dado que si los productores bajaran aún más los precios, obtendrían perdidas, no teniendo ningún incentivo para continuar dicha estrategia. Es por ello que el equilibrio de Nash, se encuentra allí donde los ingresos son iguales a los costes marginales (beneficios 0).

Si bien en la situación anterior no hay colusión económica, puesto que los precios se establecen a precios infra competitivos, en la realidad se produce un fenómeno de paralelismo consciente.

De la misma forma que todos los competidores saben, que bajando el precio es posible obtener beneficios casi monopolísticos al acaparar toda la cuota de mercado, también saben que si uno de los que forman el oligopolio sigue dicha estrategia, el resto de los competidores harán lo mismo, ganando un beneficio supra competitivo completamente efímero, propiciando que a largo plazo se minimicen las ganancias hasta su valor nulo. Esta conducta, es lo que se conoce como represalia (Ivaldi et al., 2003). Es por ello que, ante esta situación, en la práctica las empresas que forman parte de estos oligopolios, mantienen al alza los precios repartiéndose cuota de mercado, comportándose de forma coordinada sin mediar comunicación ni acuerdo por ninguna de las partes, al ser **conscientes** de que el descenso de los precios por parte de un miembro, acarrearía el descenso generalizado de los beneficios, por ello mantienen los precios al alza hasta alcanzar un equilibrio que habría resultado idéntico si se hubiera acordado un acuerdo monopolístico entre las partes (Dranove et al, 2017). Como podemos observar, en esta situación parece que se dé un equilibrio de Nash donde los precios se mantienen allí donde se alcanza el beneficio monopolístico dividido entre el número de empresas que conforman el oligopolio, así que si bien la mayoría de la literatura afirma que en el modelo de Bertrand solo existe un equilibrio de Nash allí donde los beneficios son 0, bajo esta perspectiva es posible que existan varios equilibrios de Nash.

De este modo, según Ivaldi et al. (2003) en *The Economics of Tacit Collusion*, la colusión tácita solo es sostenible en el tiempo siempre que las represalias por saltarse las condiciones de la coordinación tácita colusoria sean mayores que los beneficios obtenidos a corto plazo al dejar de coludir. Con represalias, nos referimos tanto a la perdida de beneficios supra competitivos que dejaría de percibir la empresa divergente al romper la coordinación tácita, como las decisiones que toman el resto de los competidores como reacción a la desviación colusoria de una de las empresas que conforman el oligopolio. Las represalias ante el desvío colusorio ocurren como resultado de la pérdida de confianza del resto de competidores en la sostenibilidad de la coordinación tácita al experimentar cómo una empresa decide ganar beneficios a corto plazo por encima del resto de competidores. Asimismo, las desviaciones de la colusión tácita se configuran como beneficios inmediatos puntuales, las represalias se configuran como descensos futuros prolongados de beneficios, por lo que la sostenibilidad de la colusión dependerá, en gran medida, del peso que las diferentes empresas otorgan a los beneficios futuros en detrimento de los actuales, es decir, dependerá de si su factor de descuento está más cercano a 1, que a 0.

Existe un gran abanico de represalias que comprenden desde reacciones simples, como romper con toda coordinación tácita restableciendo la competencia y maximizando beneficios a corto

plazo, a reacciones complejas, como establecer guerras de precios puntuales con las que eliminar competidores o perjudicar especialmente los beneficios de la empresa divergente para eliminarla del mercado.

5. Condiciones de mercado que propician prácticas colusivas

En el apartado anterior, hemos determinado qué es la colusión tácita, la naturaleza de su existencia, las implicaciones legales de su ejercicio y los incentivos que las empresas pueden tener para querer coludir tácitamente. Si bien es legítimo que toda empresa tenga como objetivo maximizar beneficios a niveles supra competitivos siempre que dicha práctica no sea considerada anticompetitiva, hemos de tener en cuenta que la posibilidad de coludir tácitamente depende de varias condiciones de mercado.

Ivaldi et al. (2003) en *The Economics of Tacit Collusion*, proporciona una serie de variables que permiten identificar en qué mercados existe mayor facilidad para coludir tácitamente. Si bien identifica hasta doce factores relevantes para coludir, en el presente apartado únicamente se hará mención a aquellas que resulten más relevantes para el presente estudio:

5.1. Número de competidores

Dado un mercado, a mayor número de competidores más complicada resulta la coordinación tácita. Por un lado, cuantas más empresas participen en un mercado, más difícil se vuelve el consenso y la interpretación común tácita de la conducta que debe realizar una empresa para coludir tácitamente con las demás.

Por otro lado, en un oligopolio colusorio, las ganancias oligopólicas dependen del número de empresas. Así pues, cuantos más competidores hay en un mercado, menores son los beneficios que otorga la colusión, y mayores son las ganancias de desviarse de la coordinación tácita ya que dicho desvío proporciona mayores beneficios y mayor ganancia de cuota de mercado. A modo de ejemplificación:

Consideremos un mercado con n empresas, cuyas funciones de coste son idénticas, y cuyos productos son perfectamente homogéneos. Dado un precio colusorio p_c donde los beneficios totales de mercado son B_T, los beneficios que gana cada empresa coludiendo se calculan como $B_{ic} = \frac{B_T}{n}$, y la cuota de mercado como $\pi_i = \frac{1}{n}$, siendo simétricas todas las cuotas de mercado. Para que una empresa i se desvíe de la colusión y alcance beneficios casi monopolísticos B_M, debe vender sus productos a un precio p_i siempre inferior a p_c. De este modo, $B_T > B_M > B_{ic}$.

Si las ganancias que se pueden percibir del desvío se calculan como $BD_i = B_M - B_{ic} = B_M - \frac{B_T}{n}$, la ganancia de cuota de mercado de la empresa desviada se calcula como $D_{\pi i} = \frac{1-\pi_i}{\pi_i} \cdot 100 = \frac{1-\frac{1}{n}}{\frac{1}{n}} \cdot 100$, y la cuota de mercado resultante de la empresa divergente se calcula como $\pi_D = D_{\pi i} \cdot \pi_i = \frac{1-\frac{1}{n}}{\frac{1}{n}} \cdot 100 \cdot \frac{1}{n}$. Un ejemplo numérico sería:

$B_T = 100$ $B_M = 90$ $n = 4$ $B_{ic} = \frac{100}{4} = 25$ $\pi_i = \frac{1}{4} = 0{,}25$ $BD_i = 90 - 25 = 65$ $D_{\pi i} = \frac{1-0{,}25}{0.25} \cdot 100 = 300\%$ $\pi_D = 300\% \cdot 0{,}25 = 0{,}75$	$B_T = 100$ $B_M = 90$ $n = 20$ $B_{ic} = \frac{100}{20} = 5$ $\pi_i = \frac{1}{20} = 0{,}05$ $BD_i = 90 - 5 = 85$ $D_{\pi i} = \frac{1-0{,}05}{0.05} \cdot 100 = 1.900\%$ $\pi_D = 1.900\% \cdot 0{,}05 = 0{,}95$

Como podemos observar, dado un mercado, a mayor número de empresas mayores incentivos para desviarse de la coordinación al ganar mayor cuota de mercado, haciendo más difícil que se mantenga la colusión tácita.

5.2. Asimetría de mercado

En el supuesto anterior, suponíamos que las cuotas de mercado de cada empresa eran simétricas, es decir, eran idénticas. Cuando en un mercado las cuotas de cada empresa parecen similares, suele darse porque todas las empresas de dicho mercado presentan simetrías en ciertos aspectos, como puede ser la venta de productos perfectamente homogéneos y/o sustitutivos, funciones de coste idénticas o funciones de producción iguales. Estos aspectos, pueden afectar gravemente la facilidad de coludir en cierto mercado si presentan asimetrías.

5.2.1. Asimetría de costes

Imaginemos un duopolio de Bertrand con funciones de costes distintas, donde la empresa A tiene costes altos (C_A) y la empresa B tiene costes bajos (C_B), de manera que $C_A < C_B$. Aunque *"las empresas con menores costes marginales insistirán en precios más bajos que aquel que el resto de firmas preferirían sostener"* (Ivaldi et al., 2003), imaginemos que su demanda *D* es perfectamente inelástica al ser sus consumidores especialmente insensibles al precio, cuya inelasticidad finaliza en cierto punto, en un precio crítico, donde cualquier precio por encima de dicho punto comporta que $D = 0$. De esta forma, la demanda resta inmutable a los cambios de precio siempre que estén por debajo del punto crítico, comprándose la misma cantidad de producto.

Ante esta situación, si las empresas deciden coludir, todas estarán de acuerdo en que lo harán estableciendo un precio colusorio (p_c) igual al punto crítico, pues es donde se maximizan los beneficios, cada una bajo cierta cuota de mercado establecida (π_i). De esta manera, todas las empresas establecerán el mismo precio colusorio, sin que aquellas con costes más bajos prefieran un precio colusorio más bajo.

En este escenario, solo será posible coludir si los beneficios obtenidos en colusión por ambas

empresas son mayores a los beneficios obtenidos en competencia de precios, sin embargo, la posibilidad de coludir recae más en el interés colusorio de la empresa B que de la empresa A.

Bajo el modelo de Bertrand con funciones de coste idénticas, cada firma bajaba el precio en relación al precio contrario con el fin de acaparar la mayor cuota de mercado posible, estableciendo el equilibrio de Nash allí donde los beneficios son 0, es decir, donde el coste marginal es igual al precio. Sin embargo, si ambas firmas se conciencian de que al seguir esta estrategia sólo promueve la minimización de ganancias, se puede producir un caso de paralelismo consciente donde concluyan que la mejor estrategia a seguir es no bajar el precio ya que, si compitieran en precios, acabarían obteniendo 0 beneficios siendo ésta la represalia que surgiría ante cualquier desvío de precio de la empresa contraria.

Sin embargo, cuando en un duopolio de Bertrand ambas firmas presentan funciones de costes distintas, para la empresa B es factible competir en precios y acaparar toda la cuota de mercado simplemente estableciendo un precio menor a los costes de la empresa contraria ya que, si dicha empresa estableciese el mismo precio, obtendría pérdidas. Cuando la empresa B establece un precio tal que $p_B < C_A$, acaba convirtiéndose en una especie de monopolio al ganar beneficios casi monopolísticos (B_{BM})[1]. De esta forma, para que la empresa B esté dispuesta a coludir, necesita un incentivo adicional: que los beneficios obtenidos en colusión compartiendo cierta cuota de mercado bajo un mismo precio sean superiores a los obtenidos en competencia de precios. A modo de ilustración:

Escenario 1	Escenario 2
$C_B = 5$ $C_A = 10$ $D = 100$ $p_C = 20$ $\pi_i = 0.5$	$C_B = 2$ $C_A = 10$ $D = 100$ $p_C = 15$ $\pi_i = 0.5$
Si coluden:	*Si coluden:*
Empresa B:	***Empresa B:***
$B_{Bc} = 0.5 \cdot (100 \cdot 20 - 100 \cdot 5) = 750$	$B_{Bc} = 0.5 \cdot (100 \cdot 15 - 100 \cdot 2) = 650$
Empresa A:	***Empresa A:***
$B_{Ac} = 0.5 \cdot (100 \cdot 20 - 100 \cdot 10) = 500$	$B_{Ac} = 0.5 \cdot (100 \cdot 15 - 100 \cdot 10) = 250$
Si NO coluden:	*Si NO coluden:*
Empresa B:	***Empresa B:***
$p_B < C_A \longrightarrow 9 < 10$	$p_B < C_A \longrightarrow 9 < 10$
$B_{BM} = (100 \cdot 9 - 100 \cdot 5) = 400$	$B_{BM} = (100 \cdot 9 - 100 \cdot 2) = 700$

[1] Casi monopolísticos ya que dicho estado monopolístico se alcanza estableciendo un precio infra monopolístico, pero consiguiendo beneficios supra competitivos al acaparar toda la cuota de mercado

$B_{Bc} > B_{BM}$ ⟶ *Empresa B prefiere* ***coludir***	$B_{Bc} < B_{BM}$ ⟶ *Empresa B prefiere* ***NO coludir***

En el escenario dos, la empresa B prefiere no coludir al alcanzar mayores beneficios compitiendo que coludiendo, sin embargo, si la empresa A estuviese dispuesta a reducir su cuota de mercado en colusión, la empresa B podría preferir coludir. Por ejemplo, si $\pi_A = 0{,}25$ y $\pi_B = 0.75$:

Si coluden	**Si no coluden**
Empresa B:	***Empresa B:***
$B_{Bc} = 0.75 \cdot (100 \cdot 15 - 100 \cdot 2) = 975$	$p_B < C_A \longrightarrow 9 < 10$
Empresa A:	$B_{BM} = (100 \cdot 9 - 100 \cdot 2) = 700$
$B_{Ac} = 0.25 \cdot (100 \cdot 15 - 100 \cdot 10) = 125$	

Empresa B prefiere ***coludir***

Así pues, dado un mercado con empresas que presentan asimetrías de coste, se acabará coludiendo siempre que:

$$\pi_B \cdot (D \cdot p_c - D \cdot C_B) > D \cdot p_B - D \cdot C_B \longrightarrow \pi_B \cdot (p_c - C_B) > p_B - C_B$$

$$p_B < C_A,\ p_B < p_c,\ C_B < C_A,\ \pi_i < 1$$

5.2.2. Asimetría productiva

Asumiendo el mismo escenario que el anterior, asumamos que ambas empresas tiene costes simétricos, pero que tienen capacidades productivas asimétricas. Dichas capacidades marcan la producción máxima de bien que pueden alcanzar dichas empresas. Siguiendo con la misma lógica, la empresa A tiene una alta capacidad (Cap_A) y la empresa B una baja capacidad (Cap_B) productiva, donde $Cap_A > Cap_B$.

La suma de las capacidades de todas las empresas es igual a la capacidad agregada de mercado ($Cap_{AM} = \sum Cap_i$), y ésta debe ser mayor que la demanda D para que exista cierto interés por parte de las empresas para coludir.

Para ilustrar esta condición, imaginemos que en dicho mercado, $Cap_{AM} \leq$ D, donde:

$$Cap_A = 70,\ Cap_B = 30,\ C = 10,\ D = 100,\ p_C = 20$$

Si ambas empresas venden la cantidad máxima que pueden producir al máximo precio posible:

Empresa A:

$B_A = 70 \cdot 20 - 70 \cdot 10 = 700$

Empresa B:

$B_B = 30 \cdot 20 - 30 \cdot 10 = 300$

Si empresa A decide desviarse:

Empresa A:

$p_{A'} < p_A \longrightarrow 19 < 20$

$B_A = (70 \cdot 19 - 70 \cdot 10) = 630$

Si empresa B decide desviarse:

Empresa B:

$p_{B'} < p_B \longrightarrow 19 < 20$

$B_B = (30 \cdot 19 - 30 \cdot 10) = 270$

Al ser $D \geq Cap_{AM}$, cualquier reducción de precio por debajo del precio crítico de cualquier empresa, no implica obtener mayor cuota de mercado ya que la cantidad vendida está limitada a la cantidad productiva máxima. Así pues, cualquier intento de "competir" en precios se traduce en resultados infra competitivos.

De este modo, no existe ni interés en desviarse ni interés en coludir, ya que sin acuerdo explícito o tácito se obtiene el mismo resultado que cualquier pacto colusorio aspira a obtener, es decir, el resultado económico de la colusión se da naturalmente, en este tipo de mercados, bajo máxima competencia.

En este sentido, en un mercado donde $Cap_{AM} >$ D, mediante las conclusiones de Ivaldi et al. (2003), es posible crear las condiciones de incentivo necesarias que se tienen que dar para que, en este tipo de mercados, se prefiera coludir a competir. Así pues, en este tipo de mercados, la empresa *i* preferirá coludir, si y solo si:

$$B_{ECi} \cdot (1-\delta_i) \geq B_{DVi} \cdot \delta_i$$

Donde:

$$\pi_i \cdot (D \cdot p_c - D \cdot C) \cdot (1-\delta_i) \geq (Cap_i \cdot p_{dv} - Cap_i \cdot C) \cdot \delta_i \qquad \text{Si } Cap_i < D$$

$$\pi_i \cdot (D \cdot p_c - D \cdot C) \cdot (1-\delta_i) \geq (D \cdot p_{dv} - D \cdot C) \cdot \delta_i \qquad \text{Si } Cap_i \geq D$$

$$Cap_{AM} = \sum Cap_i$$

$$Cap_{AM} > \text{D}$$

$$p_c > p_{dv}$$

Donde, B_{ECi} *y* B_{DVi}, son los beneficios obtenidos por la empresa *i* según si, respectivamente, decide seguir una estrategia colusoria o desviarse; p_c y p_{dv} son, respectivamente, los precios de colusión y los precios de desvío, π_i es la cuota de mercado de cada una, y δ_i es el factor de descuento de la empresa *i*. El factor de descuento puede definirse como la valoración que da cierta empresa a sus beneficios a corto plazo en detrimento de los de largo plazo, es decir,

cuando más cercano este dicho factor de 1, dicha empresa preferirá seguir una estrategia enfocada al corto plazo que al largo plazo, cuanto más cercano a 0, preferirá seguir una estrategia a largo plazo que a corto.

