

EQUILIBRIOS EN UNA ECONOMÍA DE MERCADO CON INFORMACIÓN IMPERFECTA Y RACIONALIDAD LIMITADA

Matías Daniel Fernandez (UNLP)* y Juan Sebastián Ivars (UNLP)**
daniel.mza@gmail.com, ** jsivars@gmail.com

ABSTRACT

El Primer y Segundo Teorema de Bienestar suponen la existencia de información perfecta en los precios de mercado, es decir, que éstos contienen toda la información relevante para la toma de decisiones y son conocidos por todos los agentes. Sin embargo, debido a que este supuesto es muy fuerte, se han desarrollado diversas corrientes de la economía que analizan el comportamiento de los agentes ante la existencia de asimetrías en la información. En el presente trabajo se busca realizar un abordaje que permita la existencia de estas asimetrías, pero además, considere que los agentes tienen sesgos en su toma de decisiones; en consecuencia desarrollan un mecanismo de aprendizaje sobre los precios del mercado. Este proceso define sus roles de compradores y vendedores, a fin de maximizar su utilidad. Para ello, se utiliza como base un modelo de sistemas complejos conocido como naming game, que permite implementar una primera etapa de aprendizaje sin costo, a partir de la interacción aleatoria de los agentes, y una segunda etapa, donde los mismos toman decisiones, participan del mercado y obtienen nueva información a través de sus experiencias particulares. Mediante este mecanismo de prueba y error los agentes alcanzan un equilibrio estable. Los primeros resultados son consecuentes con los enfoques tradicionales: la información que se tenga sobre los precios es muy importante para alcanzar un equilibrio paretiano. No obstante, el modelo también muestra que los agentes pueden construir poder de mercado por conocer tempranamente su valoración relativa tempranamente.

Clasificación JEL: D23, D4, D83, D7

Introducción

En el presente trabajo se analiza cómo se puede interpretar el juego de *Naming game*¹ para una economía de intercambio puro. En este tipo de economías se analizan circunstancias mediante las cuales diversos agentes (países, individuos, o sectores económicos) deciden intercambiar bienes, con otros con el objetivo de lograr satisfacer un mayor índice de necesidades.

El juego de *Naming game* permite analizar un mecanismo de aprendizaje mediante el cual cada individuo se encuentra dotado de un conjunto de palabras, y en el proceso de interacción con otros individuos (cuya dotación de palabras es diferente), éstos pueden obtener consenso de que un objeto se denota de una determinada manera, o aprender otro conjunto de palabras mediante las cuales se denota el mismo objeto. Finalmente, este juego desemboca en un proceso de convergencia mediante el cual todos los individuos aprenden que un mismo objeto tiene el mismo significado para todos los individuos².

En el presente trabajo se realiza una adaptación del *Naming game* con el objetivo de intentar analizar una economía en el estadio más primitivo del intercambio, en donde el mismo se produce a través de trueque entre bienes. De esta manera, cada agente lleva al mercado el conjunto de bienes que posee (dotaciones), con un conjunto de necesidades que sabe que tiene (demanda). En este caso, cuando al producirse el encuentro bilateral el intercambio es exitoso, cuando ambos tienen conjeturas coherentes de precios (uno de compra y el otro de venta). Por otro lado, el intercambio fracasa cuando alguno de esos dos individuos (o ambos) tiene conjeturas extremas sobre los precios, y mediante este proceso los individuos acotan sus conjeturas.

El artículo se encuentra estructurado en cuatro secciones. En la primera sección se presenta la motivación económica del estudio de una economía de mercado a través del *Naming Game*. En la segunda, se presenta el modelo teórico, el *Benchmark* utilizado para contrastar los resultados y las simulaciones. Finalmente se presentan las conclusiones, y en el apéndice la sintaxis desarrollada en Python.

Motivación

Los análisis sobre economías de intercambio simple consideran que los agentes tienen un conocimiento pleno sobre la información relevante del mercado, y que además se encuentra disponible para el procesamiento de los agentes en cada momento que este mismo lo requiera. A su vez, considera que estos últimos realizan sus elecciones a través de comportamientos racionales que les permiten tomar decisiones óptimas a través de la información requerida. Si bien se han desarrollado marcos analíticos que contemplan los problemas informativos propios de la conducta del hombre y las organizaciones³. Estos abordajes se encuentran sujetos a la noción de agentes que son racionales, y toman decisiones óptimas de acuerdo al conjunto informativo (limitado) que poseen.

No obstante, en el presente trabajo se considera que los agentes presentan limitaciones en esas capacidades de cálculo. Se considera que los mismos tienen una conjetura asociada a un determinado evento, y a partir de un mecanismo de aprendizaje imperfecto pueden ir mejorando sus conjeturas respecto del mismo. En particular, se considera que ese método de aprendizaje está regido por las reglas de *Naming Game*. Es decir, se presenta un entorno en el cual los agentes tienen restricciones en el conjunto de información, pero además tienen limitaciones en sus capacidades cognitivas⁴.

La idea que subyace al análisis es que los agentes no son lo suficientemente sofisticados para tomar decisiones óptimas, pero lo interesante es identificar como a pesar de eso solucionan sus problemas cotidianos. Como los entornos de información imperfecta suelen generar imperfecciones de mercado, un aspecto importante a analizar es si los mecanismos de aprendizaje pueden proveer mecanismos

¹Basado en el artículo de Lu et al. (2009). Además, existen otros artículos como por ejemplo el de Baronchelli (2011) que analizan el rol de las conversaciones y las trasmisión de mensajes bajo el modelo de *Naming Game* con el objetivo de estudiar el proceso de negociación basado en la memoria; y el de Gao et al. (2013) que utilizan una extensión de este modelo para investigar el consenso en redes auto organizadas.

²Este proceso no implica que siempre la palabra mediante la cual denotamos al objeto sea la misma en cada simulación, sino que finalmente el proceso convergerá a una palabra que puede variar según las condiciones iniciales.

³Es importante destacar que el último Premio Nobel fue otorgado en conjunto Oliver Hart y Bengt Holmström por sus análisis sobre teoría de contratos que tienen como objetivo analizar los problemas de información en distintas circunstancias.

⁴Se utilizó también como fuente bibliográfica el material de clases Heymann et al. (2013)

que se acerquen a condiciones de decisión óptimas de los agentes, que les permitan sortear esas dificultades.

El Modelo

Descripción

En primer lugar, cada uno de los agentes se encuentra dotado con una unidad de un bien que se considera homogéneo⁵. Cada uno de los individuos posee una valoración particular de ese bien, pero no saben el precio que puede tener ese bien en el mercado. Por lo tanto, tampoco pueden definir su posición en el mercado, ya sea como comprador, vendedor o simple consumidor del bien del cual poseen.

A partir de allí, los agentes aprenden de dos maneras diferentes, en primer lugar los agentes mejoran sus conjeturas de precios a través de conversaciones bilaterales entre ellos, este mecanismo no tiene costos para ellos (está basado en *Naming Game*). Mientras que el segundo mecanismo de aprendizaje se da a través de transacciones bilaterales de mercado, y lógicamente, el acceso al mercado tiene un costo asociado, tanto las transacciones exitosas como aquellas que no lo son. Finalmente, se supone que los agentes no obtienen información de los mercados de los cuales no forman parte.

