CAPÍTULO VIII

# ¿Sueña la inteligencia artificial con cárteles virtuales?

**Jose Penalva***

La inteligencia artificial (IA) se aplica como instrumento para fijar precios en muchos sectores más allá del comercio electrónico. Se utiliza para fijar tipos de interés de préstamos, seguros, billetes de hotel y avión, alquileres, etc. En este capítulo discutimos, en el contexto de nuevos resultados que analizan el comportamiento de algoritmos en el marco de la fijación de precios, el posible impacto de este uso de la IA, y hasta qué punto las políticas de competencia pueden seguir aplicando recetas antiguas o han de adaptar sus planteamientos a este nuevo contexto.

*Palabras clave:* inteligencia artificial, competencia, teoría de juegos, política de precios.

* Quiero agradecer a Álvaro Cartea y a Nuria Serrano su ayuda a la hora de revisar versiones preliminares de este artículo, además de todo el apoyo que me han brindado siempre. Añadir a Patrick Wang y Harrison Waldon que junto con Álvaro forman el resto del equipo que desarrolla esta línea de investigación, y a Juan-José Ganuza que me mantiene con los pies en el suelo. También quiero agradecer al contribuyente ya que parte de la investigación sobre la que se apoya este artículo se ha financiado con un proyecto de la Agencia Estatal de Investigación (PID2019-104649RB-I00/AEI/10.13039/501100011033). Los errores son todos míos.

## 1. LA INTELIGENCIA ARTIFICIAL Y LA FIJACIÓN DE PRECIOS

Phillip K. Dick escribió una novela corta (magníficamente reinventada como la película *Blade Runner*) en la que describe un futuro distópico donde un cazarrecompensas llamado Rick Deckard se dedica a "retirar" (matar) androides que vuelven al planeta Tierra sin autorización. La novela gira alrededor de las cuestiones morales que surgen sobre las leyes a aplicar a androides que son cada vez más difíciles de distinguir de los humanos. El título de este capítulo distorsiona libremente el título original de la novela (*¿Sueñan los androides con ovejas eléctricas?*) como analogía a los problemas mucho más concretos y limitados a los que se enfrentan las autoridades de la competencia a la hora de lidiar con la cada vez mayor participación de la inteligencia artificial (IA) en el proceso de fijación de precios. Y, en particular, si las interpretaciones de las leyes desarrolladas para lidiar con comportamientos humanos poco competitivos necesitan modificarse cuando los precios se fijan mediante IA.

A día de hoy, la IA ocupa un espacio cada vez mayor en la fijación de precios y otras decisiones económicas. La irrupción de la IA como instrumento en la fijación de precios empezó en segmentos muy especializados y cada vez más los humanos delegan en ella decisiones sobre precios en diferentes ámbitos. Las primeras aplicaciones aparecen a la hora de fijar los precios de billetes de avión y habitaciones de hotel. Hoy en día encontramos programas de IA específicamente diseñados para estos sectores que se encargan de todo el proceso de fijación de precios y lo hace de manera automática y dinámica, como por ejemplo el programa Airline Dynamic Pricing de la empresa PROS. Para hoteles existen programas parecidos (en la dimensión de fijación de precios) como Revpar Guru, de Rate Shopper.

Los mercados financieros han resultado ser un campo abonado para la IA hasta tal punto que dominan ciertos mercados como los de acciones. Empresas como Virtu y Citadel se han convertido en las principales empresas de mediación y provisión de mercados en las bolsas electrónicas de todo el mundo, y lo hacen mediante algoritmos que utilizan IA. Pero ésto no ocurre sólo en las bolsas. Por ejemplo, Munich Re es una de las empresas líderes en el sector del seguro de automoción, y utilizan la IA en su *software* de fijación de precios, AutoML. En el mercado de préstamos también encontramos programas de valoración de riesgos y fijación de precios como Sopra Banking Suite, Lendstream, o HES Loanbox. Y, naturalmente, con el crecimiento del comercio electrónico, la IA juega un papel cada vez mayor en las políticas de precios. Una parte muy significativa de las ventas en la plataforma Amazon se llevan a cabo en tiendas que usan la IA para fijar precios, con programas como Amazon Algorithmic Reprice (de Feedvisor), PricingPRO (de PROS), o Incompetitor (de Intelligence Node).

Este crecimiento en el número y uso de IA para fijar precios es algo que preocupa a los reguladores, entre otras razones porque consideran que existe un riesgo muy real que el uso de IA en la fijación de precios pueda estar facilitando que los precios aumenten y se mantengan por encima de su nivel competitivo (véase los documentos de la Competition & Markets Authority (2018, 2021), la OCDE (2017), y la Comisión Europea (2017)). Como muestra de este fenómeno podemos leer en un artículo *online* reciente en propublica.org una investigación periodística que relaciona el incremento sustancial en los precios de los alquileres de

viviendas y locales en diversas ciudades de EE. UU. con la compra y aplicación del programa RealPage, o el trabajo de investigación de Assad *et al.* (2021) que asocia aumentos de precios de la gasolina con la adopción de programas de gestión de precios automatizados.

Para poder plantearnos cómo regular la IA en su aplicación al problema de fijación de precios, un problema fundamental es entender qué tipo de comportamiento es capaz de generar y qué le lleva a fijar precios más o menos competitivos. Para esto hay que tener en cuenta que en el problema de fijación de precios los algoritmos no son pasivos. Éstos reaccionan a lo que pasa a su alrededor, y a su vez, lo que hace un algoritmo afecta a lo que observan los demás, y esta interacción puede ser muy compleja. Muchos de los algoritmos de IA se han estudiado en problemas aislados, pero cuando se implementan, muchas veces acaban interactuando entre ellos, y no sabemos hacia dónde les lleva el delicado baile de precios digital que se genera.

En este capítulo se describe el trabajo de investigación que el autor está llevando a cabo con sus compañeros del Oxford-Man Institute, Álvaro Cartea y Patrick Chang, y con Harrison Waldon de la Universidad de Texas, en Austin. En este trabajo, plasmado en dos artículos científicos (Cartea, Chang y Penalva, 2022; Cartea *et al.,* 2022) se utilizan las herramientas de aproximación estocástica para desarrollar un sistema de ecuaciones que permite describir cómo interactúan los algoritmos cuando juegan unos contra otros. Aplicando estas ecuaciones encontramos que si la IA utiliza uno de los algoritmos de referencia, *Q-learning,* la interacción entre algoritmos puede generar precios altos y colusión tácita, sin necesidad de comunicarse entre ellos. También encontramos que los algoritmos aprenden a coordinarse de maneras inesperadas dando lugar a patrones de comportamiento sorprendentes.

## 2. EL JUEGO DEL PRISIONERO: COMPETENCIA Y COLUSIÓN

El proceso de fijación de precios es muy complejo y una de las principales razones por las que la IA aporta valor en tantos sectores es porque puede incorporar de manera rápida, sistemática, y efectiva muchas variables relevantes. Pongamos por ejemplo la fijación de precios (intereses) de préstamos. En este mercado el riesgo de crédito es fundamental a la hora de elegir el tipo de interés que ofrecer. Si fijas un tipo de interés muy alto pierdes clientes, y si lo fijas muy bajo tienes muchos clientes que por su perfil de riesgo te pueden generar más costes que beneficios. Pero determinar el perfil de riesgo y, por lo tanto, un tipo de interés adecuado depende de muchas variables: la situación financiera y patrimonial del cliente, su historial, su situación laboral, etc. La IA ofrece la posibilidad de combinar de una manera rápida y eficiente mucha y muy variada información sobre los clientes, lo que facilita el análisis del perfil de riesgo de los clientes y por tanto el tipo de interés adecuado. En el eCommerce, la IA te permite recopilar y combinar información muy variada no solo de potenciales clientes sino también de la situación del mercado, sobre todo qué precios están ofertando tus competidores, y así optimizar la estrategia de precios. De la misma manera, compañías aéreas y hoteles utilizan la IA para, entre otras cosas, identificar patrones de demanda que les permita obtener una mejor combinación de precios por asiento y ocupación. En todos estos casos, además

de la importancia de una agregación y procesamiento de información adecuados, la selección de la política de precios óptima incluye dos características principales y comunes: la primera es que con un precio más bajo atraes más consumidores, y la segunda es que la IA permite recoger datos y revisar precios continuamente. Estas dos características son las que queremos capturar en nuestro modelo de competencia entre algoritmos.

Cuadro 1.

