

# Encontrar un equilibrio Walrasiano es fácil para un número de agentes fijos

Marcelo Gallardo

**PUCP**

Octubre 2024

- 1 Introducción
- 2 Revisión de literatura
- 3 Preliminares
- 4 El algoritmo
- 5 Extensiones

- ❶ Autores: [Federico Echenique](#) (UC Berkeley) y [Adam Wierman](#) (Caltech).
- ❷ [[Echenique and Wierman, 2012](#)].
- ❸ EC 12: Proceedings of the 13th ACM Conference on Electronic Commerce.
- ❹ Objetivo del artículo: computar en tiempo polinomial un  $\varepsilon$ —equilibrio Walrasiano con respecto al número de bienes en una economía y la inversa de  $\varepsilon$ .
- ❺ Aplicaciones en macroeconomía y finanzas donde el número de agentes está fijo y no es grande, mientras que el número de bienes es arbitrariamente grande.

- ① The computation of equilibrium prices: an exposition, de Herbert Scarf (Yale). Algoritmo para computar puntos fijos (Brouwer). [[Scarf, 1977](#)]
- ② The computation of fixed points and applications, de Michael J. Todd. Libro publicado en Springer. [[Todd, 1976](#)]
- ③ Combinatorial Algorithms for Market Equilibria from I - Computing in Games de Vijay V. Vazirani. Se presentan algoritmos combinatorios en tiempo polinómico para encontrar precios de equilibrio y asignaciones en el caso de utilidades lineales. [[Vazirani, 2007](#)]
- ④ Efficient Computation of Equilibrium Prices for Markets with Leontief Utilities<sup>1</sup> de Bruno Codenotti y Kasturi Varadarajan 2006. [[Codenotti and Varadarajan, 2006](#)].

---

<sup>1</sup>  $u(x) = \min\{\alpha_1 x_1, \alpha_2 x_2, \dots, \alpha_n x_n\}, \alpha_i > 0.$

- ① El mensaje es que, **para el caso general**, es complicado computar un equilibrio Walrasiano<sup>2</sup>.
- ② Settling the Complexity of Computing Two-Player Nash Equilibria de Chen et al 2007.  
Pre-print en [arXiv](#): incluso para funciones de utilidad que son lineales por tramos, cóncavas y separables, es complicado.
- ③ Lo mismo se prueba para funciones de utilidad Leontief (Bruno Codenotti y Kasturi Varadarajan 2006).
- ④ Hay un gap evidente entre la generalidad y la dificultad para computar un equilibrio Walrasiano. El artículo de Echenique y Wierman busca cerrar este gap.

---

<sup>2</sup>En los preliminares se definirá formalmente este concepto.

# Preliminares

# Preliminares

- 1 Notación.
- 2 Equilibrio General: economía de intercambio puro.
- 3  $\varepsilon$ —equilibrio Walrasiano.
- 4 Ejemplos sencillos.
- 5 Enfoque de Negishi.
- 6 Lema de Sperner.

- ❶  $X = \mathbb{R}_+^L$
- ❷ Relación de preferencias: relación binaria sobre  $X \times X$  (subconjunto tal que  $xRy \Leftrightarrow x \succeq y$ ).
- ❸ Transitiva y completa.
- ❹  $u(\cdot) : X \rightarrow \mathbb{R}$  representa a  $\succeq$  si y solo si

$$x \succeq y \Leftrightarrow u(x) \geq u(y).$$

- ❺  $\|\cdot\|_\infty$  es la norma del máximo:  $\|x\|_\infty = \max_{1 \leq i \leq L} \{|x_i|\}$ .
- ❻  $\|\cdot\|_1$  es la norma de la suma:  $\|x\|_1 = \sum_{i=1}^L |x_i|$ .
- ❼  $u \in C^1(X, \mathbb{R})$  si existe  $U$  abierto con  $X \subset U$ ,  $f : U \rightarrow \mathbb{R}$  diferenciable y con derivada continua, tal que  $f|_X = u$ .
- ❽  $Du(x)$  denota el gradiente de  $u$  en  $x$ .

La norma se restringe a  $\mathbb{R}_+^L$  de ser necesario.



Decimos que  $u$  (función de utilidad) es

① Monótona (a veces estrictamente monótona): si  $x \leq y$ ,  $x \neq y$  implica  $u(x) < u(y)$ .

② Cuasicóncava: si para cualesquiera  $x, y \in X$ ,  $x \neq y$  y  $\theta \in (0, 1)$ ,

$$u(x) \leq u(y) \implies u(x) \leq u(\theta x + (1 - \theta)y).$$

③ Cóncava: si para cualesquiera  $x, y \in X$ ,  $x \neq y$  y  $\theta \in (0, 1)$ ,

$$\theta u(x) + (1 - \theta)u(y) \leq u(\theta x + (1 - \theta)y).$$

④ Homotética: si para cualesquiera  $x, y \in X$  tales que  $u(x) = u(y)$ , entonces  $u(\alpha x) = u(\alpha y)$  para todo  $\alpha \in \mathbb{R}_+$ .

# Problema de maximización de la utilidad

Dado un ingreso  $I > 0$  y un vector de precios  $p \in \mathbb{R}_{++}^L$ , cada individuo resuelve

$$\begin{cases} \max & u(x) \\ \text{s. a} & p \cdot x \leq I \\ & x \geq 0. \end{cases}$$

- ① Cuando las preferencias son monótonas, el problema se resuelve sobre  $p \cdot x = I$ .
- ② Cuando  $\lim_{x_i \downarrow 0} \frac{\partial u}{\partial x_i} = \infty$ , entonces  $x^* > 0$ .
- ③  $u(x) = \prod_{i=1}^L x_i^{\alpha_i}$  con  $\alpha_i > 0$ , entonces  $x_i^* = \frac{\alpha_i I}{(\sum_{i=1}^L \alpha_i) p_i}$ .
- ④ La solución es invariante si se considera como función objetivo  $f(u(x))$  con  $f(\cdot)$  estrictamente creciente.