En este punto, si asumimos que ante la indiferencia de cierta empresa entre coludir o desviarse, es decir, dicha empresa elegirá coludir a competir, puede saturarse la desigualdad que confecciona la condición de incentivos igualándola.

Confeccionada esta condición de incentivos, se deben tener en cuenta varios factores. Por un lado, para toda empresa *i* el precio colusorio será el mismo ya que, siendo la demanda perfectamente inelástica hasta cierto punto, el precio máximo que puede ser establecido es el precio crítico, por ello, para toda empresa, independientemente de la asimetría en capacidad productiva, el precio colusorio será el precio crítico. Por este hecho, si las empresas decidieran coludir, el precio colusorio no sería objeto de discusión. Como los factores de descuento no pueden pactarse al ser valoraciones intrínsecas de cada compañía, el último término (y único) que resta por pactar, son las cuotas de mercado para cada empresa *i*.

Ivaldi et al. (2003), concluye que, con el fin de que cierto pacto colusorio evite los incentivos para desviarse, las cuotas de mercado de cada empresa deben decidirse según la capacidad proporcional de cada una respecto a la capacidad agregada de mercado. Es decir, $\pi_i = Cap_i / (\sum Cap_n)$. Llegados a este punto, tal y como se ha mencionado, los factores de descuento no pueden negociarse al ser valoraciones intrínsecas, pero si podemos determinar cuál debe ser, dadas las cuotas de mercado y el resto de factores, el factor de descuento máximo que pueden tener las empresas para que decidan coludir.

$$B_{ECi} - B_{ECi}\delta_i = B_{DVi}\delta_i$$

$$B_{ECi} = (B_{DVi} + B_{ECi})\delta_i$$

$$\frac{B_{ECi}}{B_{DVi} + B_{ECi}} = \delta_{MAXi}$$

Donde:

$$\frac{\pi_i(D \cdot p_c - D \cdot C)}{(Cap_i \cdot p_c - Cap_i \cdot C) + \pi_i(D \cdot p_c - D \cdot C)} = \delta_{MAXi} \qquad \text{Si } Cap_i < D$$

$$\frac{\pi_i(D \cdot p_c - D \cdot C)}{(D \cdot p_c - D \cdot C) + \pi_i(D \cdot p_c - D \cdot C)} = \delta_{MAXi} \qquad \text{Si } Cap_i \geq D$$

Como $\pi_i = Cap_i / (\sum Cap_n)$, el factor de descuento máximo es el mismo para todas las empresas, es decir, $\delta_{MAXi} = \delta_{MAX}$. Por tanto, cualquier empresa puede tener el factor de descuento que más le convenga, pero solo estarán dispuestas a coludir si $\delta_i < \delta_{MAX}$. De esta manera, la solución aportada por Ivaldi et al. (2003) para el establecimiento de las cuotas de mercado es correcta. Sin embargo, existe un caso particular donde la cuota de mercado calculada $Cap_i / \sum Cap_n$ es incorrecta. Este caso se da cuando las empresas de un mercado, aun presentando asimetrías en la capacidad productiva, cada una de ellas es capaz de satisfacer por completo la demanda de forma independiente, es decir, cuando $Cap_i \geq D$.

En este caso particular, si aplicamos el cálculo de la cuota de mercado de proporcional a la capacidad productiva, no solo se presentan asimetrías en la capacidad productiva si no en el factor de descuento máximo, pues será estrictamente necesario que el factor de descuento máximo de la empresa de menor capacidad sea menor a la de gran capacidad. Para ilustrarlo numéricamente:

Caso general

$Cap_A = 100, Cap_B = 60, C = 10, D = 100, p_C = 20, p_{dvi} < 20$

$\pi_A = 100 / 160 = 0{,}625$ $\qquad$ $\pi_B = 60 / 160 = 0{,}375$

$B_{ECA} = 0.625\ (\ 100 \cdot 20 - 100 \cdot 10\ ) = 625$ $\qquad$ $B_{ECB} = 0.375\ (\ 100 \cdot 20 - 100 \cdot 10\ ) = 375$

$B_{DVA} = (100 \cdot 19 - 100 \cdot 100) = 900$ $\qquad$ $B_{DVB} = (\ 60 \cdot 19 - 60 \cdot 10\ ) = 540$

$\delta_{MAXA} = \frac{625}{625 + 900} = \frac{25}{61}$ $\qquad$ $\delta_{MAXB} = \frac{540}{375 + 540} = \frac{25}{61}$

$$B_{ECi} \cdot (1\text{-}\delta_i) = B_{DVi} \cdot \delta_i$$

$$625 \cdot (1\text{-}\tfrac{25}{61}) = 900 \cdot \tfrac{25}{61} \quad \rightarrow \quad 368{,}85 = 368{,}85$$

$$375 \cdot (1\text{-}\tfrac{25}{61}) = 540 \cdot \tfrac{25}{61} \quad \rightarrow \quad 221{,}31 = 221{,}31$$

Si $\delta_i \leq \frac{25}{61}$: Ambas empresas coludirán

Si $\delta_i > \frac{25}{61}$: Ambas se desviarán

Caso particular

$\boldsymbol{Cap_A = 200, Cap_B = 100}$, $C = 10, D = 100, p_C = 20, p_{dvi} < 20$

$\pi_A = 200 / 300 = 2/3$ $\qquad$ $\pi_B = 100 / 300 = 1/3$

$B_{ECA} = 2/3 \cdot (\ 100 \cdot 20 - 100 \cdot 10\ ) = 666{,}67$ $\qquad$ $B_{ECB} = 1/3 \cdot (\ 100 \cdot 20 - 100 \cdot 10\ ) = 333{,}33$

$B_{DVA} = (100 \cdot 19 - 100 \cdot 100) = 900$ $\qquad$ $B_{DVB} = (\ 100 \cdot 19 - 100 \cdot 10\ ) = 900$

$\delta_{MAXA} = \frac{666{,}67}{666{,}67 + 900} = \frac{20}{47} \approx 0{,}426$ $\qquad$ $\delta_{MAXB} = \frac{333{,}33}{333{,}33 + 900} = \frac{10}{37} \approx 0{,}27$

$$B_{ECi} \cdot (1\text{-}\delta_i) = B_{DVi} \cdot \delta_i$$

$$666{,}67 \cdot (1\text{-}\tfrac{20}{47}) = 900 \cdot \tfrac{20}{47} \quad \rightarrow \quad 382{,}98 = 382{,}98$$

$$333{,}33 \cdot (1\text{-}\tfrac{10}{37}) = 900 \cdot \tfrac{10}{37} \quad \rightarrow \quad 234{,}24 = 234{,}24$$

Si $\delta_A \leq \frac{20}{47}$ y $\delta_B \leq \frac{10}{37}$: Ambas empresas coludirán

Si $\delta_A > \frac{20}{47}$ y $\delta_B > \frac{10}{37}$: Ambas se desviarán

Como se puede comprobar, en este caso particular, aplicando la cuota de mercado proporcional a las capacidades productivas, se obtiene un desequilibrio entre la empresa de mayor capacidad con la de menor capacidad, exigiendo que la de menor capacidad valore en mayor medida los beneficios a largo plazo para que no decida desviarse. En este sentido, de ambas empresas la que mayores incentivos tiene para desviarse es la empresa de menor capacidad, pues gana más desviándose que la de mayor capacidad. Es por ello que solo si su factor de descuento es menor, coludirá.

Este desajuste o desequilibrio se dá al premiar a la empresa con mayor capacidad por el simple hecho de poder ser más productiva, cuando dicho excedente productivo no implica, si decide desviarse, una mayor cuota de mercado o mayores beneficios que la empresa de menor capacidad. Es por ello, que en este caso particular no tiene sentido establecer cuotas de mercado proporcionales a la capacidad.

Asimismo, cuando todas las capacidades productivas son capaces de satisfacer la demanda por complete, las cuotas de mercado, para evitar incentivos a desviarse, deberán ser establecidas como 1/n, siendo n el número de empresas del mercado.

Si se comprueba numéricamente el caso anterior, se puede observar como las condiciones de incentive de ambas empresas son idénticas, por tanto, el factor de descuento máximo también. De esta forma, el desequilibrio causado por las cuotas de mercado proporcionales a las capacidades productivas se subsana. Así pues, el modelo resultante es el que sigue:

$$B_{ECi} \cdot (1-\delta_i) \geq B_{DVi} \cdot \delta_i$$

Donde:

$$\pi_i \cdot (D \cdot p_c - D \cdot C) \cdot (1-\delta_i) \geq (Cap_i \cdot p_{dv} - Cap_i \cdot C) \cdot \delta_i \qquad \text{Si } Cap_i < D$$

$$\pi_i \cdot (D \cdot p_c - D \cdot C) \cdot (1-\delta_i) \geq (D \cdot p_{dv} - D \cdot C) \cdot \delta_i \qquad \text{Si } Cap_i \geq D$$

$$\pi_i = \frac{Cap_i}{\sum Cap_n} \qquad \text{Si } Cap_i \leq D$$

$$\pi_i = \frac{1}{n} \qquad \text{Si } Cap_i \geq D$$

$$p_c > p_{dv}$$

$$Cap_{AM} = \sum Cap_i$$

$$Cap_{AM} > D$$

Si se comparan los resultados del caso particular con el caso general, se puede llegar a las siguientes conclusiones:

Cuando las capacidades de producción de las empresas no son capaces de satisfacer por completo la demanda, tienen menores incentivos para desviarse de la colusión al tener una capacidad limitada que no es capaz de satisfacer la demanda en su totalidad al desviarse, sino simplemente vender toda la cantidad máxima que pueden producir. Por otro lado, en caso de desvío, la empresa que seguía la estrategia colusoria tiene un poder para tomar

represalias mucho menor a si no tuviera limitaciones a su capacidad o, como mínimo, si pudiese satisfacer por completo la demanda de forma independiente.

Así pues, la introducción de limitaciones asimétricas a la capacidad productiva produce efectos contradictorios a la colusión, por un lado, menores incentivos de desvío se traducen en mayores incentivos para coludir, pero por otro lado, menor poder para tomar represalias puede suponer serias dificultades para mantener el trato colusorio, motivo por el cual, la limitación a la capacidad supone, en definitiva, una dificultad a la colusión.

5.2.3. Asimetría de productos

La asimetría de productos hace referencia a la heterogeneidad productiva de un mercado que se da cuando las empresas diferencian su oferta. Dicha diferenciación puede darse en dos supuestos distintos: el supuesto donde la diferenciación se realiza en la calidad del producto, es decir, dado un mercado, el mismo producto es ofertado por múltiples empresas con distintos grados de calidad; o el supuesto donde la diferenciación se realiza en las características de los propios productos, ofreciendo productos que bien podrías considerarse de la misma categoría pero que no son completamente sustitutivos entre sí puesto que, si bien satisfacen la misma necesidad general, no satisfacen las mismas necesidades secundarias (Caves & Williamson, 1985).

Ambos supuestos corresponden a lo que se conoce comúnmente como diferenciación vertical y horizontal, respectivamente. El análisis de estos tipos de diferenciación respecto la colusión resultan sencillos puesto que corresponden al análisis ya realizado respecto la asimetría en costes (diferenciación vertical) y la asimetría en capacidad (diferenciación horizontal).

La diferenciación vertical afecta a la colusión de forma similar a la asimetría en costes, puesto que ambas situaciones son análogas: mientras que en la asimetría en costes, los productos se venden en la misma calidad, pero a costes diferentes, obteniendo la empresa de bajos costes un mayor margen; en la diferenciación vertical, se vende el mismo producto con costes simétricos, pero con calidades diferentes, obteniendo la empresa de alta calidad un mayor margen.

En este sentido, el pacto colusorio no discutirá los precios si no la cuota de mercado que deberá tener cada empresa, otorgando mayor cuota a la empresa de alta calidad al ser la que gana mayor beneficio al desviarse del pacto colusorio. De esta forma, *"cuando las empresas se diferencian por niveles de calidad, la colusión es más difícil, cuánto mayor es la ventaja competitiva de la empresa de alta calidad"* (Ivaldi et al., 2003.).

Por otro lado, la diferenciación horizontal afecta a la colusión de forma similar a la asimetría en capacidad, puesto que ambas situaciones producen análogos efectos a la colusión: *"en general, el impacto de la diferenciación horizontal parece bastante ambiguo (…), en primer lugar, limita las ganancias a corto plazo de la subcotización de los rivales, ya que se hace más difícil atraer a sus clientes. En segundo lugar, también limita (…) la capacidad de las empresas para castigar una desviación potencial"*(Ivaldi et al., 2003).

Ivaldi et al. (2003) también afirma, que en diferenciación horizontal la colusión puede resultar más o menos compleja de mantener dependiendo el tipo de competencia, dependiendo si se compite en precios o en cantidad. A demás, este tipo de asimetría de mercado puede provocar problemas adicionales para coludir cuando no hay transparencia de la información de mercado ya que, con la misma opacidad, la inferencia de los precios o las cantidades de las empresas contrarias resulta mucho más sencilla cuando no existe diferenciación horizontal, que cuando si la hay.

5.3. Barreras de entrada y rotación competitiva

Las barreras de entrada en un mercado afectan tanto al número de competidores como la facilidad para que nuevas empresas entren en dicho mercado. Ambos efectos están intrínsecamente relacionados. Cuando en un mercado existen pocas barreras de entrada, mayor facilidad tienen los nuevos competidores para entrar en dicho mercado y por tanto mayor número de competidores habrá a medida que pase el tiempo.

Si bien la escasez de barreras de entrada dificulta la posibilidad de coludir, dicha dificultad radica en la rotación de competidores y no en el incremento de su número. Para ejemplificar este hecho, imaginemos dos mercados distintos, el mercado A y el mercado B. El mercado A, tiene actualmente 3 competidores, mientras que el mercado B tiene 20. Si bien, según el apartado anterior, de los dos mercados el que tiene más dificultades para coludir es el mercado B, también se debe atender a las barreras de entrada y a la rotación de sus competidores.

El mercado A tiene tan pocas barreras de entrada, que su rotación competitiva es muy elevada, mientras que el mercado B tiene tantas barreras de entrada, que hacen que sea imposible que un nuevo competidor entre en el mercado y por tanto su rotación competitiva es nula. Bajo esta premisa, el mercado B siempre tendrá 20 competidores, y siempre serán los mismos al no existir posibilidad de que otros competidores nuevos accedan al mercado. Por otro lado, el mercado A no solo tiene posibilidades de que su número de competidores aumente con el paso del tiempo, afectando a la facilidad para coludir, sino que es posible que parte de los competidores actuales abandonen el mercado, reemplazándose por otros. Esta rotación, hace que las posibilidades para que el mercado A coluda quede afectadas seriamente.

Si en un mercado la rotación de competidores es muy alta debido a las bajas barreras de entrada, es más difícil para las empresas establecer y mantener acuerdos colusorios a largo plazo, ya que siempre habrá nuevos competidores que intentarán ganar cuota de mercado. Además, los nuevos competidores pueden ofrecer precios más bajos que los incumbentes, lo que puede desestabilizar el acuerdo colusorio y poner en peligro los beneficios de las empresas involucradas.

5.4. Transparencia

El concepto de transparencia de mercado hace referencia a la facilidad y/o capacidad que ostentan las empresas y sus respectivos participantes de cierto mercado en observar u obtener información precisa o conocimiento confiable sobre los precios, cantidades, características de los productos, condiciones de oferta y demanda, así como cualquier otra información relevante para la toma de decisiones o negociaciones comerciales. *"En consecuencia, la transparencia tiene muchas dimensiones"* (Madhavan, 1996).

Según Ivaldi et al. (2003), respecto la transparencia lo importante *"no es lo que observan directamente las empresas, sino la información que las empresas pueden inferir de los datos de mercado disponibles"*. Esto es importante ya que, si bien, es público y notorio que la transparencia tiene efectos contradictorios en la competencia, puesto que por un lado se promueve la competitividad al estar los consumidores mejor informados de los precios y características de los productos, pero por otro lado facilita la colusión tácita entre los productores (Schultz, 2004), la facilidad colusoria que promueve la transparencia depende de múltiples factores, como las características inherentes del propio mercado del que se trate.