Castigos en el dilema del prisionero

| *Castigo* | *Jugador 2* | |
|---|---|---|
| *Jugador 1* | *Callar (C)* | *Delatar (D)* |
| Callar *(C)* | Castigo bajo, castigo bajo | Castigo máximo, sin condena |
| Delatar *(D)* | Sin condena, castigo máximo | Castigo alto, castigo alto |

Como modelo, el juego del dilema del prisionero (DdP) nos permite capturar de una manera sencilla esta interacción estratégica en un contexto de competencia entre dos participantes. En la versión clásica del DdP dos participantes deciden entre dos acciones: callar (*C*) o delatar (*D*). Dependiendo de lo que hagan los dos, los resultados que obtiene cada uno varían tal y como se describe en el cuadro 1. En caso de que ambos elijan *C*, ambos irán a prisión pero durante un periodo relativamente corto de tiempo, ya que las pruebas en su contra están limitadas. Si ambos elijen *D* acaban mucho peor, ya que los dos generan nuevas evidencias en su contra. Pero no reciben el castigo máximo por haber cooperado con el fiscal, delatando al otro. Sin embargo, lo peor que puede pasar es que uno se calle (*C*) y el otro delate (*D*). En este caso, uno recibe la pena máxima, y el otro se libra de la cárcel. La propiedad que caracteriza al DdP es que lo óptimo es *D*, independientemente de lo que haga el oponente. Si el oponente calla, lo mejor es delatar para así librarse de la cárcel. Si el oponente delata, lo mejor es delatar también para reducir condena. Por lo tanto, según el concepto de solución de Nash, el único equilibrio posible es (*D D*). A pesar de que ambos preferirían la situación en la que ambos callan, (*C C*), al decidir a nivel individual lo que es óptimo hacer acaban los dos peor, (*D D*).

La relación entre el DdP y el problema de fijación de precios se ve si cambiamos la interpretación de lo que significan *C* y *D*. En el problema de fijación de precios los participantes deciden qué precios poner a los productos que venden. La acción *C,* se interpreta como ofertar un precio alto, mientras que *D* es ofertar un precio más bajo. Si ambos ofertan un precio alto están cooperando (*C*) para sostener beneficios entre competidores, mientras que al bajar precios se desvían (*D*) a un precio más competitivo. En términos de incentivos, la estructura es la misma que en el DdP: en lugar de recibir un castigo bajo por callar, lo que ocurre es que se obtiene un beneficio alto si ambos eligen precios altos (*C C*). En lugar de recibir un castigo más alto por delatar, se obtiene un beneficio más bajo si se eligen precios bajos (*D D*). Y finalmente, si uno fija un precio alto (*C*) y el oponente uno bajo (*D*), el oponente se lleva todo el beneficio y el otro no vende nada. Para que la situación entre el DdP y el problema de fijación de precios sea esencialmente la misma, es fundamental que el beneficio del oponente que fija precio bajo (*D*) cuando el otro fija precio alto (*C*) sea mayor que el beneficio que obtendría si ambos se repartieran el mercado con precios altos (*C C*), pero esto no es difícil que se cumpla.

Por todo esto se utiliza el modelo del DdP para estudiar problemas básicos de competencia en fijación de precios. Y dada su sencillez y amplia utilidad, el modelo del DdP se ha estudiado en profundidad. Lo que hemos descrito es el modelo *one-shot* o estático. El modelo donde los participantes juegan una sola vez. En términos de implicaciones para la fijación de precios, el hecho de que el único equilibrio de Nash de este juego (estático) sea que los dos fijen precios bajos, (*D D*), implica que el resultado de la competencia en precios lleva a precios bajos, a pesar de que los competidores preferirían coordinarse para poner precios más altos y obtener mayores beneficios. Pero si el juego se repite, es decir, si los participantes juegan repetidamente al DdP la interacción estratégica se complica. Los jugadores empiezan a tener en consideración las consecuencias de sus decisiones no sólo durante la partida de DdP que están jugando ahora, sino también el efecto que puedan tener estas decisiones en futuras rondas del DdP. En tal caso la teoría de juegos también nos dice (por medio de lo que se conoce como el *Folk Theorem,* o teorema de tradición oral) que la competencia en precios repetida puede dar lugar a precios altos en el DdP. De manera más concreta, el teorema dice que cuando ambos oponentes interactúan de una manera repetida, los beneficios obtenidos de cooperar (*C C*) se pueden obtener como parte de un equilibrio de Nash del juego repetido. Esencialmente, la teoría establece que puede ser óptimo para ambos oponentes fijar y mantener precios altos (*C*) cuando hay una interacción repetida como la que permite la IA. La interacción repetida introduce la posibilidad de iniciar con precios altos (*C C*), y si alguien se desvía y baja precios (*D*), se puede responder bajando todos los precios en el futuro, lo que comúnmente se describe como una guerra de precios. La amenaza de una guerra de precios hace que el posible beneficio temporal de recortar precios hoy, no compense la futura pérdida de beneficios que provoca el abandonar la actual situación de cooperación y entrar en un guerra de precios. Por lo tanto, la posibilidad de futuras represalias en el juego repetido hace que sea posible generar incentivos para sostener acciones que no son óptimas en el corto plazo (y por lo tanto, no forman parte de un equilibrio de Nash en el juego estático). Y, en el caso de la competencia en precios, la interacción repetida puede dar lugar a una falta de competencia en precios.

Esta posible falta de competencia en precios es uno de los principales problemas al que se enfrentan las autoridades de competencia de todo el mundo, ya que daña a los consumidores, que son los que acaban teniendo que decidir entre pagar precios más altos o ser excluidos del mercado. El argumento fundamental sobre el que se basan las autoridades de competencia es la presencia de colusión. Colusión, según la teoría económica, es la práctica de sostener precios altos por medio de un mecanismo de premio-castigo (PC), o, en un lenguaje más coloquial, de palo y zanahoria. El concepto fundamental es que el uso de mecanismos de PC implica una estrategia consciente y voluntaria de intentar sostener precios por encima de sus niveles competitivos (precios supracompetitivos) y, por lo tanto, la causa de que los precios sean altos. En base a este argumento, la adopción por parte de competidores de estrategias colusivas es una práctica anticompetitiva y por lo tanto perseguible y punible.

Las autoridades de competencia han desarrollado estrategias y métodos para identificar y perseguir posibles casos de colusión entre humanos. En la práctica, a la hora de perseguir colusión las autoridades de competencia intentan demostrar que existe algún tipo de acuerdo

(contractual o no) entre competidores para implementar estos tipos de estrategias de PC. En consecuencia, lo fundamental es que haya algún acuerdo o conspiración para no competir, ya que la ley no obliga a competir. Y para demostrarlo, tradicionalmente las autoridades de la competencia tienen que encontrar evidencia convincente de que los comportamientos observados no podrían ocurrir de manera independiente, y que los actores han tomado decisiones que han servido como instrumento de comunicación entre las partes.

## 3. ¿CÓMO APRENDE LA INTELIGENCIA ARTIFICIAL MIENTRAS COMPITE?

Los métodos de las autoridades de competencia y las estrategias de prueba y evidencia se han desarrollado en un contexto donde los precios los fijan humanos. Pero la pregunta que nos surge es si estos métodos son apropiados en un contexto donde los precios se establecen por medio de algoritmos. Para intentar responder a esta pregunta, el primer paso es comprender cómo se comportan estos algoritmos y para ello hablaremos primero de cómo funcionan los algoritmos de IA de una manera general.

Como dijimos anteriormente, las empresas usan IA para combinar información y utilizarla en la toma de decisiones. Podemos pensar en la IA como una caja negra donde se introduce información y de la que se obtiene una propuesta de decisión (un precio). Existen varios tipos de cajas según el método de aprendizaje que utilicen. En el caso que nos ocupa hablamos de *Reinforcement Learning (RL)* como método de aprendizaje, el tipo de aprendizaje que hay detrás de los algoritmos de DeepMind de Google para jugar al ajedrez (AlphaZero) y al Go (AlphaGo). La IA que utiliza RL lo que hace, básicamente, es usar datos para experimentar con diversas acciones. Las acciones generan diferentes recompensas y el algoritmo las emplea para reforzar aquellas acciones que den mejor resultado y reducir el valor (debilitar) aquellas acciones que den peor resultado. Cuando hablamos de aprendizaje algorítmico o de cómo aprende la IA nos referimos a cómo cambia la estructura interna (los valores que se asignan a las diferentes acciones) de los algoritmos en respuesta a su entorno. Se ha demostrado que este tipo de algoritmos en un entorno de decisión fijo, como por ejemplo cuando se utiliza la IA para organizar horarios o clasificar imágenes, se comportan muy bien en el sentido de que son capaces de generar reglas de decisión óptimas o cuasióptimas de manera rápida y autónoma (sin supervisión humana). Esto las hace especialmente atractivas a la hora de tomar decisiones rápidamente en entornos muy complejos.