# Economías de intercambio puro

$$\mathcal{E} = \{\omega_i, u_i\}_{i=1}^n.$$

- ❶  $\omega_i \in \mathbb{R}_+^L$ .
- ❷  $u_i : \mathbb{R}_+^L \rightarrow \mathbb{R}$ .
- ❸  $L$  es el número de bienes en la economía.
- ❹  $n$  es el número de consumidores.

Se consideran economías de intercambio puro  $E$  tales que

- ❶  $\sum_{i=1}^n \omega_i \in \mathbb{R}_{++}^L$ .
- ❷  $u_i \in C^1$  cóncava, estrictamente monótona.
- ❸ Se cumplen las condiciones de Inada.
- ❹  $\forall x \in \mathbb{R}_+^{nL}$  con  $\sum_{i=1}^L \omega_i \geq \sum_{i=1}^L x_i$ ,  $u_i(x_i) \in [0, 1]$  (normalización). Simplemente dividir por  $\max_{1 \leq i \leq n} |u_i(\sum_{i=1}^n \omega_i)|$ .

# Equilibrio Walrasiano

## Definición

Sea  $\mathcal{E}$  una economía de intercambio puro. Un equilibrio Walrasiano es un par  $(p^*, x^*)$  tal que

- ❶  $p^* \in \mathbb{R}_{++}^L$
- ❷  $x^* = (x_i^*)_{i=1}^n \in \mathbb{R}_+^{nL}$  es una asignación en la que  $\sum_{i=1}^n x_i^* = \sum_{i=1}^n \omega_i$
- ❸  $\forall i = 1, \dots, n: p \cdot \omega_i = p \cdot x_i^*$  y  $u_i(y) > u_i(x_i^*) \implies p \cdot y > p \cdot x_i^*$ . Esto significa que cada agente maximiza su utilidad.

## Observación

Por el Teorema del Máximo de Berge,  $x_i^* = x^*(p, \omega_i)$  es continua, más no sabemos nada acerca de la diferenciabilidad (con miras por ejemplo a usar Newton-Raphson).

# Equilibrio Walrasiano

Matemáticamente, cada agente resuelve

$$\begin{cases} \max & u_i(x_i) \\ \text{s. a} & p \cdot x_i \leq p \cdot \omega_i \\ & x_i \geq 0. \end{cases}$$

Se obtiene  $x_i^*(p)$ . Luego, para encontrar  $p^*$ , se resuelve

$$\sum_{i=1}^n x_i^*(p) = \sum_{i=1}^n \omega_i.$$

## Ejemplo

Considere  $n = 2$  y  $L = 2$  con  $u_1(x_{11}, x_{21}) = x_{11}x_{21}$  y  $u_2(x_{12}, x_{22}) = x_{11}^{1/4}x_{22}^{3/4}$ . Considere también  $\omega_1 = (1, 2)$  y  $\omega_2 = (4, 3)$ . Entonces,

$$\textcircled{1} \quad x_{11}^* = \frac{p_1 + 2p_2}{2p_1}, \quad x_{21}^* = \frac{p_1 + 2p_2}{2p_2}.$$

$$\textcircled{2} \quad x_{12}^* = \frac{4p_1 + 3p_2}{4p_1}, \quad x_{22}^* = \frac{3(4p_1 + 3p_2)}{4p_2}.$$

$\textcircled{3}$  Ley de Walras y homogeneidad de grado cero<sup>a</sup>

$$x_{11}^* + x_{12}^* = \frac{p_1 + 2}{2p_1} + \frac{4p_1 + 3}{4p_1} = 1 + 4 = 5.$$

Resolviendo la ecuación,  $p_1^*/p_2^* = 1/2$ . Resulta que la otra ecuación es redundante y solo nos interesa el ratio  $p_1/p_2 = 1/2$ .

---

$$^a x^*(\alpha p) = x^*(p) \text{ para todo } \alpha > 0.$$

## $\varepsilon$ —equilibrio Walrasiano

Un  $\varepsilon$ —equilibrio Walrasiano es un par  $(p^*, x^*) \in \mathbb{R}_+^L \times \mathbb{R}_+^{nL}$  tal que para todo  $i = 1, \dots, n$

$$\textcircled{1} \quad u_i(y) > u_i(x_i^*) \implies p \cdot y > p \cdot x_i^*$$

$$\textcircled{2} \quad |p^* \omega_i - p \cdot x_i^*| < \varepsilon$$

y donde  $\sum_i x_i^* = \sum \omega_i$ . Informalmente, un  $\varepsilon$ —equilibrio Walrasiano consiste en un precio y una asignación en los cuales los agentes están maximizando su utilidad, la oferta es igual a la demanda y los gastos de los agentes son aproximadamente iguales a sus ingresos.

# Enfoque de Negishi

- 1 Welfare Economics and Existence of an Equilibrium for a Competitive Economy - Takashi Negishi (1960).
- 2 Se usa el Segundo Teorema del Bienestar para probar la Existencia del Equilibrio Walrasiano.
- 3 La Existencia del Equilibrio Walrasiano usualmente se prueba haciendo uso del Teorema del Punto Fijo de Brouwer (preferencias estrictamente convexas) o el de Kakutani (preferencias convexas).
- 4 Equilibrio General con Transferencias.



# Equilibrio General con Transferencias

## Definición

**Equilibrio Walrasiano con Transferencias.** Es una terna  $(p^*, x^*, T) \in \mathbb{R}_+^L \times \mathbb{R}_+^{nL} \times \mathbb{R}^n$  tal que

- ❶  $\sum_{i=1}^n T_i = 0$  (vector de transferencias)
- ❷  $\sum_{i=1}^n x_i^* = \sum_{i=1}^n \omega_i$
- ❸  $\forall i: u_i(y) > u_i(x_i^*) \implies p^* \cdot y > p \omega_i + T_i$  y  $p \cdot x_i^* = p \cdot \omega_i + T_i$ .