Especificaciones del modelo

Setup

- a) Se trata de un modelo basado en agentes, donde cada uno tiene la posibilidad de interconectarse bilateralmente entre sí, de forma tal que pueden conversar e intercambiar entre cualquiera de ellos.
- b) Cada agente tiene una valoración fija v sobre el bien (si el agente no compra ni vende, suponemos que tiene una utilidad igual a esta valoración, por el simple hecho de quedarse con el bien en su poder. Esta valoración se distribuye uniformemente e incluye el costo de acceder el mercado. A su vez, posee una conjetura sobre el precio mínimo p^{min} y máximo p^{max} que puede tener el bien dentro del mercado. Por último, puede tener tres estados: “comprar”, “vender”, “no hacer nada”.

En la parte principal del algoritmo, se procesan las decisiones e interacciones que se producen entre los agentes ($N=10$) en cada una de las etapas propuestas, que se integran en un ciclo que denominamos día. El algoritmo computa los precios y las utilidades de cada uno de los agentes tras una determinada cantidad de **días**(30), y a su vez, cada uno de estas simulaciones se repite una determinada (**repeticiones** = 100) cantidad de veces, lo que nos permite analizar la distribución de los resultados. Las variables mencionadas son inicializadas al comienzo del algoritmo, conjuntamente con el **costo** (1) de acceso al mercado, así como también los espacios de distribución de precios (**espacio_precios**=50) y preferencias (**espacio_preferencias**=100).

- c) Cada agente tiene una preferencia como comprador sobre con qué vendedor realizar transacción (suponemos que si el kiosko favorito del agente está abierto, no hay motivos para probar otro lugar), que se actualiza a partir de las transacciones exitosas y los fracasos, y funciona como una lógica de reputación de los vendedores. De esta manera, la figura 4 que almacena los vectores de utilidades y preferencias muestra un elemento indispensable para el proceso

⁵También se realizó la simulación para el caso en el cual los individuos no tienen restricciones en las dotaciones, es decir, son capaces de vender tanto bienes como los oferentes se encuentren interesados en comprar. Bajo esta circunstancia se encuentra una fuerte concentración de la oferta en aquellos individuos que logran apropiarse primero de una buena reputación, generando mercados monopólicos u oligopólicos que concentran la oferta y obtienen los niveles más altos de utilidad. Los resultados no se analizarán porque implicaría hacer supuestos importantes sobre el nivel de producción, como por ejemplo que es perfectamente flexible. Se considera que la restricción en la dotación (dada la economía de intercambio) se puede asociar mejor a una economía productiva con límites estructurales. En todo caso, el modelo a probar, podría ser permitiendo un número mayor que uno, pero menor a infinito.

de intercambio dado que los vectores, dado que el vector de preferencias almacena el ranking de reputación que cada individuo arma sobre los vendedores⁶.

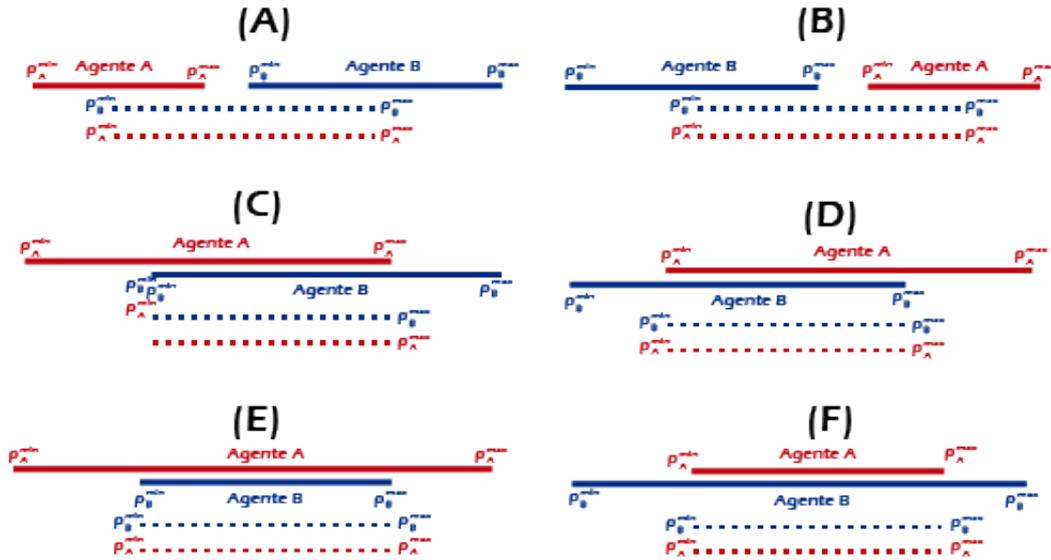
Timing

El algoritmo se repite una cantidad T de días. Cada día consta de tres etapas. Teniendo la tercer etapa (Intercambio), tres subetapas.

1. CONVERSACIÓN (basado en "Naming game")

Aquí cada uno de los agentes escoge otro agente al azar y actualiza la conjetura sobre los precios mínimos y máximos⁷. Ambos escenarios se detallan a través de la explicación de la siguiente imagen:

Figura 1: Intervalos de precios



Existe una intersección entre los intervalos: los agentes acuerdan que dicha intersección es el intervalo de precios que más se ajusta al mercado. En la figura correspondiente a Intervalos de precios, se corresponde con los casos (C), (D), (E) y (F). En los casos (C) y (D), los agentes encuentran en las conjeturas del otro un punto de referencia creíble que le permite actualizar sus propias expectativas. De esta manera, la nueva conjetura de compromiso resulta de la intersección de ambos intervalos. Por último, los casos (E) y (F) son un caso especial de (C) y (D), en tanto uno de los intervalos contiene al otro y, por lo tanto, ambos agentes encuentran dos puntos de referencia creíbles que permiten que uno de los agentes mantenga sus expectativas iniciales y sea imitado directamente por el segundo agente.

No existe una intersección entre los individuos: los agentes acuerdan que ambos tienen conjeturas muy extremas, y acuerdan un intervalo del tipo $(\min\{p^{max}\}; \max\{p^{min}\})$.

⁶el campo de Economía Conductual, este tipo de comportamiento de los agentes se corresponde con un sesgo basado en inferencias muestrales. Particularmente, el estudio de las decisiones estratégicas que llevan adelante las empresas para aprovecharse de este sesgo muestral se denomina Obfuscation (oscurecimiento), al respecto se puede consultar Spiegler (2014) y Spiegler (2006). En este apartado, no se analizarán desarrollos de este tipo, sólo se considerará que las agentes tienen ese sesgo, pero no las acciones estratégicas que puedan llevar adelante las firmas para sacar provecho.

⁷En esta instancia se considera que los agentes discuten sobre precios y no sobre valoraciones, porque los precios tienen magnitudes sujetas a discusión, mientras que las valoraciones si bien son inherentes a los agentes subyacen en un espacio más profundo de la subjetividad. Kahneman (2012) en la página 138 describe un experimento realizado en Alemania, en el cual se le pregunta a los participantes respecto de su nivel de felicidad asociado al número de citas que ha tenido en el mes, y la conclusión del experimento muestra la no independencia en el orden de las preguntas. Lo cuál permitiría argumentar que no existen posiciones determinantes en las valoraciones subjetivas. Es probable que los agentes vayan aprendiendo también cual es su verdadera valoración de un mismo bien

Este comportamiento de aprendizaje no es perfectamente flexible, es decir, si bien ambos se percatan que tienen preferencias extremas, actualizan sus conjeturas sujetos a los valores de referencia⁸ que tenían en la conjetura inicial. En los casos (A) y (B), los agentes poseen conjeturas significativamente diferentes, por lo que optan por una solución de compromiso, en las cuales cada uno de ellos descartan las posiciones más extremas de las conjeturas y hacen foco en la media de sus conjeturas. De esta manera el precio mínimo y máximo de las expectativas de compromiso se corresponden con la mínima y máxima media, respectivamente⁹.