En esta misma línea, Ivaldi et al. (2003) argumenta que la transparencia de los mercados puede afectar de maneras distintas a la facilidad para coludir si estamos ante un mercado cuyos productos y tecnologías son estándares y homogéneos, respecto si estamos a un

mercado cuyos productos y/o tecnologías productivas están altamente diferenciadas. Mientras que para el primer tipo de mercado, incluso con poca transparencia de mercado, es fácil acordar un precio colusorio común, al mismo tiempo también facilita a las diferentes empresas a desviarse de la colusión en el momento oportuno debido a la gran cantidad de información de mercado disponible. Por tanto, en este tipo de mercados *"la colusión podría ser fácil de coordinar, pero difícil de hacer cumplir"* (Ivaldi et al., 2003). Por otro lado, en el segundo tipo de mercado, incluso con una alta transparencia de mercado, es difícil proponer un precio común colusorio ante la existencia de múltiples tipos de productos con diferentes características o con diferentes niveles de calidad. Sin embargo, debido justamente a la heterogeneidad de dicho mercado, una vez establecido el pacto colusorio, *"la colusión podría ser relativamente fácil de hacer cumplir una vez acordada, pero casi imposible de coordinar"*, puesto que la transparencia no otorga suficiente seguridad para desviarse de la colusión sin asegurar que lo ganado en desvío compensa las posibles represalias, si es que éstas se pueden dar.

En definitiva, ante cualquier tipo de mercado, puede decirse que a mayor transparencia de mercado mayor facilidad o bien, para coordinar la colusión, o bien, para hacerla cumplir. Así pues, a mayor opacidad de mercado, mayor dificultad para coludir, en términos generales.

5.5. Interacción

Hasta ahora, para todos los modelos mencionados se ha considerado que cada una de las empresas interactuaban entre ellas un número infinito de veces. Es bien conocido que uno de los factores habituales que se tienen en consideración a la hora de identificar mercados colusorios, es el número de veces que interaccionan los diferentes agentes económicos de un mercado.

Con interacciones, se hace referencia no al número de veces que ciertas empresas venden en un periodo dado, si no a la frecuencia en que dichos agentes ajustan sus precios según el precio establecido por los contrarios (Ivaldi et al., 2003). Mediante esta definición, puede hallarse el motivo de porqué la alta interacción puede ser signo de colusión: ante un mercado con pocas interacciones o interacciones alejadas en el tiempo, las represalias ante una desviación colusoria se desvanecen al ser consecuencias lejanas en el tiempo y los cambios en los precios suceden de forma tan esporádica que no es posible realizar inferencias en los datos de mercado para establecer una estrategia colusoria tacita (Ivaldi et al., 2003).

Como definición formal, algunos estudios (Athey et al., 2004) entienden interacción como todo juego del modelo de Bertrand iniciado, finalizado y repetido en la que se establece un precio que maximiza ganancias según los precios establecidos por las empresas contrarias. A efectos del presente estudio, todo modelo económico mencionado considera que se han realizado un número infinito de veces un juego de Bertrand, puesto que es una condición necesaria para que se produzca la colusión tácita que dichos agentes económicos interactúen un número elevado de veces.

6. Algoritmos informáticos: qué son y su capacidad para establecer precios y coludir tácitamente

"Un algoritmo, según la definición formal de Sipser (2013), es una colección simple de instrucciones para realizar una tarea. Mediante esta definición, tal y como comenta Cormen (2013), cualquier acción realizada por una persona podría identificarse como un algoritmo, así que debemos distinguir entre algoritmos informáticos y no informáticos. Ambos tipos son una colección de pasos o instrucciones para realizar cierta tarea, la diferencia radica en que los informáticos necesitan que dichas instrucciones estén suficientemente definidas para que un dispositivo de computadora sea capaz de ejecutarlo, ofreciendo soluciones a problemas computacionales" (Sánchez, 2023), como por ejemplo establecer y fijar precios a ciertos productos según ciertas variables.

La utilización de algoritmos informáticos en detrimento de personal humano para establecer precios viene motivado por la gran ventaja que supone la computarización de dichas tareas dadas las inherentes características de los algoritmos informáticos. Según Brundage et al. (2018), la gran ventaja que supone el uso de algoritmos informáticos para la realización de ciertas tareas, es debido a que la mayoría son tanto eficientes como escalables. Un algoritmo es eficiente si *"puede cumplir cierta tarea más rápido o más barato que un humano"*, mientras que es escalable si *"completada cierta tarea, incrementando el poder computacional o haciendo copias del propio sistema permite completar un mayor número de instancias de la misma tarea"*.

En esta misma línea, un algoritmo informático fijador de precios puede ser tanto eficiente como escalable. Generalmente, toda fijación de precios se realiza mediante la confluencia entre la oferta y la demanda de un determinado producto, pero dichas variables también dependen de otros factores muy diversos y complejos que pueden variar de un mercado a otro. Para las personas, determinar un precio teniendo en cuenta la misma cantidad de factores que un algoritmo informático puede ser muy costoso tanto en tiempo como en términos económicos, pero para una computadora, es factible que tenga en cuenta una infinidad de variables. En este sentido, un algoritmo informático fijador de precios puede ser fácilmente eficiente en términos de velocidad, gracias a su gran poder de procesamiento de datos, pero también más fiable, gracias a que puede tener en cuenta muchos factores a la hora de tomar decisiones. En este sentido, a mayor número de variables tenga en cuenta dicho algoritmo, más complejo y costoso se vuelve su desarrollo, ya que cada calculo e instrucción debe definirse, por lo que la eficiencia en términos económicos puede diluirse, al menos, a corto plazo, pero a largo plazo el coste puede ser fácilmente amortizable a medida que se establecen mayor número de veces.

Si bien no es imposible que las personas repliquen la eficiencia algorítmica en términos de la cantidad de variables a la hora de establecer precios, es prácticamente inviable que puedan replicar su escalabilidad en términos de velocidad. La eficiencia algorítmica permite que un algoritmo establezca precios en períodos de tiempo muy cortos, dependiendo de la complejidad del algoritmo y del número de variables involucradas en la toma de decisiones. Un algoritmo con estas características puede ser utilizado para establecer precios de manera dinámica en tiempo real, recalculando el precio en función de los cambios experimentados por las variables que conforman el precio desde la última fijación. Este proceso no puede ser replicado por las personas de manera efectiva utilizando métodos tradicionales basados en procesos cognitivos, se necesitaría una cantidad excesiva de recursos humanos y de coordinación.

Bajo lo expuesto, la fijación de precios algorítmica no solo tiene ventajas en términos de eficiencia y escalabilidad si no que, además, esconde otras ventajas competitivas no perceptibles a simple vista desde la perspectiva colusoria. En este sentido, Ezrachi y Stucke (2015), establecen hasta cuatro categorías diferentes de colusión tácita que pueden provocar los algoritmos informáticos fijadores de precios.

Como primera categoría, los mencionados autores presentan la colusión algorítmica tipo *messenger*. Este tipo de comportamiento se da en un escenario donde las personas implicadas utilizan algoritmos computacionales para ejecutar su voluntad de coludir. El uso de dichos algoritmos no permite otra cosa que monitorizar e implementar políticas automáticas para restringir la competencia, aceptando la creación de un cartel, es decir, aceptando un acuerdo. Bajo esta perspectiva, este tipo de colusión implica realizar algún tipo de acuerdo con la que coordinar la restricción de la competencia, revistiendo mayor carácter explícito que tácito, puesto que los algoritmos solo facilitan el trabajo para coludir, siendo necesario que exista un acuerdo horizontal donde se coordine de forma no independiente la restricción de la competencia. A mayor abundamiento, las conductas englobadas en este tipo de colusión se consideran ilegales, motivo de más para categorizarlas, a efectos del presente trabajo, de colusión explicita y no tácita.

Como segunda categoría, la colusión tipo *Hub and Spoke* ocurre en escenarios donde la restricción de la competencia no se realiza mediante la coordinación horizontal algorítmica de cada una de las empresas, si no mediante acuerdos verticales. En este caso, todos los competidores no acuerdan coludir entre sí, si no que todos acuerdan que una sola empresa, mediante un algoritmo fijador de precios, sea quien restrinja la competencia. Esta empresa central es el Hub operacional, que funciona como intermediaria de las coordinaciones para restringir la competencia. Si la coordinación se efectúa bajo el paraguas de acuerdos colusorios con la empresa Hub, donde ha mediado comunicación directa, este tipo de conductas son consideradas anticompetitivas, ilegales y explícitamente colusorios. Sin embargo, a diferencia de la anterior categoría, en este tipo de escenarios no es necesaria la existencia de ningún acuerdo expreso con la empresa Hub. Ésta, puede confeccionar un algoritmo capaz de compartir cierta información con los competidores con el fin de influir en la fijación de sus precios, produciendo un efecto similar a si se hubiese producido acuerdo. Sólo en casos de similar naturaleza, se consideran tácitamente colusoria las mencionadas conductas a efectos del presente estudio. Sin embargo, cabe destacar que se haya mediado o no acuerdo, estas prácticas pueden conllevar por parte de la empresa Hub, cierta intención colusoria. De este modo, algunas legislaciones utilizan la intencionalidad para considerar ilegales este tipo de colusión, por lo que únicamente pueden eludir parcialmente las consecuencias legales de la colusión si no ha mediado intencionalidad, o de si ésta es difícilmente demostrable.

Finalmente, cómo tercera y cuarta categoría, nos encontramos con la colusión de tipo *Predictable Agent* y de tipo *Digital Eye*. Ambas categorías, comparten la característica de aparecer en escenarios donde diferentes compañías de un sector homogéneo en concreto desarrollan e implementan algoritmos fijadores de precios de forma completamente indirecta y unilateral.

En la colusión de *Predictable Agent*, donde los algoritmos fijan precios **prediciendo las cambiantes condiciones de mercado**, los precios pueden alcanzar niveles supra competitivos por dos razones principales interdependientes pero distintas. Por un lado, dichos algoritmos suelen tener diseños y sistemas funcionales muy parecidos. Por otro lado, las condiciones de mercado son de dominio público, son transparentes, propiciando a la generación de situaciones de información simétrica. Si todas las empresas tienen la misma información disponible, y ante dicha información reaccionan racionalmente de formas parecidas, lo que sucede es que se genera una situación de paralelismo consciente, donde todas actúan al unísono sin mediar acuerdo. En esta situación, la colusión no se da como resultado de colusión explícita, si no como resultado natural de la colusión tácita. Como se presume por lo expuesto, en este tipo de escenario no existe intención colusoria, cada una de las empresas deciden desarrollar dichos algoritmos con fines competitivos, pues dado un mercado, a mayor porcentaje de empresas tengan algoritmos fijadores de precios, menor incertidumbre estratégica. En este sentido, no desarrollar tu propio algoritmo supone una desventaja competitiva. Aunque éste escenario pueda presumirse completamente legal, si se demuestra que algún agente ha podido facilitar que sé de ésta situación colusoria, puede ser que se le atribuyan ciertas consecuencias legales.

En la colusión de *Digital Eye*, los algoritmos no fijan precios prediciendo y analizando las condiciones de mercado, si no que **establecen precios en función de un objetivo particular: la**

maximización de beneficios. La colusión se produce cuando los diferentes algoritmos, de forma completamente independiente y sin mediar ningún tipo de comunicación entre ellos, fijan un precio idéntico o similar al ser éste el que maximiza beneficios para la empresa. El precio escogido no ha sido resultado de ningún acuerdo, no existe intención de coludir, simplemente mediante la experiencia adquirida, cada uno de los algoritmos llegan a la misma solución que si se hubiera realizado acuerdo. Al no haber ningún rastro de acuerdo ni intención de coludir, esta situación es completamente legal no habiendo ningún sujeto directo o indirecto a quien atribuirle responsabilidad legal. Es otro caso de paralelismo consciente.

De los tipos de colusión tácita algorítmica expuestos, se extraen dos ventajas más que ostentan los algoritmos establecedores de precios que están estrechamente correlacionadas: la elusión legal algorítmica y la capacidad de coludir sin intención colusoria.

6.1. La elusión legal algorítmica y la capacidad de coludir sin intención colusoria

Un algoritmo informático fijador de precios no tiene intención colusoria cuando la intención de sus propios desarrolladores o de su particular diseño no implica inicialmente facilitar la colusión entre sus usuarios. Cuando, sin aras de coludir, el algoritmo informático acaba coludiendo, nadie es legalmente responsable de las consecuencias económicas anticompetitivas y, por tanto, la elusión legal es prácticamente completa.

En este sentido, desde la perspectiva de la responsabilidad legal, Ezrachi y Stucke, (2015) destacan la importancia de *"distinguir entre la fase de iniciación e implementación. Cuando la colaboración o la comunicación dominan la fase inicial (humana), la falta de ella en la fase de implementación (computarizada) no sirve para evadir la responsabilidad"* (colusión tipo *Messenger*), por otro lado, la falta de colaboración o comunicación en la fase inicial pero presente en la fase de implementación, tampoco la evade (colusión tipo *Hub and Spoke*), pues sigue existiendo intención colusoria, sin embargo, *"cuando la comunicación o colaboración no están presentes ni en la fase de iniciación ni en la de implementación, la cuestión de la responsabilidad se vuelve más complicada"* (colusión tipo *Predictable Agent* y *Digital Eye*).

Dicho esto, que la responsabilidad no se evada no significa que dichos tipos de colusión no tengan capacidad elusiva. En la colusión tipo *Messenger* y *Hub and Spoke*, aun careciendo colaboración o comunicación en una de sus fases, la elusión de la responsabilidad no se da por no tener intención colusoria (que la hay, sino no se habría comunicación o colaboración) si no por la complejidad a detectar y demostrar la existencia de colusión tácita algorítmica (Ezrachi & Stucke, 2015).

De esta manera, los algoritmos fijadores de precios no solo aventajan a las personas al ser capaces de establecer precios de un modo mucho más eficiente y escalable, sino que además, son capaces de coludir intencionalmente dificultando las posibilidades de ser detectados por las autoridades anticompetitivas y, a mayor abundamiento, en caso de no tener intención de coludir, siempre existe la posibilidad (bajo ciertos requisitos) de que se generen beneficios supra competitivos sin ni siquiera percibirlo, debido a la capacidad de algunos algoritmos de coludir sin haberlos diseñado expresamente para ello, sin existir ni comunicación ni colaboración por ninguna de las partes y, en consecuencia, por ser completamente irresponsables legalmente de las consecuencias legales anticompetitivas. Así pues, de todos los tipos de algoritmos fijadores de precios, los más atractivos para las empresas son aquellos capaces de coludir en *Predictable Agent* y *Digital Eye*, ante la posibilidad de que se coluda sin pretenderlo y sin consecuencias previsibles.

Sin embargo, no todo algoritmo informático es capaz de coludir en dichas modalidades. Para que un algoritmo coluda de esta forma, es necesario que, en primer lugar, no exista ni intención colusoria, ni comunicación o colaboración en ninguna de las fases, ni por parte de la empresa ni por parte de los desarrolladores del algoritmo; y que, por tanto, en segundo lugar, sin haberlo dispuesto a tal efecto, el algoritmo acabe **aprendiendo a coludir de forma autónoma y**

automática sin intervención humana, al llegar a la conclusión de que la maximización de beneficios se da al establecer cierto precio que, además, acaba siendo el colusorio.

En este sentido, un algoritmo de dichas características solo puede darse si es capaz de automatizar procesos, de tomar decisiones de forma autónoma, y si es capaz de aprender de forma autónoma y automática, en base a sus decisiones pasadas (experiencia adquirida), la mejor decisión que maximiza beneficios, es decir, si utiliza *machine learning*.

El *machine learning*, o aprendizaje automático, tiene múltiples definiciones. Dicho término, fue acuñado por primera vez por Arthur Samuel en 1959, definiendo el aprendizaje automático de manera general como el *"el campo de estudio que confiere a las computadoras la capacidad de aprender sin ser programadas explícitamente"*. Aurélien Géron (2019) o Kevin P. Murphy (2012), definen funcionalmente el *machine learning* como *"la ciencia (y el arte) de programar computadoras para que puedan aprender de los datos"* por si mismas (Géron, 2019) o como algoritmos informáticos que contienen una serie de *"métodos que pueden detectar automáticamente patrones en los datos y luego usar los patrones descubiertos para predecir datos futuros o para realizar otros tipos de toma de decisiones bajo incertidumbre"* (Murphy, 2012). Si se desea una definición más técnica, Tom Mitchell (1997) define el aprendizaje automático cómo aquel algoritmo que, a la hora de realizar cierta tarea *T*, almacena sus resultados en forma de experiencia *E*, con la cual el rendimiento *P* de la tarea *T*, mejora a medida que se adquiere experiencia *E*.

