



TUGlabR, un paquete de R para juegos coalicionales

Iago Núñez Lugilde¹, Miguel Ángel Mirás Calvo², Carmen Quinteiro Sandomingo³ y Estela Sánchez Rodríguez¹

¹CINBIO, Universidade de Vigo, SIDOR. Departamento de Estatística e Investigación Operativa.

²Universidade de Vigo, RGEAF, Departamento de Matemáticas.

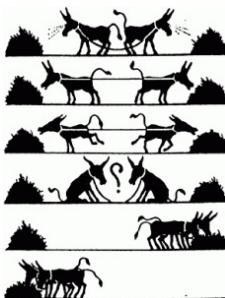
³Universidade de Vigo, Departamento de Matemáticas.

20/10/2022

Introducción

Teoría de juegos

- La Teoría de Juegos se ocupa de modelar y estudiar procesos de decisión en los que intervienen varios agentes o individuos que se comportan de modo estratégico (competitivo o no cooperativo) o de **modo cooperativo**.
- Desde la publicación, en 1944, del libro *The Theory of Games and Economic Behavior* ([Von Neumann and Morgenstern, 2007]), esta disciplina ha ido creciendo y se han desarrollado diversos modelos y aplicaciones tanto para comprender las acciones estratégicas de los “jugadores” como para proponer repartos o soluciones en contextos en los que la cooperación es determinante.



- El proyecto **TUGlab** (Transferable Utility Games laboratory, ([Mirás Calvo and Sánchez Rodríguez, 2010])) trata fundamentalmente de resaltar los aspectos geométricos de la teoría de juegos cooperativos para 3 y 4 jugadores.
- **TUGlabWeb** (<http://TUGlabweb.uvigo.es/TUGlabWEB2/index.php>, [Grande Coughil et al., 2011]) es una plataforma online en la que se encuentran implementadas las funciones básicas de TUGlab. El usuario puede experimentar con juegos cooperativos con 3 o 4 jugadores. A día de hoy tiene más de 500 usuarios registrados.
- **TUGlabR** es una extensión de TUGlab, en la que el usuario puede trabajar con juegos cooperativos con cualquier número de jugadores.
- **GameTheory**, [Cano-Berlanga et al., 2017], y **CoopGame**, [Staudacher and Anwander, 2019], son otros paquetes de R centrados en la teoría de juegos cooperativos.



Mercados de intercambio o asignación

Un mercado de intercambio surge cuando un conjunto de agentes desean comprar un bien de entre un catálogo. Cada agente asigna un valor sobre cada bien.

Problema del aeropuerto

Este problema consiste en cómo dividir los costes de mantenimiento de una pista de aterrizaje entre las distintas aerolíneas que operan en un aeropuerto.

Problemas de bancarrota (o de reparto)

Un problema de reparto de recursos escasos tiene lugar cuando tenemos que distribuir una cantidad fija entre un conjunto de demandantes con los mismos derechos que reclaman diferentes cantidades y de forma que esa cantidad inicial no llega para cubrir todas las peticiones. [ClaimsProblems](#), [Núñez Lugilde et al., 2022] es un paquete de R exclusivamente dedicado a repartir recursos escasos. Cuenta con casi 5000 descargas desde su publicación.

Problemas de árbol de expansión de coste mínimo

Los problemas de árbol de expansión de coste mínimo sirven para modelizar situaciones en las que varios usuarios (ciudades, granjas, hogares, ...) se quieren conectar a una fuente o recurso. Cada usuario quiere tener acceso al servicio, pero le es indiferente una conexión directa o a través de otros usuarios, por lo que existe la posibilidad de reducir costes a través de la construcción de una red óptima.

Y muchas más...

TUGlabR

Juegos TU

Un **juego cooperativo** es un par (N, v) donde $N = \{1, \dots, n\}$, v es una función $v : 2^N \rightarrow \mathbb{R}$ tal que $v(\emptyset) = 0$ y $|N| = n$. Los elementos de N son *jugadores*, los subconjuntos de N reciben el nombre *coaliciones*, y v es la *función característica*. Denotamos por G^N el conjunto de juegos cooperativos donde N es el conjunto de jugadores. Sea $S \in 2^N$, dado $x \in \mathbb{R}^n$, denotamos $x(S) = \sum_{i \in S} x_i$.