En esta etapa, cada agente adopta un rol activo al menos una vez por día, lo que nos asegura que no haya nadie que no reciba información para actualizar sus conjeturas. Sin embargo, el componente azaroso en el rol pasivo, que es aquel que escucha y responde en la conversación, permite que algunos agentes, en un mismo día, interactúen más veces que otros.

2. DECISIÓN:

Tras la etapa de conversaciones, cada agente está en condiciones de definir su rol en el mercado. Para evitar complejizar innecesariamente el modelo, se supone que los agentes no tienen restricciones ni fricciones para ofrecer el bien, por lo que pueden ofrecerlo el día que tienen mejores condiciones como vendedores que como consumidores. A su vez, si consideran que no tienen información suficiente para tomar una decisión, pueden optar por esperar un día más y una nueva ronda de conversaciones.

Si la valoración del agente es mayor o igual al precio máximo que él espera que se de en el mercado, el agente decide que su mejor opción es comprar otro bien. Como la demanda es del tipo TIOLI (take it or leave it), define además que comprará el bien siempre que el precio sea menor igual al precio máximo esperado¹⁰.

De la misma manera, si la valoración del agente es menor al precio mínimo esperado, define que su mejor opción es la de vender el bien. En este punto, además, debe escoger dentro del intervalos de precios posibles cuál será la mejor opción para obtener una ganancia. Con ánimos de simplificar el modelo, y poner en evidencia que los agentes pueden recurrir a una serie heterogénea de criterios para resolver el mismo problema, el precio de venta se define aleatoriamente mediante una distribución uniforme de los precios esperados por el vendedor. Por último, si la valoración del agente pertenece al intervalo de conjetura, carece de incentivos, tanto para la compra, como para la venta. En este sentido, su mejor opción será esperar un día más, privilegiando los mecanismos de aprendizaje sin costos que provee la etapa de conversaciones a los mecanismos de aprendizaje costosos del intercambio. El resultado final de esta etapa es la determinación de un vector de demandantes y precios de demanda, por un lado, y de un vector de oferentes y precios de oferta, por el otro.

3. INTERCAMBIO

En esta etapa se producen los intercambios entre oferentes y demandantes, así como también se realizan las mercancías, se actualizan las conjeturas sobre los precios mínimos y máximos y se adquiere un conocimiento más acabado sobre el funcionamiento del mercado. Como se mencionó anteriormente, esta etapa es costosa, y, por lo tanto, no todos los agentes están deseosos de participar de los intercambios si no tienen información previa que justifique la posibilidad de conseguir mayores niveles de utilidad. Esta etapa, consecuentemente consta de una serie de sub-etapas que procederemos a describir someramente:

⁸Kahneman and Tversky (1974) argumentan a través de su teoría de las perspectivas que el rol del punto de referencia es clave, resultado que presenta un fuerte contraste con la teoría de utilidad esperada, en el que las elecciones sólo dependen de las probabilidades esperadas de los estados finales. Un artículo que presenta avances formales respecto a esto es el de Koszegi (2006)

⁹Supongamos que tenemos dos individuos, el primer con un rango de precios [1,5] y el segundo [6,10], así, cada uno de ellos toma su valor medio del rango y los consideran como pmin y pmax, en este caso ambos agentes actualizarían su conjetura de precios en el rango [3,8].

¹⁰Esta idea está asociada con equilibrios psicológicos, en los cuales los agentes están interesados en los motivos detrás de la acción del otro jugador. Si advierten que la actitud es engañosa ellos prefieren ser dañinos (en este caso, no comprar a pesar de que le otorga utilidad positiva). Un artículo que analiza este tipo de comportamiento es el de Rabin (1993)

Matching

Aquí, cada uno de los demandantes, si hay al menos un oferente, decide a quién comprarle en función de su experiencia previa dentro del mercado, que se formaliza a partir de un *ranking* de preferencias. Es decir, mientras que los compradores no concurren a un oferente con un precio tan alto que no se realiza la transacción, seguirán frecuentando el mismo lugar, siempre y cuando el mismo mantenga su rol de vendedor, y no opte por un cambio de posición en el mercado. En otras palabras, los agentes privilegian la explotación de situaciones favorables a la exploración de nuevas alternativas. Si para un agente, hay más de un vendedor que tiene el mismo ranking, el comprador escogerá entre ellos al azar con una distribución uniforme. Lo que se observa es que, durante los primeros días, este factor aleatorio y el hecho que los agentes pueden no ofrecer permanentemente¹¹, los agentes exploran un pequeño conjunto de alternativas y no se quedan necesariamente con el primer afortunado.

Intercambio

Hasta este momento, los vendedores no conocen el precio de demanda de los demás agentes, así como tampoco los compradores conocen el precio de venta de los demás. Sólo cuando la relación de intercambio se ha establecido, es decir, cuando el comprador concurre al local de vendedor, estos valores se conocen. Aquí, si el precio propuesto por el vendedor es menor o igual al máximo precio esperado por el comprador se produce el intercambio. El vendedor aprende que precios menores al que se realizó la transacción no le convienen, por lo que actualiza su precio máximo. El comprador aprende que precios mayores al que compró tendrían la intención de apropiarse de su excedente, y como se pudo observar con anterioridad, ante esta situación actualiza la conjetura sobre su precio mínimo¹².

En cambio, cuando el precio propuesto por el vendedor es mayor al máximo precio esperado por el comprador el intercambio no se realiza. Sin embargo, esta situación aún garantiza a los agentes información valiosa para actualizar sus conjeturas. De tal manera, el vendedor tiene la posibilidad de preguntar al cliente con qué expectativas había concurrido al local, y actualizar su precio mínimo a la baja. Consecuentemente, el comprador, al ver el precio de venta, actualiza su precio máximo hacia el alza, en tanto considera que sus expectativas pueden haber sido excesivamente optimistas y que vale la pena volver a confrontarlas en la ronda de conversaciones.

Actualización de Preferencias

Un intercambio exitoso además genera un aumento en la preferencia del comprador por concurrir al local del vendedor, el cual está correlacionado por la diferencia que existe entre el precio esperado y el precio de venta. En cambio, un intercambio infructuoso es computado mediante una caída en la preferencia por el vendedor, proporcional a la diferencia entre el precio esperado y el precio de venta.

Benchmark

En este apartado se presenta el desarrollo de un modelo que podría ser considerado como el *Benchmark* para el modelo propuesto en el trabajo. Partiendo del resultado propuesto por Myerson (1979) en el cual si existe un mecanismo cuya función de resultados particular $y(\cdot)$ puede ser implementada por el diseñador, entonces, existe un mecanismo directo de revelación en el cual se implementa $y(\cdot)$ como parte de un equilibrio donde los agentes revelan su verdadera información, se puede asegurar que la Subasta de Segundo Precio es un mecanismo coherente para proponerlo como punto de referencia. Este modelo de Subastas de Segundo Precio con múltiples bienes, es un

¹¹El oferente puede haber optado por dejar de ser vendedor o puede haber establecido un vínculo transaccional con otro demandante.

