

IMPLEMENTACIÓN EN R DE ALGORITMOS EXACTOS Y APROXIMADOS PARA EL CÁLCULO DEL CORE-CENTER EN PROBLEMAS DEL AEROPUERTO Y DE BANCARROTA

Iago Núñez Lugilde y Nadine Helena Espinoza Burgos

Departamento de Estadística e Investigación Operativa

Universidade de Vigo. Spain

- 1 DEFINICIONES BÁSICAS
- 2 PROBLEMA DE BANCARROTA
 - Cálculo del core-center
- 3 PROBLEMA DEL AEROPUERTO
 - Core-Center
 - Algoritmos de aproximación
 - Algoritmos Exactos
- 4 BIBLIOGRAFÍA

ÍNDICE

- 1 DEFINICIONES BÁSICAS
- 2 PROBLEMA DE BANCARROTA
 - Cálculo del core-center
- 3 PROBLEMA DEL AEROPUERTO
 - Core-Center
 - Algoritmos de aproximación
 - Algoritmos Exactos
- 4 BIBLIOGRAFÍA

JUEGOS TU

Los juegos coalicionales modelizan situaciones en las que la cooperación entre los agentes frecuentemente supone un beneficio para la coalición total que debe ser repartido. Pueden ser de dos tipos:

- Juegos de costes
- Juegos de beneficios

CONJUNTOS ASOCIADOS A LOS JUEGOS TU

ASIGNACIONES EFICIENTES:

$$H = \{x \in \mathbb{R}^N : \sum_{i=1}^n x_i = v(N)\}$$

CONJUNTO DE IMPUTACIONES:

$$I(v) = \{x \in H : x_i \geq v(i), \forall i \in N\}$$

$$I(c) = \{x \in H : x_i \leq c(i), \forall i \in N\}$$

NÚCLEO:

$$C(v) = \{x \in I(v) : x(S) \geq v(S), \forall S \in 2^N\}$$

$$C(c) = \{x \in I(c) : x(S) \leq c(S), \forall S \in 2^N\}$$

CORE-CENTER

SOLUCIONES DE JUEGO COALICIONALES

Una solución ψ es una asignación que asocia, con cada juego $v \in G^N$ un vector de pagos $\psi(v) \in \mathbb{R}^N$ tal que $\sum_{i \in N} \psi_i(v) = v(N)$.

CORE CENTER DE UN JUEGO EQUILIBRADO

Una posible solución del juego es el core-center, que se define de la siguiente manera: Dado un politopo convexo (el núcleo) $K \subset H$, sea $\mu(K)$ su centro de gravedad.

El core-center de un juego equilibrado $v \in G^N$, González Díaz y Sánchez Rodríguez (2007), es el centro de gravedad del núcleo $C(v)$, i.e.,

$$\mu(v) = \mu(C(v)) \in C(v).$$

ÍNDICE

- 1 DEFINICIONES BÁSICAS
- 2 PROBLEMA DE BANCARROTA
 - Cálculo del core-center
- 3 PROBLEMA DEL AEROPUERTO
 - Core-Center
 - Algoritmos de aproximación
 - Algoritmos Exactos
- 4 BIBLIOGRAFÍA

PROBLEMA DE BANCARROTA Y JUEGO TU ASOCIADO

PROBLEMA DE BANCARROTA

Un problema de bancarrota es una tripla (N, E, d) , donde N es el conjunto de acreedores, $E \geq 0$ es la cantidad de dinero a repartir que se tiene inicialmente y $d \in \mathbb{R}^N$ es el vector de demandas que satisface que $d_i \geq 0 \forall i \in N$ y $\sum_{i \in N} d_i \geq E$. Sea B^N la clase de los problemas de bancarrota cuyo conjunto de acreedores es N .

JUEGO TU ASOCIADO (O'NEILL (1982))

Sea G^N la clase de los juegos de coalición cuyo conjunto de jugadores es N .

Dado un problema de bancarrota $(E, d) \in B^N$, su correspondiente juego coalicional asociado es el juego $v \in G^N$ definido por:

$$v(S) = \max\left\{0, E - \sum_{j \in N \setminus S} d_j\right\}, S \subset N.$$

PROBLEMA DE PAGO DE IMPUESTOS

OTRO CONTEXTO DE APLICACIÓN: (THOMSON (2003))

Podemos interpretar la tripla anterior como un juego de reparto de impuestos: N es el conjunto de contribuyentes, d es el vector de ingresos brutos y E es el dinero que necesita el estado. La repartición de E en este caso corresponderá a lo que debe pagar cada contribuyente.

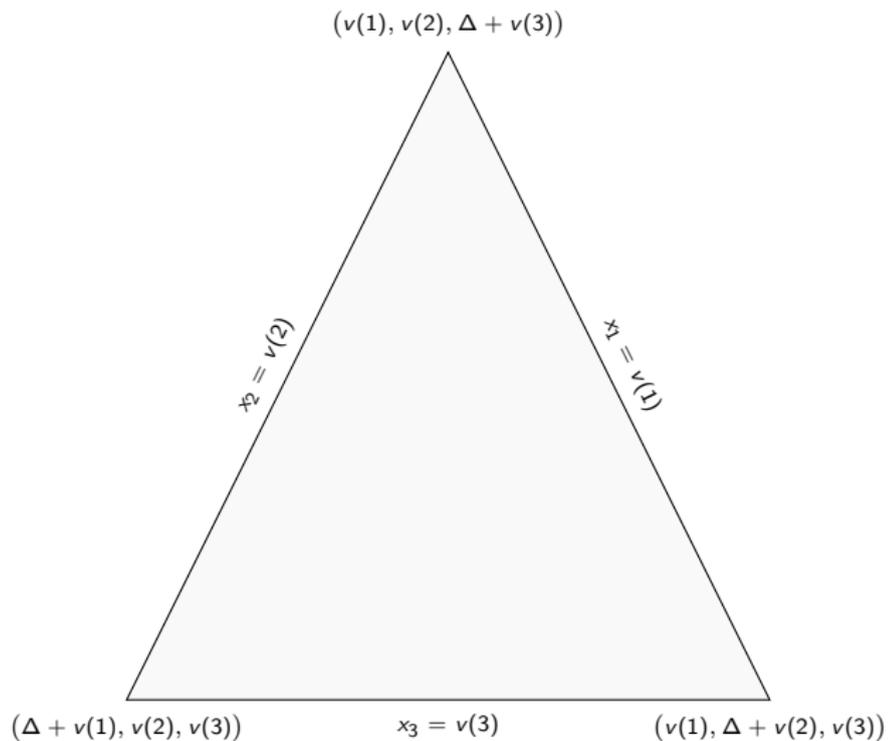
Ambos problemas son equivalentes.

REGLAS DEL PROBLEMA Y SOLUCIONES DEL JUEGO

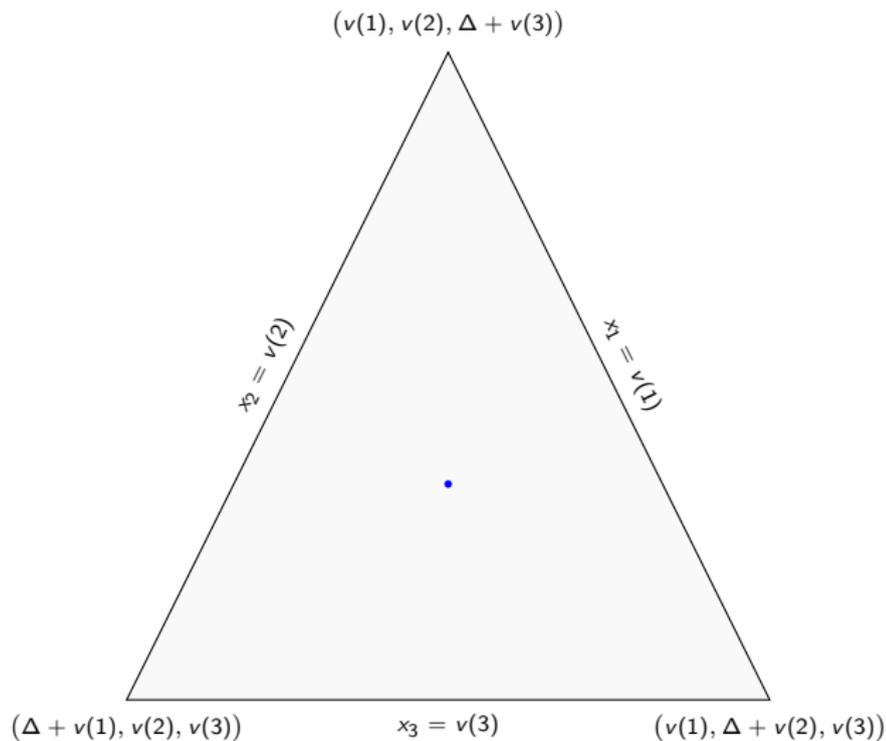
REGLAS DE BANCARROTA:

Una regla R es una función que asocia, con cada $(E, d) \in B^N$ un vector $R(E, d) \in \mathbb{R}^N$ t.q $\sum_{i \in N} R_i(E, d) = E$ y $0 \leq R(E, d) \leq d$.