Sin embargo, tenemos mucha menos información sobre el comportamiento de estos algoritmos en entornos dinámicos, como el que nos atañe. El problema de aprendizaje de un algoritmo en un contexto de interacción estratégica con otros algoritmos de aprendizaje es mucho más complejo. En nuestro caso, los algoritmos toman decisiones sobre precios que generan (o no) compras, y esto se traduce en beneficios que utiliza el algoritmo como recompensa para aprender. Al haber dos (o más) algoritmos aprendiendo a la vez en un juego donde las acciones de un participante afectan las recompensas del otro, el entorno se vuelve

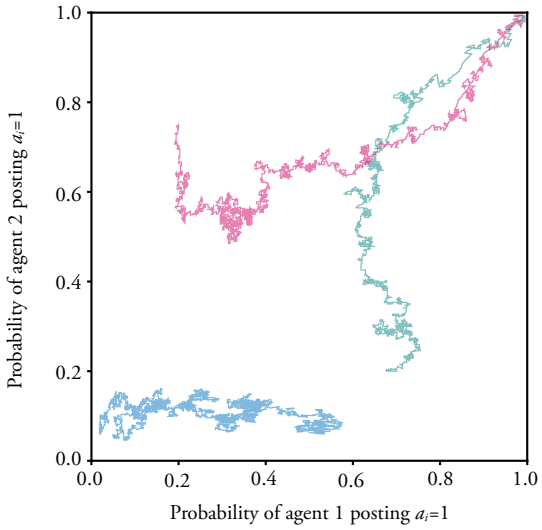
cambiante en el sentido de que las recompensas recibidas por los algoritmos cambia a la vez que los algoritmos modifican su estrategia (aprenden). En términos teóricos esto implica que el entorno se vuelve no-estacionario, lo que supone un problema a la hora de definir lo que significa una acción óptima. Lo que es óptimo en un momento dado puede dejar de serlo en el siguiente debido a que los algoritmos al aprender cambian su comportamiento y, por tanto, las recompensas que puedan recibir tanto ellos como sus contrincantes.

La mayor parte de lo que sabemos sobre el aprendizaje algorítmico en estos entornos se ha obtenido en base a estudios de simulaciones en los que se generan numerosas secuencias de interacciones entre algoritmos y se estudian las estadísticas obtenidas de las trayectorias de las estrategias de los algoritmos en cada simulación (como por ejemplo en Calvano *et al.*, 2020). En estas simulaciones el aprendizaje tiene un fuerte componente aleatorio lo que genera mucho ruido en las trayectorias, tal y como se ilustra en la figura 1. En esta figura se describen las trayectorias de dos jugadores (algoritmos 1 y 2), en un juego simétrico de fijación de precios (DdP) en el que tienen que elegir entre dos acciones, $a_1$ y $a_2$. En la abscisa de la figura 1 tenemos la probabilidad de que el jugador 1 (el algoritmo 1) elija la acción $a_1$ = C (y por lo tanto con la probabilidad complementaria elegirá la acción $a_2$ = D), y en la ordenada la probabilidad de que el segundo jugador (el algoritmo 2) elija la misma acción $a_1$ = C. Las tres trayectorias parten de diferentes puntos iniciales y nos dan una idea de cómo cambia la trayectoria en el tiempo.

Uno de los principales inconvenientes a la hora de analizar estas simulaciones es que para reducir la aleatoriedad en las trayectorias hay que hacer muchas simulaciones, y según

Figura 1.

**Simulación de tres trayectorias en un juego con dos acciones y dos jugadores**



aumenta la complejidad de la interacción, la cantidad de simulaciones necesaria para obtener una muestra representativa puede volverse rápidamente impracticable. Un segundo inconveniente es que el incremento del tamaño de la muestra obliga a que el análisis se centre en propiedades estadísticas de la muestra y en tomar medias entre las trayectorias de la muestra. Estas trayectorias medias pueden ser muy poco representativas del comportamiento de las trayectorias de la muestra.

Desde el punto de vista teórico se han utilizado diferentes técnicas para aproximar la trayectoria que siguen los algoritmos y se han conseguido algunos resultados teóricos con diferentes versiones de algoritmos de aprendizaje con refuerzo (RL) bajo ciertas circunstancias. Los resultados principales se han obtenido en contextos donde la única información usada por los algoritmos es la recompensa recibida durante el juego. Cuando los algoritmos sólo utilizan sus recompensas del juego para aprender se puede construir de manera sencilla la probabilidad de transición de los parámetros de los algoritmos. Con esta probabilidad se puede computar una transición esperada como la solución a una ecuación diferencial ordinaria (EDO), y se puede demostrar que las trayectorias de los algoritmos se aproximan a la trayectoria descrita por la EDO (la transición esperada en tiempo continuo). El método general que se utiliza se denomina aproximación estocástica (*stochastic approximation, SA*).

Para entender mejor lo que sabemos teóricamente y describirlo de una manera más precisa, volvamos al ejemplo en la figura 1. En esta figura podemos ver tres trayectorias. Estas trayectorias tienen dos componentes. Por un lado, hay un componente aleatorio que hace que la trayectoria cambie de manera no predecible. Es lo que hace que parezca que la trayectoria la haya dibujado alguien en un autobús en marcha. Por otro lado, hay un componente determinístico (estable) de la interacción esperada entre los algoritmos que le da dirección a la trayectoria. El método de SA consiste en utilizar una ecuación para describir el componente determinístico de la interacción esperada, y demostrar que la parte aleatoria deja de ser importante en ciertas circunstancias. Específicamente, la dinámica de los algoritmos se puede describir con la siguiente ecuación en diferencias:

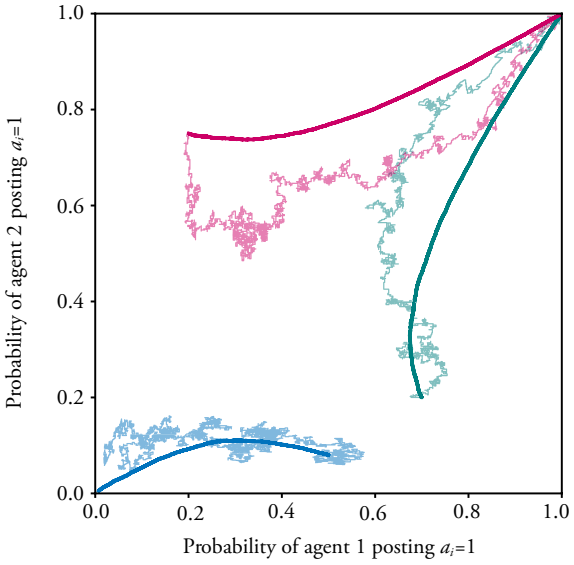
$$\theta_{n+1} = \theta_n + \gamma_{n+1} f(\theta_n, \mathrm{a}_n)\,, \qquad [1]$$

donde $\theta_n$ corresponde al valor de los parámetros que describen el comportamiento de los algoritmos en la ronda $n$ del juego, la función $f$ describe el ajuste que hacen los algoritmos después de tomar las acciones $a_n$ en la ronda $n$ (y observar las recompensas que estas acciones generan), y $\gamma_n$ que describe la velocidad de aprendizaje. La interpretación del parámetro $\gamma$ es muy importante. Cuanto menor sea la velocidad $\gamma$, los parámetros de los algoritmos serán menos sensibles a lo que ocurre en cada ronda. Esto implica que para generar un cambio en los parámetros de una cierta magnitud, con una velocidad γ menor se necesitan más interacciones. En la figura 2 hemos superimpuesto sobre la figura 1 las trayectorias de los algoritmos con el mismo punto de inicio pero con una velocidad de aprendizaje γ mucho menor. Como se puede observar, el componente aleatorio apenas es visible a simple vista. Esto es debido a que la trayectoria acumula muchas interacciones pequeñas

que en media tienden a cancelarse, lo que reduce la importancia del componente aleatorio y deja a la vista el componente dinámico de las interacciones entre jugadores[1].

Figura 2.

**Simulación de tres trayectorias en un juego con dos acciones**




Este argumento que acabamos de proponer se puede formalizar de manera rigurosa. Usando técnicas de SA, se puede demostrar que al disminuir γ la trayectoria en tiempo discreto descrita por los algoritmos y por la ecuación [1] se aproxima a la trayectoria descrita por la siguiente ecuación diferencial ordinaria (en tiempo continuo):

$$\frac{d\theta}{dt} = \dot{\theta} = F(\theta), \qquad [2]$$
$$F(\theta) = \mathbb{E}\left[ f(\theta, a) \right],$$

donde la esperanza se toma sobre la distribución de probabilidad sobre acciones de los algoritmos en la ronda $n$[2].

En la figura 3 hemos superpuesto en gris una serie de trayectorias que describe la dinámica expresada por la ecuación [2] para el juego que estamos usando como ejemplo. Como

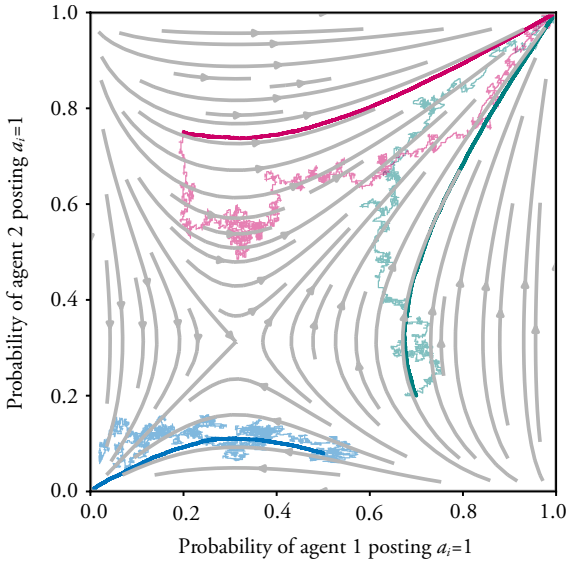
[1] Por otro lado, para simular la trayectoria descrita en la figura se han necesitado muchos más pasos que las de la figura anterior, donde la velocidad de aprendizaje γ era mucho mayor.