Ejemplo 1

Un mecánico (1), una ingeniera (2) y un informático (3) quieren construir un coche. Entre los 3 pueden sacar al mercado un coche valorado en 4000€. Sin embargo, el mecánico es capaz de hacer un coche él solo que estaría valorado en 1000€. El mecánico y la ingeniera juntos podrían hacer uno valorado en 2000€. Por otro lado, entre el mecánico y el informático son capaces de hacer un coche valorado en 3000€. Por último, el coche hecho entre la ingeniera y el informático se valoraría en 2000€. La función característica (v) del juego asociado a este problema viene dada por los siguientes valores:

$$\begin{aligned}v(\{1\}) &= 1000, \quad v(\{2\}) = 0, \quad v(\{3\}) = 0, \\v(\{1, 2\}) &= 2000, \quad v(\{1, 3\}) = 3000, \quad v(\{2, 3\}) = 2000, \\v(\{1, 2, 3\}) &= 4000\end{aligned}$$

Orden binario o lexicográfico con 3 jugadores

Posición	Binario	Orden binario	Orden lexicográfico
1	100	{1}	{1}
2	010	{2}	{2}
3	110	{1, 2}	{3}
4	001	{3}	{1, 2}
5	101	{1, 3}	{1, 3}
6	011	{2, 3}	{2, 3}
7	111	{1, 2, 3}	{1, 2, 3}

Función característica en R

```
#Orden binario:  
>v = c(1000,0,2000,0,3000,2000,4000)  
#Orden lexicografico:  
>v = c(1000,0,0,2000,3000,2000,4000)  
>v = Lex2bin(v)
```


Orden binario o lexicográfico con 4 jugadores

Posición	Binario	Orden binario	Orden lexicográfico
1	1000	{1}	{1}
2	0100	{2}	{2}
3	1100	{1, 2}	{3}
4	0010	{3}	{4}
5	1010	{1, 3}	{1, 2}
6	0110	{2, 3}	{1, 3}
7	1110	{1, 2, 3}	{1, 4}
8	0001	{4}	{2, 3}
9	1001	{1, 4}	{2, 4}
10	0101	{2, 4}	{3, 4}
11	1101	{1, 2, 4}	{1, 2, 3}
12	0011	{3, 4}	{1, 2, 4}
13	1011	{1, 3, 4}	{1, 3, 4}
14	0111	{2, 3, 4}	{2, 3, 4}
15	1111	{1, 2, 3, 4}	{1, 2, 3, 4}

Ejemplo 2

André (1), Aleixo (2), Lúa (3) y Alicia (4) tienen que realizar un trabajo para clase. Con sus conocimientos individuales André, Aleixo y Lúa obtendrían un 0 y Alicia un 1. Como individualmente no pueden aprobar, deciden colaborar. Dependiendo las colaboraciones que lleven a cabo obtendrán las siguientes notas:

$$w(\{1\}) = 0, w(\{2\}) = 0, w(\{3\}) = 0, w(\{4\}) = 1$$

$$w(\{1, 2\}) = 2, w(\{1, 3\}) = 3, w(\{1, 4\}) = 4$$

$$w(\{2, 3\}) = 4, w(\{2, 4\}) = 5, w(\{3, 4\}) = 6$$

$$w(\{1, 2, 3\}) = 5, w(\{1, 2, 4\}) = 6$$

$$w(\{1, 3, 4\}) = 7, w(\{2, 3, 4\}) = 8$$

$$w(\{1, 2, 3, 4\}) = 10$$

Función característica en R

```
>w = c(0,0,0,1,2,3,4,4,5,6,5,6,7,8,10)
>w = Lex2bin(w)
```