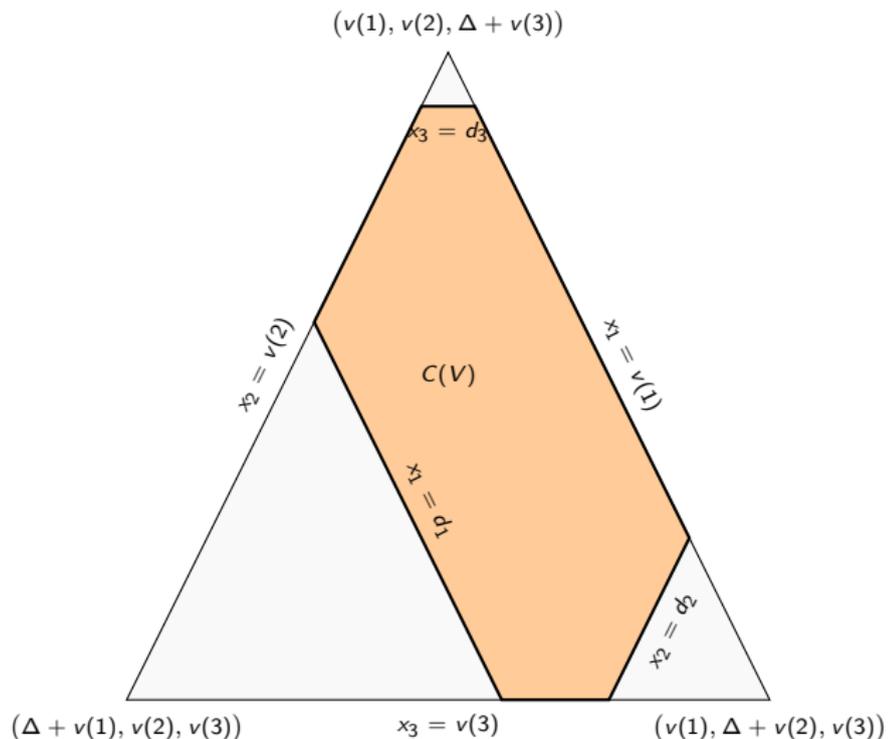
UNA INTERPRETACIÓN DEL CORE-CENTER



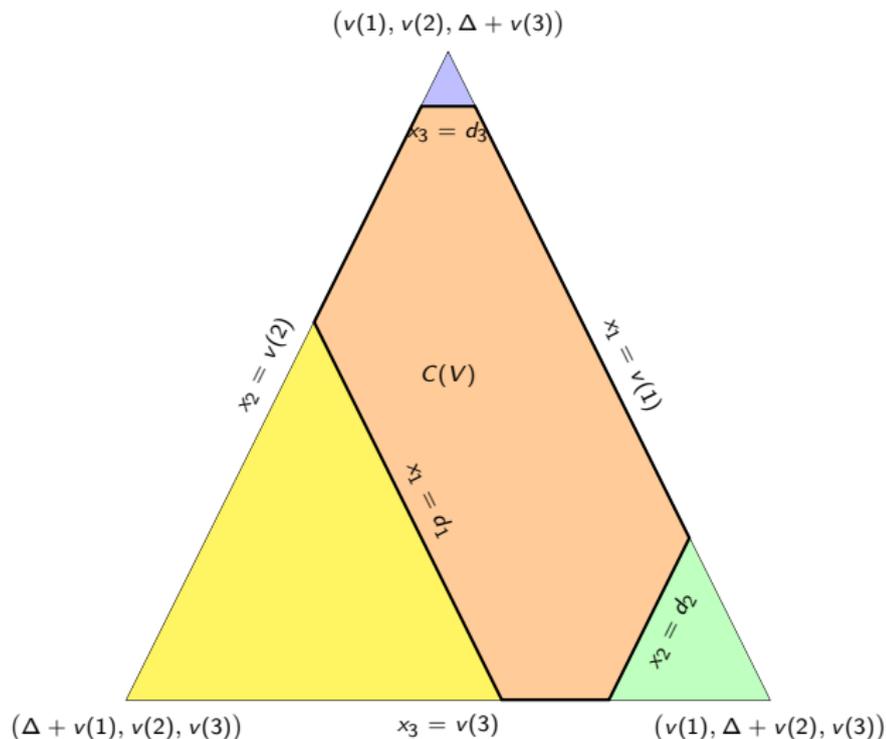
UNA INTERPRETACIÓN DEL CORE-CENTER



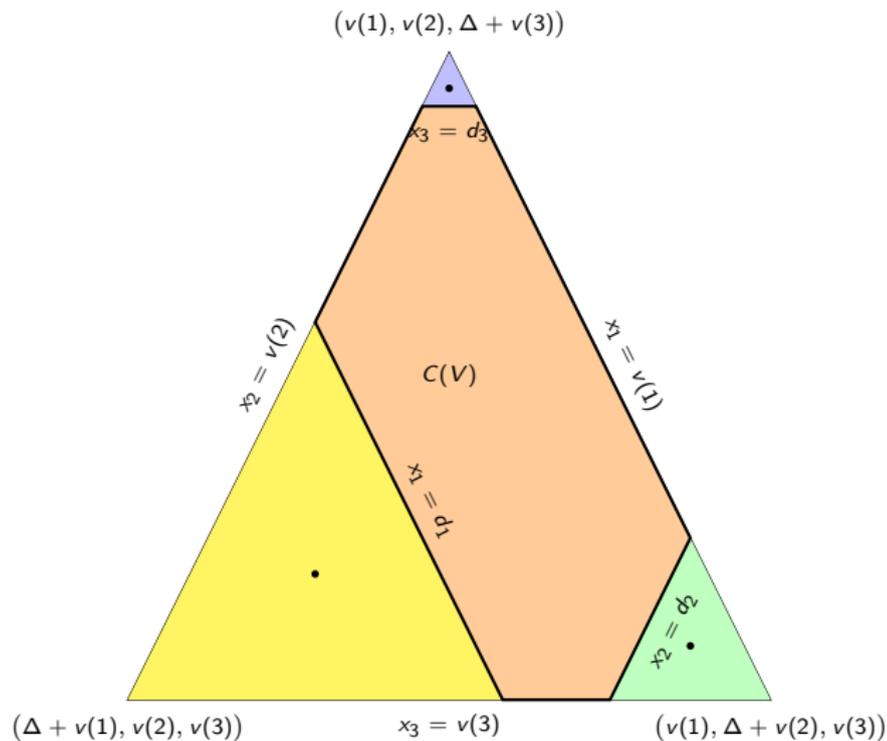
UNA INTERPRETACIÓN DEL CORE-CENTER



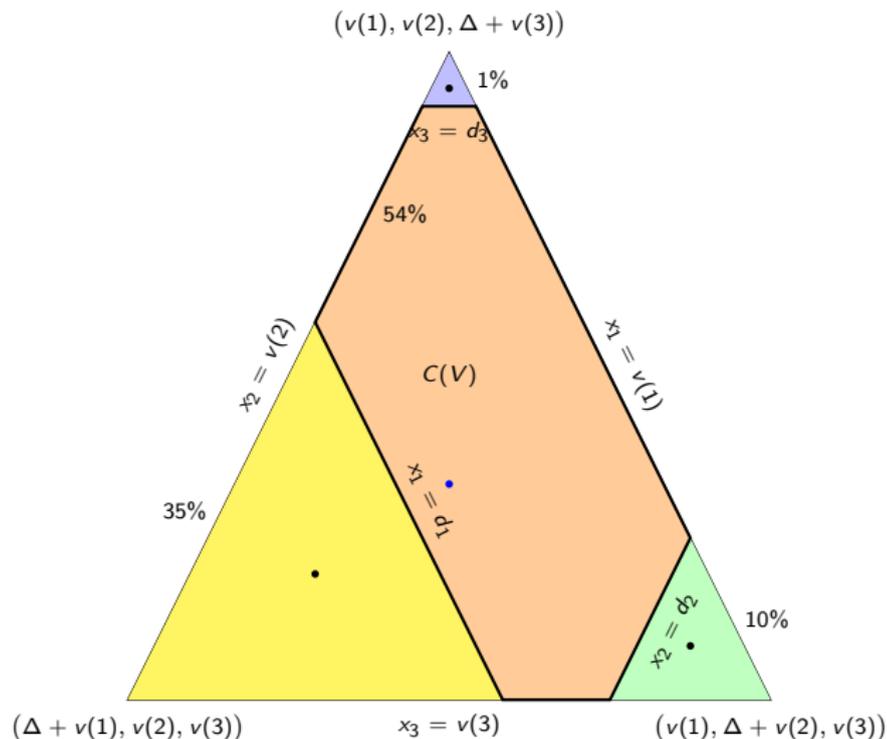
UNA INTERPRETACIÓN DEL CORE-CENTER



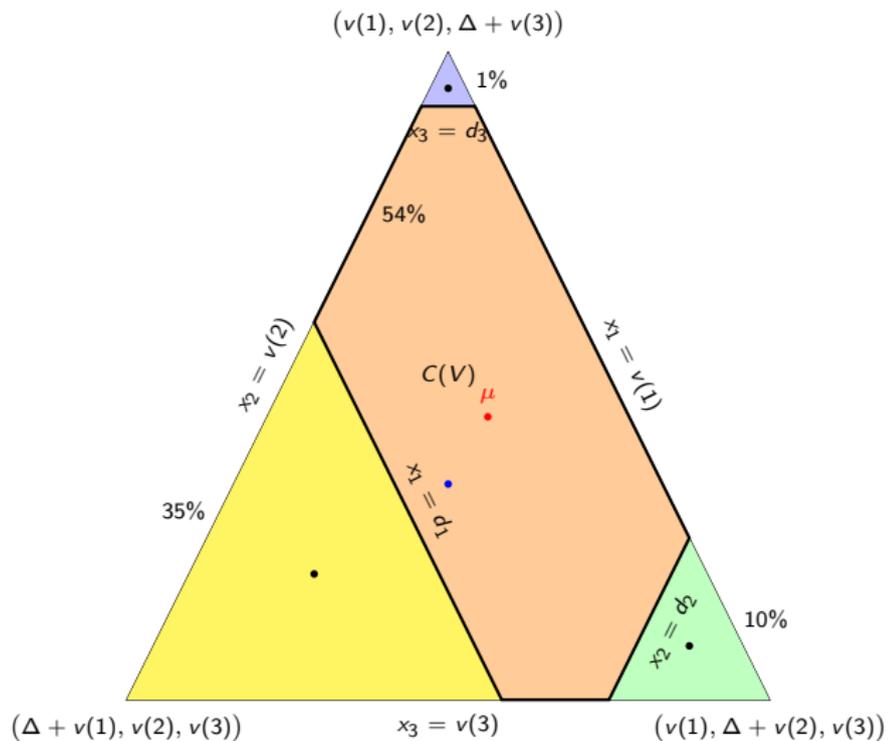
UNA INTERPRETACIÓN DEL CORE-CENTER



UNA INTERPRETACIÓN DEL CORE-CENTER



UNA INTERPRETACIÓN DEL CORE-CENTER



ALGORITMO EXTERIOR: TESELACIÓN DEL CONJUNTO DE IMPUTACIONES

NÚCLEOS DE BANCARROTA

Considerando el conjunto de las coaliciones

$$\mathcal{F} = \left\{ T \subset N : |T| \leq n - 2, \sum_{i \in T} d_i < E \right\}.$$

se tiene que $T \in \mathcal{F} \Leftrightarrow p_T = \text{Vol}(C(v_T)) > 0$.

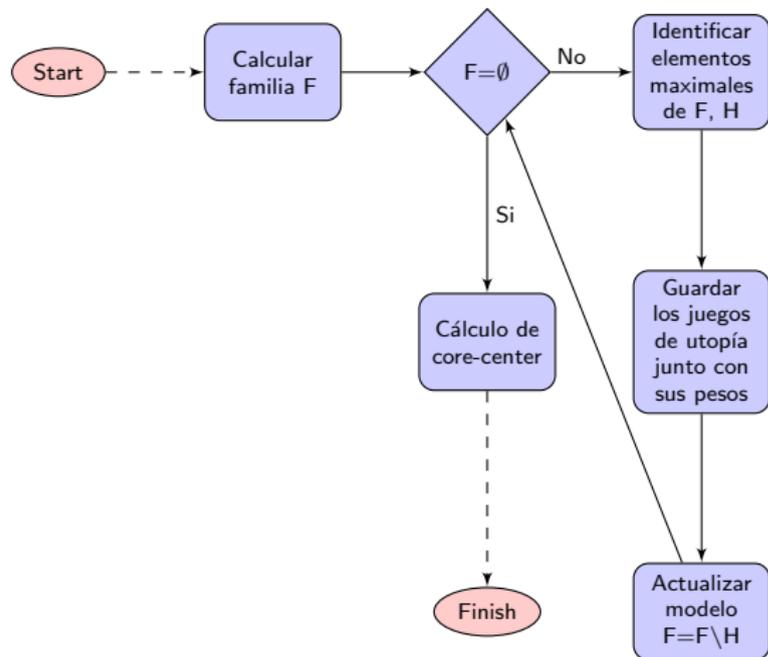
ELEMENTOS MAXIMALES

Si $T \in \mathcal{F}$ es un elemento maximal, entonces $C(v_T) = I(v_T)$

DESCOMPOSICIÓN DEL CONJUNTO DE IMPUTACIONES

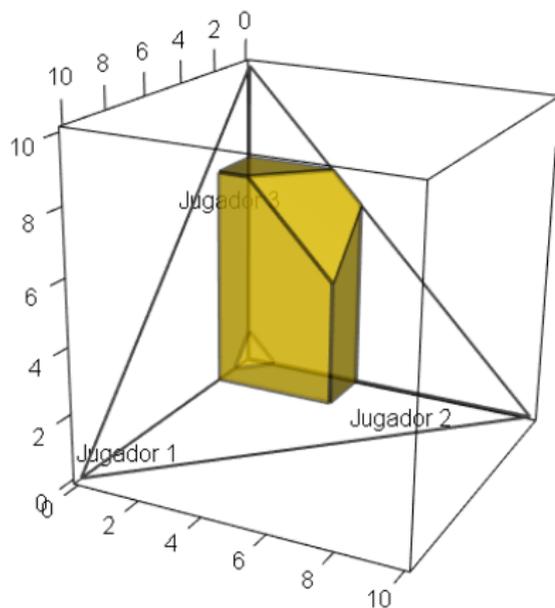
$$I(v) = C(v) \cup \left(\bigcup_{S \in \mathcal{F}} C(v_S) \right),$$

FORMULACIÓN ALGORITMO EXTERIOR (SÁNCHEZ RODRÍGUEZ ET AL (2018))



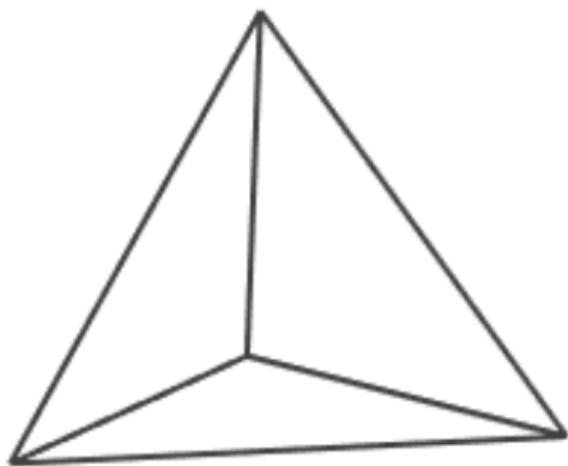
ALGORITMO EXTERIOR

El núcleo del problema $E = 10$, $d = (2, 4, 7, 9)$



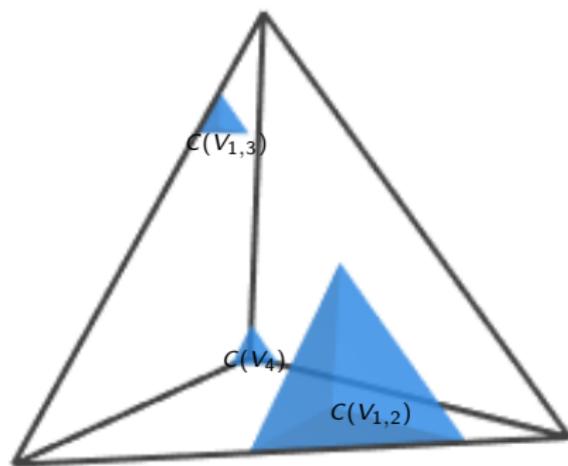
EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



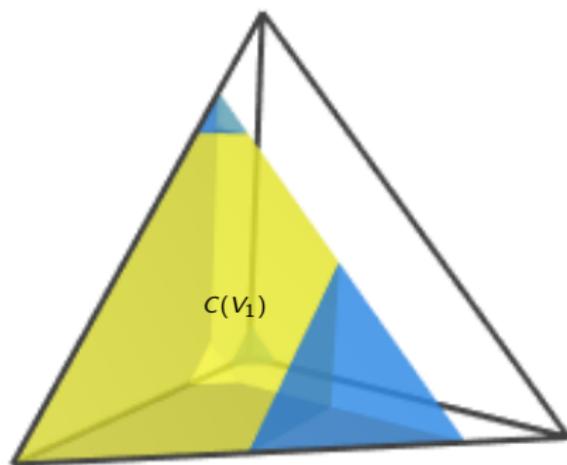
EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



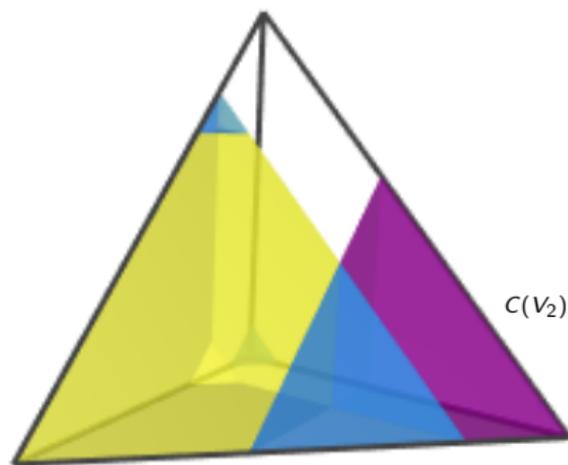
EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