## Observación

Un equilibrio Walrasiano es un equilibrio Walrasiano con transferencias en el que  $T = 0$ . Un  $\varepsilon$ -equilibrio Walrasiano es un equilibrio en el que  $\|T\|_\infty \leq \varepsilon$ .

## Segundo Teorema del Bienestar (una versión)

Para todo vector de pesos  $\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{R}_+^n$ , existe un equilibrio Walrasiano con transferencias  $(p^*, x^*, T)$  que maximiza

$$\sum_{i=1}^n \lambda_i u_i(x_i).$$

Esto se debe al **Segundo Teorema del Bienestar**.

Los pesos  $\lambda_i$  se conocen como pesos de bienestar. Es posible (ver Negishi 1960) construir una aplicación que busque deshacer las transferencias  $T_i$  otorgando diferentes pesos: mayor peso a los que se ven perjudicados  $T_i < 0$ , y menor peso a los que se ven beneficiados  $T_i > 0$ .

$$\underbrace{(\lambda_1, \dots, \lambda_n)}_{\text{asociado a } T_1, \dots, T_n} \rightarrow (w_1, \dots, w_n).$$

Los detalles de esta aplicación (función) son omitidos en el artículo, pero lo importante es que todo punto fijo cumple con  $T = 0$  pues ningún  $\lambda$  cambiaría. Aplicando Kakutani, se prueba la existencia de dicho punto fijo, y se hace sobre un espacio  $n$ —dimensional y no  $L$ —dimensional (caso clásico vía Brouwer).

# Kakutani vs Lema de Sperner

- 1 Usualmente la existencia del equilibrio Walrasiano se demuestra aplicando el [Teorema del Punto Fijo de Brouwer](#) (cuando las preferencias son estrictamente convexas, lo que asegura que  $x^*$  sea un único punto).
- 2 Cuando las preferencias no son estrictamente convexas, se procede por Kakutani.
- 3 La prueba de Brouwer tiene dos enfoques: una demostración que pasa por el Lema de Borsuk (bastante topología y análisis vectorial) y otra que pasa por el Lema de Sperner (también usa topología, pero más elemental).
- 4 Los autores hacen uso del Lema de Sperner para construir su algoritmo.

- 1 Fixed Point Theorems with Applications to Economics and Game Theory de Kim Border (1985).
- 2 Una colección de puntos  $x_0, \dots, x_m \in \mathbb{R}^n$  es afín linealmente independientes si  $\sum_{i=1}^m \theta_i x_i = 0$  y  $\sum_{i=1}^m \theta_i = 0$  implica que  $0 = \theta_0 = \dots = \theta_m$ .
- 3 La clausura de un  $m$ -simplex es la cápsula convexa de un conjunto de  $m + 1$  vectores afín linealmente independientes:

$$\Delta(x_0, \dots, x_m) = \left\{ \sum_{i=0}^m \theta_i x_i : \theta_i \geq 0, i = 0, \dots, m, \sum_{i=0}^m \theta_i = 1 \right\}.$$

- 4 El simplex y su clausura se diferencian de la siguiente forma:  $\theta_i > 0$  en vez de  $\theta_i \geq 0$ . Cada  $x_i$  es un vértice de  $\Delta(x_0, \dots, x_m)$ .
- 5 Por ejemplo, en  $\mathbb{R}^3$ , un 2-simplex es un triángulo. Un 3-simplex es un tetraedro.
- 6 Cada  $\Delta(x_{i_0}, \dots, x_{i_k})$  es una cara del simplex.
- 7 El diámetro de un simplex es la mayor distancia entre 2 de sus vértices.
- 8 Para cada  $y \in \overline{\Delta}(x_0, \dots, x_m)$ , sea  $\chi(y) = \{i : \theta_i > 0\}$ . Si  $\chi(y) = \{i_0, \dots, i_k\}$ , entonces  $y \in \Delta(x_{i_0}, \dots, x_{i_k})$ .

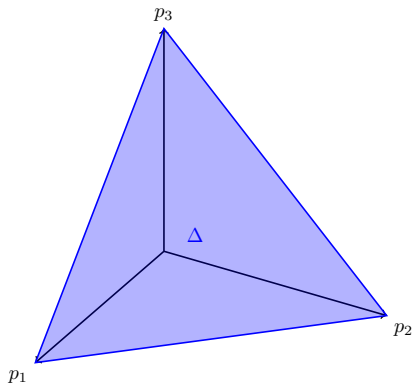


Figura Simplex en  $\mathbb{R}^3$ .

- ❶ Denotemos por  $e_i \in \mathbb{R}^n$  el clásico vector canónico. Entonces, el simplex estándar se denota  $\Delta = \Delta(e_1, \dots, e_n)$  y es tal que

$$\overline{\Delta} = \left\{ x \in \mathbb{R}^n : x \geq 0, \sum_{i=1}^n x_i = 1 \right\}.$$

- ❷ Una división simplicial de  $\overline{\Delta}$  es una colección  $A_1, \dots, A_J$  de índices tales que  $\overline{\Delta} = \bigcup_{j=1}^J A_j$  y tal que para todo  $j \neq h$ ,  $\overline{A_j} \cap \overline{A_h}$  es el vacío o la clausura de una cara en común.
- ❸ El **mesh** (malla, grilla) de una división simplicial es el diámetro más grande de los sub-simplex.
- ❹ Fijemos una división simplicial de  $\Delta$ , digamos  $A_1, \dots, A_J$ . Denotamos por  $V$  a los vértices de dicha subdivisión.
- ❺ Una función  $f : V \rightarrow \{1, \dots, n\}$  para la que  $f(v) \in \chi(v)$ ,  $\forall v \in V$  es llamado un etiquetado adecuado de la subdivisión simplicial.

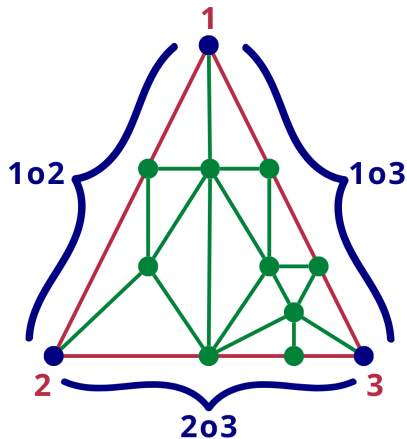


Figura Coloreo o etiquetado.