Propiedades

Función y uso	Descripción
AdmissibleGame(v)	Comprueba si el juego es compromiso admisible .
BalancedGame(v)	Comprueba si el juego es equilibrado .
Belongtocore(v,x)	Comprueba si el vector x está en el core del juego v .
ConcaveGame(v)	Comprueba si el juego es cóncavo .
ConvexGame(v)	Comprueba si el juego es convexo .
EssentialGame(v)	Comprueba si el juego es esencial y/o degenerado .
ExactGame(v)	Comprueba si el juego es exacto .
MonotonicGame(v)	Comprueba si el juego es monótono .
NonnegativeGame(v)	Comprueba si el juego es no negativo .
NormalizedGame(v)	Calcula los juegos 0-normalizado y 0-1 normalizado .
TotalBalancedGame(v)	Comprueba si el juego es totalmente equilibrado .
SuperAdditiveGame(v)	Comprueba si el juego es superaditivo .
ZeroMonotonicGame(v)	Comprueba si el juego es 0-monótono .

Propiedades

Monotonía

Un juego (N, v) es monótono si $v(T) \geq v(S)$ para todo $S, T \in 2^N$ tal que $S \subset T$.

```
> MonotonicGame(w)
[1] 1
```

Superaditividad

Un juego (N, v) es superaditivo si $v(S \cup T) \geq v(S) + v(T)$ para todo $S, T \in 2^N$ tales que $S \cap T \neq \emptyset$.

```
> SuperAdditiveGame(v)
[1] 1
> SuperAdditiveGame(w)
[1] 1
```

Convexidad (concavidad)

Un juego (N, v) es convexo (cóncavo) si $v(S \cup T) + v(S \cap T) \geq (\leq) v(S) + v(T)$ para todo $S, T \in 2^N$.

```
> ConvexGame(v)
$convex
[1] 0
$info
[1] "Fails the coalition S={1, 3} and the coalition T={2, 3}"
[2] "Fails the coalition S={2, 3} and the coalition T={1, 3}"
```

Operaciones con juegos

Operaciones con juegos

Función y uso	Descripción
AverageMarginal(v)	Calcula el vector de contribuciones marginales promedio.
DualGame(v)	Calcula el juego dual del juego v.
ExactEnvelope(v)	Calcula la envoltura exacta del juego v.
Excesses(v,x)	Calcula el vector de excesos del vector x sobre el juego v.
FacesGames(v)	Calcula los juegos de las caras de todas las coaliciones del juego v.
HarsanyiDividends(v)	Calcula los dividendos de Harsanyi.
Lexinal(v)	Calcula los vectores lexinales del juego v.
Marginal(v)	Calcula los vectores de contribuciones marginales.
UtopiaPayoffs(v)	Calcula el vector de los pagos de utopía y el vector de mínimos derechos.
SubGames(v)	Calcula los subjuegos del juego v.

Soluciones de tipo conjunto

Soluciones de tipo conjunto

Una solución definida en algún dominio $\Omega \subset G^N$, es una correspondencia Ψ , $\Psi : \Omega \rightarrow \mathbb{R}^n$, que asocia a cada juego (N, v) en Ω un subconjunto $\Psi(N, v) \subset \mathbb{R}^n$.

Función y uso	Descripción
CoreCoverSet(v,set=T,col=NULL)	Dibuja el core-cover del juego v .
CoreCoverVertices(v)	Calcula los vértices del core-cover del juego.
CoreSet(v,set=T,col=NULL)	Dibuja el core del juego v .
CoreVertices(v)	Calcula los vértices del core del juego.
EpsilonCoreSet(v,set=T,col=NULL)	Dibuja el ϵ-core del juego v .
EpsilonCoreVertices(v)	Calcula los vértices del ϵ-core del juego.
HarsanyiSet(v,set=T,col=NULL)	Dibuja el conjunto de Harsanyi del juego v .
ImputationSet(v,col=NULL)	Dibuja el conjunto de imputaciones del juego v .
ImputationVertices(v)	Calcula los vértices del conjunto de imputaciones del juego.
LeastCoreSet(v,set=T,col=NULL)	Dibuja el least-core del juego v .
LeastCoreVertices(v)	Calcula los vértices del least-core del juego.
WeberSet(v,set=T,col=NULL)	Dibuja el conjunto de Weber del juego v .
WeberVertices(v)	Calcula los vértices del conjunto de Weber del juego.