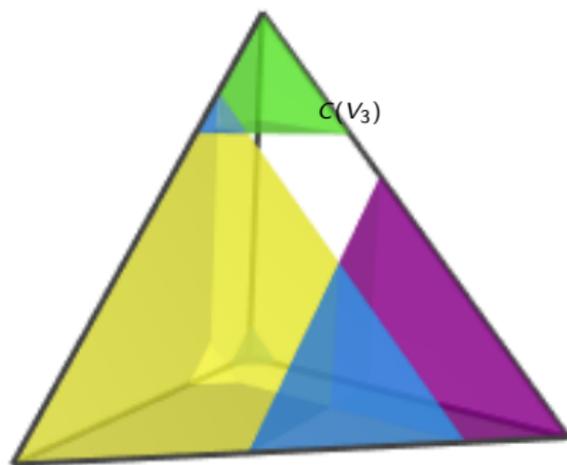
EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



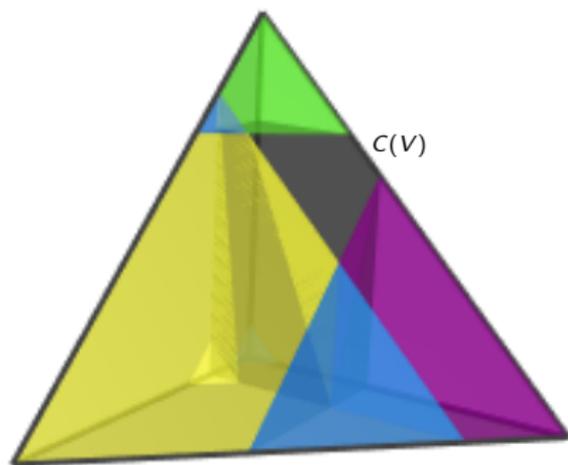
EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



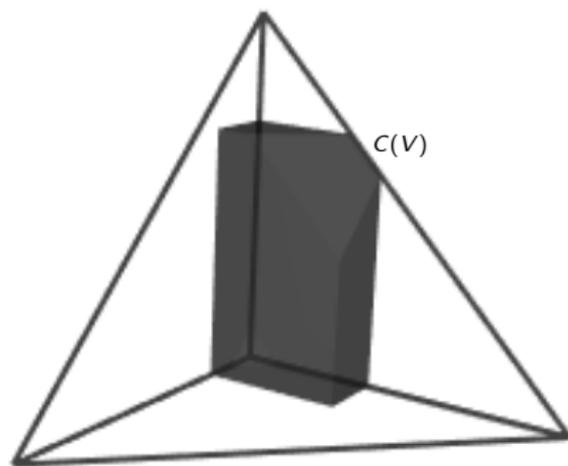
EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



EJEMPLO DE 4 JUGADORES

$$E = 10, d = (2, 4, 7, 9); \mathcal{F} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}\}$$



- Cuanto más grande sea \mathcal{F} , mayor complejidad tendrá el algoritmo.
- Si solo hay coaliciones de un jugador en \mathcal{F} , el algoritmo solo tendrá un paso.
- Se ejecuta con el código exterior(E,d)

ALGORITMO INTERIOR AL NÚCLEO

PARTICIÓN DEL NÚCLEO

Proposición: Sea $(E,d) \in B^N$ un juego de bancarrota, $v \in G^N$ su juego asociado y $C(v)$ el núcleo de dicho juego, existe un problema de bancarrota $(E,d') \in B^N$, siendo $v' \in G^N$ su juego asociado y $C(v')$ el núcleo de dicho juego t.q $\frac{\sum_{i \in N} d_i}{2} = E$ y $C(v') \subseteq C(v)$. Llamaremos al núcleo de este juego pieza base.

Una posible pieza base será el núcleo del problema (E,d') con d' :

$$d'_i = \min \left\{ \frac{2E - \sum_{i=1}^n d_i}{n - i + 1}, d_i \right\}, i \in N$$

DESCOMPOSICIÓN DEL NÚCLEO DE BANCARROTA.

Sea $(E,d) \in B^N$ un juego de bancarrota, $v \in G^N$ su juego de bancarrota asociado y (E,d') su problema base asociado. Se definen los problemas de bancarrota de cada jugador $(E_{\{i\}}, d_{\{i\}})$ según la siguiente tabla:

	$d'_i < d_i$
$E_{\{i\}}$	$E - d'_i$
$d_{\{i\}}$	$(d_{\{i\}})_j = \begin{cases} d_i - d'_i & \text{si } j = i \\ d_i & \text{si } j \neq i \end{cases}$
$a_{\{i\}}$	$a_{\{i\}}(j) = \begin{cases} d'_i & j = i \\ 0 & j \neq i \end{cases}$

Denotamos por $v_{\{i\}}, i = 1, \dots, n$ a cada juego de bancarrota asociado al problema $(E_{\{i\}}, d_{\{i\}})$.

DESCOMPOSICIÓN DEL NÚCLEO DE BANCARROTA.

PROPOSICIÓN:

$\forall i \in N$, sea $v_{\{i\}}$ el juego asociado a cada problema de bancarrota de cada jugador $(E_{\{i\}}, d_{\{i\}})$, se tiene que:

- 1 $C(v_{\{i\}}) \subset C(v)$
- 2 $\text{Vol}(C(v') \cap C(v_{\{i\}})) = 0$

PROPOSICIÓN:

Sea $(E, d) \in B^N$ un juego de bancarrota, $v \in G^N$ su juego asociado, (E, d') su juego base y $(E_{\{i\}}, d_{\{i\}})$, $i = 1, \dots, n$, sus juegos asociados a cada jugador, se tiene que:

$$\cup_{i \in N} C(v_{\{i\}}) \cup C(v') = C(v)$$

EJEMPLO DE DESCOMPOSICIÓN PARA 4 JUGADORES

Sea el problema de bancarrota (E,d) con $E=10$ y $d=(2,4,7,9)$.

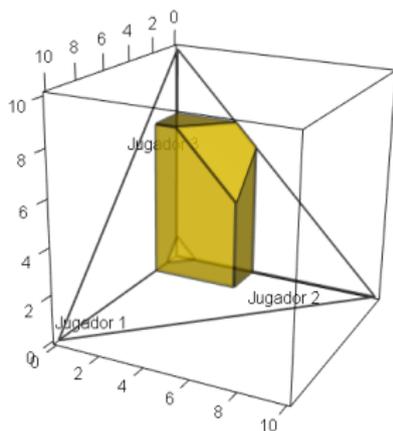
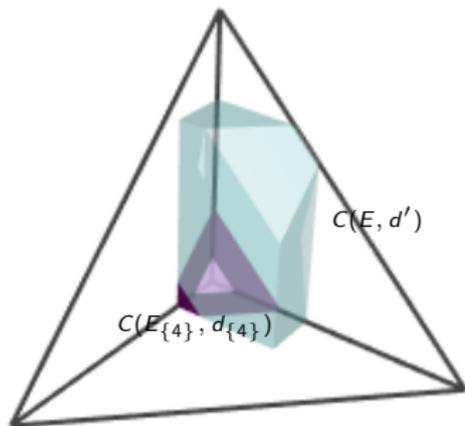


FIGURE: Nucleo del juego (E,d)

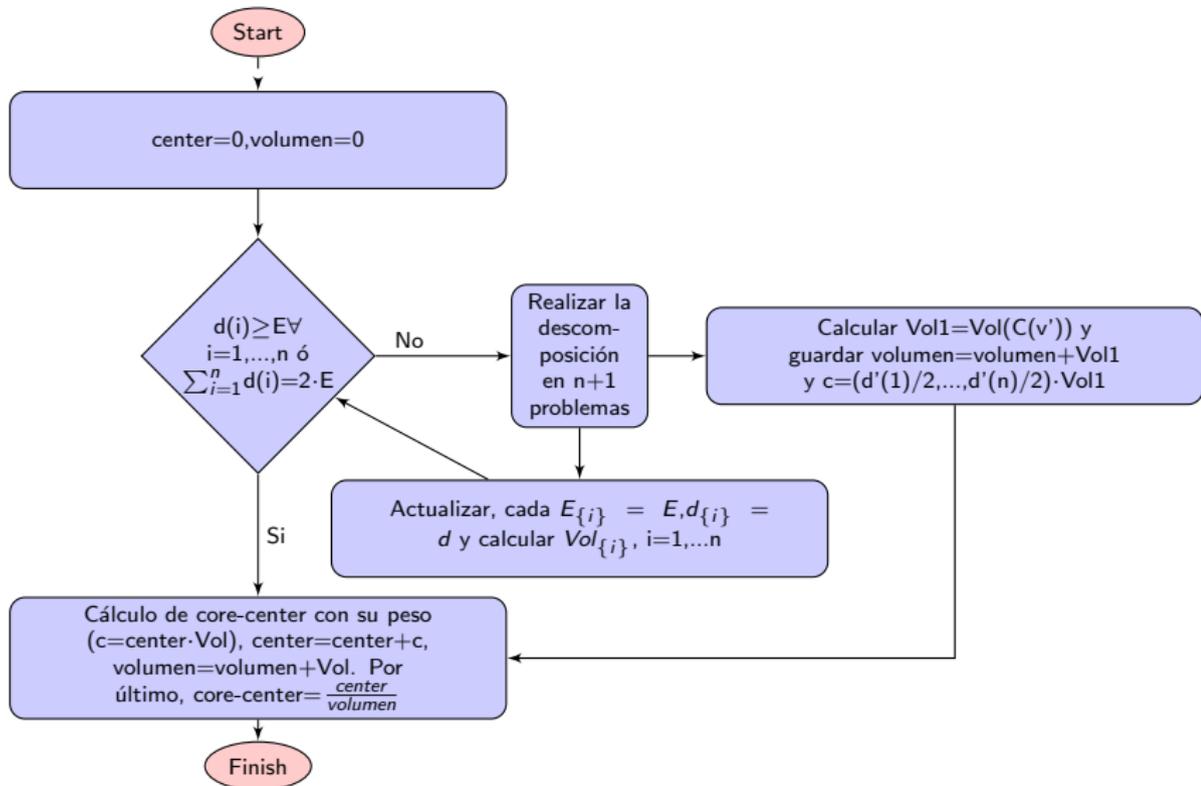
EJEMPLO DE DESCOMPOSICIÓN PARA 4 JUGADORES (2)

Dicho juego se descompone en 2:

- (E, d') , con $d' = (2, 4, 7, 7)$
- $(E_{\{4\}}, d_{\{4\}})$ con $E_{\{4\}} = 3$ y $d_{\{4\}} = (2, 4, 7, 2)$
- Nótese que el resto de problemas son vacíos ya que $E_{\{i\}} - (E - d_i) = 0$

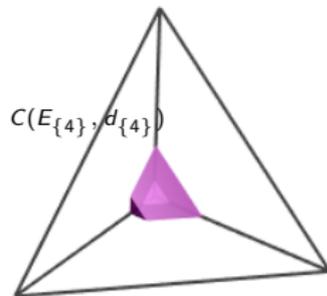
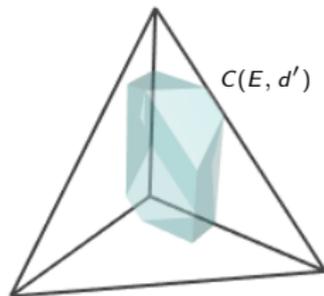
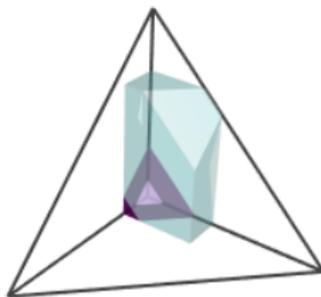


FORMULACIÓN DEL ALGORITMO



EJEMPLO ALGORITMO INTERIOR

$$E=10, d=(2,4,7,9)$$



- Cuantos más lejos estén las demandas del estado, menor número de juegos aparecerán en la descomposición.
- La familia \mathcal{F} de los juegos de la descomposición serán más simples que la familia \mathcal{F} inicial
- Se ejecuta con el código interior(E,d)

ALGORITMO COMBINADO

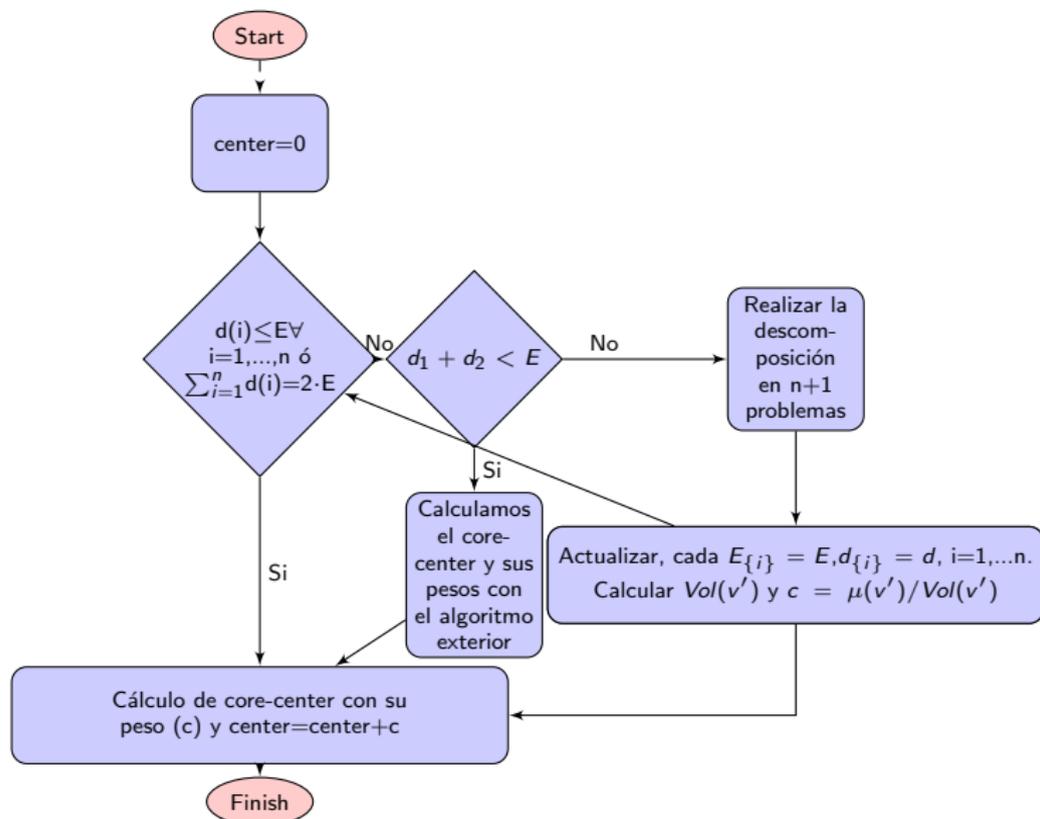
ALGORITMO INTERIOR

Los problemas $(E_{\{i\}}, d_{\{i\}})$ tienen menos cortes que el problema inicial.