¹²En caso que el precio de compra sea menor al precio mínimo esperado por el comprador, el agente además actualizará este valor, que contemplará la excelente transacción que logró realizar.

mecanismo particular de la familia de mecanismos Vickrey-Groves-Clarke (VCG) que cumplen lo mencionado.

En el modelo presentado se considera una totalidad de N agentes, por lo tanto se considera $N = 2$ agentes en cada grupo. Es decir, $N = 2$ individuos son vendedores, y los restantes compradores. Los individuos que deciden comprar el bien son los $N = 2$ que más valoran el bien, ya que perciben una utilidad mayor de la compra del bien, que del pago que les implica pagarlo. Mientras que los vendedores se considera a aquellos $N = 2$ que tienen la menor valoración del bien, y perciben una utilidad mayor por el pago que reciben.

De este modo, para una totalidad de 10 agentes, hay 5 compradores y 5 vendedores. Así, el bien se vende a la sexta mayor valoración (dado que los 5 que más valoran el bien se definen como compradores), la sexta mayor valoración corresponde con la segunda valoración más alta a la que el último de ellos compraría.

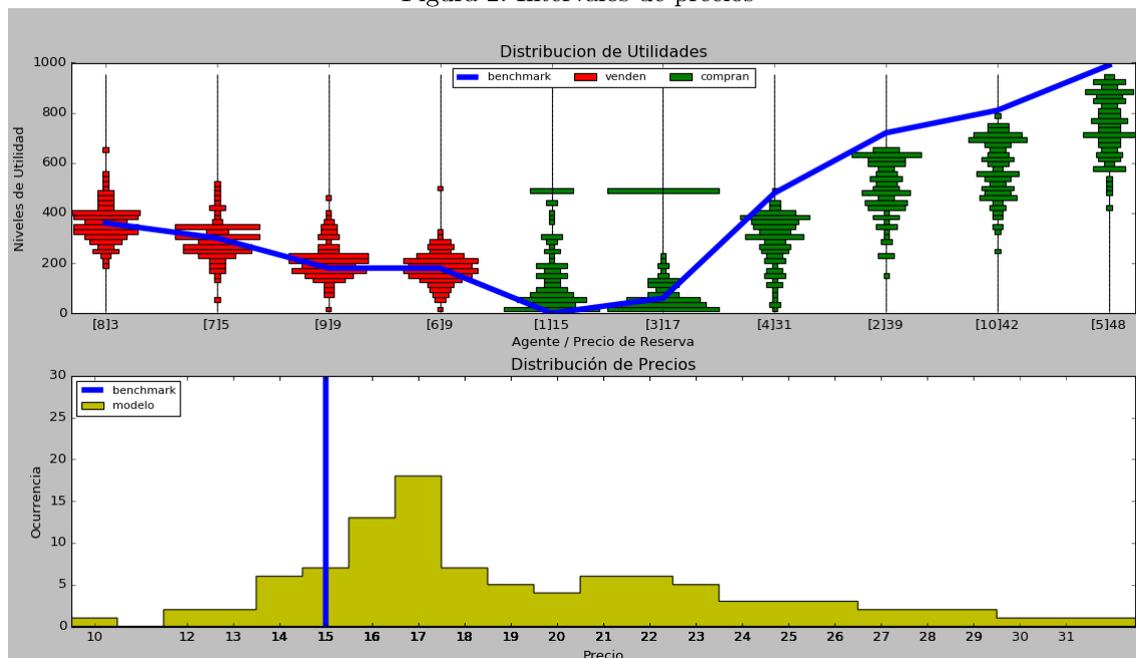
A continuación, se presentará en los gráficos los resultados del Benchmark en contraste con los resultados obtenidos por el modelo propuesto.

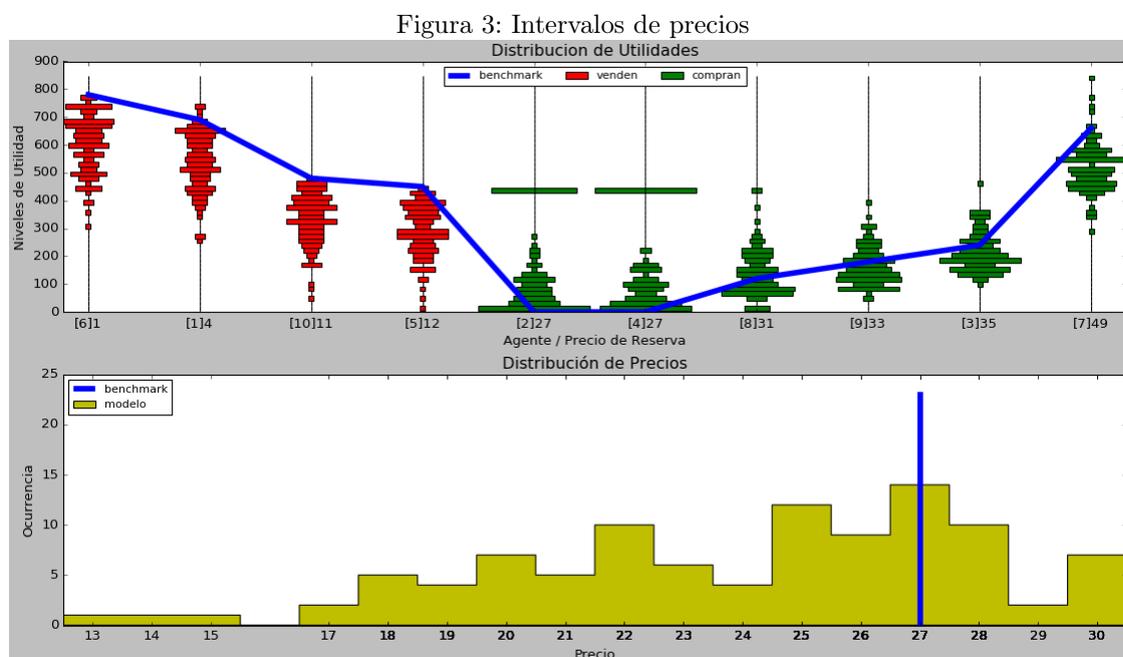
Simulación de resultados

Las principales salidas del programa versan sobre dos gráficos que muestran la distribución de utilidades de cada uno de los agentes y los precios de equilibrio, respectivamente. En el primer gráfico, en las abscisas ubicamos a cada uno de los agentes (marcados entre corchetes) y su valoración individual sobre el bien, mientras que en las ordenadas los niveles de utilidad que se alcanzan en cada uno de los 100 ciclos de 30 días cada uno.

De tal manera, encontramos la distribución de utilidades, agrupada en escalones de 20 unidades, para cada uno de los agentes. En rojo, aquellos que a lo largo de las simulaciones optaron predominantemente por ser vendedores, mientras que en verde los que optaron por ser compradores. El algoritmo permite mostrar en amarillo aquellos casos en que pudiera suceder que un agente predominantemente optara por esperar durante un ciclo, sin embargo, una primera observación que podemos realizar es que esto no sucede nunca, es decir, todos los agentes más temprano que tarde adquieren información suficiente sobre cuál es su rol óptimo dentro del mercado del bien. En azul, se ha dibujado una línea azul que une los niveles de utilidad que los agentes alcanzan en el benchmark, lo que permite comparar los resultados fácilmente.