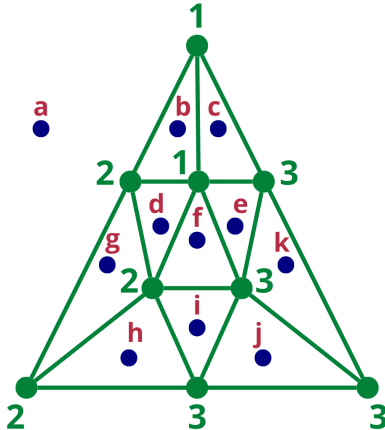


Figura Coloreo completo.



## Lema de Sperner

## Lema

**Sperner.** Sean  $A_1, \dots, A_J$  una división simplicial de  $\overline{\Delta}$  y  $f$  un etiquetado adecuado. Entonces, existe un número impar de  $A_j$  tales que  $f$  alcanza todos los valores de  $\{1, \dots, n\}$  en los vértices de  $A_j$ .<sup>a</sup>

<sup>a</sup>A esto se le conoce como etiquetado completo.

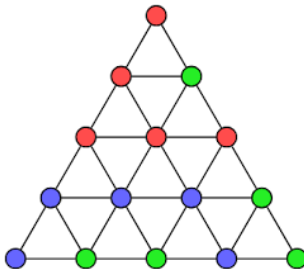


Figura Lema de Sperner.

## Resultado principal e implicancias

## Teorema

Sea  $\varepsilon > 0$ . Existe un algoritmo tal que, para toda economía  $E$ , encuentra un  $\varepsilon$ -equilibrio Walrasiano en tiempo polinomial en  $L$  y  $1/\varepsilon$ , para  $n$  fijo.

- ① Número arbitrariamente grande de bienes  $L$ : problemas de consumo inter-temporal.
- ② Tiempo polinomial y no exponencial.
- ③ Aproximado:  $\varepsilon$ —equilibrio Walrasiano.
- ④ Supuestos razonables: no separabilidad, concavidad, diferenciabilidad, monotonía.

# El Algoritmo

## El algoritmo

Sea  $\varepsilon > 0$  y fijemos una economía de intercambio puro con las propiedades establecidas. Considere el siguiente problema de optimización donde  $\lambda \in \overline{\Delta}$

$$\mathcal{P}_\lambda : \begin{cases} \max & \sum_{i=1}^n \lambda_i u_i(x_i) \\ \text{s. a} & \sum_{i=1}^n x_i \leq \sum_{i=1}^n \omega_i \\ & x_i \geq 0. \end{cases}$$

Sea  $x(\lambda)$  una solución de  $\mathcal{P}_\lambda$ . Como  $\lambda \in \overline{\Delta}$ , existe al menos un  $h$  tal que  $\lambda_h > 0$ . Por las condiciones de Inada, sabemos que  $x_h(\lambda) \in \mathbb{R}_{++}^L$ . Luego, definamos

$$p(\lambda) = \lambda_h Du_h(x_h(\lambda)) > 0.$$

Las CPO aplicadas a

$$L(x, \mu, \gamma; \lambda) = \sum_{i=1}^n \lambda_i u_i(x_i) + \mu \left[ \sum_{i=1}^n \omega_i - \sum_{i=1}^n x_i \right] + \sum_{i=1}^n \gamma_i x_i$$

aseguran que  $p(\lambda)$  sea independiente del  $h$ :

$$\lambda_{h_1} Du_{h_1}(x_{h_1}^*) = \mu = \lambda_{h_2} Du_{h_2}(x_{h_2}^*), \quad x_{h_i}^* > 0.$$

# El algoritmo

Defina para todo  $i$

$$T_i(\lambda) = p(\lambda) \cdot (x_i(\lambda) - \omega_i)$$

y sea  $g : \bar{\Delta} \rightarrow \mathbb{R}^n$  dada por  $g(\lambda)_i = T_i(\lambda)$ . Como  $\sum_i x_i = \sum_i \omega_i$ , entonces se tiene que  $\sum_i T_i = 0$ .

## Observación

La terna  $(x(\lambda), p(\lambda), T(\lambda))$  es un equilibrio Walrasiano con transferencias pues  $p(\lambda)$  fue escogido para satisfacer las CPO de primer orden de los agentes con  $\lambda_i > 0$  y aquellos con  $\lambda_i = 0$  maximizan trivialmente su utilidad. En efecto, recordemos que el problema del consumidor en el contexto de equilibrio general con transferencias es

$$\begin{aligned} \max u_i(x_i) \\ \text{s. a } p \cdot x_i \leq p \cdot \omega_i + T_i \\ x_i \geq 0. \end{aligned}$$

El Lagrangiano es

$$L(x_i, \zeta, \gamma) = u_i(x_i) + \zeta[T_i + p \cdot \omega_i - p \cdot x_i] + \gamma \cdot x_i.$$

Cuando  $x_i^* > 0$ , se tiene que  $Du_i(x_i) = p\zeta \implies \zeta = \lambda^{-1}$ .

## El algoritmo

- Construya una subdivisión simplicial  $S$  de  $\overline{\Delta}$  con tamaño del mesh (malla/grilla)

$$\frac{\varepsilon}{(M - m)(n - 1)^4}$$

con  $M$  y  $m$  dados por (1) y (2) respectivamente.