ALGORITMO EXTERIOR

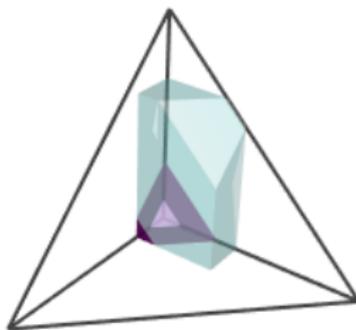
Mayor complejidad cuando entran más cortes ya que falta más espacio por cubrir del conjunto de imputaciones.

FORMULACIÓN



EJEMPLO

$$E=10, d=(2,4,7,9).$$



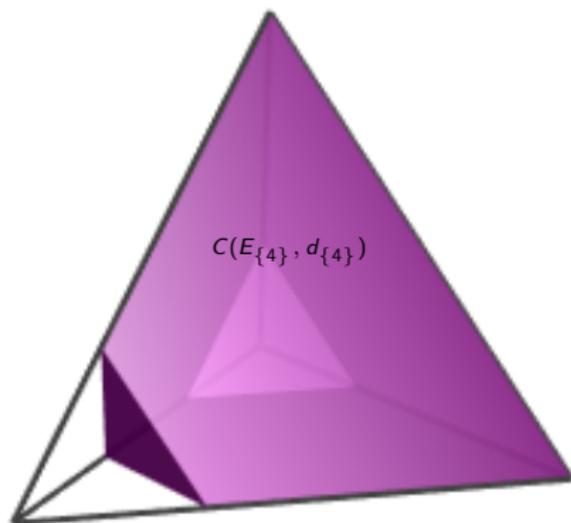
CASO 2.1:

Calculamos $\mu(v')$ y $\text{Vol}(E',d')$. Guardamos dicho core-center promediado con su volumen y, el otro problema es:

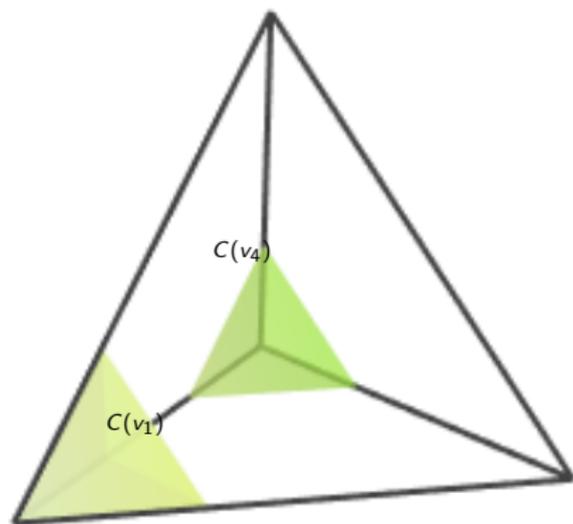
$$E_{\{4\}} = 3, d_{\{4\}} = (2, 4, 7, 2), a_{\{4\}} = (0, 0, 0, 7)$$

CASO 2.2:

Aplicamos algoritmo exterior del siguiente core:



Iteración número 2: $E=3, d=(2,4,7,2)$. Algoritmo exterior: $\mathcal{F}=\{\{1\}, \{4\}\}$:



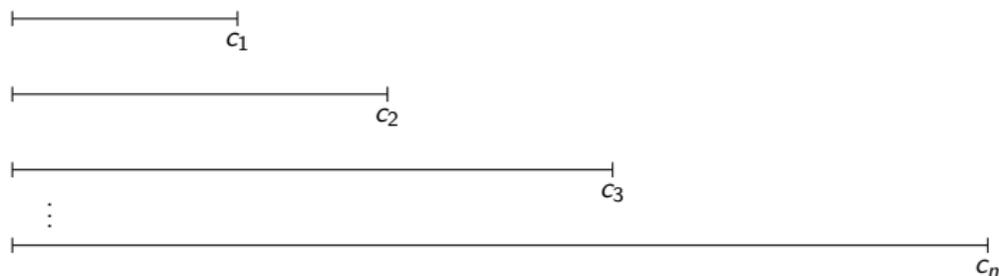
- Usamos las partes positivas de los dos algoritmos anteriores.
- Tenemos que determinar en que problemas será mejor descomponer y en cuales será mejor usar el algoritmo exterior.
- Dependerá de las demandas y del número de jugadores.
- Se ejecuta con el código combinación(E,d,t), siendo t la coalición a partir de la cual se realizará la descomposición, determinada por el usuario.
- Con un número de jugadores alto, es una buena forma de aproximación.

ÍNDICE

- 1 DEFINICIONES BÁSICAS
- 2 PROBLEMA DE BANCARROTA
 - Cálculo del core-center
- 3 PROBLEMA DEL AEROPUERTO
 - Core-Center
 - Algoritmos de aproximación
 - Algoritmos Exactos
- 4 BIBLIOGRAFÍA

PROBLEMA DEL AEROPUERTO

Un problema del aeropuerto es un par (N, C) siendo $N = \{1, \dots, n\}$ el conjunto de todas las compañías que operan en el aeropuerto, ordenadas según la necesidad de la pista de menor a mayor, y $C \in \mathbb{R}^n$ el vector donde cada componente se corresponde con el coste de mantenimiento de la pista de aterrizaje de cada agente, es decir, cada compañía, $i \in N$, se caracteriza por el coste de mantenimiento de los tramos de pista necesarios, el i -ésimo elemento del vector C , c_i .



JUEGO TU ASOCIADO (JUEGO DEL AEROPUERTO)

Sea (N, C) problema del aeropuerto, su correspondiente juego de costes asociado es el juego (N, c) , con función característica

$$c(S) = \max_{i \in S} c_i \quad \forall S \subset N$$

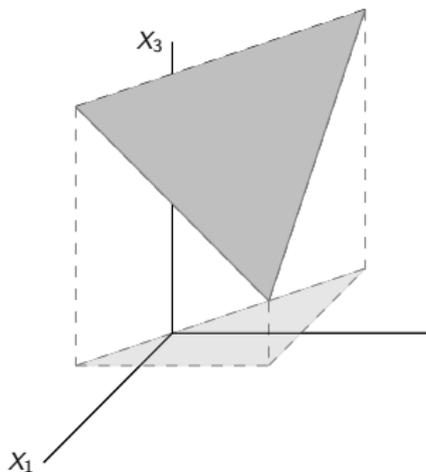
JUEGO TU ASOCIADO (JUEGO DEL AEROPUERTO)

Sea (N, C) problema del aeropuerto, su correspondiente juego de costes asociado es el juego (N, c) , con función característica

$$c(S) = \max_{i \in S} c_i \quad \forall S \subset N$$

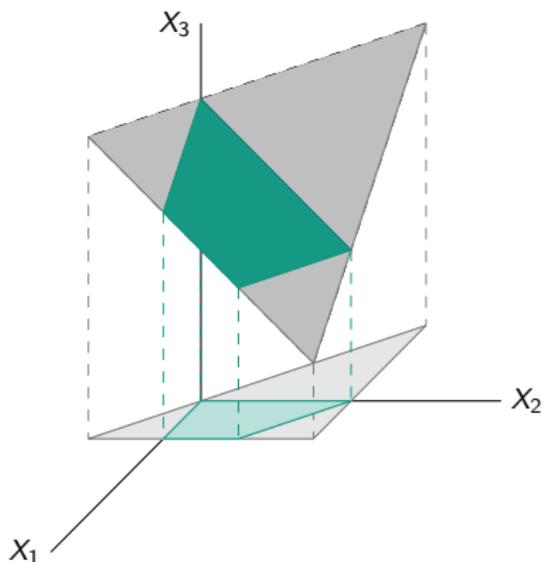
Conjunto de imputaciones:

$$I(N, c) = \left\{ \mathbf{x} \in \mathbb{R}^n : x_i \leq c_i, \sum_{i \in N} x_i = c_n \right\}$$



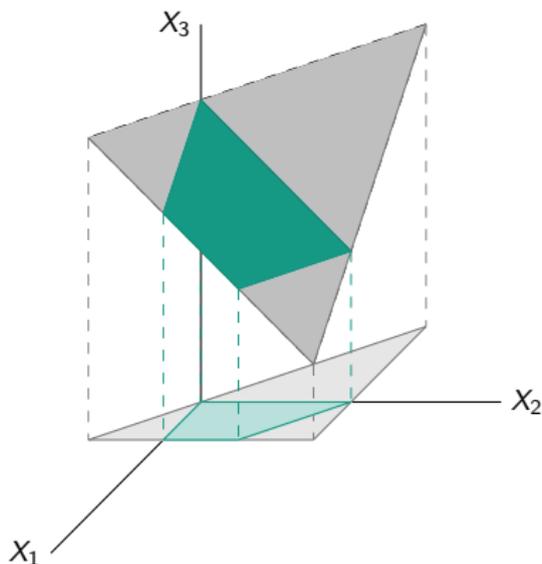
Core del juego:

$$\begin{aligned}
 C(N, c) &= \{ \mathbf{x} \in I(N, c) : x(S) \leq c(S), \forall S \subseteq N \} \\
 &= \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i \in S} x_i \leq \max_{i \in S} c_i, \forall S \subseteq N, \sum_{i \in N} x_i = c_n \right\}
 \end{aligned}$$



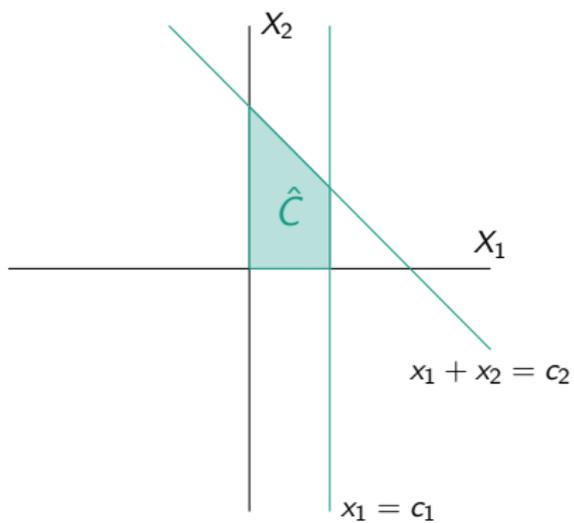
Core del juego:

$$\hat{C}(N, c) = \left\{ \mathbf{x} \in \mathbb{R}^{n-1} : \mathbf{x} \geq 0, \sum_{j \leq i} x_j \leq c_i \forall i < n \right\}$$

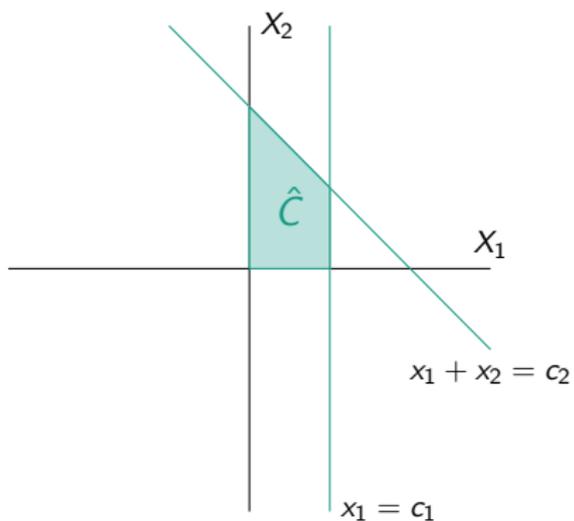


$$\hat{C}(N, c) = \{\mathbf{x} \in \mathbb{R}^{n-1} : A\mathbf{x} \leq \mathbf{b}\}$$

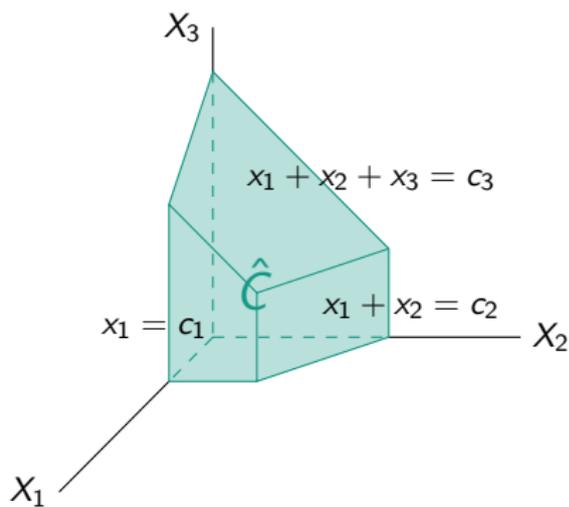
$$A = \begin{pmatrix} -I_{n-1} \\ A' \end{pmatrix} = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \\ 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix}$$



Core proyectado (3 agentes)



Core proyectado (3 agentes)



Core proyectado (4 agentes)