Figura 2: Intervalos de precios





En el segundo ejemplo, en amarillo se observa la distribución de precios de equilibrio de cada uno de los ciclos, mientras que la línea azul marca el precio de equilibrio para la situación del *benchmark*. Una primera aproximación a los resultados muestra que precios por encima del *benchmark* son más frecuentes, aunque no es una tendencia que se sostenga para todos los conjuntos de valoraciones de los agentes.

Como puede apreciarse en la Figura 2, el precio de equilibrio más frecuente coincide con el *benchmark*, pero éste valor alcanza apenas un 15% de las ocurrencias, mientras que en la mayoría de los casos se dan precios de equilibrio por debajo del ya mencionado, lo que redundaría en una mayor cantidad de casos donde los compradores obtienen niveles de utilidad superiores a los que obtienen en el *benchmark* y donde, contrariamente, los vendedores se encuentran en una situación inferior.

Sin embargo, en las Figuras 3 y 4, se observa no sólo que el modelo no arroja como precio más frecuente el resultado del *benchmark*, sino que la mayor cantidad de ocurrencias se corresponden con precios de equilibrio superiores. De tal manera, el análisis de la distribución de utilidades de los agentes se encuentra en las antípodas de lo relatado en el párrafo anterior.

Sin embargo, vemos que esta situación no nos garantiza la predominancia de un rol sobre otro, sino que, dependiendo de la distribución inicial de valoraciones, podremos encontrar un caso como el de la Figura 3, donde predomina una situación de 6 vendedores y 4 compradores, mientras que en la Figura 4, predomina la situación de 4 vendedores y 6 compradores.

En la Figura 5, nuevamente el resultado más frecuente coincide con el *benchmark*, y a su vez no puede afirmarse que exista una tendencia de precios de equilibrio por encima ni por debajo de dicho valor. En este punto, lo que puede afirmarse es que, si bien predominan resultados donde el precio de equilibrio se encuentra por encima del observado en el escenario de *benchmark*, donde los vendedores alcanzan mayores niveles de utilidad en detrimento de los compradores; las simulaciones muestran que la existencia de valoraciones extremas, empujan el equilibrio en su dirección.

Sin embargo, hay una tipología de agente que casi siempre se ve beneficiado por la falta de información, y se trata del comprador (o vendedor) marginal, que en el *benchmark* obtiene una utilidad nula, dada que su valoración es la que finalmente determina el precio de equilibrio, impidiéndole la captación de excedentes. Este motivo produce que sea el agente más sensible a las distorsiones que se producen en el precio de equilibrio, en función del sendero que se construye a partir de los mecanismos de aprendizaje, lo que le permite obtener mayores niveles de utilidad que en la situación de *benchmark*.

Figura 4: Intervalos de precios

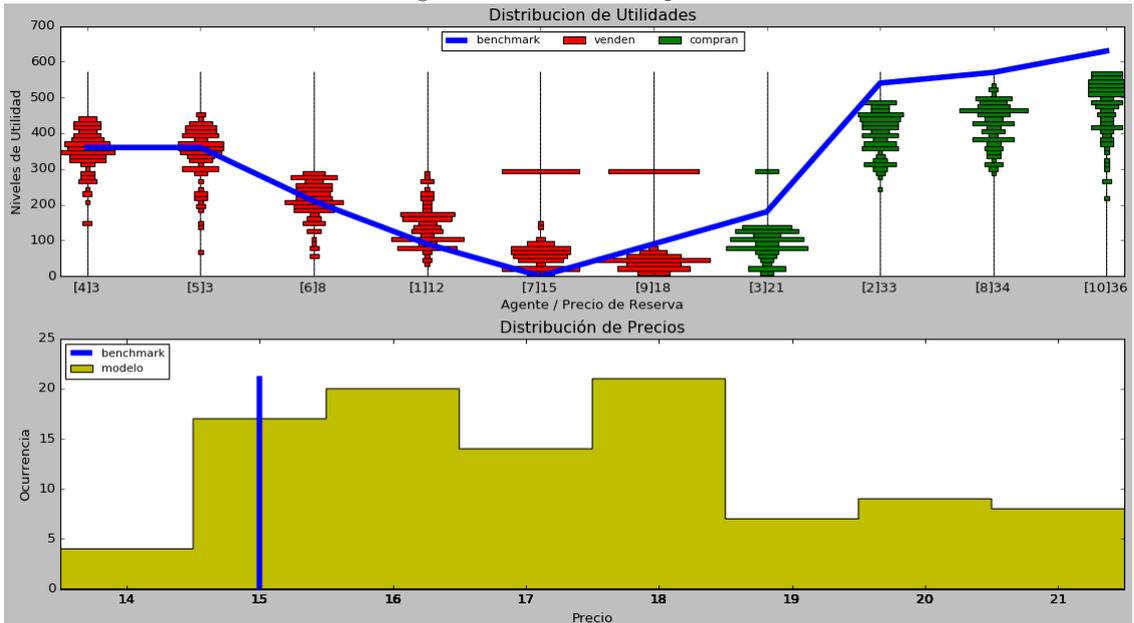
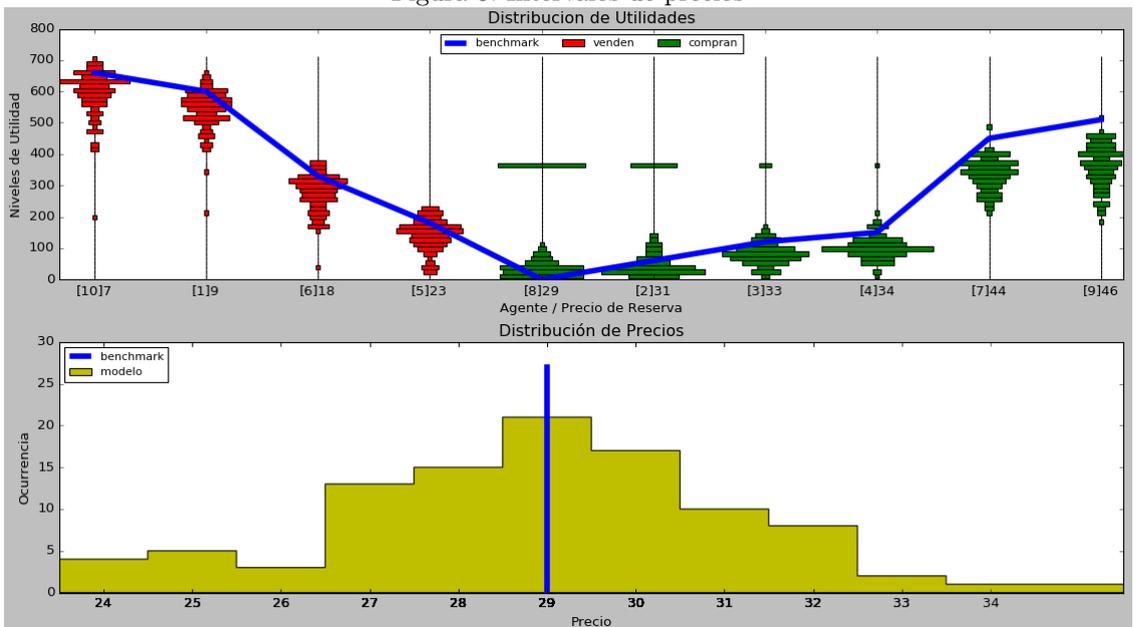


Figura 5: Intervalos de precios



Conclusiones

Como se mencionara, los análisis sobre economías de intercambio simple consideran que los agentes tienen un conocimiento pleno sobre la información relevante del mercado, y se encuentra siempre disponible para el procesamiento de los agentes. Además, considera que no existen problemas de coordinación entre los agentes que eligen racionalmente lo que les permiten tomar decisiones óptimas con la información disponible. No obstante, en múltiples circunstancias estos agentes se encuentran limitados a conjuntos de información acotados y a reglas cognitivas que requieren comportamientos heurísticos.