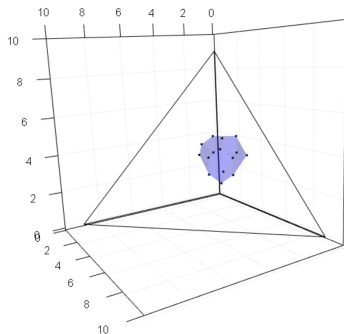
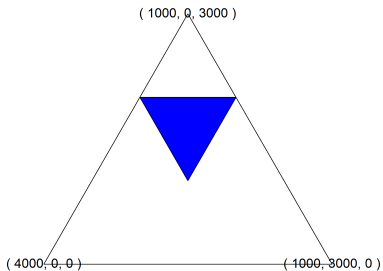
Si set=T, dibuja el gráfico correspondiente dentro del conjunto de imputaciones.
Si set=F, lo dibuja dentro del gráfico anterior.

Sea $(N, v) \in G^N$. Definimos los siguientes conjuntos:

- **Conjunto de asignaciones eficientes:** $H(v) = \{x \in \mathbb{R}^n : x(N) = v(N)\}$.
- **Conjunto de imputaciones:**
 $I(v) = \{x \in H(v) : x_i \geq v(i) \text{ para todo } i \in N\}$.
- **Core (o núcleo)**, [Gillies, 1953]:
 $C(v) = \{x \in H(v) : x(S) \geq v(S) \text{ para todo } S \in 2^N\}$

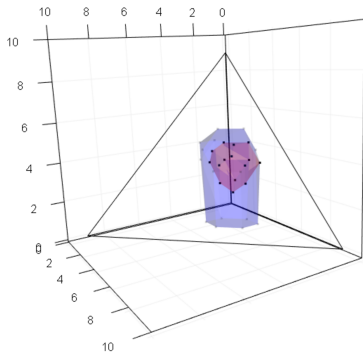
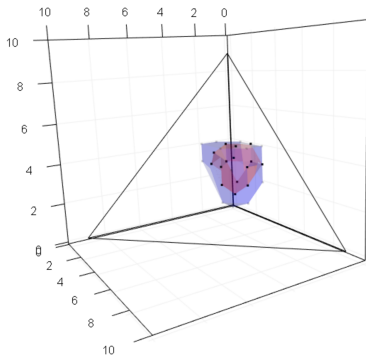
Soluciones de tipo conjunto

```
> CoreSet(v, col="blue")  
> CoreSet(w, col="blue")
```



Soluciones de tipo conjunto

- > CoreSet(w, set=T, col="red")
- > CoreCoverSet(w, set=F, col="blue")
- > CoreSet(w, set=T, col="red")
- > WeberSet(w, set=F, col="blue")



Soluciones puntuales

Soluciones puntuales

Función y uso	Descripción
Alexia(v)	Calcula el valor de Alexia del juego v .
Corecenter(v)	Calcula el core-center del juego v .
DuttaRay(v)	Calcula el valor de Dutta-Ray del juego v .
Nucleolus(v)	Calcula el nucleolus del juego v .
OneNucleolus(v)	Calcula el 1-nucleolus del juego v .
PerCapitaNucleolus(v)	Calcula el nucleolus per cápita del juego v .
PlotSolution(v,solution,awards,set,col)	Dibuja el vector dado por solution o awards dentro del gráfico elegido.
PreNucleolus(v)	Calcula el pre-nucleolus del juego v .
Shapley(v)	Calcula el valor de Shapley del juego v .
Solidarity(v)	Calcula el valor de solidaridad del juego v .
Tau(v)	Calcula el τ-valor del juego v .

Valor de Shapley, [Shapley, 1953]

Se define el valor de Shapley como el promedio de los vectores de contribuciones marginales asociados a todas las posibles ordenaciones de los jugadores.

$$Sh(N, v) = \frac{1}{n!} \sum_{\pi \in \Pi^N} m^\pi(N, v),$$

donde $\pi : N \rightarrow \{1, \dots, n\}$ denota una ordenación de los jugadores de N , $\pi \in \Pi^N$ y el reparto de contribuciones marginales asociado al orden π , $m^\pi(v) = (m_i^\pi(v))_{i \in N} \in \mathbb{R}^N$, viene dado por

$$m_i^\pi = v(\text{Pre}^\pi(i) \cup i) - v(\text{Pre}^\pi(i))$$

para todo $i \in N$, donde $\text{Pre}^\pi(i) = \{j \in N : \pi(j) < \pi(i)\}$.