CORE-CENTER DEL JUEGO DEL AEROPUERTO

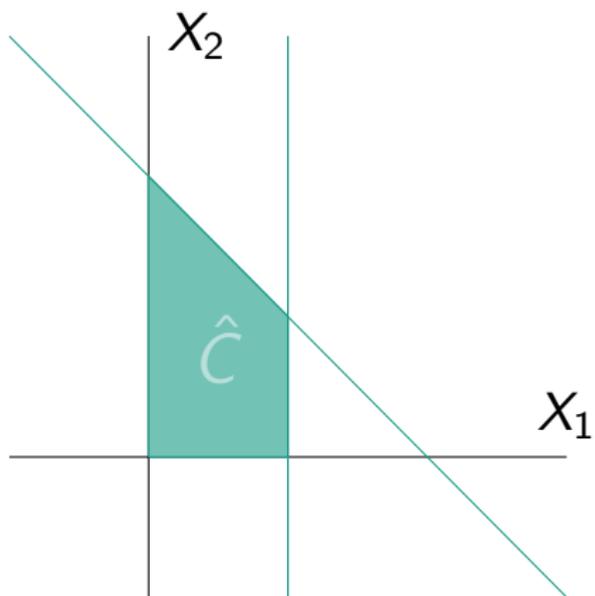
En el artículo González Díaz y Sánchez Rodríguez (2016) se obtuvo que

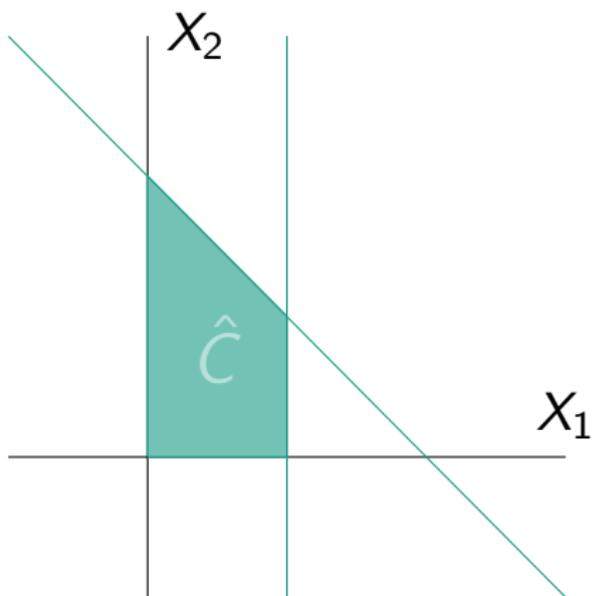
$$\hat{\mu}_j(C) = \frac{V_n(c_1, \dots, c_j, c_j, \dots, c_{n-1})}{V_{n-1}(c_1, \dots, c_j, \dots, c_{n-1})} \quad \forall j < n \quad (1)$$

$$\mu_n(C) = \frac{V_n(c_1, \dots, c_n)}{V_{n-1}(c_1, \dots, c_{n-1})} \quad (2)$$

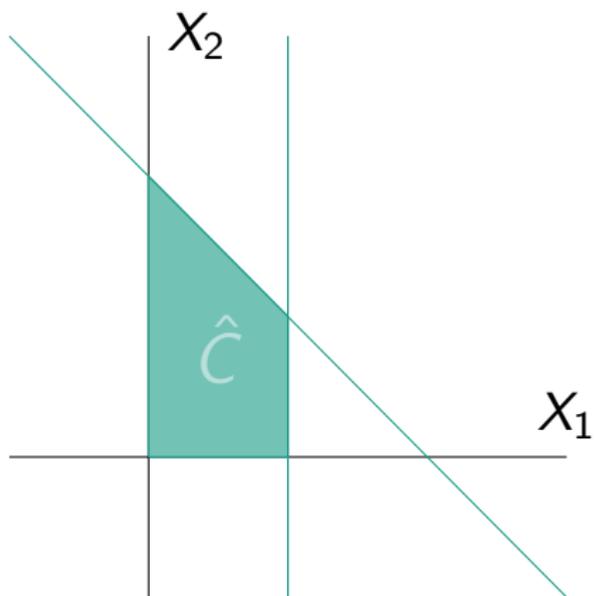
Donde el volumen el core proyectado se puede calcular de forma recursiva mediante la siguiente expresión

$$V_k(c_1, \dots, c_k) = \begin{cases} c_1 & \text{si } k = 1 \\ \frac{c_2^2}{2} - \frac{(c_2 - c_1)^2}{2} & \text{si } k = 2 \\ \frac{c_k^k}{k!} - \frac{(c_k - c_1)^k}{k!} - \sum_{i=2}^{k-1} \frac{(c_k - c_i)^{k-i+1}}{(k-i+1)!} & V_{i-1}(c_1, \dots, c_{i-1}) \end{cases} \quad (3)$$

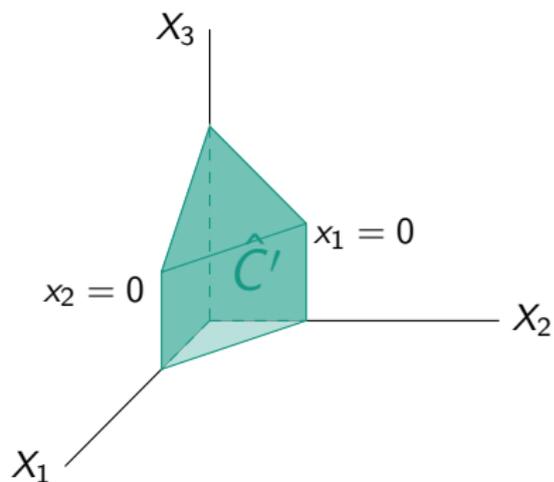


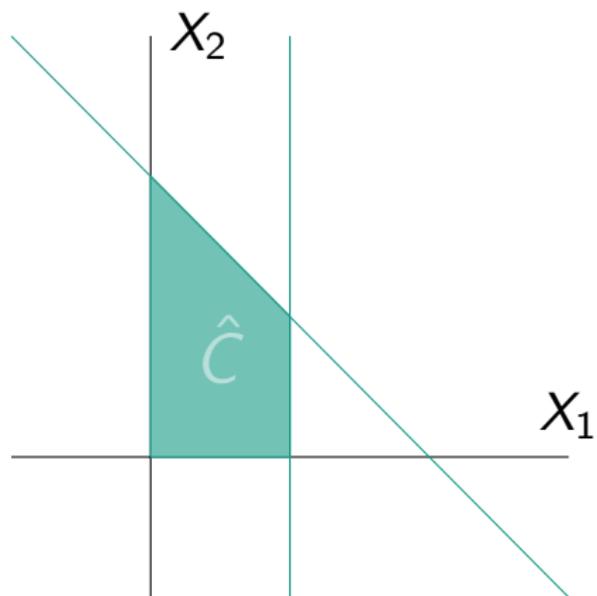


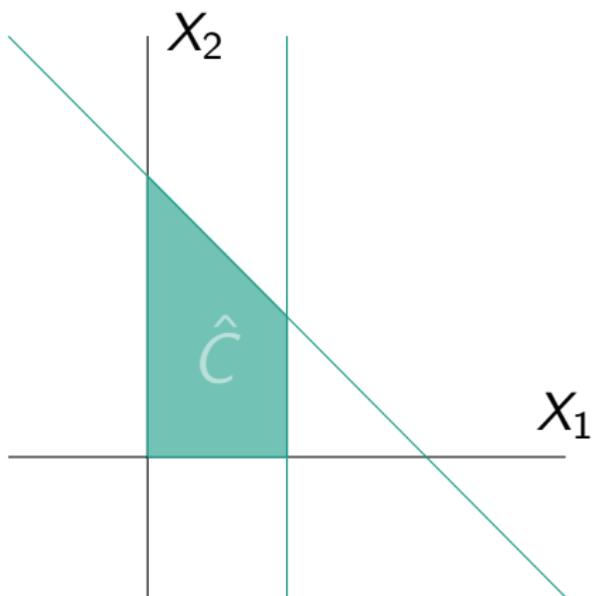
$$\mu_1 = \frac{V_3(c_1, c_1, c_2)}{V_2(c_1, c_2)} \quad (4)$$



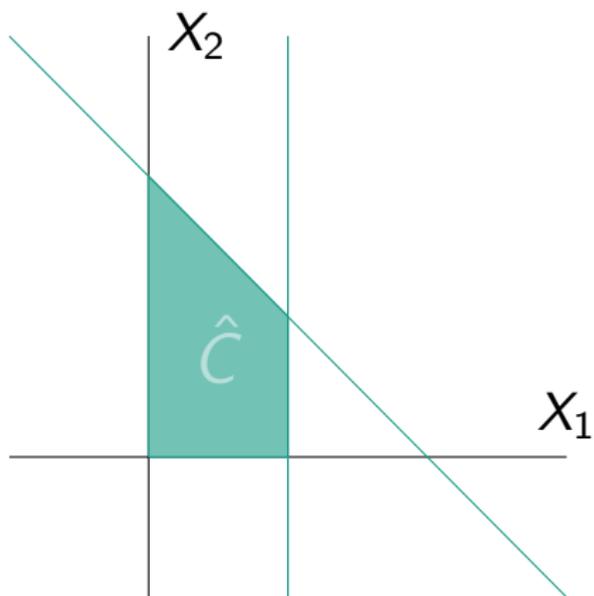
$$\mu_1 = \frac{V_3(c_1, c_1, c_2)}{V_2(c_1, c_2)} \quad (4)$$



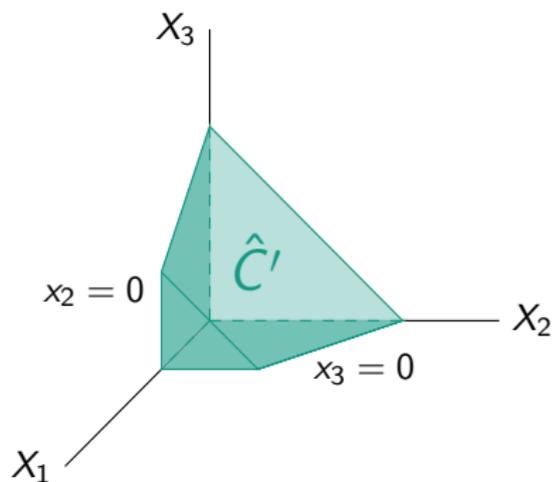




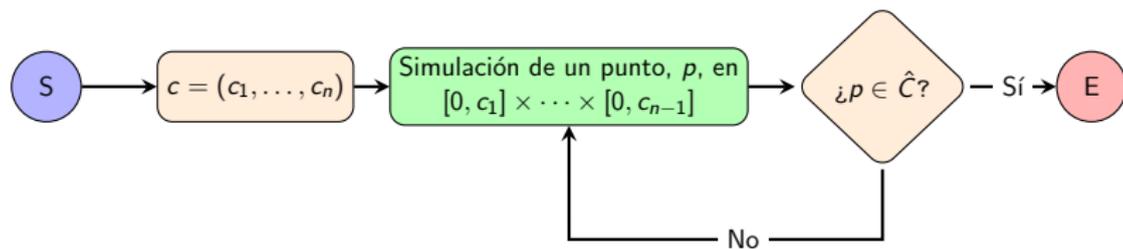
$$\mu_2 = \frac{V_3(c_1, c_2, c_2)}{V_2(c_1, c_2)} \quad (5)$$



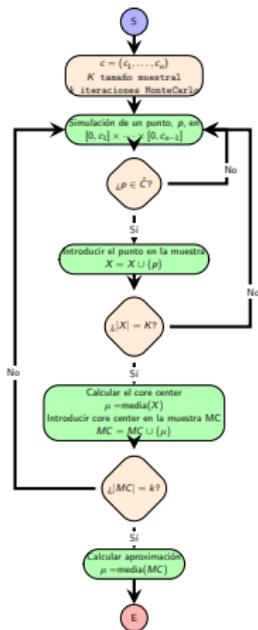
$$\mu_2 = \frac{V_3(c_1, c_2, c_2)}{V_2(c_1, c_2)} \quad (5)$$



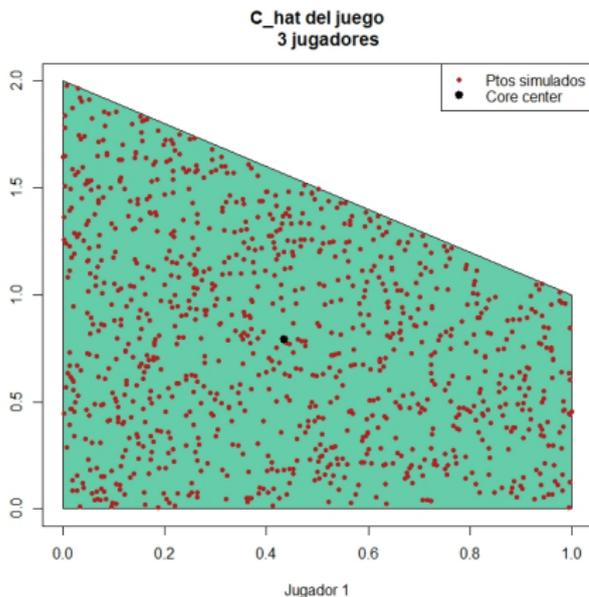
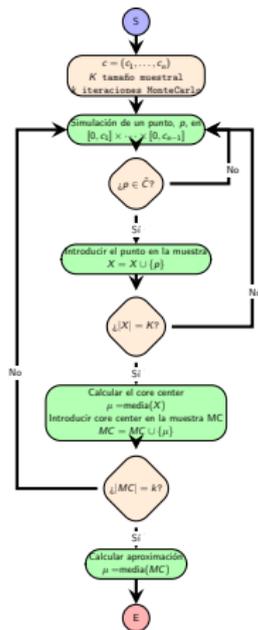
MÉTODO DE ACEPTACIÓN/RECHAZO