Cuando los límites a los cuáles se enfrentan los agentes son informativos, los resultados a los que ha concluido la literatura se asocian a equilibrios sub óptimos de Second Best, en los cuáles aquellos individuos que tienen una mayor (menor) valoración obtienen beneficios mayores al comprar (vender) dado que el vendedor (comprador) no puede diseñar un mecanismo que le permita separar perfectamente las valoraciones de los individuos. Este mecanismo, produce resultados de *Non distortion at the bottom (top)* para los compradores (vendedores), que implican el pago de una renta informativa a los individuos que se pueden apropiarse de los beneficios de la información privada.

No obstante, cuando adicionalmente se le agregan límites cognitivos se observan algunas particularidades no triviales a destacar. En primer lugar, que a pesar de incorporar un aspecto que produce distorsiones respecto del mecanismos de asignación original, existe una trayectoria de equilibrio que deja a los individuos satisfechos ante los resultados del mercado. Dicho de otra manera, los agentes logran sortear las dificultades cognitivas e informativas y solucionar sus problemas de mercado transaccionalmente. En segundo lugar, la trayectoria de equilibrio que genera las limitaciones cognitivas sumado a los problemas informativos tienen resultados diferentes a los mencionados sólo en los casos que hay límites de información. En esta trayectoria, existe también una divergencia en los consumidores marginales los cuáles con límites informativos tendrían beneficios nulos y en estas circunstancias pueden apropiarse de beneficios, con lo cual la distorsión se genera en ambos extremos.

Finalmente, se pueden destacar dos resultados que se desprenden de los mencionado. En primer lugar, se destacan implicancias distributivas, si bien en el primer caso el individuo de mayor (menor) valoración obtiene utilidades más altas y explota los problemas informativos de la valoración privada con la percepción de una renta informativa. Ahora los que gozan de mayores beneficios en términos relativos, son los consumidores marginales quienes tenían beneficio nulo, soportado principalmente por los estratos medios de valoración de compra y venta. En segundo lugar, se puede notar que ahora los agentes dejan de tener interés en “el verdadero precio de equilibrio” y su preocupación principal se focaliza en que la conjetura sobre el precio de mercado no difiera demasiado de la conjetura de los demás, lo que resulte transacciones sin éxitos en el mercado dado que aquellos que individuos que no pueden realizar transacciones del mercado (en varias oportunidades) obtienen beneficios menores. Como conclusión, el complemento de ambas restricciones a los agentes individuales producen resultados diferentes y significativos a los trabajados con anterioridad.

Apéndice

```
1
2 #importar libreria para generacion de numeros aleatorios
3 import random
4 #importar libreria para manipulacion de matrices
5 import numpy as np
6 #importar libreria para graficos
7 import matplotlib.pyplot as py
8
9 N = 10
10 repeticiones = 100
11 #vector de agentes: precio de reserva, precio minimo, precio maximo, estado
12 agentes = np.zeros((N+1,5),dtype=np.int)
13 #vector de preferencias: cuanto prefiere cada agente comprarle a otro
```

```

14 ranking = np.zeros((N+1,N+1),dtype=np.int)
15 #vector de demanda: demandante, precio de compra, potencial vendedor
16 demanda = np.zeros((N+1,4),dtype=np.int)
17 #vector de oferta: oferente y precios de venta
18 oferta = np.zeros((N+1,3),dtype=np.int)
19 #vector auxiliar para guardar mejores oferentes
20 oferentes = np.zeros((N+1),dtype=np.int)
21 #vector de utilidades para los agentes
22 utilidad = np.zeros((N+1),dtype=np.int)
23 #vector de valoraciones (benchmark)
24 valoraciones = np.zeros((N+1),dtype=np.int)
25 #vector de distribucion de utilidades
26 utilidades = np.zeros((N+1,repeticiones),dtype=np.int)
27 #vector de distribucion de precios, cantidad de vendedores y compradores
28 precios = np.zeros((repeticiones),dtype=np.int)
29 #vector de utilidades para el caso benchmark
30 utilidades_benchmark = np.zeros((N+1),dtype=np.int)
31
32 #inicializa costo, precio maximo del bien y valor inicial de preferencias
33 costo=1
34 espacio_precios=50
35 espacio_preferencias=100
36 #cantidad de dias que se repite la secuencia
37 dias = 30
38
39 #inicializa valores de reserva
40 for individuo in range(1,N+1):
41     agentes[individuo,1] = random.randrange(espacio_precios)+costo
42
43 #inicializacion de cada ciclo
44 def inicializacion():
45     ##inicializa vector de preferencias
46     for j in range(1,N+1):
47         for k in range(1,N+1):
48             ranking[j,k] = espacio_preferencias
49
50     ##inicializa vector de agentes
51     for individuo in range(1,N+1):
52         agentes[individuo,2] = agentes[individuo,1]
53         agentes[individuo,3] = agentes[individuo,1]
54         agentes[individuo,4] = 0
55
56     ##inicializa vector de utilidades
57     for individuo in range(1,N+1):
58         utilidad[individuo]=0
59
60 #funciones que permitieron visualizar las matrices durante la codificacion
61 def escribir_agentes():
62     print("AGENTES")
63     for individuo in range(1,N+1):
64         print("[",individuo,"]: \t",end="(")
65         for j in range(1,3):
66             print(agentes[individuo,j],end=" \t ")
67         print(agentes[individuo,3],end=") \t - ")
68         if agentes[individuo,4] == 6:
69             print("vendedor")
70         elif agentes[individuo,4] == 9:
71             print("comprador")
72         else:
73             print("espera")
74     print("")