[2] Formalmente, $\theta$ define unas estrategias para cada uno de los agentes lo que genera la probabilidad conjunta sobre el espacio de acciones de los agentes, y es sobre esta probabilidad sobre la que se toma la esperanza.

podemos observar, las trayectorias de los algoritmos con una γ pequeña se comportan igual que las trayectorias expresadas por la ecuación [2].

Figura 3.

**Simulación de tres trayectorias en un juego con dos acciones**




La diferencia entre las ecuaciones [1, EDS] y [2, EDO] es que la primera (la EDS) tiene un componente aleatorio que la segunda (la EDO) no. Ambas se describen computacionalmente, pero la EDO, al no tener ningún componente aleatorio nos da varias ventajas. La primera es que para describir lo que se espera que ocurra desde un punto de inicio, basta con hacer una sola simulación. Con la EDS hay que hacer muchas para poder diferenciar el componente aleatorio del que no lo es. Además, la trayectoria que describe la EDO es única, mientras que cuando analizamos una muestra de las EDS hay que aplicar técnicas estadísticas para describir la muestra, y ello puede distorsionar los resultados. En concreto, el mapa de trayectorias descritas por la EDO nos indica que el espacio se puede clasificar en diferentes zonas en base a los puntos a los que tienden a converger las trayectorias (hacia la esquina superior derecha o la esquina inferior izquierda de la figura). En el caso de las trayectorias aleatorias, vemos que si tomásemos la media de trayectorias cerca del área de transición entre las dos zonas, estaríamos mezclando trayectorias con propiedades cualitativas muy distintas y su media nos daría resultados incompatibles con las propiedades fundamentales de la dinámica de la interacción entre los algoritmos. Una tercera ventaja es que el análisis numérico de las EDO se ha ido desarrollando mucho y existen técnicas muy eficientes y con características teóricas conocidas que nos permiten analizar las propiedades con mucha mayor rapidez y confianza que con las EDS. Y aunque la trayectoria de la EDS es esencialmente la misma que

la de la EDO cuando la γ es muy pequeña, el número de pasos necesarios para construir la trayectoria de la EDS es dos o más órdenes de magnitud mayor que la de la EDO, lo que supone un coste computacional mucho mayor.

En nuestro primer trabajo (Cartea *et al.,* 2022a) logramos integrar muchos resultados existentes en la literatura en el contexto de algoritmos que aprenden solamente en base a la recompensa recibida, establecer rigurosamente los resultados de aproximación para una gama muy amplia de algoritmos de aprendizaje, y demostrar que estos resultados de aproximación son válidos para casi todos los algoritmos de aprendizaje de RL y la interacción entre ellos, incluso si los algoritmos no son de la misma familia. Además, usando la EDO para describir la dinámica de los algoritmos en un juego de competencia (a la Bertrand) identificamos familias de algoritmos que convergen a los equilibrios de Nash del juego estático. Éstos, a pesar de la interacción repetida, no generan precios muy alejados de los precios competitivos. Por otro lado, dentro de la familia de algoritmos conocidos como *Q-learning algorithms* encontramos que algunos convergen a equilibrios de Nash del juego estático, mientras que otros pueden converger a acciones muy diferentes, y en concreto a equilibrios con precios significativamente por encima de los precios competitivos. Además, encontramos que los modelos clásicos de *Q-learning* pueden converger a equilibrios de acciones asimétricas, cuando el juego y los equilibrios de Nash son simétricos.

Una de las mayores limitaciones de los resultados que obtenemos en Cartea *et al.* (2022a) es que los algoritmos que estudiamos aprenden de una manera muy limitada. Sólo utilizan información de las recompensas recibidas en cada ronda del juego. La literatura que utiliza simulaciones ha estudiado algoritmos que además de la información sobre las recompensas, también incorporan información adicional, como por ejemplo qué acciones tomó cada uno de los participantes en la ronda anterior (por ejemplo, Calvano *et al.,* 2022). Nuestro segundo trabajo de investigación (Cartea *et al.,* 2022b) obtiene por primera vez resultados teóricos sobre qué tipo de ecuaciones ordinarias describen la trayectoria de los algoritmos que aprenden con información adicional, cuando esta información puede describirse utilizando estados.

Como ejemplo aplicamos la metodología al juego del prisionero repetido lo que nos permite hablar de colusión en precios. Los algoritmos en este segundo trabajo pueden incorporar información adicional, descritos como una señal privada que sólo ve un jugador (cada jugador *i* observa $s^i$), y una señal pública (todos los jugadores observan *s*). Además de la información que ya reciben de la recompensa obtenida en cada ronda. Por lo tanto, la dinámica de los (parámetros de los) algoritmos ya no se caracteriza con la ecuación [1] si no con una serie de ecuaciones como la siguiente que incluye más variables:

$$\theta^i_{n+1}\left(a \mid s^i, s\right) = \theta^i_n\left(a \mid s^i, s\right) + \gamma_{n+1} f^i_{a|s^i,s}\left(\theta_n, s_n, \mathrm{a}_n, s_{n+1}\right) \qquad [3]$$

La estructura es similar pero ahora tenemos un componente adicional que es $s_n$, un parámetro que recoge toda la información en las señales $\left(s, \left(s^i\right)_{i=1}^I\right)$.

No teníamos resultados teóricos sobre este tipo de algoritmos porque la metodología tradicional no puede resolver el problema que genera tener la información adicional recogida en la variable *s*. Si aplicamos la técnica de tomar la media como hicimos en la ecuación [3] nos queda la función: $\overline{f}^{i}_{a|s^i,s}(\theta_n, s_n)$ que no es determinística. Esta función incluye un componente aleatorio en su dependencia del estado, $s_n$.

La dependencia de la función $\overline{f}$ del estado, $s_n$, genera un problema muy significativo desde el punto de vista teórico. En principio podríamos tomar la esperanza sobre *s* tal y como hacemos sobre las acciones, *a*. El problema es que para tomar esa esperanza habría que computar la medida sobre *s* y esa medida es muy difícil de computar. La dificultad estriba en la riqueza que aporta la variable *s* a los algoritmos. Cuando dejamos que los algoritmos condicionen su comportamiento en la información en *s*, permitimos una variedad mucho más amplia de comportamiento, pero a la vez generamos un problema. Al permitir que las acciones dependan de información adicional, y a su vez, permitir que esa información adicional incluya las acciones de los algoritmos, se genera un bucle de interacción entre información y acciones que complica el tomar expectativas. En concreto, para conseguir una versión de $\overline{f}$ que sea determinística, en cada ronda tendríamos que reconstruir la medida sobre $s_n$ paso por paso desde el estado inicial $s_0$, seguido de las primeras acciones $a_1$, después el estado resultante, $s_1$, después las acciones siguientes $a_2$, y así sucesivamente hasta llegar a la ronda *n*. Y en cada paso habría que tener en cuenta todas las posibles combinaciones. Esto es analítica y numéricamente impracticable.

Nosotros proponemos una solución: en lugar de utilizar la medida correcta, apliquemos la medida que se aplicaría en la ronda *n* si de ahí en adelante los algoritmos dejasen de aprender, $\Gamma_{\theta n}$. De esta manera tenemos una medida sencilla de construir que nos permite tomar la esperanza y obtener una función determinística:

$$F^{i}_{a|s^i,s}(\theta_n) = \mathbb{E}^{\Gamma_{\theta n}}\left[\overline{f}^{i}_{a|s^i,s}(\theta_n, s)\right] \sum_{s} \Gamma_{\theta n}(s)\, \overline{f}^{i}_{a|s^i,s}(\theta_n, s) \qquad [4]$$

De esta manera podemos construir una función *F* que describe una ecuación diferencial ordinaria como la EDO anterior (donde obviamente este *F* será una función diferente del *F* que solamente utiliza información de la recompensa en nuestro trabajo anterior):

$$\dot{\theta} = F(\theta)$$

La parte más compleja de nuestro trabajo es demostrar que, a pesar de que la esperanza que estamos tomando en este segundo paso no es la correcta, las trayectorias de los algoritmos se aproximan, según $\gamma$ se hace más pequeña, a las trayectorias de esta nueva EDO. La demostración se puede comprobar en Cartea *et al.* (2022b).

Por lo tanto, hemos conseguido una herramienta que nos permite caracterizar las trayectorias de los algoritmos en un contexto más complejo que el existente anteriormente. Además, la ventaja de la EDO sobre la ecuación [3] es mucho mayor que la que hay entre la EDO y la ecuación [1], ya que el componente aleatorio en la ecuación [3] es mucho mayor. Las

ventajas de computación numérica, reducción a trayectorias determinísticas, y la eliminación de la agregación estadística de trayectorias suponen un gran avance. Y éste lo aplicamos al problema de competencia en el juego del prisionero.