Supongamos que los jugadores van llegando a un lugar de modo aleatorio, y al llegar cada jugador recibe como utilidad su aportación marginal a los jugadores que ya estaban presentes. El valor de Shapley se puede ver como el vector de utilidades esperadas de los jugadores bajo este procedimiento.

```
> Marginal(v)
      [,1] [,2] [,3]
[1,] 2000 2000    0
[2,] 3000 1000    0
[3,] 2000    0 2000
[4,] 2000    0 2000
[5,] 1000 1000 2000
[6,] 1000 1000 2000

> Shapley(v)
[1] 1833.3333 833.3333 1333.3333
```

Tiempos de computación ([Mirás Calvo et al., 2022b])

Nº jugadores	Shapley (TUGlabR)	Shapley (CoopGame)	Shapley (GameTheory)
3	0.01"	0.01"	0.06"
4	0.01"	0.01"	0.06"
5	0.01"	0.02"	0.08"
10	0.01"	0.71"	0.1" (aproximación?)
13	0.02"	7.14"	0.37" (aproximación?)
15	0.07"	32.9"	5.41" (aproximación?)
20	2.86"	32'	-

Core-center, [González-Díaz and Sánchez Rodríguez, 2007]

El core-center es el centro de gravedad del núcleo considerando que su “masa” está uniformemente distribuida.

```
> Corecenter(v)
[1] 1666.6667 666.6667 1666.6667
```

Nucleolus, [Schmeidler, 1969]

El exceso de una coalición S y vector $x \in \mathbb{R}^n$, $e(S, x)$, es la diferencia entre el valor de la coalición y la asignación del vector x , es decir, $e(S, x) = v(S) - x(S)$.

```
> Excesses(v, Shapley(v))
$excesses
[1] -166.6667 -166.6667 -666.6667 -833.3333 -833.3333 -1333.3333
$coalition
[1] "1 3" "2 3" "1 2" "1" "2" "3"
```

Dado un reparto $x \in \mathbb{R}^n$, $\theta(x)$ es la 2^N -tupla cuyas componentes son los excesos del valor de todas las coaliciones con respecto a x ordenados de mayor a menor, es decir, $\theta_i(x) \geq \theta_j(x)$ si $1 \leq i \leq j \leq 2^N$. Definimos el nucleolus del juego como $Nu(N, v) = \{x \in I(N, v) : \theta(x) \leq_L \theta(y) \text{ para todo } y \in I(N, v)\}$.

```
> Nucleolus(v)
[1] 1666.6667 666.6667 1666.6667
```

Situaciones asimétricas

Situaciones asimétricas

Función y uso	Descripción
<code>CoalitionweightedShapley(v,lambda)</code>	Calcula el valor de Shapley coalicional ponderado del juego v con pesos λ .
<code>Myerson(v,communication)</code>	Calcula el valor de Myerson del juego v con comunicación restringida.
<code>Owen(v,partition)</code>	Calcula el valor de Owen del juego v con uniones a priori.
<code>StrongPriorityGame(v,x,coalition)</code>	Calcula el juego prioritario fuerte de coalition con respecto a x .
<code>WeakPriorityGame(v,x,coalition)</code>	Calcula el juego prioritario débil de coalition con respecto a x .
<code>WeightedShapley(v,w)</code>	Calcula el valor de Shapley ponderado del juego v con vector de pesos w .

Valor de Shapley ponderado

Un sistema ordenado de pesos $p = (\omega, \Omega)$ consiste en:

- 1 Un vector de pesos estrictamente positivos, $\omega \in \mathbb{R}_{++}^n$.
- 2 Una partición ordenada de N , $\Omega = \{S_1, \dots, S_m\}$.