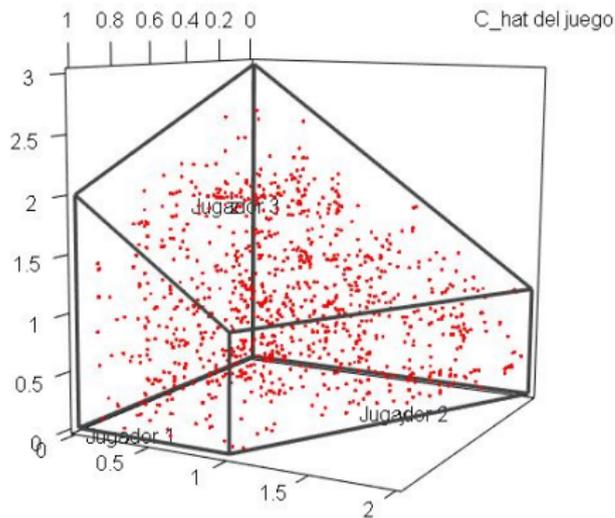
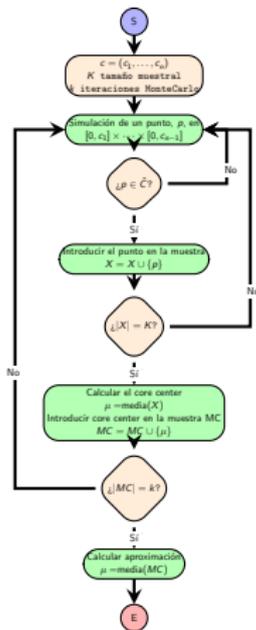
MÉTODO DE ACEPTACIÓN/RECHAZO



MÉTODO DE ACEPTACIÓN/RECHAZO



MÉTODO DE ACEPTACIÓN/RECHAZO



NÚMERO MEDIO DE REPETICIONES

El número medio de comparaciones necesarias para obtener un punto dentro del conjunto deseado sería

$$c_{opt} = \frac{\prod_{i=1}^{n-1} c_i}{V_{n-1}(c_1, \dots, c_{n-1})}$$

Por esto, una de los inconvenientes más relevantes de este método es que para dimensiones elevadas este algoritmo resultará muy lento.

MÉTODO DE INVERSIÓN

IDEA GENERAL

En este método, detallado en Cao Abad (2002), a partir de la función de densidad $f(x)$, se calcula la función de distribución $F(x)$, esta se podría considerar como una nueva variable con distribución uniforme, es decir, $U = F(x) \sim U(0, 1)$, por el teorema anterior se sabe que $F^{-1}(U)$ tiene como función de distribución F . La idea sería generar puntos uniformemente distribuidos, u , y devolver $F^{-1}(u)$ como simulación.

MÉTODO DE INVERSIÓN

IDEA GENERAL

En este método, detallado en Cao Abad (2002), a partir de la función de densidad $f(x)$, se calcula la función de distribución $F(x)$, esta se podría considerar como una nueva variable con distribución uniforme, es decir, $U = F(x) \sim U(0, 1)$, por el teorema anterior se sabe que $F^{-1}(U)$ tiene como función de distribución F . La idea sería generar puntos uniformemente distribuidos, u , y devolver $F^{-1}(u)$ como simulación.

$$f(x_1, \dots, x_n) = f_1(x_1) \cdot f_2(x_2|x_1) \cdot f_3(x_3|x_1, x_2) \cdot \dots \cdot f_n(x_n/x_1, \dots, x_{n-1})$$

MÉTODO DE INVERSIÓN

En resumen, tenemos las siguientes funciones de distribuciones

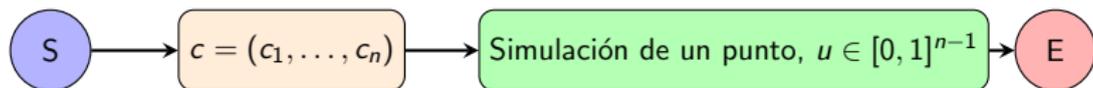
$$F_1(x_1) = \frac{V_{n-1}(x_1, c_2, c_3, \dots, c_{n-1})}{V_{n-1}(c_1, \dots, c_{n-1})}$$

con $0 < x_1 < c_1$

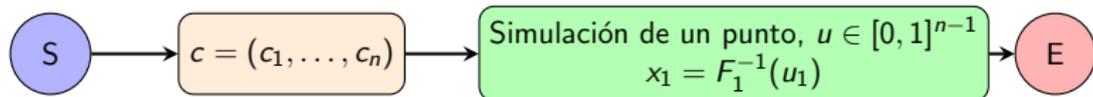
$$F_i(x_i | x_1, \dots, x_{i-1}) = \begin{cases} \frac{V_{n-i}(x_i, c_{i+1} - \sum_{j=1}^{i-1} x_j, \dots, c_{n-1} - \sum_{j=1}^{i-1} x_j)}{V_{n-i}(c_i - \sum_{j=1}^{i-1} x_j, \dots, c_{n-1} - \sum_{j=1}^{i-1} x_j)} & \text{si } i < n - 1 \\ \frac{x_i}{V_{n-i}(c_i - \sum_{j=1}^{i-1} x_j, \dots, c_{n-1} - \sum_{j=1}^{i-1} x_j)} & \text{si } i = n - 1 \end{cases}$$

con $0 < x_i < c_i - \sum_{j=1}^{i-1} x_j$.

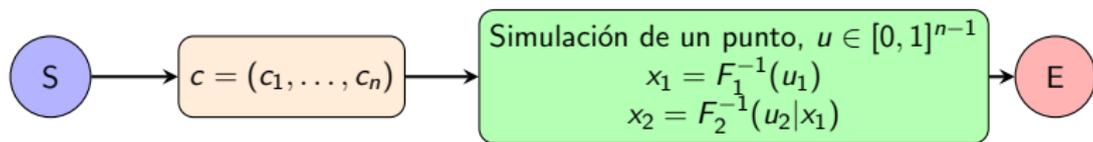
MÉTODO DE INVERSIÓN



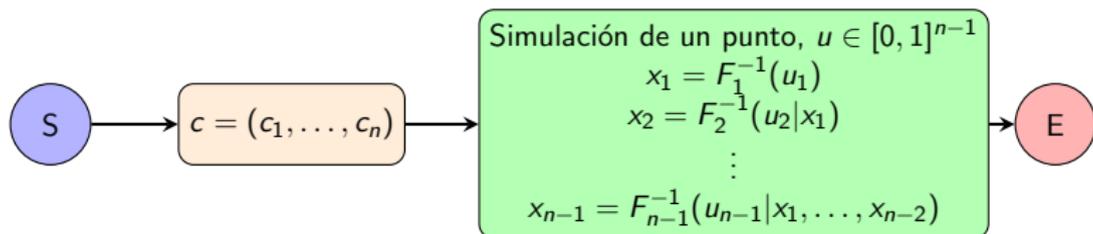
MÉTODO DE INVERSIÓN



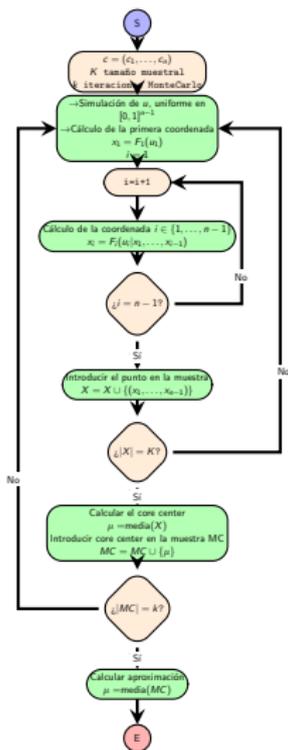
MÉTODO DE INVERSIÓN



MÉTODO DE INVERSIÓN



MÉTODO DE INVERSIÓN



$$F_1(x_1) = \frac{V_3(x_1, c_2, c_3)}{V_3(c_1, c_2, c_3)}$$

$$F_2(x_2|x_1) = \frac{V_2(x_2, c_3 - x_1)}{V_2(c_2 - x_1, c_3 - x_1)}$$

$$F_3(x_3|x_1, x_2) = \frac{x_3}{V_1(c_3 - x_1 - x_2)}$$

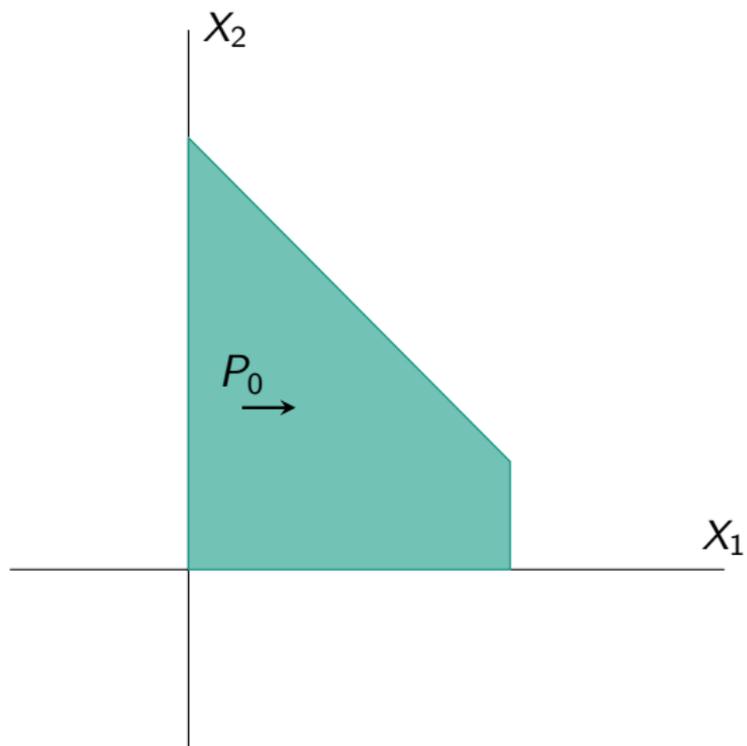
Este caso es más complejo para obtener las expresiones explícitas de las funciones inversas.

MÉTODO HIT-AND-RUN

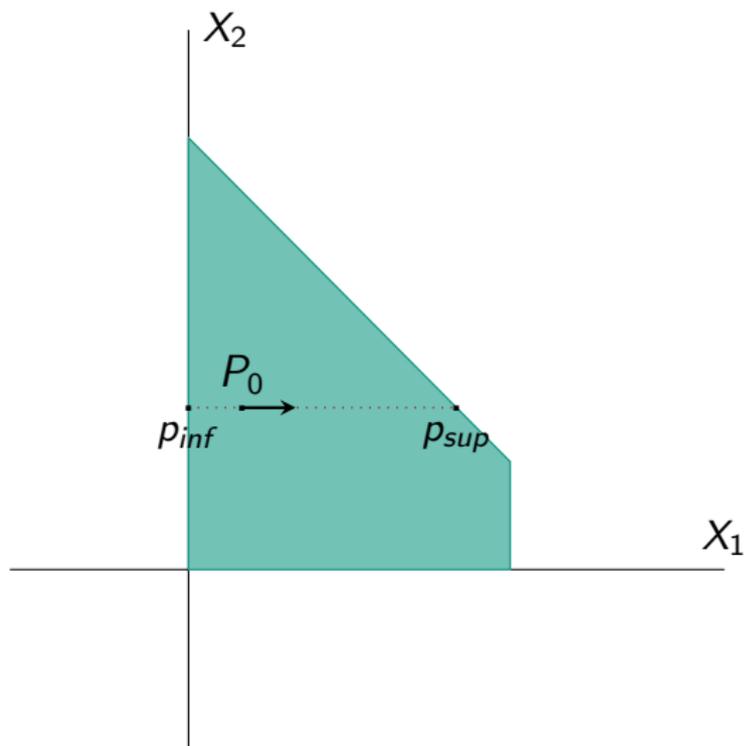
IDEA GENERAL

En este algoritmo (ver Smith (1984) y Emiris y Fisikopoulos (2018)) a partir de un punto interior se simula uniformemente una dirección en la que nos podemos mover, es ahora cuando simulamos un punto uniformemente distribuido en la intersección de la recta que define dicho vector director y el punto original, con nuestro core proyectado.