- Para cada vértice de  $S$  calcular  $g(\lambda)$ . Si para cierto  $\lambda \in S$  se cumple que  $g(\lambda) \leq 0$ , entonces  $T_i(\lambda) \leq 0$  para todo  $i$ . Así,  $\sum_{i=1}^n T_i = 0$  en conjunto con  $T_i \leq 0$  implican que  $T_i = 0$ . Así, habremos encontrado un equilibrio Walrasiano.
- Si no existe tal vértice, esto quiere decir que para cualquier  $\lambda \in S$ ,  $g(\lambda)_i > 0$  para algún  $i \in \{1, \dots, n\}$ . En dicho caso, definamos el siguiente etiquetado: la etiqueta de un vértice  $\lambda$  es el  $i$  para el cual la transferencia a  $i$  en  $g(\lambda)$  es la más grande (es decir el mayor componente de  $g(\lambda)$ ). Si hay más de uno, escoger el menor en términos de  $i$  (del índice).
- El etiquetado descrito en el ítem previo es un etiquetado adecuado. Esto vamos a probarlo más adelante.
- Dado que el etiquetado es adecuado, por el Lema de Sperner existe un sum-simplex completamente etiquetado  $\lambda^1, \dots, \lambda^n$ . Este se puede encontrar aplicando el algoritmo de Scarf.
- Sea  $\eta^i = g(\lambda^i)$ .
- Se reporta  $\eta^1$  como el  $\varepsilon$ -equilibrio Walrasiano.



---

**Algorithm 1:** Algoritmo compacto para encontrar un equilibrio Walrasiano.

---

```
1 Construir subdivisión simplicial  $S$  de  $\overline{\Delta}$  con tamaño de mesh  $\frac{\varepsilon}{(M-m)(n-1)^4}$ ;  
2 foreach vértice  $\lambda$  en  $S$  do  
3   Calcular  $g(\lambda)$ ;  
4   if  $g(\lambda) \leq 0$  then  
5      $T_i(\lambda) \leq 0$  para todo  $i$ , y  $\sum_{i=1}^n T_i = 0$ ;  
6     return Equilibrio Walrasiano;  
7 if ningún vértice satisface la condición then  
8   Etiquetar cada vértice  $\lambda$  con el índice  $i$  donde  $g(\lambda)_i$  es máximo;  
Result: Por el Lema de Sperner, usar el algoritmo de Scarf para encontrar el sum-simplex etiquetado  $\lambda^1, \dots, \lambda^n$ . Reportar  $\eta^1 = g(\lambda^1)$  como el  $\varepsilon$ -equilibrio.
```

---

# El algoritmo

Lo pendiente es:

- 1 Definir  $m$  y  $M$  y justificar el mesh (Lema 3.2 del artículo).
- 2 Probar que el etiquetado es adecuado. (Lema 3.3 del artículo).
- 3 Probar que  $\|\eta^i\|_\infty = \|g(\lambda^i)\|_\infty < \varepsilon$  (esto asegura por definición que se haya encontrado un  $\varepsilon$ -equilibrio Walrasiano) - ya tenemos los precios, las canastas en función de  $\lambda$ , solo falta definir el  $\lambda$  ( $\lambda^1$  en este caso) y los  $T$ . Ver Definición 3.2. (Lema 3.4 del artículo).
- 4 Verificar que se computa en tiempo polinomial.

# El algoritmo

Definimos

$$M = \max_{1 \leq i \leq n} \left\{ u_i(x_i) : (x_1, \dots, x_n) \geq 0, \sum_j x_j \leq \sum_j \omega_j \right\} \leq 1 \quad (1)$$

$$m = \min_{1 \leq i \leq n} \left\{ u_i(x_i) : (x_1, \dots, x_n) \geq 0, \sum_j x_j \leq \sum_j \omega_j \right\} \geq 0. \quad (2)$$

Tanto  $m$  como  $M$  están bien definidos pues  $u_i$  es continua para todo  $i$  y el conjunto de las asignaciones factibles es un conjunto compacto. Además, nada la normalización que se efectuó al inicio, se tienen las cotas propuestas.

# El algoritmo

## Lema

Dados  $\lambda, \lambda' \in \Delta$ ,  $\|g(\lambda) - g(\lambda')\|_\infty \leq (n-1)(M-m)\|\lambda - \lambda'\|_\infty$ .<sup>a</sup>

<sup>a</sup>Esto es,  $g$  es Lipschitz continua con constante  $(n-1)(M-m)$ .

## Proof.

Recordemos que  $g(\lambda)_i = T_i(\lambda)$ . Como  $u_i$  es cóncava,  $Du_i(y)(z - y) \geq u_i(y) - u_i(x)$ . De este modo,

$$\begin{aligned} T_i(\lambda) &= p(\lambda) \cdot (x_i(\lambda) - \omega_i) \\ &= \lambda_i Du_i(x_i(\lambda))(x_i(\lambda) - \omega_i) \\ &\leq \lambda_i (u_i(x_i(\lambda)) - u_i(\omega_i)) \\ &\leq \lambda_i (M - m). \end{aligned}$$



## El algoritmo

Proof.

Por otro lado,

$$\begin{aligned}\sum_j T_j(\lambda) &= 0 \\ T_i(\lambda) + \sum_{j \neq i} T_j(\lambda) &= 0 \\ T_i(\lambda) &= - \sum_{j \neq i} T_j(\lambda) \\ &\geq - \sum_{j \neq i} (M - m) \lambda_j \\ &\geq -(n - 1)(M - m) \|\lambda\|_\infty.\end{aligned}$$

Así,

$$-(n - 1)(M - m) \|\lambda\|_\infty \leq T_i(\lambda) \leq \lambda_i(M - m) \leq (n - 1)(M - m) \|\lambda\|_\infty.$$

O sea,  $|T_i(\lambda)| \leq (n - 1)(M - m) \|\lambda\|_\infty$ .



# El algoritmo

Complicaciones:

- 1 El resultado anterior no permite concluir que  $g$  sea Lipschitz, al menos no directamente. Primero que nada,  $T_i$  no es lineal, combinando las desigualdades, queda una cota en  $|T_i(\lambda) - T_i(\lambda')| \leq C(\lambda_i + \lambda'_i)$  y no se puede acotar esto por  $\|\lambda - \lambda'\|_\infty$ .
- 2 A priori, no tenemos que  $g \in C^1(\Omega)$ , con  $\Delta \subset \Omega$ .
- 3 ¿Alguna propiedad sobre operador acotados linealmente? Ver [KOROVKIN TYPE THEOREMS FOR WEAKLY NONLINEAR AND MONOTONE OPERATORS](#).
- 4  $\Delta$  tiene propiedades especiales: compacto, convexo.