Aurélien Géron (2019) distingue hasta cuatro tipos diferentes de aprendizaje automático: el aprendizaje supervisado, no supervisado, semi supervisado, y el aprendizaje por refuerzo. La diferencia principal entre los tres primeros tipos de *machine learning*, radica en el tipo de datos que se utilizan para aprender y mejorar el desempeño de cierta tarea. Por otro lado, el aprendizaje por refuerzo, tal y como indica Gerón (2019), *"es una bestia muy diferente"*.

Según el mismo autor, el aprendizaje supervisado consiste en conferirle al algoritmo un conjunto de datos **etiquetados** en los que se incluye la respuesta correcta esperada que debería dar el programa a ciertos problemas que se proporcionan de ejemplo. De esta manera, el algoritmo debe ser capaz de establecer patrones o mapear correctamente las entradas a las salidas esperadas con el objetivo de que, ante una entrada desconocida, el algoritmo sea capaz de predecir estadísticamente la solución correcta. Por otro lado, en el aprendizaje no supervisado se confieren conjuntos de datos no etiquetados sin proporcionar soluciones deseadas explicitas, puesto que el objetivo en este tipo de aprendizaje es simplemente la detección de patrones, anomalías, estructuras de datos, correlaciones u otras métricas estadísticas. Finalmente, y tal y como se presume de lo expuesto, el aprendizaje semi supervisado se configura como una técnica de aprendizaje automático la cual utiliza datos tanto etiquetados como no etiquetados para mejorar la capacidad del algoritmo para detectar patrones y realizar predicciones precisas. En este tipo de aprendizaje, se proporcionan algunos datos etiquetados que indican la solución deseada para ciertos problemas, y datos no etiquetados sin proporcionar la respuesta correcta. La combinación de ambos tipos de datos permite al algoritmo aprender útiles patrones complejos en los datos no etiquetados que ayudan a mejorar su capacidad para hacer predicciones precisas ante nuevas entradas a través de la información proporcionada por los datos etiquetados.

En todo caso, los tres tipos de aprendizaje automático (supervisado, no supervisado y semi supervisado) tienen en común que utilizan datos de entrenamiento para entrenar el algoritmo con el fin de que proporcione soluciones correctas, con la diferencia de que en cada tipo de aprendizaje se utilizan datos de diferente naturaleza. Por otro lado, lo que diferencia estos tres tipos de aprendizaje con el aprendizaje por refuerzo, no radica en la naturaleza de los datos sino el modo en que se entrena el algoritmo.

Mientras que en los algoritmos que utilizan datos de entrenamiento se les "entrena" para que hallen patrones con los que encontrar soluciones, en los algoritmos que utilizan aprendizaje por refuerzo, no se les confiere datos si no que estos los extraen del entorno al interactuar con él tomando diferentes decisiones de acuerdo a su experiencia adquirida. En este sentido, a dicho

algoritmo se le confieren recompensas o penalizaciones en función de si la decisión que ha tomado ha generado resultados deseados o no deseados. En este tipo de algoritmos, los datos del entorno son desconocidos, por lo que mediante la estadística y la retroalimentación obtenida al interactuar con dicho entorno, se obtiene mediante la iteración prolongada de tomas de decisiones una política de actuación optima que maximiza a largo plazo las recompensas esperadas.

En este sentido, puede observarse que el único tipo de aprendizaje automático que puede llegar a aprender a coludir tácitamente (de forma autónoma, sin intervención humana y de manera automática) en la modalidad de *Predictable Agent* o *Digital Eye*, sin mediar comunicación, colaboración o intención colusoria en ninguna de las fases de iniciación o implementación, es decir, siendo completamente legal, es el **aprendizaje por refuerzo**. Existen varios motivos principales a la hora de realizar tal afirmación.

En primer lugar, los aprendizajes basados en datos que tienen algún tipo de supervisión, por definición, no son completamente autónomos. En su fase inicial, se requiere conferir a dichos algoritmos de datos de entrenamiento para que aprenda que soluciones son las deseadas ante ciertos datos etiquetados, de manera que, ante nuevas entradas, prediga el resultado en base a lo aprendido. De esta manera, dichos algoritmos requieren de intervención humana para su correcto funcionamiento en, como mínimo, su fase inicial. Eso no quiere decir, que posteriormente, en su fase de inferencia no aprenda de forma completamente autónoma, pero implica que si cierto algoritmo acaba aprendiendo a coludir, exista la posibilidad de que parte de dicho aprendizaje pueda responsabilizarse a sus desarrolladores debido a los datos proporcionados (sesgos), a la manera en que se han proporcionado, en la manera en que el algoritmo interactúa con dichos datos (solución deseada: colusión), etc.

Por otra parte, el aprendizaje supervisado tiene como objetivo *"extrapolar o generalizar sus respuestas para que actúe correctamente en situaciones no presentes en el conjunto de entrenamiento. Es un tipo importante de aprendizaje, pero por sí solo no es adecuado para aprender de la interacción"* (Sutton, 1998). En este sentido, como ya se ha mencionado en los factores económicos que permiten la colusión, la interacción es un requisito indispensable para ello por lo que este tipo de aprendizaje no resulta idóneo desde el punto de vista funcional.

En segundo lugar, el aprendizaje no supervisado sí que puede considerarse, por definición, autónomo en todas sus fases, pues no requiere de intervención humana. Como se ha indicado, este tipo de aprendizaje trata de encontrar patrones, correlaciones y otras métricas estadísticas en entradas de datos no etiquetados. En este sentido, la solución deseada es el hallazgo mismo de dichos patrones, patrones que posteriormente se utilizarían para que el algoritmo estableciera un precio. Sin embargo, este tipo de algoritmos tienen limitaciones según el tipo de mercado en el que operen, por ejemplo, en mercados cuya transparencia es limitada, existiría la imposibilidad de obtener una gran cantidad de datos de mercado necesarios para una fijación de precios adecuada y satisfactoria.

Sin embargo, dada la gran habilidad de este tipo de algoritmos para detectar patrones, pueden llegar a ser muy aptos para coludir en la modalidad de *Hub and Spoke*. En este sentido, las empresas subsidiarias a la empresa *Hub*, podrían utilizar algoritmos de aprendizaje no supervisado para detectar patrones en la información compartida por la empresa central con el fin de averiguar en que situaciones debe bajarse, mantener o subir el precio establecido.

Finalmente, teniendo en cuenta todo lo expuesto, el aprendizaje por refuerzo es el más indicado para coludir en *Predictable Agent* y *Digital Eye*. En primer lugar, debe destacarse que aunque este tipo de aprendizaje automático pueda considerarse dentro de la rama de los **no supervisados** ya que, directamente, no utiliza datos de entrenamiento para hallar patrones y establecer predicciones a partir de ellos, sino que interactúa con el entorno y, a partir de las decisiones tomadas en el pasado en una situación idéntica o similar, toma la decisión que maximice beneficios, el aprendizaje por refuerzo es considerado *"un tercer tipo de paradigma de aprendizaje automático, junto con el aprendizaje supervisado, no supervisado y quizás otros paradigmas"* (Sutton, 1998).

Como este tipo de aprendizaje requiere interaccionar con el entorno, cuanta menor transparencia de mercado más dificultad tendrá un algoritmo por refuerzo a la hora de establecer precios, de igual manera que un algoritmo no supervisado, sin embargo, la ventaja del aprendizaje por refuerzo es que permite interactuar en entornos cuyos datos se desconocen, es decir, en entornos desconocidos. En el resto de aprendizajes automáticos, se requiere, al menos, desde el inicio, cierta cantidad de datos para entrenar el algoritmo. En el aprendizaje por refuerzo no es necesario puesto que, la primera decisión, suele ser aleatoria y, en función de su impacto, valorar si fue una buena decisión no. Así pues, no es que la poca transparencia de mercado suponga una dificultad para el aprendizaje por refuerzo, si no que a mayor transparencia mayor facilidad para tomar decisiones acertadas.

Por otro lado, gracias a que su aprendizaje se basa en la retroalimentación de sus propias decisiones tomadas, un algoritmo por refuerzo es capaz de operar en entornos muy dinámicos y en tiempo real, característica fundamental para coludir en la modalidad de *Digital Eye*, al tener en cuenta que en este tipo de colusión los agentes económicos están constantemente monitoreando los movimientos de los contrarios y tomando decisiones al respecto. En este sentido, es discutible si este tipo de aprendizaje es indicado para coludir en la modalidad *Predictable Agent*, pues en este escenario lo que se espera de los algoritmos es que predigan las acciones de los competidores con el fin de establecer un precio en consecuencia, cosa que los aprendizajes automáticos supervisados y no supervisados están diseñados para ello. En esta línea, el aprendizaje por refuerzo está diseñado de tal manera que es innecesario predecir los movimientos del contrario a fin de establecer el mejor precio, pues con este tipo de algoritmos se evalúan todas las decisiones tomadas en el pasado por lo que, por inferencia, se acaba estableciendo el mejor precio incluso desconociendo los movimientos contrarios.

6.2. Aprendizaje por Refuerzo: elementos, procesos de decisión de Markov, y política Epsilon-greedy. Q-Learning como mejor algoritmo de aprendizaje por refuerzo fijador de precios

Anteriormente se han introducido ciertas ideas y conceptos clave sobre el aprendizaje por refuerzo a fin de justificar la concentración del presente estudio a esta rama de aprendizaje supervisado. Sin embargo, no se ha proporcionado una definición formal ni se han mencionado todos los elementos y extremos de esta modalidad de *machine learning*.

El aprendizaje por refuerzo, o *reinforcement learning*, es un tipo de aprendizaje automático que permite un algoritmo mejorar sus resultados o tomas de decisiones a medida que dicho algoritmo interacciona repetidamente en cierto entorno. El "aprendizaje" surge como resultado directo de las consecuencias de la decisión tomada en el entorno. Ante cualquier decisión, el entorno reacciona a dicha decisión otorgando ciertos resultados, que son almacenados por el algoritmo según las decisiones tomadas y calcula, en consecuencia, cual es la estrategia o la toma de decisión óptima que maximiza los resultados a largo plazo. En este sentido la interacción con el entorno produce dos tipos de resultados: una recompensa inmediata (o coste) y una evolución del entorno debido a la decisión tomada (Puterman, 2005). En pocas palabras, es un algoritmo que, a base de **prueba y error** (Sutton, 1998), elige la mejor decisión según la experiencia adquirida. Esto tiene varias implicaciones.

En primer lugar, existe cierto intervalo de aprendizaje donde el resultado esperado por las decisiones tomadas por el algoritmo no son acertadas. Es habitual en este tipo de algoritmos establecer decisiones iniciales aleatorias con el fin de "tentar" al entorno y ver sus reacciones. A medida que se interacciona, mayor información se recoge y, por tanto, más adelante podrán tomarse decisiones maximizadoras de beneficio. Por tanto, no permite garantizar buenas tomas de decisiones iniciales.

En segundo lugar, al ser un aprendizaje basado en el prueba y error, es necesario establecer un equilibrio de lo que se conoce comúnmente en aprendizaje automático como "exploración" y "explotación". La exploración, consiste en tomar decisiones o interactuar con el entorno de forma aleatoria y de manera consciente, al ser el único modo de adquirir cierta experiencia que, de otra forma, *"nunca podría ser vista"* (Sutton, 1998). La necesidad de tomar decisiones aleatorias radica en el hecho de que, si solo se toman decisiones explotadoras, nunca se sabrá si exista una decisión mejor que la establecida como óptima. Por otro lado, la explotación consiste en tomar aquella decisión que maximiza el resultado a largo plazo.

En este sentido, el equilibrio entre la explotación y la exploración es dinámico conforme se interacciona con el entorno. A menores interacciones, más se explora con el fin de adquirir experiencia y, conforme se explora el entorno, más se explota con el fin de obtener beneficios. Así pues, es muy común para este tipo de algoritmos establecer tasas de exploración elevadas en las primeras interacciones y disminuir dicha tasa a medida que se interacciona.

En tercer lugar, es posible que incluso en fases tardías, teniendo vasta experiencia y habiendo interaccionado repetidamente un número elevado de veces, el algoritmo siga "explorando" con el fin de averiguar si en ese mismo momento, las mejores decisiones registradas siguen siendo las mejores, o si algo en el entorno a cambiado. Esto supone una ventaja y un inconveniente a la vez. La ventaja radica en que, incluso en entornos cambiantes, el algoritmo es capaz de adaptarse y mejorar sus decisiones. La desventaja es que, con el fin de explorar nuevas opciones, es posible que dicha decisión acarree perjuicios ocasionales y, por otro lado, que ante un número elevado de interacciones la tasa exploratoria sea tan baja, que el algoritmo no sea capaz de seguir el ritmo evolutivo del entorno. Todo esto varía según el caso concreto.

De esta forma, se configura un problema típico de aprendizaje por refuerzo. Formalmente, este tipo de problemas se basan en los procesos de decisión de Markov o, por sus siglas en inglés, MDP (*Markov decisssion processess*). El MDP, es un método o técnica matemática de modelizar un problema secuencial de tomas de decisiones (Puterman, 2005). Para identificar la diferencia entre los MDP y el aprendizaje por refuerzo, es necesario conocer los elementos comunes que los conforman. Tal y como se extrae de *Reinforcement Learning. An introduction* (Sutton, 1998):

1. **Agente:** es el sujeto que interactúa con el entorno tomando decisiones y aprendiendo de ellas, es decir, el propio algoritmo.

2. **Entorno:** es el marco en el cual el agente interactúa. Ante cada interacción el entorno devuelve al agente una recompensa o un perjuicio en base a la decisión tomada. El agente en el entorno interactúa en una secuencia de pasos discretos *t*.

3. **Recompensa o perjuicio (R_{t+1}):** es el valor numérico inmediato que el entorno devuelve al agente al interactuar en él, mide cuan buenas o malas han sido las acciones tomadas por el agente. El objetivo del agente, y lo que define todo el problema del aprendizaje por refuerzo, es maximizar a largo plazo la cantidad total de recompensa recibida.

4. **Estado (S_t):** es la situación particular en la que se encuentra el entorno en un momento *t* concreto.

5. **Acción:** es la decisión que toma el agente para interactuar con el entorno..

6. **Función de valor ($V(s)$):** es la función que determina la recompensa acumulada a largo plazo que puede obtenerse al elegir cierta acción en cierto estado *t*.

7. **Factor de descuento (γ):** determina el peso que el algoritmo confiere a la obtención de recompensas futuras en detrimento de las inmediatas.

8. **Política épsilon-greedy:** determina el equilibrio entre la tasa de exploración (α) y explotación (1-α). "Epsilon" (ε) es el parámetro que controla la probabilidad de que el agente elija una acción aleatoria en lugar de la acción que produce la mayor función valor *V(s)*. Por tanto, "greedy" es el parámetro que controla la probabilidad de que el agente elija aquella función que maximiza la función valor *V(s)*, es decir, 1- ε.