```

```

75
76 def escribir_demanda():
77     print("DEMANDA")
78     for individuo in range(1,demanda[0,1]):
79         print(demanda[individuo,1],demanda[individuo,2],demanda[individuo
80             ,3],end=" ")
81     if demanda[0,1]>0:
82         print(demanda[demanda[0,1],1],demanda[demanda[0,1],2],demanda[
83             demanda[0,1],3])
84
85 def escribir_oferta():
86     print("OFERTA")
87     for individuo in range(1,oferta[0,1]):
88         print(oferta[individuo,1],oferta[individuo,2],end=" ")
89     if oferta[0,1]>0:
90         print(oferta[oferta[0,1],1],oferta[oferta[0,1],2])
91     print("")
92
93 def escribir_ranking():
94     print("PREFERENCIAS")
95     for individuo in range(1,N+1):
96         print("[",individuo,"]: ",end="\t")
97         for j in range(1,N+1):
98             print(ranking[individuo,j],end="\t")
99         print("")
100     print("")
101
102 def escribir_utilidades():
103     print("UTILIDADES")
104     for individuo in range(1,N+1):
105         print("[",individuo,"]: \t",utilidad[individuo])
106     print("")
107
108 def escribir_benchmark(equilibrio):
109     print("BENCHMARK")
110     print("precio:\t",equilibrio)
111
112     print("vende:\t",end="")
113     for individuo in range(1,N//2 +1):
114         print("[",valoraciones[individuo],"]",end="\t")
115     print("")
116
117     print("compra:\t",end="")
118     for individuo in range(N//2 +1,N+1):
119         print("[", valoraciones[individuo],"]",end="\t")
120     print("")
121
122     print("")
123
124 #algoritmo de ordenamiento rapido: q-sort
125 def qsort(l,r):
126     i=l
127     j=r
128     x=agentes[valoraciones[(l+r)//2],1]
129
130     while i<=j:
131         while agentes[valoraciones[i],1]<x:
132             i+=1
133         while agentes[valoraciones[j],1]>x:

```

```

134         if i<=j:
135             aux = valoraciones[i]
136             valoraciones[i] = valoraciones[j]
137             valoraciones[j] = aux
138
139             i+=1
140             j-=1
141
142         if l<j:
143             qsort(l,j)
144         if r>i:
145             qsort(i,r)
146
147     #benchmark
148     def benchmark():
149
150         ##inicializa vector de valoraciones
151         for individuo in range(1,N+1):
152             valoraciones[individuo] = individuo
153
154         ##ordena de menor a mayor
155         qsort(1,N)
156
157         precio_benchmark=agentes[valoraciones[N//2],1]
158         for individuo in range(1,N+1):
159             utilidades_benchmark[individuo-1] = abs(agentes[valoraciones[
160                 individuo],1] -precio_benchmark)*dias
161
162         return (precio_benchmark)
163
164     #actualizacion de conjeturas: mecanismo de aprendizaje sin costo (parloteo)
165     def actualizar_expectativas(i,e):
166         ##p_min(agente A)>p_max(agente B): no se intersectan
167         if agentes[i,2]>=agentes[e,3]:
168             agentes[e,2]=(agentes[e,3]+agentes[e,2]+1)//2
169             agentes[i,3]=(agentes[i,2]+agentes[i,3])//2
170             agentes[i,2]=agentes[e,2]
171             agentes[e,3]=agentes[i,3]
172         ##p_max(agente A)<p_min(agente B): no se intersectan
173         elif agentes[i,3]<agentes[e,2]:
174             agentes[i,2]=(agentes[i,3]+agentes[i,2]+1)//2
175             agentes[e,3]=(agentes[e,2]+agentes[e,3])//2
176             agentes[e,2]=agentes[i,2]
177             agentes[i,3]=agentes[e,3]
178         else:
179             ###p_min(agente A)<p_min(agente B): P_MIN = p_min(agente B)
180             if agentes[i,2]<agentes[e,2]:
181                 agentes[i,2]=agentes[e,2]
182             ###P_MIN = p_min(agente A)
183             else:
184                 agentes[e,2]=agentes[i,2]
185             ###p_max(agente A)>p_max(agente B): P_MAX = p_max(agente B)
186             if agentes[i,3]>agentes[e,3]:
187                 agentes[i,3]=agentes[e,3]
188             ###P_MAX = p_max(agente A)
189             else:
190                 agentes[e,3]=agentes[i,3]
191
192     #etapa de conversacion
193     def conversacion():
194         for individuo in range(1,N+1):

```

```

194     ##escoger nodos al azar
195     escucha = individuo
196     while escucha == individuo:
197         escucha = random.randrange(N)+1
198
199     ##actualiza las expectativas sobre el precio
200     actualizar_expectativas(individuo,escucha)
201
202 #etapa de decision
203 def decision():
204     ##inicializo oferta y demanda
205     oferta[0,1] = 0
206     demanda[0,1] = 0
207     for individuo in range(1,N+1):
208         ##decido si comprar, vender o esperar
209         if agentes[individuo,1] < agentes[individuo,2]:
210             ###vendo
211             agentes[individuo,4] = 6
212             oferta [0,1] +=1
213             oferta[oferta[0,1],1] = individuo
214             oferta[oferta[0,1],2] = random.randrange(agentes[individuo,3] -
215                 agentes[individuo,2]+1)+agentes[individuo,2]
216         elif agentes[individuo,1] >= agentes[individuo,3]:
217             ##compro
218             agentes[individuo,4] = 9
219             demanda[0,1] +=1
220             demanda[demanda[0,1],1] = individuo
221             demanda[demanda[0,1],2] = agentes[individuo,3]
222         else:
223             ##consumo
224             agentes[individuo,4] = 0
225
226 #funcion para aleatorizar el vector de demanda
227 def mezclar():
228     ##Aleatoriza el vector de demanda
229     for individuo in range(1,demanda[0,1]+1):
230         otro = random.randrange(demanda[0,1])+1
231
232         ###intercambia agentes
233         aux = demanda[individuo,1]
234         demanda[individuo,1] = demanda[otro,1]
235         demanda[otro,1] = aux
236
237         ###intercambia el valor de los precios
238         aux = demanda[individuo,2]
239         demanda[individuo,2] = demanda[otro,2]
240         demanda[otro,2] = aux
241
242 #etapa de intercambio
243 def intercambio(compra,vende):
244     precio_venta = oferta[vende,2]
245     precio_esperado =demanda[compra,2]
246
247     ##hay intercambio
248     if precio_esperado >= precio_venta:
249         ###actualizacion de utilidades
250         utilidad[demanda[compra,1]] += agentes[demanda[compra,1],1] -
251             precio_venta
252         utilidad[oferta[vende,1]] += precio_venta - agentes[oferta[vende
253             ,1],1]
254     ###buena transaccion: aumento en el nivel de preferencias

```

```

252 ranking[demanda[compra,1],oferta[vende,1]] *= 1 + (precio_esperado
    - precio_venta) / precio_esperado
253 ###el comprador reduce su precio maximo
254 agentes[demanda[compra,1],3] = precio_venta
255 ###el comprador chequea su precio minimo
256 if agentes[demanda[compra,1],2] > precio_venta:
257     agentes[demanda[compra,1],2] = precio_venta
258 ###el vendedor aumenta su precio minimo
259 agentes[oferta[vende,1],2] = precio_venta
260 ###se elimina de la oferta la unidad vendida
261 oferta[vende,1] = oferta[oferta[0,1],1]
262 oferta[vende,2] = oferta[oferta[0,1],2]
263 oferta[0,1] -=1
264
265 ##no hay intercambio
266 else:
267     ###actualizacion de utilidades
268     utilidad[demanda[compra,1]] -= costo
269     utilidad[oferta[vende,1]] -= costo
270     ###NO transaccion: caida en el nivel de preferencias
271     ranking[demanda[compra,1],oferta[vende,1]] *= 1 - (precio_venta -
        precio_esperado) / precio_venta
272     ##el comprador aumenta su precio maximo
273     agentes[demanda[compra,1],3] = precio_venta
274     ##el vendedor reduce su precio minimo
275     agentes[oferta[vende,1],2] = precio_esperado
276     ###se computa el intercambio infructuoso
277     demanda[compra,3] = -1
278
279 #Enlaza a cada demandante con un potencial vendedor segun sus preferencias
280 def matching():
281     ##para cada demandante:
282     for dem in range(1,demanda[0,1]+1):
283         maximo = 0
284         if oferta[0,1] == 0:
285             break
286         ###busco los oferentes de mayor preferencia
287         for of in range(1,oferta[0,1]+1):
288             if ranking[dem,of] > maximo:
289                 ###encuentra que existe un vendedor que prefiere mas
290                 maximo = ranking[dem,of]
291                 oferentes[0]=1
292                 oferentes[oferentes[0]]=of
293             elif ranking[dem,of] == maximo:
294                 ###encuentra que existe un vendedor que prefiere igual
295                 oferentes[0] +=1
296                 oferentes[oferentes[0]]=of
297
298         ###aleatoriza entre los vendedores de maxima preferencia
299         individuo = random.randrange(oferentes[0])+1
300         demanda[dem,3] = oferta[oferentes[individuo],1]
301         intercambio(dem,individuo)
302
303 #funcion para graficar los resultados
304 def graficar():
305     ##configuracion del grafico
306     py.figure(figsize=(16,10))
307
308     ##configuracion de la distribucion de utilidades
309     py.subplot(211)
310     bloques = 50