## 4. COLUSIÓN TÁCITA Y CÁRTELES VIRTUALES

El DdP nos ofrece un modelo muy estilizado de la competencia en precios. Con nuestra investigación buscamos utilizar este modelo para entender cómo se comporta la IA, y si puede dar lugar a situaciones poco competitivas, e incluso a un comportamiento propio de acuerdos anticompetitivos (colusión tácita). En particular, nos centramos en el DdP repetido donde los contrincantes usan un tipo de algoritmo de aprendizaje concreto: *Q-learning.* Este algoritmo lo hemos elegido porque es un algoritmo muy representativo de los que se utilizan en la práctica y relativamente sencillo. Esencialmente, el algoritmo de aprendizaje fija unos pesos, que identificamos con la letra Q, para cada acción: $Q(C)$ y $Q(D)$. Cuando juega una acción $a$, por ejemplo $a = C$, el algoritmo actualiza el valor de $Q(C)$ pero no el de $Q(D)$, aumentándolo en relación al valor que obtiene después de haber tomado esa acción. Si siempre toma la acción con el valor de $Q$ más alto nunca aprendería porque siempre estaría tomando la misma acción. Para evitar esto, el algoritmo "explora", es decir, prueba acciones que tienen valores de $Q$ más bajos por si acaso podrían dar mejor resultado (y actualiza el valor de la acción que ha probado). La exploración permite al algoritmo ajustar los valores de $Q$ para así aprender y elegir (con mayor probabilidad) la mejor acción.

En nuestro caso, y para incorporar el hecho que los algoritmos en la práctica usan información adicional además de las acciones que han tomado, aumentamos la información que recibe el algoritmo de *Q-learning* en el juego repetido. La literatura teórica que analiza algoritmos de *Q-learning* que actualizan sólo los pesos de las acciones ($Q(C)$ y $Q(D)$ en el DdP). Nosotros le damos más flexibilidad al algoritmo, permitiéndole tener pesos para $C$ y para $D$ diferentes, dependiendo de lo que haya pasado en la ronda anterior. Esto implica que hay cuatro pesos para la acción $C$: $C|CC$, $C|CD$, $C|DC$, y $C|DD$, donde $Q(C|CC)$ es el peso de la acción $C$ si en la ronda anterior tanto un algoritmo como su oponente tomaron la acción $C$. Igualmente, la acción $D$ también tiene cuatro pesos, uno por cada combinación de acciones pasadas. Formalmente, usamos la anotación $a_i$ y $a_j$ para referirnos a las acciones de los oponentes/algoritmos $i$ y $j$, y el estado actual $s_n$ lo describen las acciones de los dos participantes en la ronda anterior, en $n$-1 $\left(s_n = a_{n-1} = \left(a^1_{n-1}, a^2_{n-1}\right)\right)$. En base a nuestros resultados construimos el sistema de ecuaciones (EDO) $\dot{\theta} = F\left(\theta\right)$.

Este sistema está compuesto de una serie de ecuaciones que incluimos aquí para aquellos lectores que quieran saber exactamente cuáles son:

$$\dot{Q}^1\left(a \mid s\right) = \Gamma_{\bar{Q}}\left(s\right)\pi^1_{\bar{Q}}\left(a \mid s\right)\sum_{a^2 \in A^j}\pi^2_{\bar{Q}}\left(a^2 \mid s\right)\left(R^1\left(a, a^2\right) + \delta \max_{a'}\bar{Q}^1\left(a' \mid \left(a, a^2\right)\right) - \bar{Q}^1\left(a \mid s\right)\right),$$
$$\dot{Q}^2\left(a \mid s\right) = \Gamma_{\bar{Q}}\left(s\right)\pi^2_{\bar{Q}}\left(a \mid s\right)\sum_{a^1 \in A^i}\pi^1_{\bar{Q}}\left(a^1 \mid s\right)\left(R^2\left(a, a^1\right) + \delta \max_{a'}\bar{Q}^2\left(a' \mid \left(a^1, a\right)\right) - \bar{Q}^2\left(a \mid s\right)\right). \quad [5]$$

Estas dos ecuaciones tienen cuatro versiones por cada una de la diferentes combinaciones de $s$ = ((*CC*), (*CD*), (*DC*), (*DD*)) multiplicadas por las dos posibles acciones, $a \in \{C, C\}$. En total tenemos 2 por 4 por 2 = 16 ecuaciones.

Desde cualquier punto de partida (descrito por los parámetros $\left(Q_0^i(a \mid s)\right)$) donde iniciemos los algoritmos, estas ecuaciones (las EDO) nos describen cómo evolucionan los algoritmos, y por consiguiente cómo aprenden y qué es lo que acaban aprendiendo a hacer. En particular, lo que nos importa saber es la estrategia de los algoritmos y el resultado de la interacción de las estrategias. Para facilitar la visualización de los resultados vamos a reducir el número de ecuaciones imponiendo supuestos de simetría, y en lugar de centrarnos en los parámetros $Q^i$ nos centramos en la probabilidad de cada acción condicional a cada estado que estos parametros generan: $\pi^i_{Q(n)}(a \mid s)$, o más sencillamente, $\pi^i_n(a \mid s)$. En primer lugar, aprovechamos que $\pi^i_n(D \mid s) = 1 - \pi^i_n(C \mid s)$ para reducir el número de ecuaciones a la mitad. Después asumimos simetría entre los dos algoritmos: $Q^1_0(a \mid s) = Q^2_0(a \mid s)$, lo que implica que $\pi^1_0(a \mid s) = \pi^2_0(a \mid s)$. Y por último, reducimos el número de ecuaciones relevantes a tres asumiendo simetría en las desviaciones unilaterales: $Q^i$ ($D|(DC)$) = $Q^i$ ($D|(CD)$), lo que implica que $\pi^1_n(a \mid CD) = \pi^2_n(a \mid DC)$. Con estos supuestos el número de estados y ecuaciones que necesitamos para describir el sistema completo son tres: cooperación (*C C*), desvío unilateral (*C D*) o (*D C*), y competencia/castigo (*D D*). Asociados a estos estados tenemos las dos variables que nos importan: la estrategia de los algoritmos, descrita con la variable $\pi_n$ ($C|s$) que nos dice la probabilidad tomar la acción *C* (cooperar/precio alto) si el algoritmo recibe la información: el estado es $s$ (dados los parámetros $Q$), y el resultado de la interacción de estas estrategias, $\Gamma_{Q(n)}(s)$, que nos dice dadas estas estrategias, cuál es la probabilidad de que el estado resultante sea $s$ (si los algoritmos dejan de aprender y por lo tanto los parámetros se quedan en $Q_n$ y $\pi^1_t(a \mid s) = \pi^i_n(a \mid s)$ para cualquier $t \geq n$).

Para visualizar lo que nos permite nuestra metodología vamos a considerar dos puntos de partida diferentes y ver lo que hacen los algoritmos. Los resultados aparecen en las figuras 4 y 5. Los parámetros iniciales son (A) figura 4: $\pi^0_{C|CC} = 0{,}8$, $\pi^0_{C|DD} = 0{,}4$ y $\pi^0_{C|CD} = 0{,}5$ y (B) figura 5: $\pi^0_{C|CC} = 0{,}7$, $\pi^0_{C|DD} = 0{,}7$ y $\pi^0_{C|CD} = 0{,}5$.

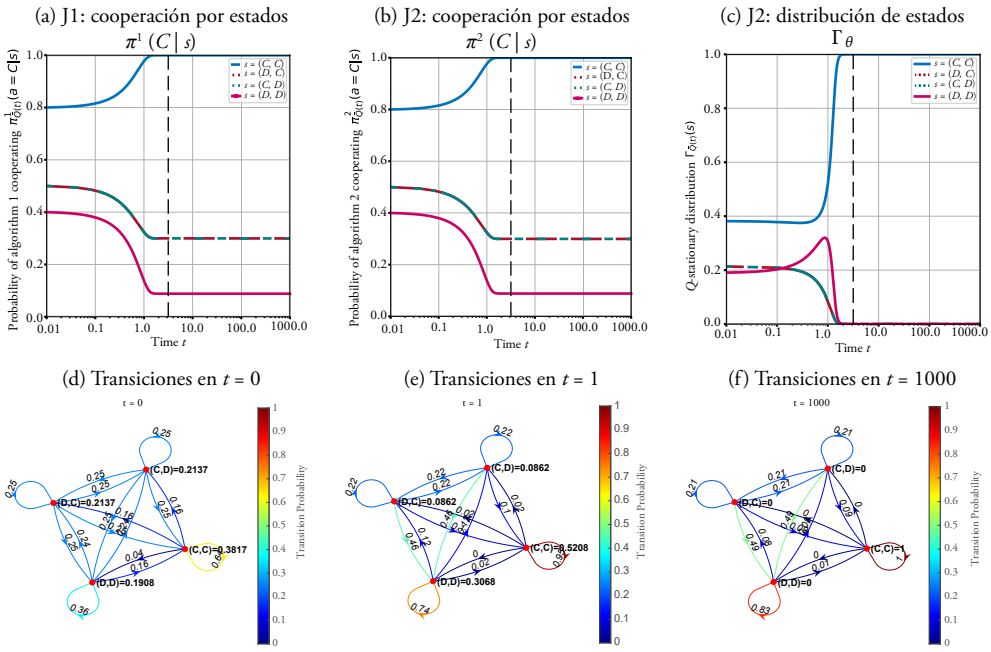
Las dos figuras reflejan el comportamiento de los algoritmos a lo largo del tiempo (en escala logarítmica). Las figuras están divididas en seis paneles. En los paneles (a) y (b) encontramos las estrategias, la probabilidad de cooperar. Las diferentes líneas describen la probabilidad de cooperar dependiendo de la información recibida. Si reciben la información (*C C*), es decir que en la última ronda ambos cooperaron, la probabilidad de volver a cooperar la recoge la línea continua azul. La línea rosa describe la probabilidad de cooperar si reciben la información (*D D*), y las líneas intermitentes las probabilidades de cooperar si uno de los dos tomó la acción *C* y el otro la *D*. En el panel (*C*) tenemos las consecuencias de la interacción de esas estrategias: la probabilidad de los estados (*C C*) en azul, (*D D*) en rosa, y (*C D*) o (*D C*) la línea intermitente. En la figura 4 podemos ver cómo los dos jugadores empiezan (para $t < 0{,}01$) mezclando la decisión entre precios altos (*C*) y bajos (*D*) dependiendo del estado: precios altos (*C*) con probabilidad 0.8 si los precios han sido altos (*C C*), con probabilidad 0.4 si los precios habían sido bajos (*D D*), y con probabilidad 0.5 si uno había ofertado precio alto y el