9. **Probabilidad de transición (*p*):** es la probabilidad de que estando en cierto estado actual S_a, se pase a cierto estado S_t según la acción realizada A_t

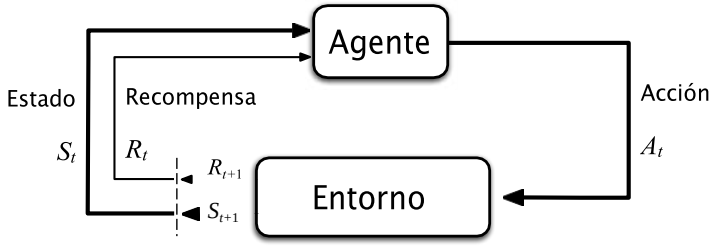

Figura 1: Proceso de decisión de Markov. Figura extraída de Reinforcement Learning. An introduction (Sutton, 1998)

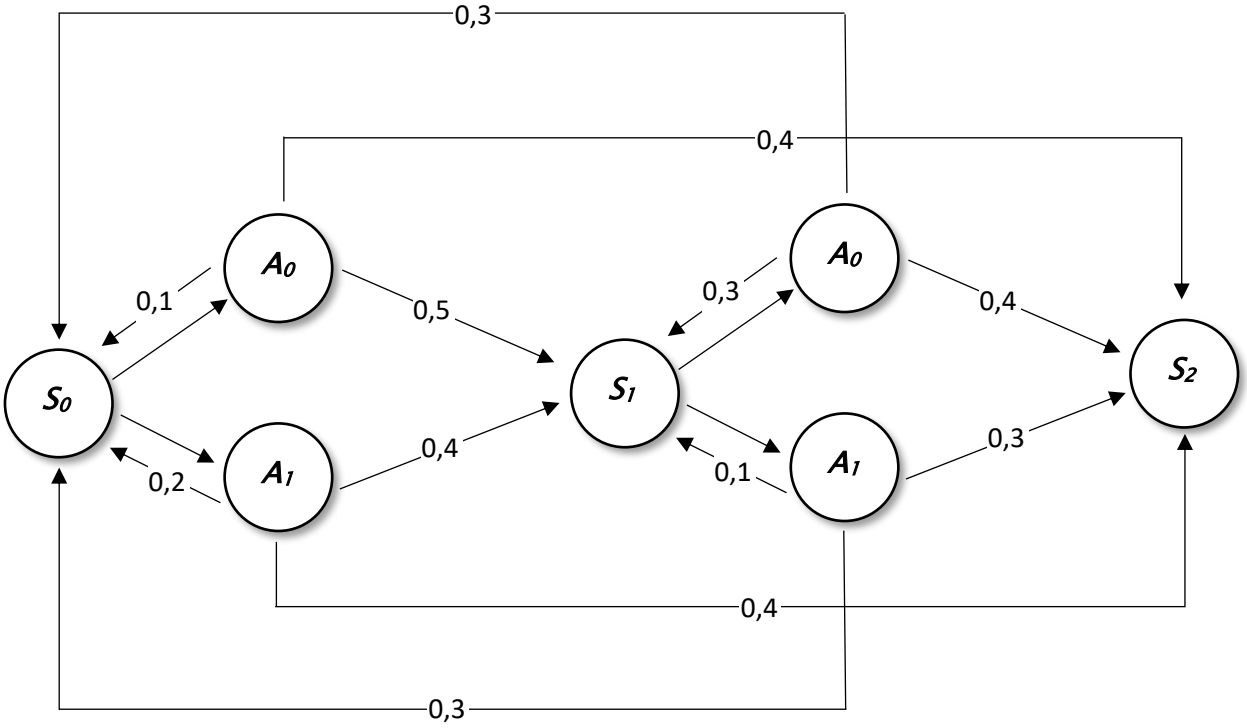

Figura 2: Árbol de decisión de Markov, probabilidades de transción entre estados. Figura de elaboración propia

$$V(s) = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$$

"El aprendizaje por refuerzo se presenta en muchas variedades diferentes", però, con el objeto del presente estudio, la elección del tipo de aprendizaje por refuerzo que debe usarse *"debe estar restringida a algoritmos que tengan memoria y, por lo tanto, puedan realizar un seguimiento de las acciones pasadas de los rivales (...), los agentes sin memoria no pueden castigar las desviaciones de los rivales y, por tanto, no permite coludir genuinamente".* Por ello, la opción idónea es el Q-learning. Según los mismos autores, *"hay varias razones para esta elección"*:

1) Los algoritmos de Q-learning maximizan *"el valor presente de un flujo de recompensas en problemas de elección repetida".*
2) Es un tipo de algoritmo extensamente utilizado por los informáticos.
3) Son algoritmos sencillos que no requieren de diseños complejos.
4) Este tipo de algoritmos han sido utilizados para obtener resultados muy prometedores *"logrando actuaciones sobre humanas, en tales tareas como jugar al antiguo juego de mesa Go (Silver et al. 2017), los videojuegos de Atari (Mnih et al. 2015) y, más recientemente, ajedrez (Silver et al. 2018)".*

Los algoritmos de Q-learning utilizan la misma ecuación de Bellman sin probabildiades de transición, con la peculiaridad de que el valor función *V(s)* que se obtiene es en realidad el valor *Q(s,a)*. Así pues, la ecuación resultante:

$$Q_{t+1}(s,a) = (1-\alpha) \cdot Q_t(s,a) + [R_t + \gamma \max_{a \in A} Q_t(s',a)]$$

Con esta ecuación, el algortimo más que calcular el valor función, actualiza dicho valor que tiene almacenado ante cierto estado-acción específico realizado. En este sentido, otra peculiaridad de este algoritmo es su forma de almacenar las decisiones tomadas y de elegir las mejores decisiones. El algoritmo almacena todas las decisiones tomadas en una tabla o matriz, llamada *Qtable*. En dicha tabla, hay tantas filas como estados posibles y tantas columnas como acciones. Lo que hace el algoritmo con la función Q es actualizar el valor de cada celda cada vez que se toma cierta acción A_t en cierto estado S_t. Entonces, cuando tiene que tomar una decisión, con una probabilidad de (*1 - α*) se dirige a la tabla Q, busca en qué estado actual se encuentra el agente y cual fue la decisión que maximizó la recompensa obtenida anteriormente en un estado anterior idéntico. Una vez elegida la acción, ya sea la que maximiza el valor función o una acción aleatoria, actualiza el valor de la tabla para ese estado-acción en concreto con la ecuación anteriormente descrita y, así, sucesivamente.

A este respecto y, como ya se ha adelantado, deben realizarse varias aclaraciones:

1) El factor de descuento γ debe ser menor que 1.

2) Es común establecer una disminución gradual de la tasa de aprendizaje o exploración conforme se realizan iteraciones. El objetivo de esto es que una vez el algoritmo haya explorado suficiente el entorno, sea capaz de explotarlo. Si bien no es recomendable que cuando las iteraciones tienden a infinito la tasa de exploración sea 0, es decir, $\lim_{t \to \infty} \alpha = 0$, si que es cierto que tiende cada vez a un valor más pequeño. En este sentido es importante que incluso en los casos donde el algoritmo haya explorado mucho el entorno, se deje

cierta tasa de exploración con el fin de que el algoritmo siga explorando otras opciones y pueda adaptarse a entornos cambiantes. Esto es especialmente importante en entornos dinámicos. Así, es como se conforma la política *epsilon-greedy*.

7. Estudio particular: colusión tácita algorítmica y duopolio de Bertrand

En el presente apartado, se introducirán las concepciones teóricas implícitas en la parte empírica del presente trabajo. Como podrá observarse a continuación, la parte empírica se basa en la simulación de un duopolio de Bertrand mediante el desarrollo de un algoritmo de Q-learning capaz fijar precios monitorizando las acciones que realizan cada una de las empresas. El objetivo de la parte empírica es mostrar como un algoritmo es capaz de aprender a coludir de forma autónoma, sin intervención humana y sin comunicación entre empresas.

Si bien un duopolio requiere que el número de empresas sea igual a 2, el algoritmo desarrollado soporta un número *n* de empresas. Sin embargo, con el fin de simular el modelo oligopólico de Bertrand y facilitar conclusiones, se ha establecido un número *n* de empresas igual a 2.

Para que el algoritmo tenga mayores probabilidades de aprender a coludir, se han establecido las condiciones de mercado idóneas relatadas en el apartado 4 del presente estudio. Es decir, funciones de coste y capacidades productivas simétricas sin limitaciones, productos completamente sustitutivos y homogéneos, un número elevado de interacciones, etc. La única condición que reúne el presente modelo que no puede concebirse como idónea para coludir, es la transparencia de mercado. Como el objetivo es que el algoritmo aprenda a coludir por sí mismo sin que exista comunicación o coordinación, es necesario establecer una transparencia de mercado mínima. Por ello, la única información pública disponible son los precios que interponen cada una de las empresas, el número de consumidores en el mercado en el que operan y su correspondiente renta disponible para gastar en las empresas.

Para simplificar el algoritmo, se han establecido ciertas asunciones:

1) Solo se ha establecido una función de coste, pero no una función de demanda. Si una empresa es capaz de establecer una función de demanda, quiere decir que ha conseguido modelizar por completo el entorno y, por tanto, sería capaz de establecer un precio sin necesidad de monitorizar los movimientos de los rivales. Como el objetivo es simular un mercado en el que las empresas tengan que aprender a fijar precios de forma autónoma y sin tener información completa sobre la demanda del mercado, es conveniente no proporcionar una función de demanda para simular un mercado incierto en el que las empresas tienen que aprender a establecer precios basándose en la información limitada que tienen sobre el mercado y sobre sus competidores. Esto hace que el problema sea más desafiante y realista, ya que las empresas tendrán que ajustar sus precios y aprender de forma autónoma a partir de los resultados obtenidos.

   La función de costes establecida es:

   $CT = 100 + q^2 - 10q$

   Siendo *q* la cantidad de producto.

2) Se han establecido seis únicas acciones posibles que puede realizar el algoritmo: subir el precio, mantener el precio o bajar el precio produciendo el 100% de cantidad total demandada

a cierto precio *p*, y subir, mantener y bajar el precio produciendo el 50% total demandada a cierto precio *p.* Se han establecido estas acciones por varios motivos:

a. Si solo se permitiera al algoritmo subir, mantener o bajar el precio, no se permite al algoritmo variar la cantidad. Si bien en el duopolio de Bertrand se compite en precios, en los tratos colusorios tanto tácitos como explícitos, se fija un precio colusorio y se reparte la cuota de mercado para obtener precios supra competitivos restringiendo la competencia. Si no se permite al algoritmo variar la cantidad, únicamente podrían contemplarse dos escenarios: el escenario donde siempre se colude (cantidad al 50%) o donde siempre se compite (100%).

b. Establecer la posibilidad de que el algoritmo decida producir al 50%[2] o al 100%, no es equivalente a darle la posibilidad de que coluda, si no que es equivalente a darle la posibilidad de que se desvíe. En el caso de que el algoritmo aprenda que la estrategia que maximiza beneficios es coludir a cierto precio *p* repartiendo la cuota de mercado a 1/número empresas, también se le ofrece la posibilidad de obtener beneficios supra colusorios desviándose de dicha política y estableciendo un precio inferior al colusorio, pero vendiendo al 100% de la cantidad demandada.

3) Hay tantos estados posibles (y filas en la tabla Q) como precios pueden establecerse. Para no incurrir en un número infinitesimal de estados posibles, se han limitado en varios sentidos:

a. Solo se tienen en consideración precios enteros, pues si se permiten precios con decimales, entre un precio *p* y un precio *p+1* habría 100 precios diferentes.

b. El primer precio y estado que puede establecerse es aquel cuyos beneficios son 0. Según Krugman (2020), el precio de beneficio nulo (B_0) puede obtenerse cuando los costes totales medios igualan los costes marginales. Así pues:

Costes totales medios: $CTM = \frac{CT}{q} = \frac{100+q^2-10q}{q}$

Costes marginales: $CM = \frac{\partial CT}{\partial q} = 2q - 10$

Cantidad beneficio nulo (q_0)[3]: $CTM = CM \rightarrow \frac{100+q^2-10q}{q} = 2q - 10 \rightarrow \boldsymbol{q_0 = 10}$

Precio beneficio nulo (p_0): $p \cdot q_0 - (100 + {q_0}^2 - 10q_0) = 0 \rightarrow \boldsymbol{p_0 = 10}$

[2] La cuota de mercado colusoria se establece al 50% debido a que se han establecido únicamente dos empresas en el mercado. Dicha cuota obedece $\pi_i = \frac{1}{n}$, siendo π_i la cuota de mercado resultante, y *n* el número de empresas del mercado. Dicho cálculo, como se ha desarrollado en el apartado 4.b.ii del presente estudio, obedece al hecho de que en el modelo simulado no existen limitaciones a la capacidad productiva de las empresas, es decir, son capaces de satisfacer por completo la demanda unilateralmente.

[3] La cantidad de beneficio nulo es 10 y -10. Sin embargo, como no puede haber cantidades negativas, solo se ha contemplado la cantidad 10 de beneficio nulo.

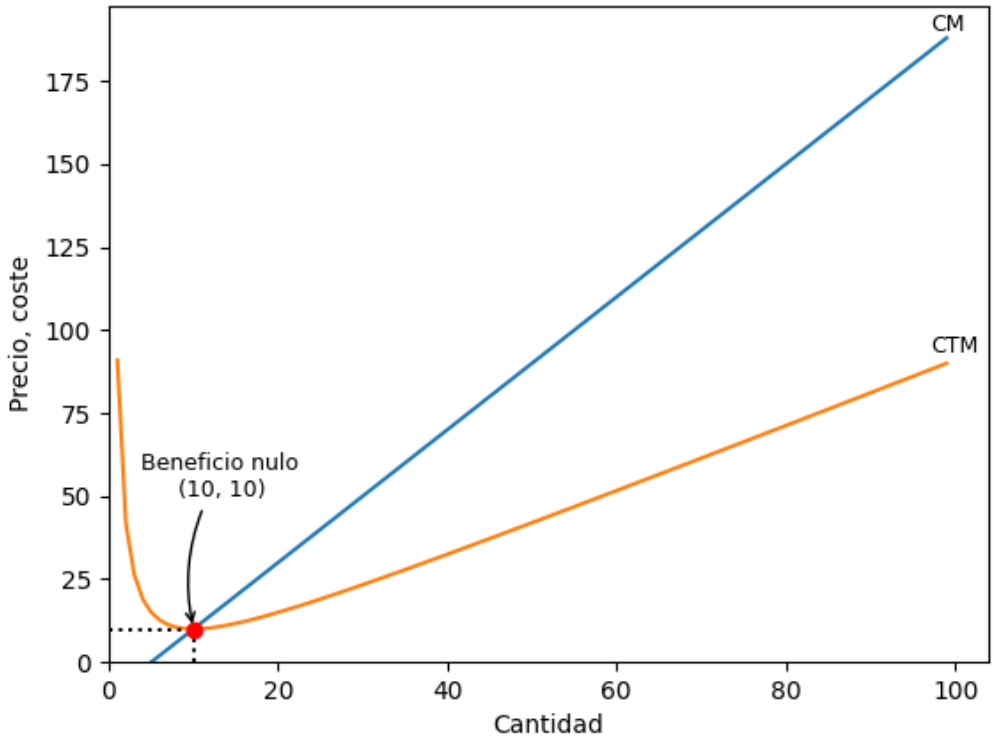

Figura 3: Gráfico de beneficio nulo del modelo simulado de Bertrand. Elaboración propia mediante Python.

c. Por otro lado, el precio máximo es el precio monopolístico que se obtiene al encontrar la tangente de todos los beneficios obtenidos respecto el precio *p*.

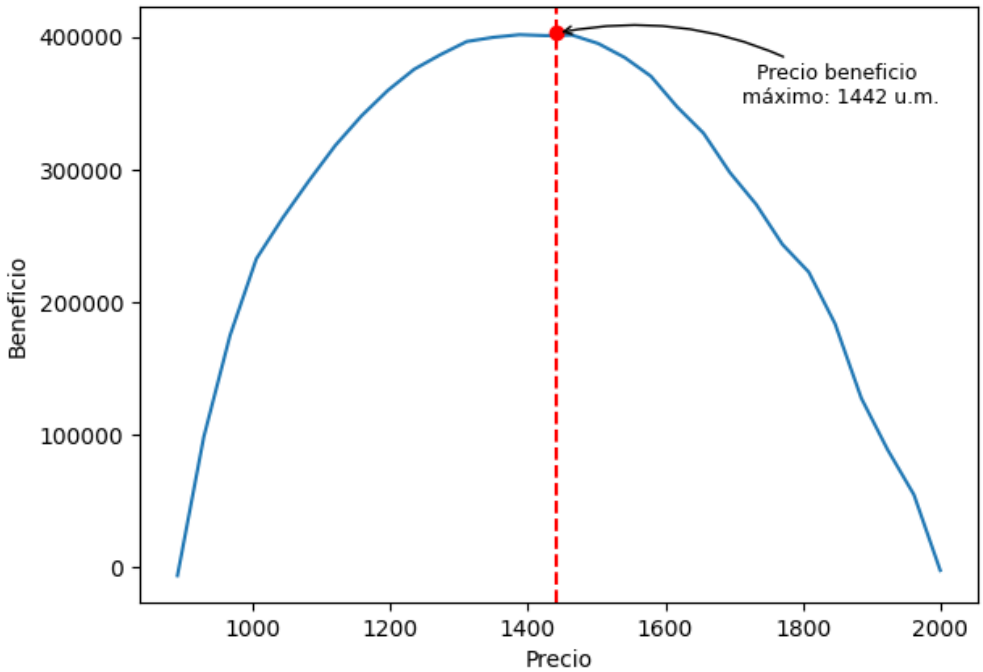

Figura 4: Gráfico de ejemplo sobre beneficio monopolístico máximo de modelo de Bertrand simulado. Elaboración propia mediante Python.

Para obtener los precios y sus respectivos beneficios, se establece un número *n* de consumidores cuya respectiva renta disponible se conforma por un rango de valores aleatorios entre *i* y *j* (en el código desarrollado, *i* = 5*00, j = 2000*). Debido a su aleatoriedad, el beneficio máximo del gráfico únicamente es un ejemplo, pues cada vez que se ejecuta el código los valores y las representaciones gráficas varían.