# El algoritmo

Forma de resolverlo, hacer que  $g \in C^1$  pues  $\overline{\Delta}$  es compacto.

- ❶  $T_i = p(\lambda) \cdot (x_i(\lambda) - \omega_i)$
- ❷  $p(\lambda) = \lambda_i Du_i(x_i(\lambda))$
- ❸ Tomemos  $u \in C^2$  y además, usemos los [teoremas de diferenciabilidad de las soluciones con respecto a los parámetros](#).
- ❹ Tendríamos que  $x_i(\lambda)$  es  $C^1$  es una vecindad  $V_\lambda$  de  $\lambda$ . Como  $\Delta$  es compacto, podemos extraer un cubrimiento finito de  $\bigcup_{\lambda \in \Delta} V_\lambda$ . Digamos  $V_{\lambda_1}, \dots, V_{\lambda_k}$ .
- ❺ Para cada  $V_{\lambda_j}$ , escoger  $C_j$  tal que  $T_i$  es  $C^1(V_{\lambda_j})$  por composición y aritmética de funciones clase  $C^1$ .
- ❻ ¿Combinar continuidad uniforme con crecimiento lineal?

## El algoritmo

Tomemos el resultado (Lema 2) como válido. En dicho caso, si tomamos un mesh de forma que

$$\|\lambda - \lambda'\|_{\infty} < \frac{\varepsilon}{(M - m)(n - 1)^3},$$

entonces

$$\|g(\lambda) - g(\lambda')\|_{\infty} < (M - m)(n - 1)\|\lambda - \lambda'\|_{\infty} < \frac{\varepsilon}{(n - 1)^2}.$$



# El algoritmo

## Lema

Si  $\lambda_i = 0$ , entonces  $g(\lambda)_i \leq 0$ .

## Proof.

Si  $x(\lambda)$  es una solución de  $\mathcal{P}_\lambda$ , entonces, como las funciones  $u_i$  son crecientes,  $x_i(\lambda) = 0$  (si no los otros no maximizan). De este modo,

$$T_i(\lambda) = -p(\lambda) \cdot \omega_i \leq 0.$$



## Observación

Si  $i \notin \chi(\lambda)$ , i.e.  $\lambda_i = 0$ , entonces  $g(\lambda)_i \leq 0$ , por lo que la etiqueta de  $\lambda$  (menor coordenada con  $(g(\lambda))_j$  máximo estrictamente positivo está en  $\chi(\lambda)$ , y aquellas con  $(g(\lambda))_k \leq 0$ , no están).

## Observación

Por construcción, dado que  $\eta^i = g(\lambda^i)$  y  $\eta_j = g(\lambda^j)$ , se tiene que

$$\|\eta^i - \eta^j\|_\infty < \frac{\varepsilon}{(n-1)^2}.$$

# El algoritmo

## Lema

Se tiene que  $\|\eta^i\|_\infty < \varepsilon$ .

## Proof.

Por definición del etiquetado, si probamos que  $\eta_i^i \leq \frac{\varepsilon}{n-1}$ , dado que  $\eta_i^i$  es la mayor componente de  $\eta^i$ , se cumple que si  $\eta_j^i > 0$ ,  $\eta_j^i \leq \varepsilon$ , y si  $\eta_j^i \leq 0$ , como  $\sum_h \eta_h^i = 0^a$ ,

$$\eta_j^i \geq - \sum_{h, \eta_h^i > 0} \eta_h^i \geq -(n-1)\eta_i^i \geq -\varepsilon.$$

Así, tendremos que  $|\eta_j^i| \leq \varepsilon$ . Supongamos por contradicción que existe  $i$  con  $\eta_i^i > \varepsilon/(n-1)$ . Como  $\sum_j \eta_j^i = 0$ , debe existir  $\eta_j^i < -\varepsilon/(n-1)^2$ . Entonces, para cualquier otro  $\eta^j$

$$\|\eta^i - \eta^j\|_\infty \geq |\eta_j^i - \eta_j^j| \geq |\eta_j^i| > \frac{\varepsilon}{(n-1)^2},$$

donde la primera desigualdad se debe a la definición de  $\|\cdot\|_\infty$  y la segunda pues  $\eta_j^i > 0$  y  $\eta_j^j < 0$ . □

---

<sup>a</sup>Pues las transferencias suman cero.

# El algoritmo

## Proof.

Pero entonces,

$$\|\eta^i - \eta^j\|_\infty > \frac{\varepsilon}{(n-1)^2},$$

lo cual contradice la construcción del mesh. □

## Observación

El Lema 4 permite concluir pues, si  $\|\eta^i\| < \varepsilon$ , entonces  $\|g(\lambda^i)\|_\infty < \varepsilon$ , y por ende  $\|T(\lambda^i)\|_\infty < \varepsilon$ . Como esto vale para  $i = 1$  (en particular), concluimos.

- ❶  $\mathcal{O}(g(n)) = \{f(n) : \exists c > 0, n_0 \in \mathbb{N} : 0 \leq f(n) \leq cg(n), \forall n \geq n_0\}$ . Informalmente,  $f(n) = \mathcal{O}(g(n))$  significa que  $f(n)$  es asintóticamente menor o igual a  $g(n)$ . Por ejemplo  $n^2 \in \mathcal{O}(n^3)$ .
- ❷  $\Omega(g(n)) = \{f(n) : \exists c > 0, n_0 \in \mathbb{N} : 0 \leq cg(n) \leq f(n), \forall n \geq n_0\}$ . Informalmente,  $f(n) = \Omega(g(n))$  significa que  $f(n)$  es asintóticamente mayor que  $g(n)$ . Por ejemplo,  $n^2 \in \Omega(n)$ .
- ❸  $f(n) = \Theta(g(n))$  si y solo si  $f(n) = \mathcal{O}(g(n))$  y  $f(n) = \Omega(g(n))$ . Por ejemplo  $n^2 + 1 \in \Theta(n^2 - 1)$ .