otro bajo. Según va pasando el tiempo vemos que las líneas intermitente y rosa bajan, lo que implica que disminuye la probabilidad de un precio alto (*C*) y por consiguiente aumenta la de un precio bajo (*D*) –los algoritmos "aprenden a penalizar" tanto si es una desviación unilateral (se observe C,D o D,*C*)– la línea intermitente, o si ya estaban penalizando (*D D*), pero sin llegar al extremo de penalizar con probabilidad uno. El resultado de esto lo vemos en el panel 4c donde la probabilidad de observar cooperación/precios altos (*C C*) se mantiene estable, pero aumenta la probabilidad de observar competencia (*D* D-la línea rosa aumenta entre $t = 0$ y $t = 1$). Pero, llega un momento en que los algoritmos empiezan a cooperar con mayor frecuencia, la línea azul sube hasta llegar al punto donde en el panel 4c la probabilidad de (*C C*) llega a uno y las demás líneas bajan a cero –los algoritmos sólo están ofertando precios altos. En los paneles 4a y 4b vemos que los algoritmos han aprendido una estrategia que les lleva a fijar precios altos si observan precios altos (*C C*), pero castigan con una probabilidad alta cualquier desviación, con estrategias que dan una probabilidad baja a cooperar si los estados son (*C D*), (*D C*) o (*D D*).

En los paneles inferiores, las figuras 4d-4f describen las probabilidades de transición entre estados condicional en el estado de partida, en diferentes momentos en el tiempo (inicialmente $t = 0$, en un punto intermedio $t = 1$, y cuando ya ha aprendido y los parámetros no cambian $t = 1000$). Al principio (figura 4d) hay mucho movimiento entre estados. En el

Figura 4.

**Ejemplo de colusión (A)**



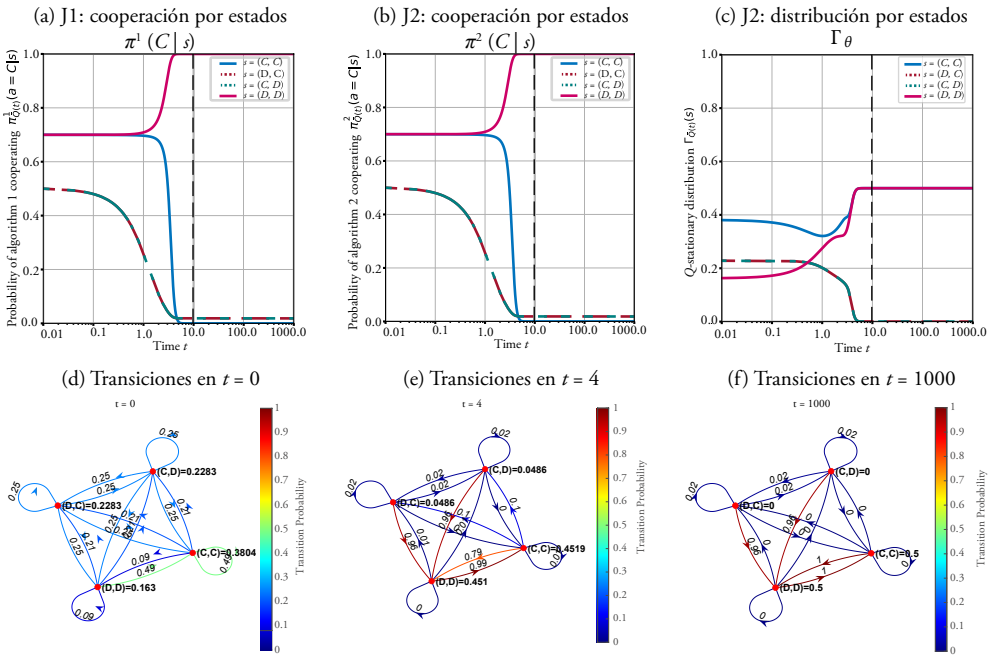
paso intermedio (figura 4e) vemos que las transiciones están concentradas en quedarse en (*C C*) o en transicionar a (*D D*). Este patrón es el que acaban aprendiendo los algoritmos, pero evitando la transición a (*D D*) y quedándose en (*C C*).

En resumen, vemos cómo los algoritmos "aprenden" a mantener precios altos bajo la "amenaza" de pasar un tiempo significativo en una guerra de precios bajos (en D,*D*). Esto es un patrón de comportamiento tradicionalmente descrito como colusión tácita: los participantes sostienen precios altos con una estrategia de PC sin comunicarse entre ellos. Esto es especialmente sorprendente, ya que los algoritmos además de encontrar una acción que les beneficia a ambos, además consiguen coordinarse para penalizar al otro si el otro no se comporta correctamente.

En la figura 5 encontramos un comportamiento diferente, a raíz de un punto de partida diferente. La estructura de los paneles es la misma que la anterior por lo que empezamos con las estrategias iniciales de ambos jugadores en los paneles 5a y 5b. Al principio, ambos jugadores cooperan con una probabilidad elevada (0.75) si ambos toman la misma acción, tanto en precios altos (*C C*) o en precios bajos (*D D*). Por otro lado, si se observa una desviación unilateral se empieza cooperando con probabilidad 0.5, pero enseguida esta probabilidad empieza a disminuir. En el panel 5c vemos que en este momento inicial los algoritmos van cambiando de estado sin quedarse mucho tiempo en ninguno de ellos, aunque, como vemos en el panel 5d

Figura 5.

**Ejemplo de coordinación (B)**



la tendencia es a estar en (*C C*) y retornar allí desde (*D D*) cuando hay desviaciones. Según pasa el tiempo y los algoritmos aumentan la penalización por desviaciones unilaterales los algoritmos empiezan a reforzar esta transición a (*C C*) desde (*D D*) en respuesta a cualquier desviación, aunque a su vez, también observamos que se refuerza la transición de (*C C*) a (*D D*). Esto provoca un cambio en la dinámica que observamos en los paneles panel 5a-panel 5c y panel 5f. Los algoritmos "aprenden" a responder con precios bajos con probabilidad uno cuando ambos juegan (*C C*), mientras que en cuanto ambos juegan (*D D*) responden con precios altos con probabilidad uno, mientras que cualquier desviación unilateral se responde con precios bajos. El resultado final es que los algoritmos entran en un bucle de alternancia entre precios altos y bajos. Ambos se coordinan en un comportamiento cíclico, que alterna precios altos un día con precios bajos el siguiente. Este tipo de comportamiento se había observado en simulaciones pero existían dudas sobre si esto es un fenómeno común o no.

Hemos visto dos ejemplos. En uno los algoritmos aprenden a sostener colusión de una manera tácita, en el segundo aprenden a coordinarse en un comportamiento cíclico. Esto nos lleva a preguntarnos, ¿cuál es el comportamiento general que deberíamos observar de estos algoritmos?, ¿se coordinan?, ¿cooperan?, ¿coluden? Para contestar a estas preguntas, en la figura 6 exploramos lo que ocurre cambiando los parámetros de inicio. En la fila superior, paneles 6a a 6c, estudiamos lo que ocurre cuando empezamos con una probabilidad de cooperar de 0.8 después de observar cooperación, $\pi^0_{C|CC} = 0{,}8$, si variamos la probabilidad de desviarse (*D*) después de observar los otros estados: (*D D*) en la ordenada y (*D C*) en la abscisa. Estas probabilidades las variamos de 0.1 a 0.9, lo que nos genera una malla de 81 valores iniciales, cada cuadrado coloreado de estas figuras corresponde a una combinación de parámetros iniciales diferente. En la fila inferior, paneles 6d a 6f, estudiamos lo mismo pero partiendo de una probabilidad de cooperación tras observar (*C C*) igual a 0.7. En total analizamos 162 parámetros iniciales que nos dan una idea de cuán representativos son nuestros dos ejemplos.