4) El número de consumidores y los respectivos niveles de renta no varían conforme se aumenta el número de episodios, una vez establecidos en el inicio de la ejecución del algoritmo se mantienen inmutables en el tiempo. Sin embargo, la generación de rentas es completamente aleatoria otorgando un nivel de renta a cada consumidor entre 200 y 5000 unidades monetarias.

5) Cada consumidor no tiene preferencias ni valora la utilidad del producto respecto su precio, se asume que si cierto consumidor tiene la capacidad adquisitiva suficiente para comprar, como mínimo, una unidad entera de producto éste la adquiere automáticamente.

MARCO EMPÍRICO

8. Metodología

El objetivo principal de la parte empírica es simular computacionalmente un duopolio de Bertrand mediante un algoritmo de *Q-learning*, con el fin de mostrar como un algoritmo informático fijador de precios es capaz de aprender a coludir por si mismo conforme aumenta el número de iteraciones o episodios, sin mediar ningún tipo de comunicación ni trato explícito entre empresas.

Existen varios trabajos académicos que han desarrollado y estudiado simulaciones computacionales parecidas para demostrar que los algoritmos son capaces de aprender a coludir autónomamente, sin embargo, *Artificial Intelligence, Algorithmic Pricing, and Collusion* de Emilio Calvano et al. (2020) ha asentado gran parte de las bases de este tipo de estudios. En dicho artículo, se utilizó un lenguaje computacional llamado FORTRAN que es utilizado ampliamente para aplicaciones científicas y de ingeniería capaz de realizar cálculos numéricos complejos y de alto nivel. Para el presente estudio, se ha optado por diseñar el algoritmo mediante Python, otro lenguaje computacional que, si bien, no está enfocado específicamente para el desarrollo de aplicaciones científicas o de ingeniería, resulta idóneo para el diseño del presente algoritmo por varios motivos:

1) Es intuitivo, fácil de usar y tiene una amplia gama de bibliotecas y módulos predefinidos que facilitan la programación.
2) Algunos estudios también han utilizado este lenguaje para diseñar algoritmos capaces de fijar precios en una simulación de duopolio de Bertrand y mostrar sus acciones colusorias.
3) Por el conocimiento previo de este tipo de lenguaje de programación.

Para "medir" si el algoritmo aprende a coludir, se utilizarán representaciones visuales basadas en los resultados obtenidos por Lepore (2021), pero con ciertos matices y modificaciones. Lepore, de entre los resultados que obtuvo en su estudio, son de especial interés los gráficos donde pueden visualizarse la evolución de los precios conforme aumentan el número de iteraciones. Dichos gráficos se conforman de varias líneas: evolución de precios de la empresa 0 y 1, promedio móvil[4] de la empresa 0 y 1, precio de equilibrio de Nash y precio monopolístico.

Respecto estos gráficos, se han adoptado todas las métricas, pero se han añadido otras. En particular, se han establecido "áreas de convergencia algorítmica". La convergencia algorítmica, determina la "estabilidad óptima" de las decisiones tomadas por el algoritmo. Para el presente caso, se ha medido dicha estabilidad si la diferencia absoluta entre cierto valor Q y su actualización varían menos de 50 puntos. Estas áreas de convergencia muestran que, si el algoritmo ha tomado la decisión de coludir, dicha decisión se ha tomado al ser la solución óptima al problema de maximización de beneficios.

Por otro lado, se ha establecido una línea punteada vertical que indica el momento en que la tasa de aprendizaje alcanza su valor mínimo. Como se ha adelantado, es común en los algoritmos de aprendizaje por refuerzo que la tasa de aprendizaje varie conforme pasa el tiempo para que dicho algoritmo pase de "explorar" el entorno a "explotarlo". Al establecer un número de iteraciones igual a 300.000, se ha establecido una tasa de aprendizaje inicial de 0,99 que disminuye en

[4] Los promedios móviles es una métrica estadística para suavizar los datos obtenidos. En el presente caso, al haber un número de iteraciones tan elevadas que pueden llegar al millón de episodios, no pueden distinguirse bien los diferentes precios debido a que las iteraciones están demasiado juntas. Por ello, los promedios móviles son necesarios para visualizar correctamente la evolución.

0,000005 puntos conforme se aumenta en 1 el número de iteraciones. Sin embargo, en el momento en que dicha tasa de aprendizaje obtiene un valor inferior a 0,01, la tasa de aprendizaje deja de disminuir. De este modo, si bien el algoritmo al final de la ejecución prácticamente solo "explota" el entorno, también da cabida a cierta exploración para averiguar si otras opciones son más óptimas por si el entorno resultase ser dinámico y cambiante. Así pues, la línea vertical siempre se sitúa en el episodio 198.000 (0,99/0,000005).

Si bien estos gráficos son útiles para visualizar la evolución de precios, no miden la colusión de las acciones del algoritmo. Para ello, se ha establecido que el algoritmo genere otros dos tipos de gráficos a tal efecto: un gráfico donde se mide la evolución de beneficios conforme aumentan el número de iteraciones, y otro gráfico donde se mide explícitamente el nivel colusorio de las acciones del algoritmo conforme aumentan el número de iteraciones.

Para calcular el nivel colusorio de las acciones propiciadas por el algoritmo, se ha utilizado la formula proporcionada por Calvano et al. (2020):

$$col = \frac{B_p - B_N}{B_M - B_N}$$

Siendo B_p el beneficio promedio, B_N el beneficio obtenido al precio de Nash y B_M el beneficio monopolístico. El beneficio promedio, es representado por una línea horizontal discontinua azul en el gráfico de beneficios. Sin embargo, para cada episodio se debe realizar el sumatorio de las tasas colusorias de cada agente para determinar el nivel colusorio del mercado. Si no se realiza el sumatorio, se obtienen frecuentemente tasas dispares. Por otro lado, en el gráfico colusorio, se muestran dos líneas horizontales discontinuas en *y = 0* e *y = 1*. Cuando la línea colusoria se acerca más 1, mayor nivel colusorio muestra la decisión tomada por el algoritmo, por otro lado, cuanto más se acerca a 0 mayor competitividad muestra la decisión tomada.

9. Resultados e interpretación

Tras varias ejecuciones del algoritmo, se han obtenido dos tipos de resultados que concuerdan con los obtenidos por Lepore (2021): colusión de equilibrio cooperativo y colusión de equilibrio dominante.

La colusión de equilibrio cooperativo se da cuando ambas empresas o agentes establecen el mismo precio o uno similar adoptando y experimentando cambios conjuntos tanto en los precios como en los beneficios obtenidos. Por otro lado, la colusión de equilibrio dominante se da cuando ambas empresas o agentes establecen un precio notoriamente diferente al de la otra empresa. De esta forma, el agente dominante es aquel cuyo precio es más elevado que el de la empresa no dominante. En este escenario también se colude, pero de distinta forma, pues mientras el agente dominante obtiene un mayor precio, el agente no dominante suele vender mayor cantidad de producto, por lo que suelen compensarse las diferencias hasta alcanzar un resultado colusorio satisfactorio para ambas (Lepore, 2021).

9.1. Resultados obtenidos

a. Colusión de equilibrio cooperativo

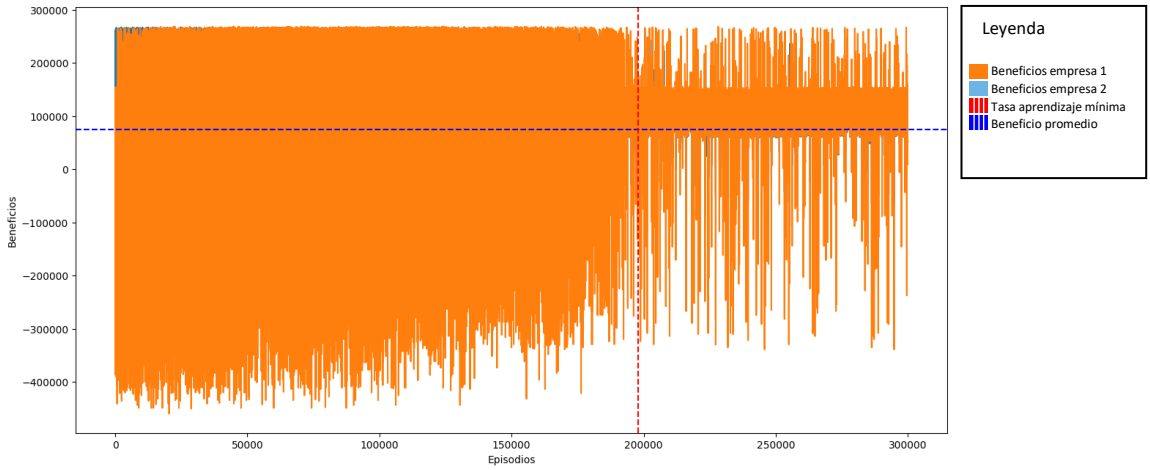

Figura 5: Evolución de beneficios en colusión cooperativa. Elaboración propia mediante Python

Figura 6: Evolución de precios en colusión cooperativa. Elaboración propia mediante Python

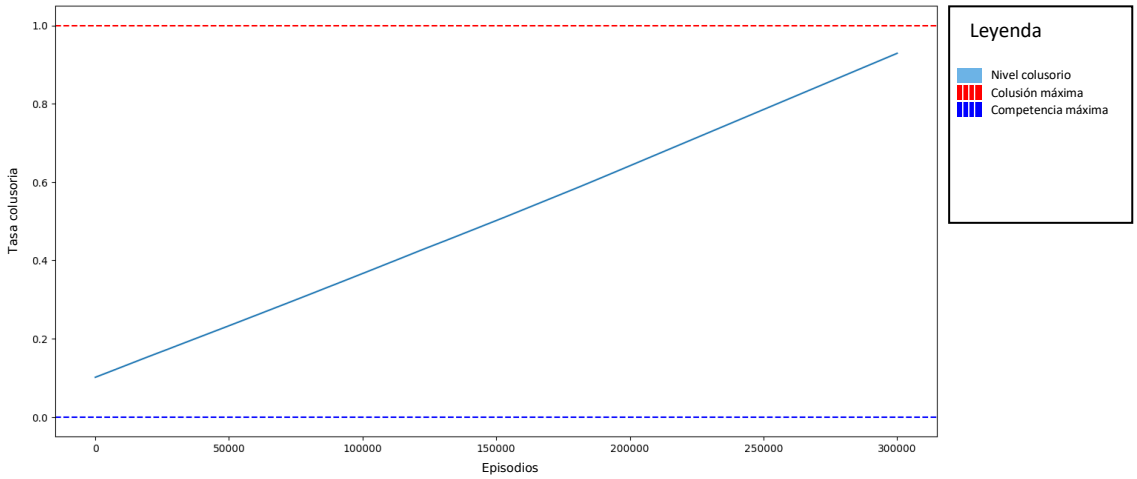

Figura 7: Evolución de colusoria cooperativa. Elaboración propia mediante Python

b. Colusión de equilibrio dominante

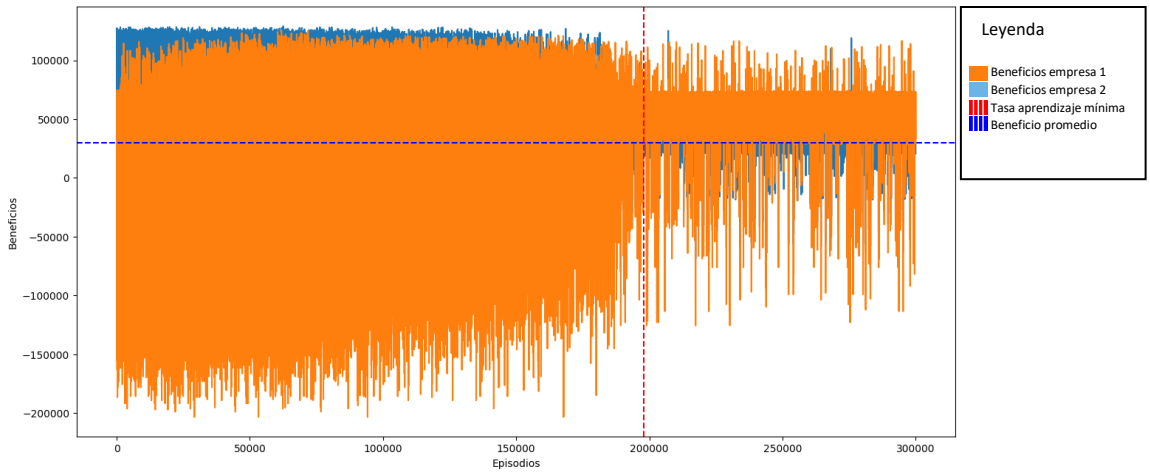

Figura 8: Evolución de beneficios en colusión dominante. Elaboración propia mediante Python

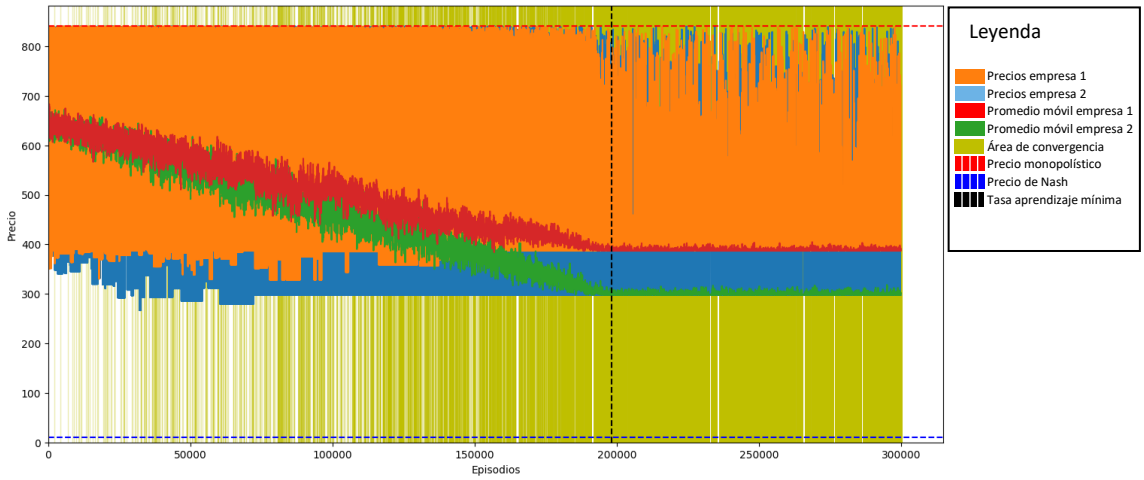

Figura 9: Evolución de precios en colusión dominante. Elaboración propia mediante Python

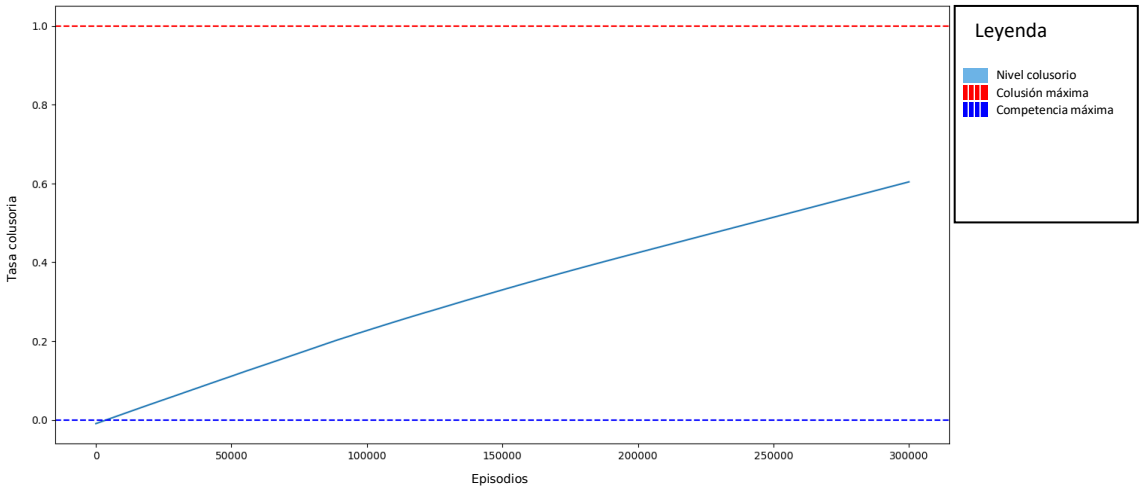

Figura 10: Evolución colusoria dominante. Elaboración propia mediante Python

9.2. Interpretación

Como puede observarse de los gráficos generados, el algoritmo comienza a explotar el entorno en el momento en que la tasa de aprendizaje alcanza su valor mínimo. A partir de ese momento, los beneficios suelen situarse a niveles superiores al nivel de beneficios promedio, donde los beneficios obtenidos por debajo de ese nivel se deben a que el algoritmo sigue explorando el entorno con una probabilidad de 1 sobre 100.