Sean $S \in 2^N$, (N, u_S) el juego de unanimidad de la coalición S y $p = (\omega, \Omega)$ un sistema ordenado de pesos. Definimos el valor de Shapley ponderado por el sistema de pesos p del juego de unanimidad (N, u_S) mediante

$$Sh_i^{(\omega, \Omega)}(N, u_S) = \begin{cases} \frac{\omega_i}{\sum_{j \in S \cap S_k} \omega_j} & \text{si } i \in S \cap S_k, \\ 0 & \text{en otro caso} \end{cases}, \text{ con } k = \max\{j : S_j \cap S \neq \emptyset\} \text{ e } i \in N.$$

Se define el valor de Shapley ponderado de un juego (N, v) , [Kalai and Samet, 1987], con sistema de pesos p como el valor que a cada $i \in N$ le asigna

$$Sh_i^{(\omega, \Omega)}(N, v) = \sum_{S \subset N} c_S Sh_i^{(\omega, \Omega)}(N, u_S),$$

donde, para cada $S \in 2^N$, c_S son los correspondientes dividendos de Harsanyi del juego (N, v) y (N, u_S) el juego de unanimidad de la coalición $S \in 2^N$.

```
> WeightedShapley(v, c(0.5, 0.25, 0.25))
[1] 1833.3333 583.3333 1583.3333
> WeightedShapley(w, c(0.3, 0.3, 0.3, 0.1))
[1] 1.680952 2.502381 3.323810 2.492857
```

Situaciones asimétricas

Juegos de las caras, [Mirás Calvo et al., 2020]

Para cada $T \in 2^N$, se define el juego de la T -cara como

$$v_T(S) = v(S \cup (N \setminus T)) - v(N \setminus T) + v(S \cap (N \setminus T)), \text{ para todo } S \in 2^N.$$

```
> FacesGames(v)
$Faces
      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 2000    0 2000 1000 3000 2000 4000
[2,] 1000 1000 2000 1000 3000 2000 4000
[3,] 2000 1000 3000 1000 3000 2000 4000
[4,] 1000    0 2000 2000 3000 2000 4000
[5,] 2000    0 2000 2000 4000 2000 4000
[6,] 1000 1000 2000 2000 3000 3000 4000
```

Valor de Shapley coalicional ponderado, [Mirás Calvo et al., 2022a]

Se define el valor de Shapley coalicional ponderado con vector de pesos $\lambda \in \mathbb{R}^{2^n-1}$ tal que $\lambda(N) = 1$, como el valor

$$\Psi^\lambda(N, v) = \sum_{T \subset N} \lambda_T \cdot Sh(v_T).$$

```
> CoalitionweightedShapley(v, c(0.5, 0.25, 0, 0.25, 0, 0, 0))
[1] 1750 625 1625
> CoalitionweightedShapley(w, c(0.3, 0.3, 0, 0.3, 0, 0, 0, 0.1, 0, 0, 0, 0, 0, 0))
[1] 1.066667 2.066667 3.066667 3.800000
```

Uniones a priori

Puede ocurrir que, por mecanismos externos al propio juego, no contemplados en la función característica, se sepa que **ciertos jugadores tienen afinidades** (de tipo personal, político, etc.) **que provocan una mayor tendencia a cooperar entre ellos**.

Sea un juego v y $P = \{P_1, \dots, P_m\}$ un sistema de coaliciones “a priori”, se define el valor de Owen, [Owen, 1977], como

$$\varphi(N, v, P) = \frac{1}{|\Pi_P^N|} \sum_{\pi \in \Pi_P^N} m^\pi(N, v),$$

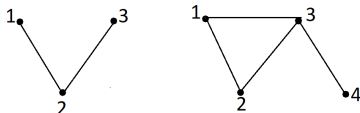
siendo Π_P^N el conjunto de ordenaciones admisibles con respecto a P .