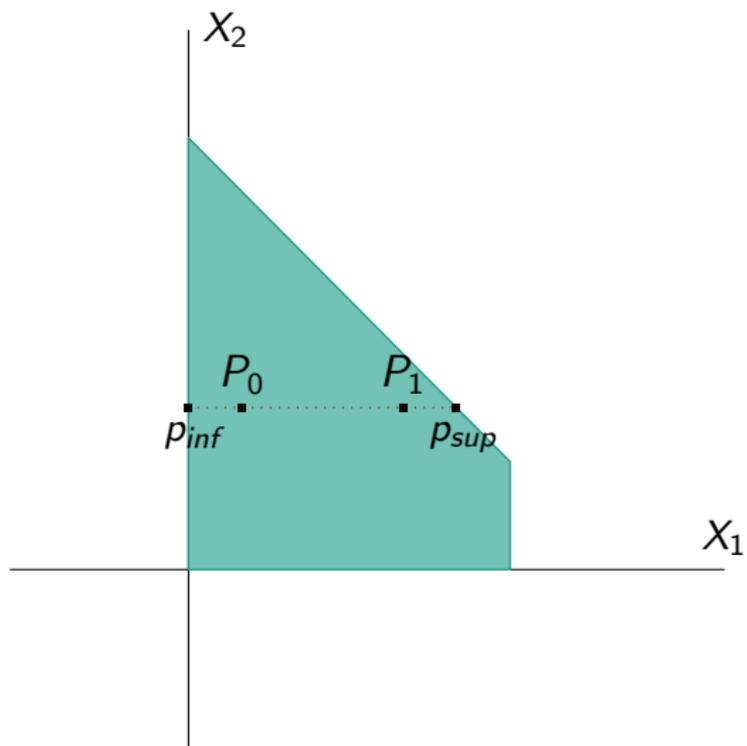
MÉTODO HIT-AND-RUN



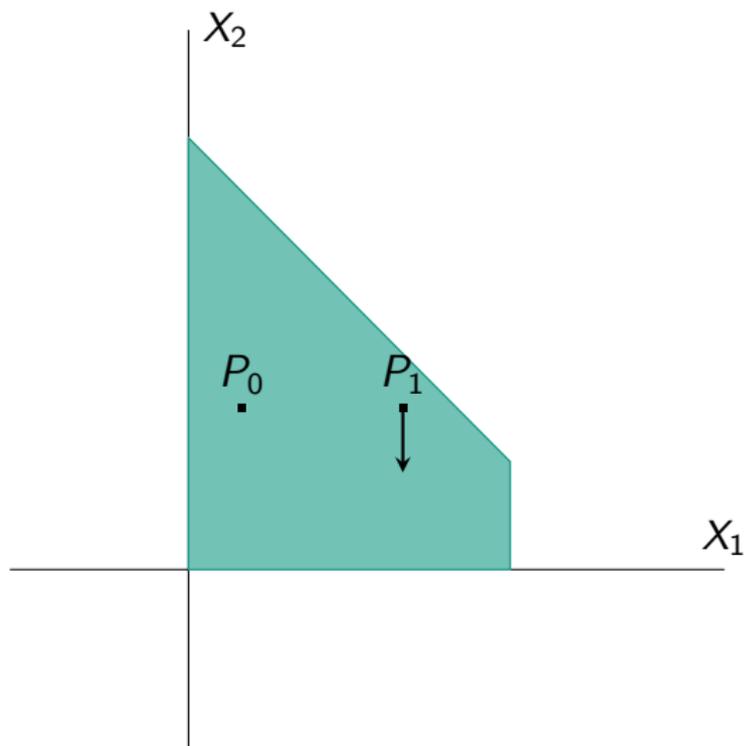
MÉTODO HIT-AND-RUN



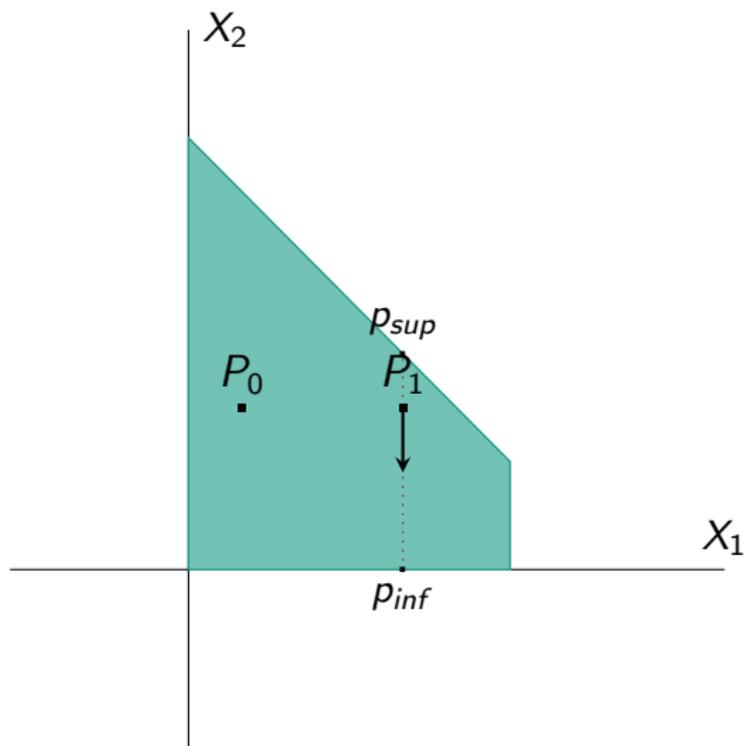
MÉTODO HIT-AND-RUN



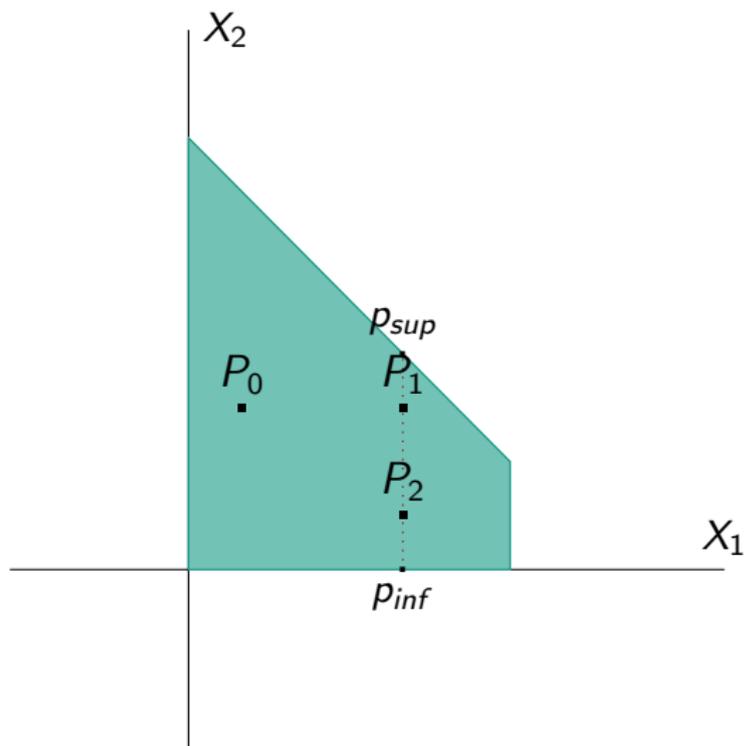
MÉTODO HIT-AND-RUN



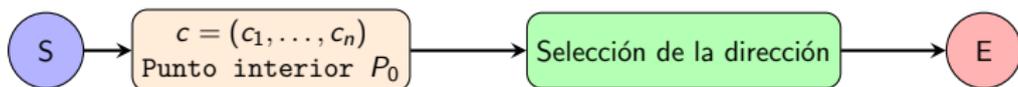
MÉTODO HIT-AND-RUN



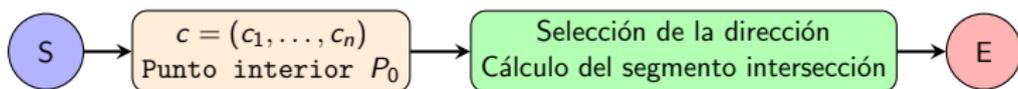
MÉTODO HIT-AND-RUN



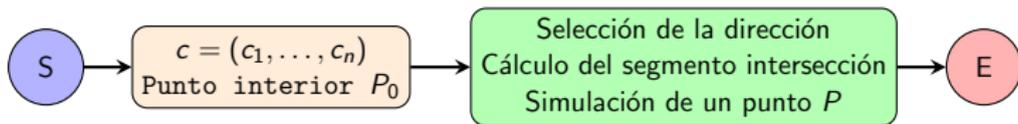
MÉTODO DE HIT-AND-RUN



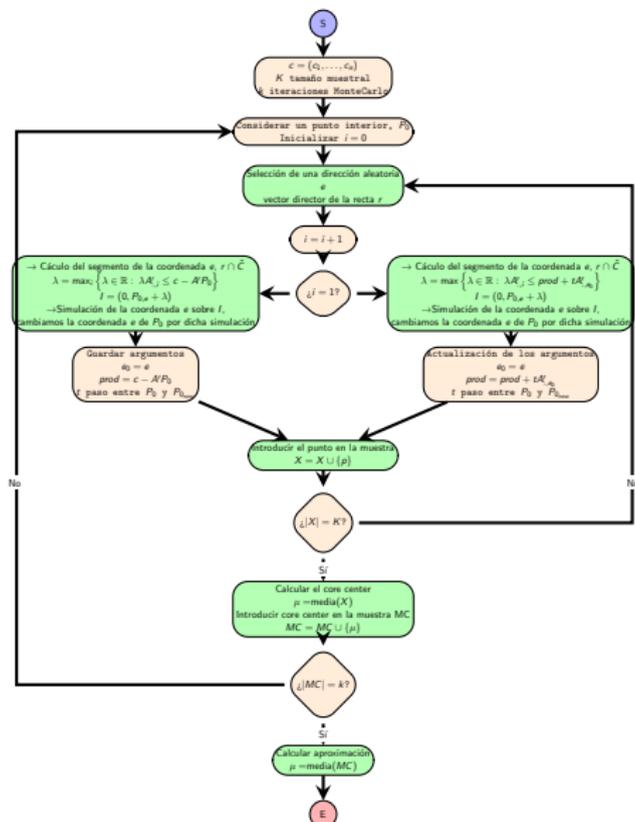
MÉTODO DE HIT-AND-RUN



MÉTODO DE HIT-AND-RUN

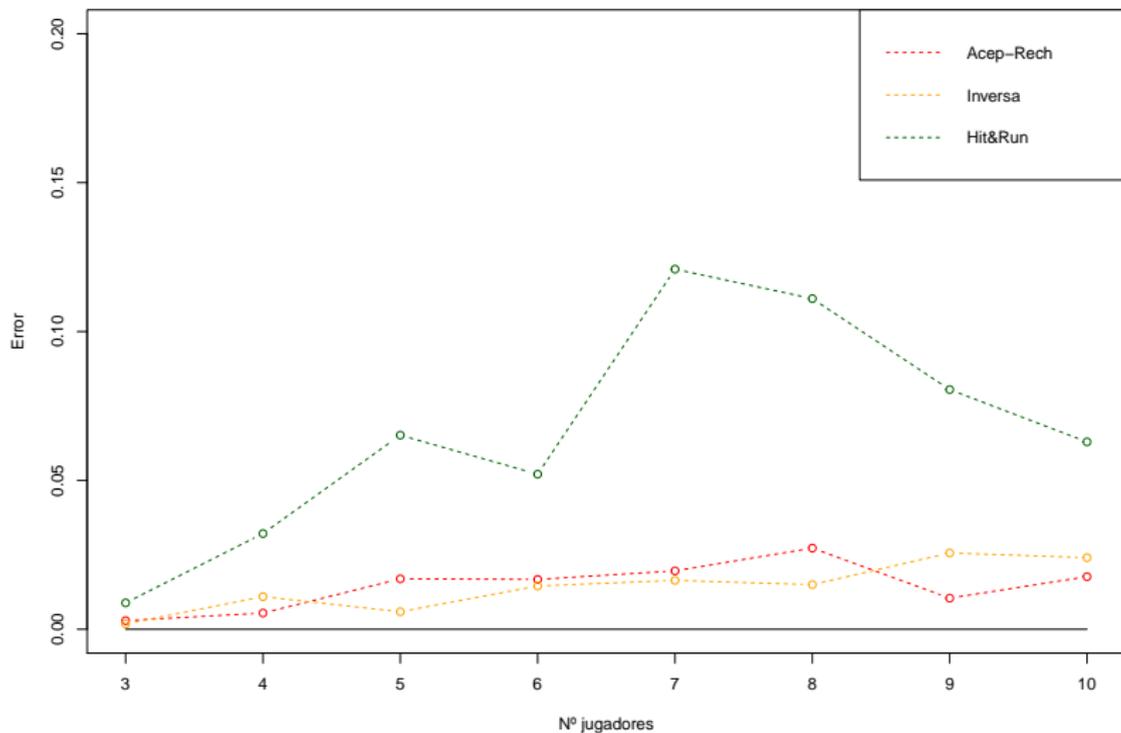


MÉTODO DE HIT-AND-RUN



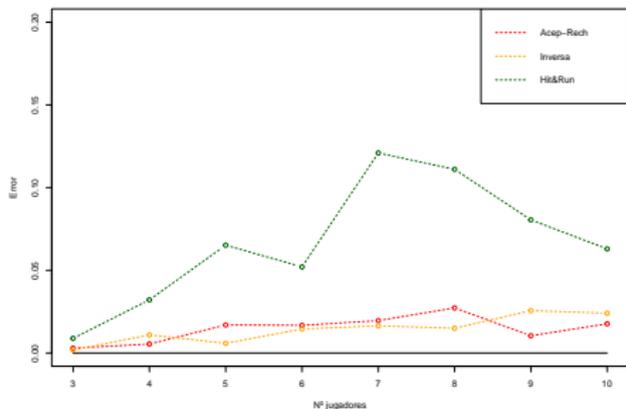
COMPARACIÓN ENTRE MÉTODOS

Error de Aproximación (nº iter MonteCarlo=10, tamaño muestral=1e3)

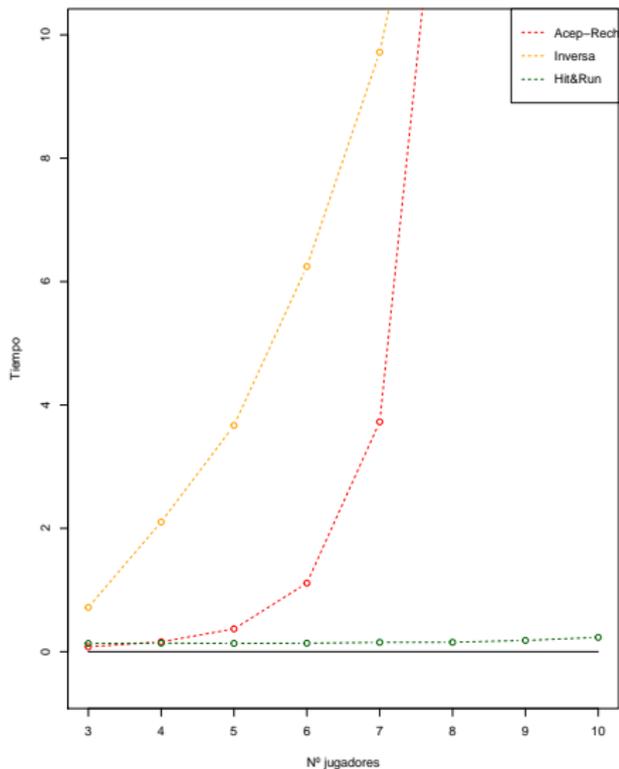


COMPARACIÓN ENTRE MÉTODOS

Error de Aproximación (nº iter MonteCarlo=10, tamaño muestral=1e3)