Los paneles 6a y 6d describen el punto de partida: la probabilidad que las estrategias iniciales de ambos den lugar a precios altos (*C C*), $\Gamma_0$(CC). En ambos casos esta probabilidad es inicialmente alta (azul-colores fríos) solo si la probabilidad de precios bajos (*D*) es baja (y la complementaria, la de precios alta, *C*, es alta) –en la esquina inferior izquierda. Los paneles 6c y 6f describen el punto final: la probabilidad de que las estrategias que ambos aprenden den lugar a precios altos (*C C*), $\Gamma_T \star$ (CC). En el panel 6c vemos que el espacio se divide en dos: en rojo aprenden precios competitivos, y en azul precios de colusión. Esto nos indica que los valores iniciales son importantes a la hora de predecir lo que van a hacer los algoritmos. Cuando los parámetros iniciales tienden a ofrecer precios altos, es muy probable que los algoritmos acaben en (*C C*), mientras que si tienden a ofrecer precios competitivos, es más probable que los algoritmos acaben ofreciendo precios competitivos, (*D D*). Sin embargo, en el panel 6f aparece un tercer color, el blanco. Para estos parámetros iniciales los algoritmos no acaban ni en (*C C*) ni en (*D D*). Lo que observamos es que aprenden a alternarse tal y como vimos en el ejemplo B. Con esto vemos que los algoritmos pueden coordinarse en ciclos y no es un comportamiento inusual, sino que ocurre de manera robusta para un número significativo de parámetros iniciales.

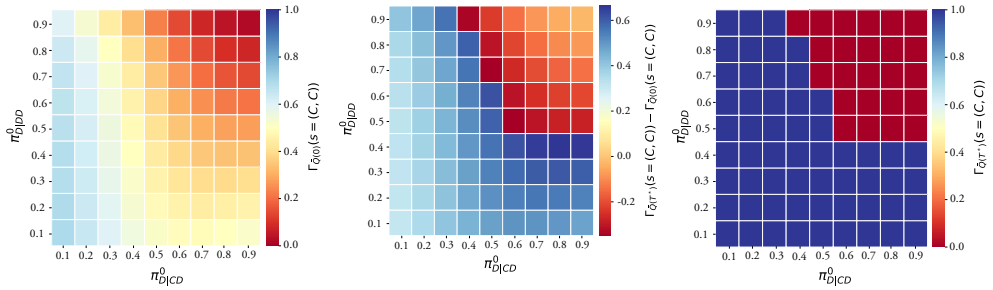
Figura 6.

**Espacio de cooperación**

(a) $\pi^0_{C|CC} = 0{,}8$, Prob de cooperación ($C$ $C$) inicial $\Gamma_\theta(\mathrm{C})$

(b) $\pi^0_{C|CC} = 0{,}8$, Ganancia en cooperación

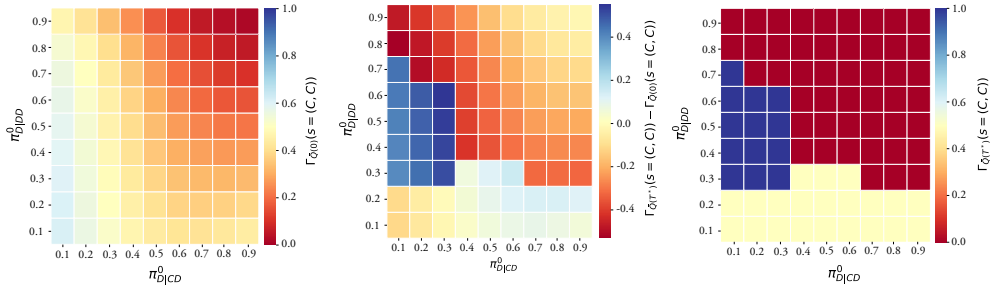
(c) $\pi^0_{C|CC} = 0{,}8$, Cooperación ($C$ $C$) aprendida




(d) $\pi^0_{C|CC} = 0{,}7$, Prob de cooperación ($C$ $C$) inicial $\Gamma_\theta(\mathrm{C})$

(e) $\pi^0_{C|CC} = 0{,}7$, Ganancia en cooperación

(f) $\pi^0_{C|CC} = 0{,}7$, Cooperación ($C$ $C$) aprendida




Los paneles intermedios 6b y 6e describen el cambio en la probabilidad de ofrecer precios altos ($\Gamma_T \star (CC) - \Gamma_0(CC)$). Colores claros, cercanos al blanco, indican que esta probabilidad cambia poco, y los encontramos en las esquinas superior derecha e inferior izquierda, así como en la zona correspondiente al aprendizaje de ciclos. Estos colores claros indican que los algoritmos no cambian mucho su comportamiento y sugiere que los algoritmos empiezan con cierto comportamiento y éste se hace permanente. Es como si se coordinasen para quedarse como empezaron. En cambio, hay grupos de parámetros iniciales, sobre todo alrededor de las zonas de transición de un color a otro (en los paneles 6c y 6f), donde observamos grandes cambios entre el comportamiento observado inicialmente y el observado cuando los algoritmos se estabilizan.

Para analizar en más detalle lo que pasa en estas zonas de grandes cambios miramos cómo cambia el comportamiento de los algoritmos en base a los cambios que observamos en las estrategias, recogidos en la figura 7. Para facilitar la transición entre la figura 6 y la 7 repetimos los paneles 6c y 6f en los paneles 7a y 7d. Éstos los completamos con los paneles 7b y 7e

donde capturamos el cambio en el comportamiento cuando se observa una desviación unilateral, cuando $s = (C\,D)$ or $(D\,C)$, y en concreto, miramos como cambia la probabilidad de ofrecer precios bajos, que denominamos el castigo inmediato, $\pi(D|(CD))$. Lo que observamos en nuestra muestra de parámetros es que los algoritmos tienden a incrementar esta probabilidad de castigo, sobre todo cuando ésta es inicialmente baja. La mayor variación la encontramos en los valores de inicio intermedios, en las zonas de transición entre diferentes áreas coloreadas en los paneles 7a y 7d. En el panel 7b vemos cómo esta probabilidad de castigo inmediato aumenta (se vuelve azul) a ambos lados de la línea que separa los parámetros que llevan a $(C\,C)$ o a $(D\,D)$ para valores intermedios de las probabilidades de castigo. En el panel 7c donde se reflejan los cambios en la probabilidad de precios bajos después de observar $(D\,D)$, lo que llamamos la duración del castigo, podemos observar un patrón similar: la mayor variación la encontramos en los valores de inicio intermedios, y en las zonas de transición entre diferentes áreas coloreadas en los paneles 7c y 7f. En el panel 7c la duración del castigo también aumenta (se vuelve azul) a ambos lados de la línea que separa los parámetros que llevan a $(C\,C)$ o a $(D\,D)$ especialmente para los valores iniciales intermedios, y observamos una gran diferencia en la ganancia de la duración del castigo (el cambio en la duración del castigo) a ambos lados de la zona donde acaban alternándose entre $(C\,C)$ y $(D\,D)$, y las otras zonas.

Estos cambios en las estrategias de los algoritmos las interpretamos de la siguiente manera: en primer lugar diferenciamos tres áreas: (i) la de precios altos $(C\,C)$, (ii) la de precios bajos, $(D\,D)$, y (iii) la de alternancia. En segundo lugar observamos que en las estrategias que tienen una tendencia inicial fuerte hacia áreas (i) o (ii), las estrategias varían poco (colores rojos y rojizos). Esto lo interpretamos como un comportamiento de coordinación entre algoritmos sobre valores iniciales. En tercer lugar observamos importantes cambios en las zonas de transición entre las áreas. En estas zonas el comportamiento lo entendemos como aprendido. Los algoritmos se adaptan hacia un comportamiento u otro. Esta adaptación se centra en un aumento de la estrategia de castigo: aumento tanto del castigo inmediato como la duración de castigo. Esta adaptación puede tener dos consecuencias: si el castigo es adecuado, ayuda a sostener precios altos, pero por el otro lado, si el castigo es demasiado duro éste empuja a los algoritmos a precios bajos. Por lo tanto, para parámetros donde no hay una tendencia clara hacia un lado u otro, los algoritmos endurecen los castigos y esto puede dar lugar a dos situaciones: colusión tácita o guerra de precios.

Por lo tanto, con la metodología que hemos desarrollado podemos analizar en detalle cómo se comportan los algoritmos que compiten entre sí. Hemos visto en un contexto estilizado del DdP las políticas de precios que implementan los algoritmos de *Q-learning.* En particular, hemos visto que los algoritmos tienden a mantener y reforzar su comportamiento inicial, si sus parámetros iniciales son similares a los de uno de los tres comportamientos que observamos en el límite: sostener precios altos $(C\,C)$, precios bajos $(D\,D)$, o alternar entre uno y otro en un ciclo determinístico. Con estos parámetros iniciales, el comportamiento que percibimos lo interpretamos como acomodación y/o coordinación a los parámetros iniciales. Por otro lado, analizamos que para parámetros iniciales sin una tendencia inicial clara hacia uno de estos tres comportamientos, lo que observamos es que los algoritmos tienden a aumentar las estrategias de bajar precios y que esto puede dar lugar a dos tipos de comportamiento.