Debido al factor aleatorio del algoritmo respecto las rentas disponibles de los consumidores, aparecen diferentes tipos de colusión, la cooperativa y la dominante. Puede observarse como la colusión dominante ofrece un nivel colusorio inferior debido a que los beneficios obtenidos por el agente dominado suelen ser inferiores a los del dominante.

Las tasas colusorias se acercan más al nivel competitivo al inicio de la ejecución algorítmica que tras su finalización, observando la importancia de la influencia del número de episodios sobre la posibilidad de que surja la colusión tácita. En este sentido, no es una coincidencia que a partir del momento en que la tasa de aprendizaje alcanza su valor mínimo, las áreas de convergencia suelen darse en mayor medida y de manera más estable en el tiempo. Esto se debe a la actualización de los valores Q. Cuando por primera vez se explora una acción en cierto estado predeterminado, el valor Q pasa de 0 a un valor determinado, por lo que su diferencia será muy superior a si se actualiza un valor Q que ya había sido explorado con anterioridad. En el momento en que la tasa de aprendizaje alcanza su valor mínimo, es muy probable que se vuelva a elegir la misma acción anterior en cierto estado determinado que maximizó la recompensa obtenida, por ello la diferencia entre el valor Q anterior y su actualización va decayendo.

Tampoco es una casualidad que el nivel de precios en los gráficos de evolución de precios se sitúe en un valor medio entre el precio de Nash y el monopolístico. Desde el plano teórico, y contradiciendo lo que se estipula en el modelo oligopólico de Bertrand, por paralelismo consciente ambas firmas deberían producir al 50% y establecer el precio monopolístico, es ahí donde se alcanzan los mayores beneficios. Sin embargo, esto solo es posible de mantener en el tiempo bajo un trato explícito colusorio y, aun así, existen incentivos dependiendo del factor de descuento a que dicho trato sea roto por una empresa divergente.

Si se observan con detalle los gráficos, puede observarse como dicha opción ha sido explorada por el algoritmo, sin embargo, no se ha mantenido en el tiempo. Esto es gracias al factor de descuento establecido en cada empresa y a que el algoritmo también contempla la posibilidad de desviarse de la tendencia colusoria, es decir, de seguir con lo estipulado por el modelo de Bertrand: establecer un precio inferior al de la empresa contraria y acaparar toda la cuota de mercado, es decir, producir al 100%. Este es el motivo de por qué la mayoría de las decisiones se concentran en un nivel de precios medio entre el precio de Nash y el precio monopolístico, por ser el precio que compensa las desviaciones y por ser el precio que acumula mayor beneficio o recompensa a largo plazo.

9.3. Influencia del número de competidores al nivel colusorio de la toma de decisiones

Tal y como se expone en el marco teórico, el número de competidores o agentes económicos en un mercado es determinante para facilitar la aparición de colusión tácita. Si ejecutamos el algoritmo desarrollado, con un número superior a dos empresas ($n > 2$), se obtienen los siguientes resultados:

n = 3

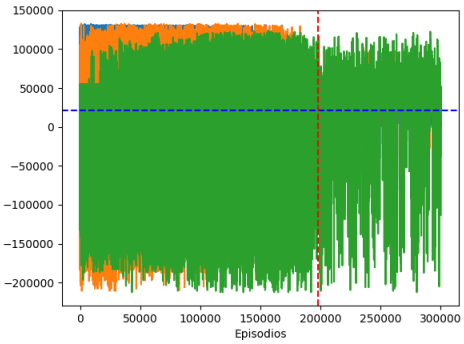

Figura 11: Evolución de beneficios en colusión con n=3. Elaboración propia mediante Python

n = 4

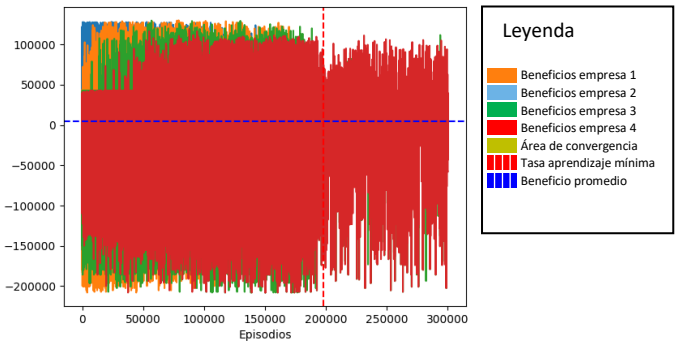

Figura 12: Evolución de beneficios en colusión con n=4. Elaboración propia mediante Python

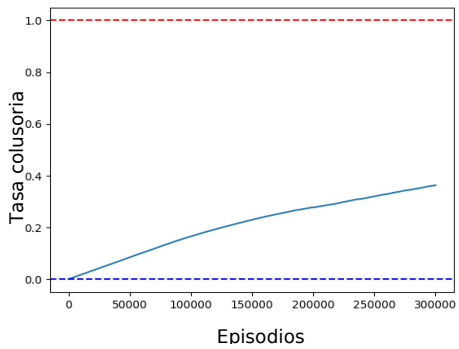

Figura 13: Evolución colusoria con n=3. Elaboración propia mediante Python

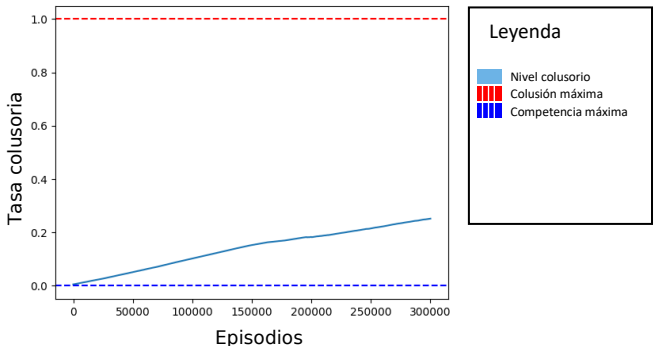

Figura 14: Evolución colusoria con n=4. Elaboración propia mediante Python

n = 3

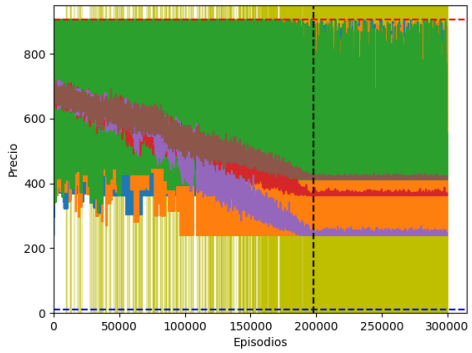

Figura 15: Evolución de precios en colusión con n=3. Elaboración propia mediante Python

n = 4

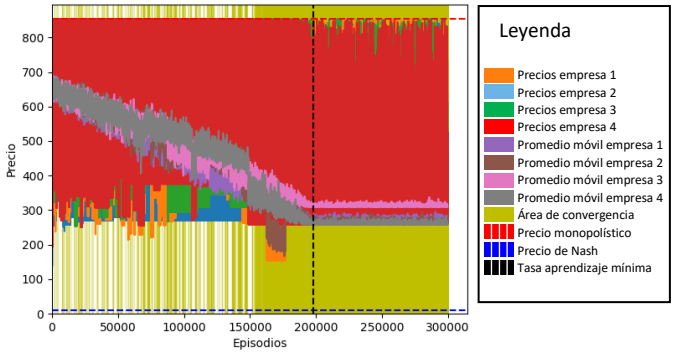

Figura 16: Evolución de precios en colusión con n=4. Elaboración propia mediante Python

Como puede observarse, a medida que se añaden nuevos agentes económicos al mercado la tasa colusoria decrece, resultados que también coinciden en su límite inferior con los obtenidos por Lepore (2021)

10. Conclusiones

El objetivo principal del presente estudio consistía en desarrollar un algoritmo de Q-learning capaz de simular un duopolio de Bertrand y de fijar precios, con el fin de monitorizar la evolución de los precios para detectar si un algoritmo informático es capaz de aprender a coludir de forma autónoma, sin intervención humana y sin comunicación entre empresas.

Para hacerlo posible, se ha desarrollado un algoritmo de juegos de Bertrand repetidos, y se han recopilado todos los precios establecidos y beneficios obtenidos en diferentes gráficos, incluyendo un gráfico donde se observa la tendencia colusoria del algoritmo en torno a sus decisiones tomadas conforme aumentan el número de iteraciones.

Los resultados obtenidos tienen ciertas limitaciones. Tal y como se ha descrito en el último apartado de la parte teórica, se han establecido ciertas asunciones que limitan el algoritmo con el fin de simplificar sus resultados, y además se ha simulado un mercado con las condiciones óptimas e idóneas para que aflorara la colusión tácita. Por tanto, los resultados obtenidos no pueden extrapolarse a la generalidad de los casos, pero si sirve para mostrar la capacidad de ciertos algoritmos, sobre todo los de aprendizaje por refuerzo, de aprender a coludir de forma autónoma gracias a su capacidad de obtener datos con la interacción del entorno y de elegir las decisiones optimas maximizando las recompensas obtenidas a largo plazo.

Tras lo expuesto, puede concluirse que el objetivo principal del presente estudio ha sido cumplido y satisfecho exitosamente. Se han abordado multitud de implicaciones teóricas sobre la colusión tácita, y se han llevado al terreno empírico desarrollando un algoritmo que muestra una tendencia colusoria de aproximadamente el 60% u 80% en las últimas iteraciones, lo que ha ayudado en sobremedida al entendimiento de la colusión tácita algorítmica y sus implicaciones. Las mencionadas tasas colusorias coinciden con los resultados obtenidos por Lepore, incluso en aquellas simulaciones con más de 2 agentes.

Asimismo, se ha desafiado el modelo oligopólico de Bertrand (tal y cómo sugiere el título del presente texto) al observar cómo, a medida que aumentan el número de episodios, los precios no

tienden a establecerse a un nivel de beneficio cero (equilibrio de Nash) si no que el algoritmo busca maximizar el beneficio incluso realizando practicas colusorias.

11. Futuras líneas de investigación

Debido al límite de tiempo y de capacidad de procesamiento de información de los dispositivos computacionales a disposición, se han realizado simulaciones con un número bajo de iteraciones a comparación con otros estudios. Por ello, tras los resultados obtenidos, sería recomendable para futuras investigaciones realizar simulaciones con mayor número de iteraciones, empleando mayor cantidad de tiempo o desarrollando un algoritmo más eficiente que no consuma tanto tiempo de ejecución.

Otras líneas de investigación futuras podrían considerar la posibilidad de simular mercados con otras condiciones de mercado no óptimas para la colusión, como funciones de producción asimétricas, capacidades de producción limitadas, diferentes productos, etc., con el fin de mostrar cómo afectan dichas condiciones a la facilidad para coludir. También sería interesante abordarlo desde un punto de vista anticompetitivo, estableciendo formas de medir la colusión tácita algorítmica en diferentes mercados, o también desde un punto de vista jurídico, sobre medidas preventivas y regulatorias sobre los algoritmos fijadores de precios para evitar la colusión en los mercados.

En definitiva, con los resultados obtenidos, puede concluirse que la colusión tácita algorítmica es una realidad, que los algoritmos informáticos son capaces de aprender a coludir autónomamente y que abre un paradigma completamente nuevo sobre posibles y futuras nuevas políticas anticompetitivas. La capacidad de los algoritmos de aprendizaje por refuerzo a aprender por su cuenta a coludir, los hacen muy atractivos para las empresas ya que las blinda de posibles consecuencias legales si no existe intención colusoria por parte de ningún implicado, sobre todo si se tiene en cuenta la gran dificultad de detectar la colusión tácita algorítmica de Digital Eye o Predictable Agent.

12. Bibliografía

Athey, S., Bagwell, K., & Sanchirico, C. (2004). Collusion and Price Rigidity. *The Review of Economic Studies, 71*(2), 317-349.

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G. C., Steinhardt, J., Flynn, C., hÉigeartaigh, S. Ó., Beard, S. J., Belfield, H., Farquhar, S., … Amodei, D. (2018). *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation* [Report]. https://doi.org/10.17863/CAM.22520

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review, 110*(10), 3267-3297. https://doi.org/10.1257/aer.20190623

Caves, R. E., & Williamson, P. J. (1985). What is Product Differentiation, Really? *The Journal of Industrial Economics, 34*(2), 113. https://doi.org/10.2307/2098677

Dranove, D. S., Besanko, D. A., Shanley, M., & Schaefer, M. (2017). Economics of Strategy. (7th ed.) John Wiley & Sons, Inc.

Ezrachi, A., & Stucke, M. E. (2015). Artificial Intelligence & Collusion: When Computers Inhibit Competition. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2591874

Garrod, L., & Olczak, M. (2018). Explicit vs tacit collusion: The effects of firm numbers and asymmetries. *International Journal of Industrial Organization, 56,* 1-25. https://doi.org/10.1016/j.ijindorg.2017.10.006

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems / Aurélien Géron* (Second edition). O'Reily.

Gibbons, R. (2011). *Un primer curso de teoría de juegos / Robert Gibbons*. Antoni Bosch.

Ivaldi, M., Jullien, B., Rey, P., Seabright, P., & Tirole, J. (2003). *The Economics of Tacit Collusion*.

Joseph E. Harrington, Jr., 2012. "A Theory of Tacit Collusion," Economics Working Paper Archive 588, The Johns Hopkins University,Department of Economics.

Krugman, P. (2020). *Fundamentos de Economía / Paul Krugman, Robin Wells, Kathryn Graddy ; versión española de Jimena García-Pardo y Alonso de Ojeda.* (tercera edición.). Editorial Reverté.

Lepore, Nicolas. 2021. AI Pricing Collusion: Multi-Agent Reinforcement Learning Algorithms in Bertrand Competition. Bachelor's thesis, Harvard College.

Madhavan, A. (1996). Security Prices and Market Transparency. *Journal of Financial Intermediation*, *5*(3), 255-283. https://doi.org/10.1006/jfin.1996.0015

Mitchell, T.M. and Tom, M. (1997) Machine Learning. McGraw-Hill, New York.

Motta, M. (2004). Collusion and Horizontal Agreements. In *Competition Policy: Theory and Practice* (pp. 137-230). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511804038.005

Murphy, K. P. (2012). *Machine learning a probabilistic perspective / Kevin P. Murphy.* MIT Press.

OECD (2017), Algorithms and Collusion: Competition Policy in the Digital Age www.oecd.org/competition/algorithms-collusion-competition-policy-in-the-digital-age.htm

Puterman, M. L. (2005). *Markov decision processes discrete stochastic dynamic programming / Martin L. Puterman.* John Wiley & Sons.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210-229. https://doi.org/10.1147/rd.33.0210

Sánchez, G. (2023). *El Plan General Algorítmico* [Trabajo de final de grado no publicado]. Universidad Autónoma de Barcelona

Schultz, C. (2004). Market transparency and product differentiation. *Economics Letters*, *83*(2), 173-178. https://doi.org/10.1016/j.econlet.2003.11.004

Stigler, G. J. (1964). A Theory of Oligopoly. *Journal of Political Economy*, *72*(1), 44-61.

Sutton, R. S. (1998). *Reinforcement learning an introduction / Richard S. Sutton and Andrew G. Barto.* MIT Press.