Tiempo de ejecución (nº iter MonteCarlo=10, tamaño muestral=1e3)



MÉTODO GONZÁLEZ-DÍAZ AT AL.(2016)

CORE-CENTER DEL JUEGO DEL AEROPUERTO

En el artículo González Díaz y Sánchez Rodríguez (2016) se obtuvo que

$$\hat{\mu}_j(C) = \frac{V_n(c_1, \dots, c_j, c_j, \dots, c_{n-1})}{V_{n-1}(c_1, \dots, c_j, \dots, c_{n-1})} \quad \forall j < n \quad (6)$$

$$\mu_n(C) = \frac{V_n(c_1, \dots, c_n)}{V_{n-1}(c_1, \dots, c_{n-1})} \quad (7)$$

Donde el volumen el core proyectado se puede calcular de forma recursiva mediante la siguiente expresión

$$V_k(c_1, \dots, c_k) = \begin{cases} c_1 & \text{si } k = 1 \\ \frac{c_2^2}{2} - \frac{(c_2 - c_1)^2}{2} & \text{si } k = 2 \\ \frac{c_k^k}{k!} - \frac{(c_k - c_1)^k}{k!} - \sum_{i=2}^{k-1} \frac{(c_k - c_i)^{k-i+1}}{(k-i+1)!} & V_{i-1}(c_1, \dots, c_{i-1}) \end{cases} \quad (8)$$

MÉTODO LASSERRE

CÁLCULO DE VOLUMEN LASSERRE (1983)

Analizamos el cálculo de un politopo para nuestro caso y tendríamos la siguiente expresión

$$\text{Vol}(n-1, A, b) = \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{b_i}{|A_{i,i}|} V_i(n-2, \tilde{A}_{t(i)}, \tilde{b}) \quad (9)$$

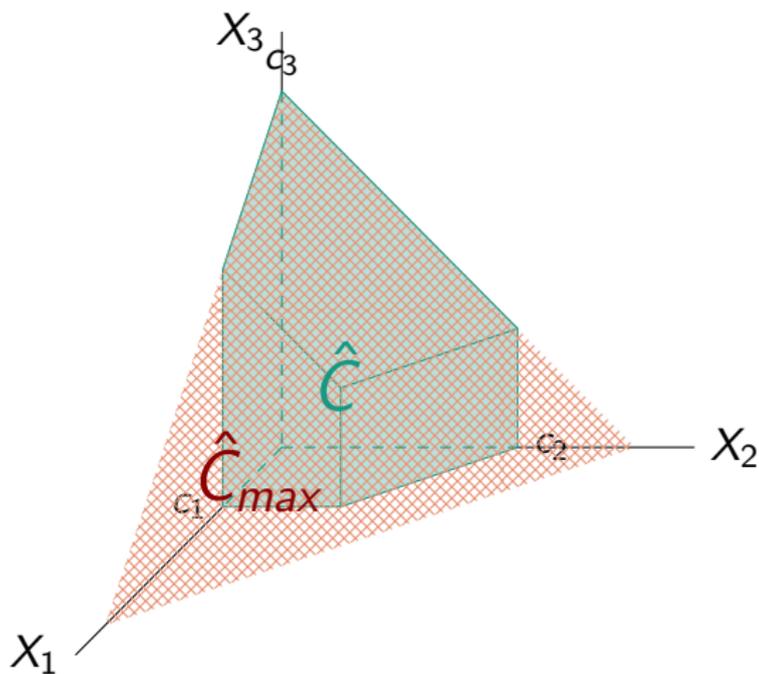
MÉTODO DE LAS CARAS

Desarrollando el anterior método tendríamos

$$\mu_j(N, c) = \frac{1}{n V_{n-1}(c)} \sum_{i=1}^{n-1} \frac{c_i}{\sqrt{i}} \mu_j(F_i) V(F_i) \quad (10)$$

donde F_i corresponde con el juego de las caras importantes del problema del aeropuerto González Díaz y Sánchez Rodríguez (2016). Cada cara se coincide con el core del producto de dos cores de aeropuertos.

TESELACIÓN



MÉTODO DE TESELACIÓN

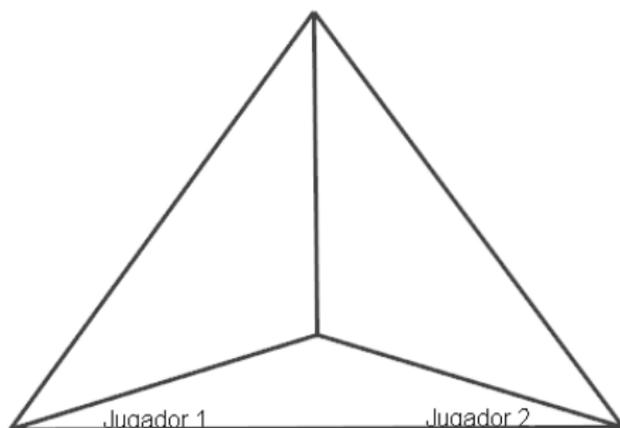
Sea el juego suma formado por los aeropuertos de n jugadores y vectores de costes

$$c^{(i)} = (c_1, \dots, c_i, 0, \dots, 0)$$

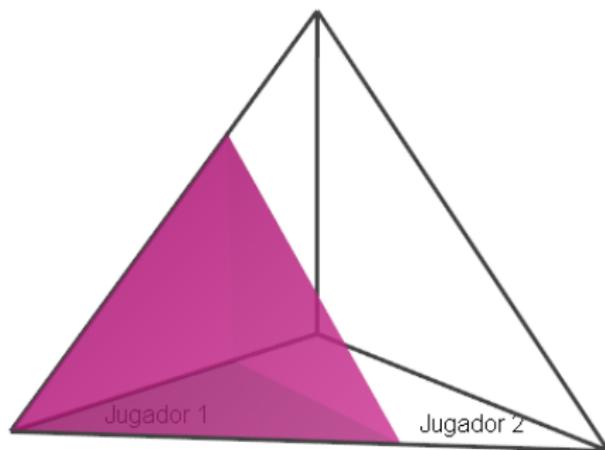
$$c^{(n-i)} = (0, \dots, 0, c_{n-1} - c_i, \dots, c_{n-1} - c_i)$$

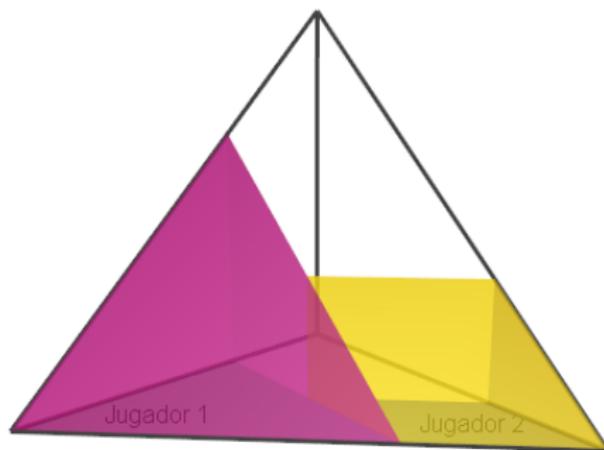
respectivamente. Vemos que el jugador i está involucrado en ambos juegos, es decir, se reparten el coste c_i entre los i primeros agentes y además a este jugador le corresponde otro reparto de la deuda restante en en segundo juego. El juego aditivo será entonces $v^{(i)} = c^{(i)} + c^{(n-i)}$.

EJEMPLO 4 JUGADORES

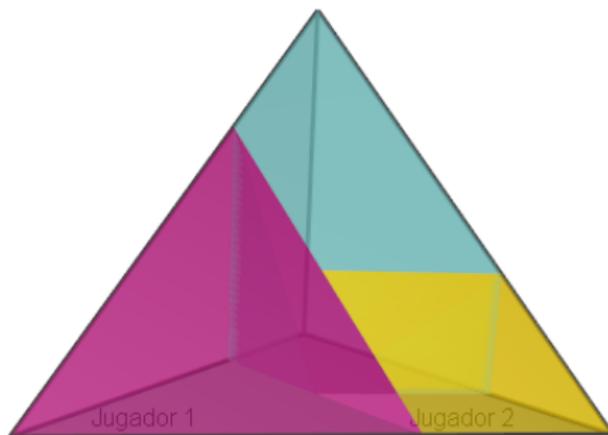


Primer elemento de la partición, $\hat{C}(u^{(1)})$



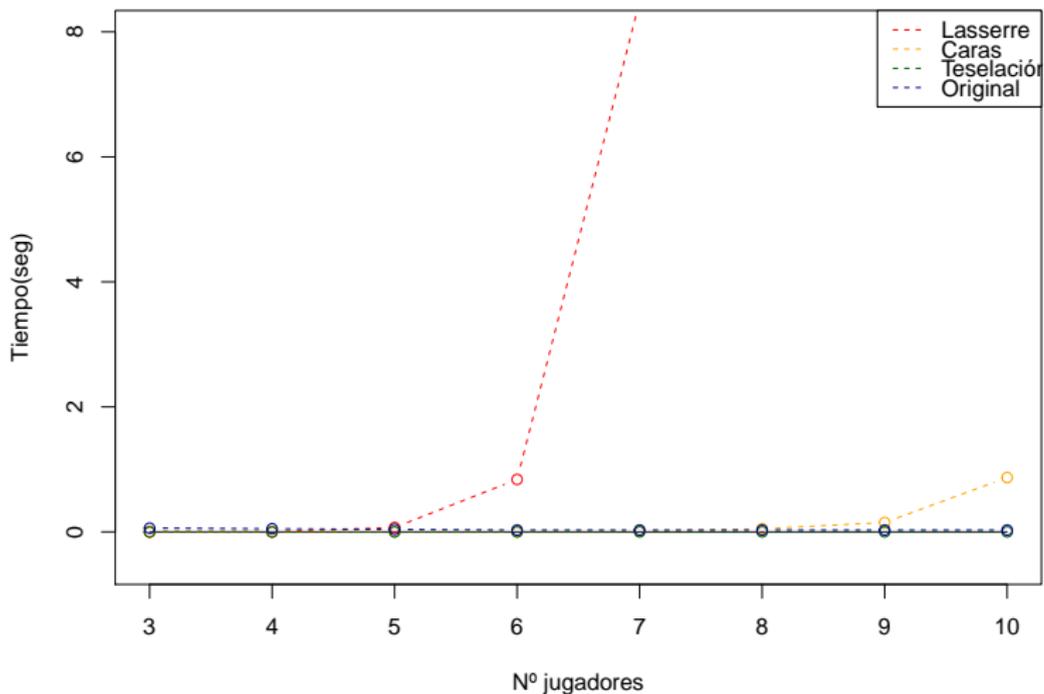
Segundo elemento de la partición, $\hat{C}(u^{(2)})$ 

El último elemento, $\hat{C}(u^{(4)})$, coincide con nuestro core original.



COMPARACIÓN ENTRE LOS MÉTODOS EXACTOS

Tiempo de ejecución



OBJETIVO

- Cálculo del core-center en esta clase de juegos

OBJETIVO

- Cálculo del core-center en esta clase de juegos
- Extrapolación de los métodos
 - Algoritmo Hit-and-Run
 - Algoritmo de Lasserre

OBJETIVO

- Cálculo del core-center en esta clase de juegos
- Extrapolación de los métodos
 - Algoritmo Hit-and-Run
 - Algoritmo de Lasserre
- Caso particular: Agentes simétricos
 - Algoritmo Hit-and-Run
 - Algoritmo de Caras
 - Algoritmo de Teselación

ÍNDICE

- 1 DEFINICIONES BÁSICAS
- 2 PROBLEMA DE BANCARROTA
 - Cálculo del core-center
- 3 PROBLEMA DEL AEROPUERTO
 - Core-Center
 - Algoritmos de aproximación
 - Algoritmos Exactos
- 4 BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Cao Abad, R. (2002) Introducción a la simulación y a la teoría de colas. Netbiblo.
- Emiris, I. y Fisikopoulos, V. Practical polytope volume approximation. ACM Transactions on Mathematical Software (TOMS) 44(4) 38–59.
- Estévez-Fernández, A., Fiestras-Janeiro, M. G., Mosquera, M. A., y Sánchez-Rodríguez, E. (2012). A bankruptcy approach to the core cover. Mathematical Methods of Operations Research, 76(3), 343-359.
- González Díaz, J. y Sánchez Rodríguez, E. (2007) A natural selection from the core of a TU game: the core-center. International Journal of Game Theory 36, 27–46.
- González Díaz, J. y Sánchez Rodríguez, E. (2016) Airport games: the core and its center. Mathematical Social Sciences 82, 105–115.

- Lasserre, J. An analytical expression and an algorithm for the volume of a convex polyhedron. *Journal of optimization theory and applications*, 39(3), 363–377.
- Mirás Calvo, M. Á., Quinteiro Sandomingo, C. y Sánchez Rodríguez, E. (2018). The core-center of a bankruptcy problem.
- O'Neill, B. (1982). A problem of rights arbitration from the Talmud. *Mathematical Social Sciences* 2, 345–371.
- Smith, R. (1984). Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research* 32(6), 1296–1308.
- Thomson, W. (2003). Axiomatic and game-theoretic analysis of bankruptcy and taxation problems: a survey. *Mathematical social sciences*, 45(3), 249-297.