Por un lado, esta bajada de precios puede dar lugar a una guerra de precios que lleva a los algoritmos a jugar (*D D*). Por otro lado, si la estrategia de penalización implícita en la bajada de precios no es demasiado agresiva, los algoritmos acaban replicando comportamiento de colusión tácita: sostienen precios altos bajo una amenaza de penalización sustancial en respuesta a bajadas de precio.

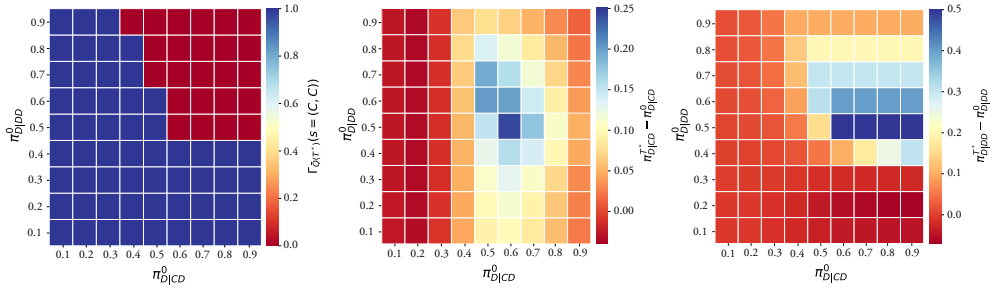
Figura 7.

**Cooperación y castigo**

(a) $\pi^0_{C|CC} = 0{,}8$, Cooperación ($C\,C$) aprendida $\Gamma_T \star (C)$

(b) $\pi^0_{C|CC} = 0{,}8$, Ganancia de castigo inmediato $\Delta \star \pi\,(D|CD)$

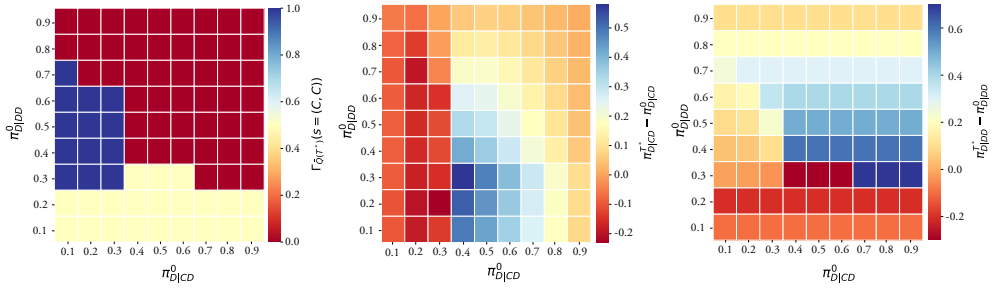
(c) $\pi^0_{C|CC} = 0{,}8$, Ganancia en duración de castigo $\Delta \star \pi\,(D|DD)$




(d) $\pi^0_{C|CC} = 0{,}7$, Cooperación ($C\,C$) aprendida $\Gamma_T \star (C)$

(e) $\pi^0_{C|CC} = 0{,}7$, Ganancia de castigo inmediato $\Delta \star \pi\,(D|CD)$

(f) $\pi^0_{C|CC} = 0{,}7$, Ganancia en duración de castigo $\Delta \star \pi\,(D|DD)$




## 5. LA IA: NUEVOS DESAFÍOS PARA LAS AUTORIDADES DE COMPETENCIA

Nuestros resultados en el contexto limitado del juego del DdP con dos algoritmos de *Q-learning* compitiendo entre sí sugieren que las autoridades de la competencia se enfrentan a nuevos desafíos, ya que la IA puede generar comportamientos poco competitivos, con las características propias de colusión, pero sin generar las señales que se utilizan hoy en día para perseguir casos de colusión.

Primero, hemos visto que los algoritmos, partiendo de ciertos parámetros iniciales, cambian de estrategia ($\pi(a|s)$) y de comportamiento ($\Gamma(s)$) de una manera que es consistente con el aprendizaje de estrategias de colusión. Para intentar aproximarnos a las implicaciones de estos resultados, sigamos la propuesta de Maureen Ohlhausen de la Federal Trade Commission y sustituyamos el término "algoritmo" por la frase "un tipo llamado Pepe". ¿Qué supone que dos tipos llamados Pepe ajusten sus estrategias de tal manera que ofrezcan precios altos bajo la amenaza de castigo con precios más competitivos en respuesta a una reducción de precios unilateral? La regulación de competencia tradicional llama a este tipo de comportamiento colusión tácita. Pero, la aplicación de la legislación vigente a estos dos tipos llamados Pepe suele requerir algún tipo de pruebas de comunicación entre ellos. En el caso de algoritmos no podemos quedarnos en la perspectiva humana y hemos de dejar de hablar de tipos llamados Pepe y hablar de algoritmos. Es difícil argumentar que los algoritmos estén comunicándose ya que hemos visto que no lo necesitan. Si partimos de una perspectiva humana exclusiva, la autoridad de la competencia no puede perseguir esta situación de comportamiento de colusión tácita de libro que encontramos. La colusión tácita está clara y es claramente demostrable, ya que se puede acceder a los parámetros de comportamiento de la IA, pero esta colusión tácita se establece en un contexto de falta de comunicación igualmente demostrable.

El segundo desafío viene de determinar cómo han llegado los algoritmos a la situación de comportamiento colusivo, una pregunta que ni siquiera se plantea cuando hablamos de colusión entre humanos. Hemos determinado que una elección estratégica de los parámetros iniciales puede colocar a los algoritmos en unas trayectorias con una tendencia muy fuerte a generar un comportamiento u otro. Las autoridades se enfrentan a una nueva situación donde los humanos interactúan en un metajuego estratégico de selección de los valores iniciales de los parámetros con los que los algoritmos arrancan. Es ahí donde se ha de buscar la intencionalidad que está ausente del comportamiento mecánico de los algoritmos. El desafío está en qué criterios establecer para determinar hasta qué punto ciertos parámetros iniciales se pueden considerar causantes de una situación de colusión tácita o no.

Un tercer desafío surge de la práctica del uso de la IA. La situación descrita aquí, en la que se implementan los algoritmos con unos parámetros iniciales y se les deja actuar hasta el final, no es común en la práctica. Lo normal es que los algoritmos se desarrollen previamente, se implementen, y una vez en funcionamiento, éstos estén supervisados por humanos que de vez en cuando intervienen para ajustar diferentes parámetros de los algoritmos. Estas decisiones se toman para mejorar el funcionamiento de los algoritmos, que es establecer precios que generan mayores beneficios. Está claro que la selección de parámetros que llevan a los algoritmos a aprender a coludir son los más rentables y por lo tanto los que acabarán siendo seleccionados. El desafío estriba en que la ley no obliga a competir, lo que hace imposible castigar este tipo de comportamiento si no hay comunicación entre las empresas (y por lo tanto una conspiración para no competir). ¿Hasta qué punto puede o debe la autoridad de la competencia establecer un criterio que determine que los precios de los algoritmos son demasiado altos y que las empresas están obligadas a intervenir para corregir el comportamiento de estos algoritmos?

No vamos hacia el futuro que previó Phillip K. Dick donde los replicantes se hacen humanos, si no hacia uno donde la IA replica comportamientos desde una lógica propia. Esta lógica nos obliga a replantear qué es y cómo se defiende la competencia, ya que parece que la IA a veces sueña con sus propios cárteles virtuales.

## Referencias


Assad, S., Clark, R., Ershov, D. y Xu, L. (2020). Algorithmic pricing and competition: Empirical evidence from the German retail gasoline market. *CESifa Warking Paper;* Na. 8521.

Calvano, E., Calzolari, G., Denicolo, V. y Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review, 110*(10), pp. 3267-97.

Cartea, Á., Chang, P. y Penalva, J. (2022). *Algorithmic Collusion in Electronic Markets: The Impact of Tick Size.* Available at SSRN 41 5954.

Cartea, Á., Chang P., Penalva J. y Waldon, H. (2022). *The Algorithmic Learning Equations: Evolving Strategies in Dynamic Games.* Available at SSRN 4175239.

Competition & Markets Authority. (2018). Pricing Algorithms. Available at https://www.gov.uk/government/publications/pricing-algorithms-research-collusion-and-personalised-pricing

Competition & Markets Authority. (2021). Algorithms: How they can reduce competition and harm consumers. Available at https://www.gov.uk/government/publications/algorithms-how-they-can-reduce-competition-and-harm-consumers

European Commission. (2017). Algorithms and Competition. Speech by Commissioner Margrethe Vestager at Bundeskartellamt 18th Conference on Competition, Berlin.

Harrington, J. E. (2018). Developing competition law for collusion by autonomous artificial agents. *Journal of Competition Law & Economics, 14*(3), pp. 331-363.

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

Ohlhausen, M. K. (2017). Should We Fear The Things That Go Beep In the Night"! Some Initial Thoughts on the Intersection of Antitrust Law and Algorithmic Pricing. Remarks fram the Cancurrences Antitrust in the Financial Sectar Canference. New York, NY. May 23, 2017.