```

```

311     etiquetas = ["["+str(valoraciones[i])+"]"+str(agentes[valoraciones[i]
312                ],1)] for i in range(1,N+1)]
313     py.subplots_adjust(hspace=0.25)
314     py.title("Distribucion de Utilidades")
315     py.ylabel("Niveles de Utilidad")
316     py.xlabel("Agente / Precio de Reserva")
317
318     ##computacion de cada uno de los histogramas
319     rango = (np.min(utilidades),np.max(utilidades))
320     histogramas = [np.histogram(utilidades[i],range=rango, bins=bloques)[0]
321                  for i in range(1,N+1)]
322     topes = np.max(histogramas, axis=1)
323     frecuencia_maxima = np.max(topes)
324     separacion_x = np.arange(frecuencia_maxima,frecuencia_maxima*(N+2),
325                             frecuencia_maxima)
326     py.xticks(separacion_x,etiquetas)
327
328     ##distribucion espacial de cada uno de los histogramas
329     separacion_y = np.linspace(rango[0], rango[1],bloques+1)
330     centro = 0.5 * (separacion_y + np.roll(separacion_y,1))[:-1]
331     altura = np.diff(separacion_y)
332
333     ##grafica cada una de las distribuciones de utilidades
334     bandera = 0
335     for x, histograma, individuo in zip(separacion_x, histogramas, np.roll(
336         valoraciones,-1)):
337         simetria = x - 0.5 * histograma
338         if agentes[individuo,4] == 6:
339             if bandera // 4 == 1:
340                 py.barh(centro,histograma, height=altura,left=simetria,
341                        color="r")
342             else:
343                 bandera += 4
344                 py.barh(centro,histograma, height=altura,left=simetria,
345                        color="r",label="venden")
346         elif agentes[individuo,4] == 9:
347             if bandera % 4 // 2 == 1:
348                 py.barh(centro,histograma, height=altura,left=simetria,
349                        color="g")
350             else:
351                 bandera += 2
352                 py.barh(centro,histograma, height=altura,left=simetria,
353                        color="g",label="compran")
354         else:
355             if bandera % 2 == 1:
356                 py.barh(centro,histograma, height=altura,left=simetria,
357                        color="y")
358             else:
359                 bandera += 1
360                 py.barh(centro,histograma, height=altura,left=simetria,
361                        color="y",label="esperan")
362
363     ##grafica el nivel de utilidad de los agentes en el benchmark
364     linea_x = [separacion_x[i] for i in range (0,N)]
365     linea_y = [utilidades_benchmark[i] for i in range(0,N)]
366     py.plot(linea_x,linea_y,"b-",linewidth=5.0,label="benchmark")
367     py.legend(loc="upper center",ncol=4,fontsize="small")
368
369     ##configuracion de la distribucion de precios
370     py.subplot(212)
371

```

```

362     bloques = np.max(precios) - np.min(precios)
363     py.ylabel("Ocurrencia")
364     py.xlabel("Precio")
365     py.title("Distribucion de Precios")
366     py.xticks(precios+0.5,precios)
367     ##grafica la distribucion de precios
368     py.hist(precios,histtype="stepfilled",bins=bloques,color="y",label="
        modelo")
369
370     ##grafica el nivel de precios en el benchmark
371     rango = (np.min(precios),np.max(precios))
372     histograma = [np.histogram(precios,range=rango)[0]]
373     linea_x = [precio_benchmark+0.5 for i in range (0,1+np.max(histograma))
        ]
374     linea_y = [i for i in range (0,1+np.max(histograma))]
375     py.plot(linea_x,linea_y,"b-",linewidth=5.0,label="benchmark")
376
377     ##muestra el grafico
378     py.legend(loc="upper left",fontsize="small")
379     py.show()
380
381     #programa principal
382     inicializacion()
383     ##benchmark
384     precio_benchmark=benchmark()
385
386     #computar determinada cantidad de repeticiones
387     for ciclo in range(1,repeticiones+1):
388         ##inicializo el vector de agentes y preferencias
389         inicializacion()
390         ##cada ciclo tiene cierta cantidad de dias
391         for individuo in range (1,dias+1):
392             ###cada dia cuenta con las etapas descriptas
393             conversacion()
394             decision()
395             if (oferta[0,1]>0) and (demanda[0,1]>0):
396                 mezclar()
397                 matching()
398
399             ##al final de cada dia se computa el precio de equilibrio
400             precios[ciclo-1] = agentes [2,3]
401             # #y la distribucion de utilidades
402             for individuo in range (1,N+1):
403                 utilidades[individuo,ciclo-1] = utilidad[valoraciones[individuo]]
404
405     graficar()

```

Referencias

- Baronchelli, A. (2011). Role of feedback and broadcasting in the naming game. *Physical Review E*, 83(4):046103.
- Gao, Y., Chen, G., and Chan, R. H. (2013). Naming game on networks: let everyone be both speaker and hearer. *arXiv preprint arXiv:1310.4169*.
- Heymann, D., Perazzo, R., and Zimmermann, M. (2013). Economía de fronteras abiertas, exploraciones en sistemas sociales complejos. *Editorial Teseo-Udesa*.
- Kahneman, D. (2012). Pensar rápido, pensar despacio. *Editorial Debate*.
- Kahneman, D. and Tversky, A. (1974). Judge under uncertainty: heuristic biases. *Science*.
- Koszegi, B.; Rabin, M. (2006). A model of reference-dependent preferences. *Quarterly Journal of Economics*.
- Lu, Q., Korniss, G., and Szymanski, B. (2009). The naming game in social networks: Community formation and consensus engineering. *Journal of Economic Interaction and Coordination*.
- Myerson, R. B. (1979). Incentive compatibility and bargaining problem. *Econometrica*.
- Rabin, M. (1993). Incorporating fairness into game theory and economics. *The American Economic Review*.
- Spiegler, R. (2006). Competition over agents with boundedly rational expectations. *Theoretical Economics*.
- Spiegler, R. (2014). Bounded rationality and industrial organization. *Oxford University Press*.