## El tiempo de ejecución del algoritmo

- 1 El número de vértices en el mesh está acotado por  $\Theta\left(\left(\frac{n^4}{\epsilon}\right)^n\right)$ . La cota es  $\left(\frac{1}{\text{diámetro del mesh}}\right)^n$ . En este caso,  $\left(\frac{(n-1)^4(M-m)}{\epsilon}\right)^n \leq \left(\frac{(n-1)^4}{\epsilon}\right)^n$ .
- 2 Para cada vértice, se resuelve un problema de maximización con función objetivo cóncava en un espacio  $L$  dimensional. Esto se puede hacer en tiempo polinomial en  $L$ . En efecto, maximizar una función cóncava es equivalente a minimizar una función convexa.
- 3 Para cada vértice, hay  $\mathcal{O}\left(\left(\frac{n^4}{\epsilon}\right)^n\right)$  sub-simplexes que contienen ese vértice.
- 4 Encontrar por fuerza bruta los  $\lambda^1, \dots, \lambda^n$  requiere  $\Theta\left(\left(\frac{n^4}{\epsilon}\right)^{2n}\right)$  operaciones. Tenemos que multiplicar  $\mathcal{O}\left(\left(\frac{n^4}{\epsilon}\right)^n\right)$  por  $\Theta\left(\frac{n^4}{\epsilon}\right)^n$ .
- 5 De este modo, como  $n$  está fijo, el algoritmo corre en tiempo polinomial en  $L$  y  $1/\epsilon$ .

En conclusión:

$$\mathcal{O}\left(\frac{n^4}{\epsilon}\right)^n \mathcal{O}(L^k) + \mathcal{O}\left(\frac{n^4}{\epsilon}\right)^{2n} = \mathcal{O}(L^k \epsilon^{-m}).$$

Ver [Eaves, 1976], [Todd, 1976] y [Scarf, 1973].

Problema de maximización de una función cóncava sobre la intersección de  $\sum x_i = \sum \omega_i$  y  $\mathbb{R}_+^{nL}$ , es equivalente a minimizar  $-\sum_i \lambda_i u_i(x_i)$  sobre el mismo conjunto de oportunidad  $\Lambda$ . Es sabido de [Convex Optimization: Algorithms and Complexity](#) de Sébastien Bubeck Theory Group, Microsoft Research o [Interior Point Polynomial Time Methods in Convex Programming](#) de A. Nemirovski que existen algoritmos para resolver el problema planteado con tiempo de ejecución es polinomial en la dimensión del espacio ( $L$ ).

# Extensiones

Hasta el momento se ha:

- 1 Introducido el contexto del modelo (preliminares).
- 2 Propuesto un algoritmo que computa un  $\varepsilon$ —equilibrio Walrasiano.
- 3 Comprobado que el algoritmo cumple su objetivo es y polinomial en  $L$  y  $1/\varepsilon$ .
- 4 Comentado acerca de las aplicaciones del algoritmo.

Luego, los autores buscan generalizar su resultado para otras dos nociones diferentes de equilibrio.



### Definición

Un  $\varepsilon$ -equilibrio aproximado en una economía de intercambio puro  $(u_i, \omega_i)_{i=1}^n$  es un par  $(p, x)$  donde  $p \in \mathbb{R}_+^L$  y  $x$  es una asignación tal que

$$p \cdot y \leq p \cdot \omega_i \implies u_i(y) \leq u_i(x_i) + \varepsilon$$

$$|p \cdot \omega_i - p \cdot x_i| < \varepsilon \text{ y } \sum_i x_i = \sum_i \omega_i.$$

### Definición

Un  $\varepsilon$ -equilibrio fuerte en una economía de intercambio puro  $(u_i, \omega_i)_{i=1}^n$  es un par  $(p, x)$  donde  $p \in \mathbb{R}_+^L$  y  $x$  es una asignación  $x \in \mathbb{R}_+^{nL}$  con  $\|\sum_{i=1}^n x_i - \sum_{i=1}^n \omega_i\|_\infty < \varepsilon$  en la que

$$p \cdot y \leq p \cdot \omega_i \implies u_i(y) \leq u_i(x_i), \quad p \cdot \omega_i = p \cdot x_i.$$

Denotamos por  $E \in \tilde{E}$  aquellas economías de intercambio puro que cumplan con los supuestos básicos y tales que, existe  $\theta > 0$  y  $\pi > 0$  para los que

$$\sup_{p \in \Delta} \left( p \cdot \sum_{i=1}^n \omega_i \right) \leq \theta, \quad D_s u_i(x_i) > \pi, \quad s \in \{1, \dots, L\}.$$

### Teorema

Sea  $\varepsilon > 0$ . Entonces, existe un algoritmo que, para cualquier economía en  $\tilde{E}$ , encuentra un  $\varepsilon$ -equilibrio aproximado y un  $\varepsilon$ -equilibrio aproximado fuerte en tiempo polinomial en  $L$  y  $\varepsilon^{-1}$  cuando  $n$  está fijo.

Idea de la prueba:

- 1 Usar  $\overline{\Delta}^\eta$ , donde  $\Delta^\eta = \{\lambda \in \Delta : \lambda_i > \eta\}$ .
- 2 Lema: escoger  $\eta$  tal que  $-\Theta + (n-1)\eta < 0$ . Entonces, para cualquier  $\lambda \in \Delta$ ,  $\lambda_i \leq \eta \implies g(\lambda)_i < 0$ .
- 3 Se define  $\varepsilon' \in (0, \varepsilon\eta)$ . El algoritmo computa un  $\varepsilon$ -equilibrio Walrasiano. Notar que  $\eta$  y  $\varepsilon'$  dependen solo de  $(n-1)$  y  $\Theta$ , por lo que son constantes.

### Lema

Considere una economía en  $\tilde{E}$ . Sea  $\eta$  tal que  $-\Theta + (n-1)\eta < 0$ . Entonces, para cualquier  $\lambda \in \Delta$ ,  $\lambda_i \leq \eta \implies g(\lambda)_i < 0$ .