13. Anexos

a. Código del algoritmo

```
from sympy import symbols, Eq, solve
import sympy as sp
from scipy import optimize, arange
from numpy import array
import random
import matplotlib.pyplot as plt
import math
import numpy as np
import pylab as plt
from operator import itemgetter
from time import time
from scipy.interpolate import make_interp_spline

start2 = time()

q = sp.Symbol('q')
costes_totales = 100 + q**2 -10*q
costes_marginales = sp.diff(costes_totales, q)
costes_totales_medios = (costes_totales)/q

ecuacion_equilibrio = costes_marginales - costes_totales_medios

solucion_cantidad_equilibrio = solve(ecuacion_equilibrio)
cantidad_minima = solucion_cantidad_equilibrio[1]

q = cantidad_minima
costes_totales = 100 + q**2 -10*q

p = sp.Symbol('p')
ecuacion_precio = p*cantidad_minima - costes_totales
precio_minimo = solve(ecuacion_precio)[0]

costes_totales_medios_por_cantidad =[]
costes_marginales_por_cantidad = []
cantidades = []

q = sp.Symbol('q')
costes_totales = 100 + q**2 -10*q
coste_marginal = sp.diff(costes_totales, q)

i = 100

coste_marginal = coste_marginal.subs(q, i)
```

```
for i in range(100):

    q = sp.Symbol('q')
    costes_totales = 100 + q**2 -10*q
    coste_marginal = sp.diff(costes_totales, q)

    q = i
    costes_totales = 100 + q**2 -10*q

    if i == 0:
        pass
    else:
        costes_totales_medios = (costes_totales)/q
        q = sp.Symbol('q')
        coste_marginal = coste_marginal.subs(q, i)
        cantidades.append(i)
        costes_marginales_por_cantidad.append(coste_marginal)
        costes_totales_medios_por_cantidad.append(costes_totales_medios)

plt.plot(cantidades, costes_marginales_por_cantidad)
plt.plot(cantidades, costes_totales_medios_por_cantidad)
plt.plot(cantidad_minima, precio_minimo, "ro")
plt.show()

class consumidor:

    def __init__(self, numero, renta):
        self.numero = numero
        self.renta = renta

    def __str__(self):
        return '\nConsumidor: ' + str(self.numero) + '\nRenta: ' +
str(self.renta)

numero_consumidores = 200

lista_consumidor = []

for i in range(numero_consumidores):
    rentax = random.randint(500,2000)
    consumidorx = consumidor(i+1, rentax)
    lista_consumidor.append(rentax)

lista_consumidor_ordenada_por_renta = sorted(lista_consumidor)

beneficios_obtenidos_por_precio = []
precio_maximo = max(lista_consumidor_ordenada_por_renta)

for j in range(precio_minimo, precio_maximo):
```

```
    cantidad_total_vendida = 0
    for i in range(numero_consumidores):
        if lista_consumidor_ordenada_por_renta[i] < j:
            pass
        else:
            cantidad_vendida =
math.floor(lista_consumidor_ordenada_por_renta[i]/j)

            cantidad_total_vendida = cantidad_total_vendida +
cantidad_vendida

            if i == len(lista_consumidor_ordenada_por_renta)-1:
                beneficios_obtenidos_por_precio.append([j,
cantidad_total_vendida])

beneficio_nash = beneficios_obtenidos_por_precio[0][1]

costes_totales = 100 + q**2 -10*q
costes_totales_medios = costes_totales/q

lista_beneficios_posibles = []
lista_beneficios_posibles_segun_precio_cantidad = []
lista_beneficios_cantidad = []
lista_beneficios_precio = []

for i in range(len(beneficios_obtenidos_por_precio)):
    p = beneficios_obtenidos_por_precio[i][0]
    q = beneficios_obtenidos_por_precio[i][1]
    ingresos = p*q
    costes = costes_totales = 100 + q**2 -10*q
    beneficios = ingresos - costes
    if beneficios >= 0:
        lista_beneficios_posibles.append(beneficios)
        lista_beneficios_cantidad.append(q)
        lista_beneficios_precio.append(p)
        lista_beneficios_posibles_segun_precio_cantidad.append([p, q,
beneficios])

beneficio_maximo = max(lista_beneficios_posibles)

for i in range(len(lista_beneficios_posibles_segun_precio_cantidad)):
    if lista_beneficios_posibles_segun_precio_cantidad[i][2] ==
beneficio_maximo:
        cantidad_beneficio_maximo =
lista_beneficios_posibles_segun_precio_cantidad[i][1]
        precio_beneficio_maximo =
lista_beneficios_posibles_segun_precio_cantidad[i][0]
    else:
        pass
```

```
X_Y_Spline = make_interp_spline(lista_beneficios_precio,
lista_beneficios_posibles)
X_ = np.linspace(min(lista_consumidor_ordenada_por_renta),
max(lista_consumidor_ordenada_por_renta), 30)
Y_ = X_Y_Spline(X_)

#plt.plot(lista_beneficios_cantidad, lista_beneficios_posibles)
#plt.plot(lista_beneficios_precio, lista_beneficios_posibles)
#plt.plot(lista_beneficios_cantidad, lista_beneficios_precio)

plt.plot(X_, Y_)

plt.plot(precio_beneficio_maximo, beneficio_maximo, "ro")
plt.axvline(x=precio_beneficio_maximo, color='red', linestyle='--')

plt.xlabel('Precio')
plt.ylabel('Beneficio')

plt.show()

lista_estados = []

for i in range(len(lista_beneficios_posibles_segun_precio_cantidad)):
    if lista_beneficios_posibles_segun_precio_cantidad[i][0] <=
precio_beneficio_maximo:
        lista_estados.append(lista_beneficios_posibles_segun_precio_canti
dad[i][0])

episodios = 300000
empresas = 2

lista_consumidor_episodio = []

q_tables = []
for i in range(empresas):
    matriz = np.zeros((len(lista_estados), 6))
    q_tables.append([matriz])

lista_beneficios=[]

lista_acciones_numericas = []

lista_numero_episodios = []
lista_evolucion_precios_episodio = []

lista_cuotas = [1, (1/empresas)]
lista_episodios_convergentes = []
accion = 0
```

```
for i in range(episodios):
    tasa_aprendizaje = 0.99 - 0.000005*i
    if tasa_aprendizaje <= 0.01:
        tasa_aprendizaje = 0.01
    factor_descuento = 0.95
    lista_numero_episodios.append(i)
    lista_consumidor_episodio =
lista_consumidor_ordenada_por_renta.copy()
    lista_acciones = []
    lista_estados_empresa = []
    lista_precios_por_empresa = []

    for j in range(empresas):

        if i == 0:
            estado_anterior = random.choice(lista_estados)
            estado_actual = random.choice(lista_estados)
            cuota = random.choice(lista_cuotas)
            if estado_anterior < estado_actual and cuota== 1: #aumentar
con cantidad 100%
                accion = 0
            if estado_anterior > estado_actual and cuota== 1: #disminuir
con cantidad 100%
                accion = 1
            if estado_anterior == estado_actual and cuota== 1: #mantener
con cantidad 100%
                accion = 2
            if estado_anterior < estado_actual and cuota== (1/empresas):
#aumentar con cantidad (1/empresas)
                accion = 3
            if estado_anterior > estado_actual and cuota== (1/empresas):
#disminuir con cantidad (1/empresas)
                accion = 4
            if estado_anterior == estado_actual and cuota== (1/empresas):
#mantener con cantidad (1/empresas)
                accion = 5
        else:
            max_q_greedy = np.max(q_tables[j][0])

            if random.random() <= tasa_aprendizaje:

                estado_anterior = estado_actual
                estado_actual = random.choice(lista_estados)

                cuota = random.choice(lista_cuotas)
                if estado_anterior < estado_actual and cuota== 1:
#aumentar con cantidad 100%
                    accion = 0
```

```
                if estado_anterior > estado_actual and cuota== 1:
#disminuir con cantidad 100%
                    accion = 1
                if estado_anterior == estado_actual and cuota== 1:
#mantener con cantidad 100%
                    accion = 2
                if estado_anterior < estado_actual and cuota==
(1/empresas): #aumentar con cantidad (1/empresas)
                    accion = 3
                if estado_anterior > estado_actual and cuota==
(1/empresas): #disminuir con cantidad (1/empresas)
                    accion = 4
                if estado_anterior == estado_actual and cuota==
(1/empresas): #mantener con cantidad (1/empresas)
                    accion = 5
            else:

                for x in range(len(lista_estados)):
                    for h in range(6):
                        if q_tables[j][0][x, h] == max_q_greedy:
                            estado_anterior = estado_actual
                            estado_actual = x
                            if estado_anterior < estado_actual: #aumentar
precio con cantidad 100%
                                accion = 0
                            if estado_anterior > estado_actual:
#disminuir precio con cantidad 100%
                                accion = 1
                            if estado_anterior == estado_actual:
#mantener precio con cantidad 100%
                                accion = 2
                            if estado_anterior < estado_actual: #aumentar
precio con cantidad (1/empresas)
                                accion = 3
                            if estado_anterior > estado_actual:
#disminuir precio con cantidad (1/empresas)
                                accion = 4
                            if estado_anterior == estado_actual:
#mantener precio con cantidad (1/empresas)
                                accion = 5

        lista_acciones.append(accion)
        lista_estados_empresa.append([estado_anterior, estado_actual])
        lista_precios_por_empresa.append([j, estado_actual])
        lista_evolucion_precios_episodio.append([j, estado_actual])

        lista_precios_ordenada_por_precio =
sorted(lista_precios_por_empresa, key=lambda x: x[1])
```

```
    if accion == 0 or accion == 1 or accion == 2:
        for j in range(empresas):
            for c in
range(len(lista_beneficios_posibles_segun_precio_cantidad)):
                if lista_precios_ordenada_por_precio[j][1] ==
lista_beneficios_posibles_segun_precio_cantidad[c][0]:
                    cantidad =
math.floor(lista_beneficios_posibles_segun_precio_cantidad[c][1])
                else:
                    pass
            lista_precios_ordenada_por_precio[j].append(cantidad)
    else:
        for j in range(empresas):
            for c in
range(len(lista_beneficios_posibles_segun_precio_cantidad)):
                if lista_precios_ordenada_por_precio[j][1] ==
lista_beneficios_posibles_segun_precio_cantidad[c][0]:
                    cantidad =
math.floor(lista_beneficios_posibles_segun_precio_cantidad[c][1]/(empresa
s))
                else:
                    pass

            lista_precios_ordenada_por_precio[j].append(cantidad)

    for j in range(empresas):
        lista_beneficios.append([])

        cantidad_vendida_total = 0
        cantidad_ofertada = lista_precios_ordenada_por_precio[j][2]

        for a in range(len(lista_consumidor_episodio)):

            if cantidad_vendida_total >=
lista_precios_ordenada_por_precio[j][2]:
                pass
            else:

                precio_ofertado = lista_precios_ordenada_por_precio[j][1]

                if precio_ofertado == 0:
                    pass
                else:

                    cantidad_consumidor_dispuesto_compra =
math.floor(lista_consumidor_episodio[a]/precio_ofertado)

                    if cantidad_ofertada >=
cantidad_consumidor_dispuesto_compra:
```

```
                        cantidad_comprada =
cantidad_consumidor_dispuesto_compra
                        lista_consumidor_episodio[a] =
lista_consumidor_episodio[a] - (cantidad_comprada*precio_ofertado)
                        cantidad_ofertada = cantidad_ofertada -
cantidad_comprada
                        cantidad_vendida_total = cantidad_vendida_total +
cantidad_comprada

                    else:
                        cantidad_comprada = cantidad_ofertada
                        lista_consumidor_episodio[a] =
lista_consumidor_episodio[a] - (cantidad_comprada*precio_ofertado)
                        cantidad_ofertada = cantidad_ofertada -
cantidad_comprada
                        cantidad_vendida_total = cantidad_vendida_total +
cantidad_comprada

        ingresos_empresa = lista_precios_ordenada_por_precio[j][1] *
cantidad_vendida_total
        costes_empresa = 100 + lista_precios_ordenada_por_precio[j][2]**2
-10*lista_precios_ordenada_por_precio[j][2]
        beneficios_empresa =  ingresos_empresa - costes_empresa
        lista_precios_ordenada_por_precio[j].append(beneficios_empresa)
        lista_beneficios[j].append(beneficios_empresa)

    for j in range(empresas):
        lista_precios_ordenada_por_precio =
sorted(lista_precios_ordenada_por_precio)

        for g in range(len(lista_estados)):

            fila_proximo_q = 0
            fila_actual_q = 0

            if lista_estados[g] ==
lista_precios_ordenada_por_precio[j][1]:
                fila_proximo_q = g
                max_q = np.max(q_tables[j][0][fila_proximo_q])

            if lista_estados[g] == lista_estados_empresa[0][1]:
                fila_actual_q = g
                valor_q_actual =
q_tables[j][0][fila_actual_q][lista_acciones[j]]

                valor_q_actualizado = valor_q_actual + tasa_aprendizaje *
(lista_precios_ordenada_por_precio[j][3] + factor_descuento -
valor_q_actual)
```

```
                q_tables[j][0][fila_actual_q][lista_acciones[j]] =
valor_q_actualizado
                convergencia = abs(valor_q_actual - valor_q_actualizado)
    if convergencia <= 50:

        lista_episodios_convergentes.append(1)
    else:
        lista_episodios_convergentes.append(0)

lista_precios_evolucion_empresa = []
lista_precios = []

for j in range(empresas):
    lista_precios.append([])

for i in range(len(lista_evolucion_precios_episodio)):
    for j in range(empresas):
        if lista_evolucion_precios_episodio[i][0] == j:
            lista_precios[j].append(lista_evolucion_precios_episodio[i][1
])

n = 100

sumatorio_beneficios = 0
sumatorio_número_beneficios = 1
lista_colusion = []

for i in range(empresas):
    lista_colusion.append([])

    plt.plot(lista_numero_episodios, lista_beneficios[i])
    sumatorio_número_beneficios = sumatorio_número_beneficios +
len(lista_beneficios[i])

    for j in range(len(lista_beneficios[i])):

        sumatorio_beneficios = sumatorio_beneficios +
lista_beneficios[i][j]

        beneficios_promedio =
sumatorio_beneficios/sumatorio_número_beneficios
        lista_colusion[i].append(beneficios_promedio)

lista_colusion_definitiva = []
lista_colusion_predef = []

for j in range(len(lista_beneficios[0])):
    promedio_final = 0
```

```
    for i in range(empresas):
        promedio_final = promedio_final + lista_colusion[i][j]
        if i == (empresas-1):
            lista_colusion_predef.append(promedio_final)
    colusion_final = ((lista_colusion_predef[j]) - (beneficio_nash)) /
((beneficio_maximo) - (beneficio_nash))
    lista_colusion_definitiva.append(colusion_final)

plt.axhline(y=beneficios_promedio, color='blue', linestyle='--', label =
'Límite colusorio')
plt.axvline(x=198000, color='red', linestyle='--', label = 'Greedy')
plt.xlabel('Episodios')
plt.ylabel('Beneficios')
plt.show()

plt.plot(lista_numero_episodios, lista_colusion_definitiva)
plt.axhline(y=0, color='blue', linestyle='--', label = 'Competencia')
plt.axhline(y=1, color='red', linestyle='--', label = 'Colusion')
plt.show()

xmin = 0
ymin = 0
promedios_moviles = []

for j in range(empresas):
    promedios_moviles.append([])

for i in range(n-1, len(lista_precios[0])):
    for j in range(empresas):
        promedio = sum(lista_precios[j][i-n+1:i+1])/n
        promedios_moviles[j].append(promedio)

delete = n - 1
for sublista in lista_precios:
    sublista[:] = sublista[:-delete]

lista_numero_episodios = lista_numero_episodios[:-delete]

for i in range(len(lista_precios)):
    plt.plot(lista_numero_episodios, lista_precios[i], label='Evolución
de precios empresa {}'.format(i+1))

for i in range(len(promedios_moviles)):
    plt.plot(lista_numero_episodios, promedios_moviles[i],
label='Movimientos promedio empresa {}'.format(i+1))

plt.axhline(y=precio_minimo, color='blue', linestyle='--', label =
'Equilibrio de Nash')
```

```
plt.axhline(y=precio_beneficio_maximo, color='red', linestyle='--', label
= 'Beneficio monopolístico')
plt.axvline(x=198000, color='black', linestyle='--', label = 'Greedy')

plt.xlabel('Episodios')
plt.ylabel('Precio')

for i in range(len(lista_episodios_convergentes)-n):
    if lista_episodios_convergentes[i]==1 and
lista_episodios_convergentes[i+1]==1 and
i!=(len(lista_episodios_convergentes)-n):
        plt.axvspan(i, i+1, alpha=0.3, color='y')

plt.xlim(left=xmin)
plt.ylim(bottom=ymin)
plt.show()

for i in range(len(lista_precios)):
    plt.plot(lista_numero_episodios, lista_precios[i], alpha = 0.5,
label='Evolución de precios empresa {}'.format(i+1))

for i in range(len(promedios_moviles)):
    plt.plot(lista_numero_episodios, promedios_moviles[i], alpha = 0.5,
label='Movimientos promedio empresa {}'.format(i+1))

plt.axhline(y=precio_minimo, color='blue', linestyle='--', label =
'Equilibrio de Nash')
plt.axhline(y=precio_beneficio_maximo, color='red', linestyle='--', label
= 'Beneficio monopolístico')
plt.axvline(x=198000, color='black', linestyle='--', label = 'Greedy')

plt.xlabel('Episodios')
plt.ylabel('Precio')

for i in range(len(lista_episodios_convergentes)-n):
    if lista_episodios_convergentes[i]==1 and
lista_episodios_convergentes[i+1]==1 and
i!=(len(lista_episodios_convergentes)-n):
        plt.axvspan(i, i+1, alpha=0.3, color='y')

plt.xlim(left=xmin)
plt.ylim(bottom=ymin)

plt.show()

end2 = time()
print(f"Se han tardado {end2-start2} segundos en completar la simulación
de Bertrand")
```