El valor coalicional puede ser visto como la media de las contribuciones marginales, dándoles **probabilidad cero a aquellas ordenaciones en las cuales entre dos jugadores de una coalición “a priori” hay jugadores de otras coaliciones “a priori”**, e igual probabilidad al resto de ordenaciones, es decir, a aquellas en las que los jugadores de cada coalición “a priori” aparecen consecutivos.

```
>partition=matrix(c(1,0,0,0,1,1),nrow=2,byrow=T)
>partition
      [,1] [,2] [,3]
[1,]     1     0     0
[2,]     0     1     1
>Owen(v,partition)
[1] 1500 1000 1500
```

Comunicación restringida

En determinadas situaciones pueden existir **limitaciones en la comunicación que no permitan que los jugadores cooperen libremente**. El valor de Myerson, [Myerson, 1977], es **el valor de Shapley del juego restringido por el grafo**. Este juego representa el valor que puede obtener cada coalición si la restricción en la cooperación viene dada por un grafo.



```
> communication=matrix(c(1,1,0,0,1,1),nrow=2,byrow=T)
> communication
      [,1] [,2] [,3]
[1,]    1    1    0
[2,]    0    1    1
> Myerson(v,communication)
[1] 1333.3333 1833.3333 833.3333
```

Próximos pasos

- Compartir el paquete en el CRAN de R.
- Actualización de TUGlabWeb con las funciones de TUGlabR.
- Complementar el paquete con una batería de juegos aplicados a problemas de Investigación Operativa.

References I



Cano-Berlanga, S., Gimenez-Gomez, J. M., and Vilella, C. (2017).

Enjoying cooperative games: The r package gametheory.

Applied Mathematics and Computation, 10.



Gillies, D. B. (1953).

Some theorems on n-person games.

Princeton University.



González-Díaz, J. and Sánchez Rodríguez, E. (2007).

A natural selection from the core of a tu game: the core-center.

International Journal of Game Theory, 36(1):27–46.



Grande Cougil, R., Mosquera Rodríguez, M., and Ramos Valcarcel, D. (2011).

Tuglabweb: Interfaz web para juegos tu.

X Congreso Galego de Estatística e Investigación de Operacións.



Kalai, E. and Samet, D. (1987).

On weighted shapley values.

International Journal of Game Theory, 3(16):205–222.



Mirás Calvo, M. A., Núñez Lugilde, I., Quinteiro Sandomingo, C., and Sánchez Rodríguez, E. (2022a).

The coalition-weighted shapley values.

Working paper.



Mirás Calvo, M. A., Nuñez Lugilde, I., Quinteiro Sandomingo, C., and Sánchez Rodríguez, E. (2022b).

On the computation of the shapley value and the random arrival rule.

preprint.

References II



Mirás Calvo, M. A., Quinteiro Sandomingo, C., and Sánchez Rodríguez, E. (2020).

The boundary of the core of a balanced game: face games.

International Journal of Game Theory, 49:579–599.



Mirás Calvo, M. A. and Sánchez Rodríguez, E. (2010).

Herramientas informáticas de cálculo y representación gráfica para juegos tu.

La Gaceta de la RSME, 13(1):89–108.



Myerson, R. B. (1977).

Graphs and cooperation in games.

Mathematics of operations research, 2(3):225–229.



Núñez Lugilde, I., Mirás Calvo, M. A., Quinteiro Sandomingo, C., and Sánchez Rodríguez, E. (2022).

ClaimsProblems: Analysis of Conflicting Claims.

R package version 0.2.0.



Owen, G. (1977).

Values of games with a priori unions.

In Mathematical economics and game theory. Springer, Berlin, Heidelberg., pages 76–88.



Schmeidler, D. (1969).

The nucleolus of a characteristic function game.

SIAM J Appl Math, 17:1163–1170.



Shapley, L. (1953).

A value for n-person games.

Annals of mathematics studies, Princeton University Press., 28:307–317.

References III



Staudacher, J. and Anwander, J. (2019).

Using the R package CoopGame for the analysis, solution and visualization of cooperative games with transferable utility.

R Vignette.



Von Neumann, J. and Morgenstern, O. (2007).

Theory of games and economic behavior.

In Theory of games and economic behavior. Princeton university press.



TUGlabR, un paquete de R para juegos coalicionales

Iago Núñez Lugilde¹, Miguel Ángel Mirás Calvo², Carmen Quinteiro Sandomingo³ y Estela Sánchez Rodríguez¹

¹CINBIO, Universidade de Vigo, SIDOR. Departamento de Estatística e Investigación Operativa.

²Universidade de Vigo, RGEAF, Departamento de Matemáticas.

³Universidade de Vigo, Departamento de Matemáticas.

20/10/2022