### Proof.

Sea  $\lambda \in \Delta - \Delta^\eta$ . Sea  $i$  tal que  $\lambda_i \leq \eta$ . Defina

$$\hat{\lambda}_j = \begin{cases} \lambda_j + \lambda_i/(n-1), & \text{si } j \neq i \\ 0, & \text{si } j = i. \end{cases}$$

Entonces, por el mismo argumento del Lema 3,  $g(\hat{\lambda})_i = -p(\hat{\lambda})\omega_i$ . Luego,  $\|\hat{\lambda} - \lambda\|_\infty = \lambda_i \leq \eta$ . Así,  $\|g(\hat{\lambda}) - g(\lambda)\|_\infty \leq (n-1)\eta$ . Entonces,  $|g(\hat{\lambda})_i - g(\lambda)_i| \leq (n-1)\eta$ , por lo que  $g(\hat{\lambda})_i = -p(\hat{\lambda})\omega_i$  implica que

$$g(\lambda)_i \leq -p(\hat{\lambda})\omega_i + (n-1)\eta \leq -\Theta + (n-1)\eta < 0.$$



### Caso 1: $\varepsilon$ -equilibrio aproximado.

#### Proof.

Sea  $\varepsilon' > 0$  con  $\varepsilon' < \varepsilon\eta$ . Tanto  $\eta$  como  $\varepsilon'$  solo dependen de  $\Theta$  y  $n - 1$ . Luego, en  $\Delta^\eta$ ,  $\lambda_i \geq \eta > 0$  para todo  $i$ , por lo que  $x_i \in \mathbb{R}_{++}^L$ . Luego, sea  $\tilde{x}_i^*$  solución de

$$\begin{aligned} \max u_i(\tilde{x}_i) \\ \text{s.a } p\tilde{x} \leq p\omega_i. \end{aligned}$$

Basta probar que  $u_i(x_i^*) - u_i(x_i) < \varepsilon$ . Si  $px \geq p\omega_i$ , concluimos. Si no, usando la concavidad de  $u_i$ ,

$$\begin{aligned} u_i(x_i^*) - u_i(x_i) &\leq Du_i(x_i)(x_i^* - x_i) \\ &= (1/\lambda_i)p(x_i^* - x_i) \\ &\leq (1/\eta)(p\omega_i - px_i) \\ &\leq (1/\eta)\varepsilon' < \varepsilon. \end{aligned}$$



## Caso 2: $\varepsilon$ —equilibrio aproximado fuerte.

### Proof.

Sea  $\eta > 0$  como antes y  $\varepsilon' > 0$  tal que  $\varepsilon' < \varepsilon\eta\pi/n$ . El algoritmo inicial aplicado sobre  $\overline{\Delta}^\eta$ , por lo expuesto previamente, devuelve un  $\varepsilon'$ —equilibrio  $(x, p)$ . Nuevamente, como  $\eta, \varepsilon'$  dependen de  $n - 1$  y  $\Theta$ , están fijo. Luego, sea  $x_i^*$  una solución a

$$\begin{aligned} \max u_i(\tilde{x}_i) \\ \text{s. a } p\tilde{x} \leq p\omega_i. \end{aligned}$$

Esto define  $x^* \in \mathbb{R}_+^L$  (para cada  $i$ ). El objetivo es probar que  $(x^*, p)$  es un  $\varepsilon$ —equilibrio aproximado fuerte. Definamos para  $1 \leq s \leq L$ ,  $y^s \in \mathbb{R}_+^L$  dado por

$$y_h^s = \begin{cases} x_h, & \text{si } h \neq s \\ x_h + \frac{p(x_i^* - x_i)}{p_h}, & \text{si } h = s \end{cases}$$

y  $\theta_s = \frac{p_s(x_{si}^* - x_{si})}{p(x_i^* - x_i)}$  ( $x_{si}$  es la cantidad del bien  $s$  consumido por  $i$ ). Entonces,

$$\sum_s \theta_s = 1, \quad y \quad x^* = \sum_{s=1}^L \theta_s y^s.$$



### Proof.

Como  $p = \lambda_i Du_i(x_i)$ ,  $p_s \geq \eta\pi$  (por el Lema 5). Luego,

$$\begin{aligned} \|x_i^* - x_i\|_\infty &\leq \max \left\{ \frac{|p(x_i^* - x_i)|}{p_h} \right\} \leq \sup \left\{ \frac{|p\omega_i p - x_i|}{\eta\pi} \right\} < \frac{\varepsilon'}{\eta\pi} \\ \|x^* - x\|_\infty &\leq \sum_i \|x_i^* - x_i\|_\infty < \eta \frac{\varepsilon'}{\eta\pi} < \varepsilon.x \end{aligned}$$





Gracias



Codenotti, B. and Varadarajan, K. R. (2006).

Efficient computation of equilibrium prices for markets with leontief utilities.

In Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP), pages 371–382. Springer.



Eaves, B. C. (1976).

A finite algorithm for the linear exchange model.

Journal of Mathematical Economics, 3(2):197–203.



Echenique, F. and Wierman, A. (2012).

Finding a walrasian equilibrium is easy for a fixed number of agents.

In Proceedings of the 13th ACM Conference on Electronic Commerce (EC 2012), pages 495–496. ACM, Association for Computing Machinery.



Scarf, H. E. (1973).

The Computation of Economic Equilibria.

Number 24 in Cowles Foundation Monograph. Yale University Press.



Scarf, H. E. (1977).

The computation of equilibrium prices: an exposition.

Handbook of Mathematical Economics, 1:1001–1061.



Todd, M. J. (1976).

The computation of fixed points and applications, volume 124 of Lecture Notes in Economics and Mathematical Systems.

Springer-Verlag.



Vazirani, V. V. (2007).

Combinatorial algorithms for market equilibria from i - computing in games.

In Nash Equilibrium: Computation and Applications, pages 101–114. American Mathematical Society.