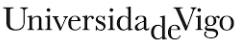


Trabajo fin de máster

El juego de producción lineal

Alumno Prieto Rodríguez, Daniel
Universidade de Santiago de Compostela

Directora Lorenzo Freire, Silvia María
Universidade da Coruña

Máster en Técnicas Estadísticas
2017-2018

El presente documento recoge el Trabajo Fin de Máster para el Máster en Técnicas Estadísticas realizado por D. Daniel Prieto Rodríguez bajo el título "El juego de producción lineal". Ha sido realizado bajo la dirección de Dña. Silvia María Lorenzo Freire, que lo considera terminado y da su conformidad para su presentación y defensa.

Fdo: Silvia María Lorenzo Freire

Índice general

Resumen

Este trabajo de fin de máster consiste en una revisión bibliográfica de los juegos de producción lineal descritos en la literatura. Supongamos que tenemos una empresa que dispone de varios recursos para fabricar determinados productos: un problema de producción lineal es un problema de programación lineal en el que se pretenden maximizar los beneficios derivados del proceso de producción teniendo en cuenta los recursos disponibles. Si ahora asumimos que hay varias empresas que deciden asociarse y aportar sus recursos a un proceso productivo común, tendríamos el juego de producción lineal asociado, principal objetivo del trabajo. Este trabajo se complementa con el desarrollo de la librería "coopProductGame" en R que resuelve diferentes versiones del problema, las cuales se describen a lo largo del trabajo, así como una aplicación en Shiny que permite obtener soluciones gráficas para problemas en dos dimensiones de una forma rápida e interactiva.

Introducción

Algunos problemas de la Investigación Operativa pueden abordarse desde el punto de vista de la Teoría de Juegos Cooperativos; ejemplos de ello son los problemas de conexión, los problemas de rutas, los problemas de secuenciación, los problemas de inventario o los problemas de producción. Este trabajo fin de máster se centra en el estudio de la cooperación en los problemas de producción.

Los primeros estudios sobre juegos cooperativos datan del año 1944 de la mano de von Neumann y Morgenstern con el libro "Theory of games and economic behaviour". Este tipo de juegos son de gran utilidad para modelar situaciones de cooperación entre diferentes agentes o personas.

Este trabajo se centra en el estudio de una clase particular de juegos cooperativos, los juegos de producción lineal introducidos por Owen en 1975 [17].

Los juegos cooperativos constan de un conjunto finito y ordenado, el conjunto de jugadores y una función, función característica, que asigna a cada subconjunto de jugadores un número real. En las situaciones modeladas con este tipo de juegos, los jugadores o agentes pueden decidir trabajar solos, en subgrupos o todos en colaboración. A cada una de estas combinaciones se le conoce con el nombre de coalición.

Por otro lado, la función característica define la ganancia que pueden obtener los jugadores para cada una de las posibles coaliciones. Una de las principales preguntas sobre la que se han centrado muchos estudios en el ámbito de la Teoría de Juegos Cooperativos es, en el caso que todos los jugadores decidiesen cooperar y formar la gran coalición, ¿cómo se repartirían las ganancias? Es lógico desear que los repartos satisfagan la propiedad de *estabilidad*, es decir, que para todos los jugadores sea conveniente trabajar juntos porque el beneficio obtenido por la gran coalición es mayor que la suma de los valores de las coaliciones en cualquier partición del grupo. Existen muchas situaciones en las que esto no ocurre y, para ciertos subgrupos, será más rentable dejar a los demás y trabajar por su cuenta, ya que ganarían más de esta manera. Por

tanto, la división que verifique estabilidad no siempre debe existir y, en otros casos sin embargo, puede existir más de un reparto con esta propiedad. Este tipo de situaciones han dado lugar a la definición de reglas de división basadas en otras propiedades diferentes a la estabilidad.

Este trabajo está estructurado en cuatro capítulos. El primero de ellos se centra en los problemas de programación lineal, introduciendo los problemas de producción lineal como caso particular de estos. En este capítulo se describe la notación necesaria para tratar con este tipo de problemas, la cual se empleará a lo largo de todo el trabajo. Se presentan aquí también varios métodos de resolución, tales como el método gráfico y el Símplex. Todo ello de la mano de ejemplos que permitan poner de manifiesto todos estos conceptos. Además, también se hace una introducción a la librería programada en R, así como los primeros casos de uso en los que podemos ver su aplicabilidad.

El segundo capítulo comienza mostrando las nociones básicas de los juegos cooperativos, introduciendo el concepto de solución para estos juegos y describiendo algunas de las más importantes. A continuación se introducen los juegos de producción lineal introducidos por Owen, se describen los conjuntos de Owen y se analizan sus características y propiedades.

En el capítulo 3 se describe brevemente una aplicación web desarrollada en Shiny y publicada en shinyapps.io que permite al usuario interactuar de forma dinámica y visual con las soluciones gráficas para los problemas de producción lineal y los juegos correspondientes asociados.

Por último, en el capítulo 4 se resumen brevemente las conclusiones obtenidas y se describen algunas de las posibles líneas de trabajo futuras, que permitan completar y mejorar lo estudiado en este documento.

Capítulo 1

El problema de producción lineal

Un problema de programación se refiere, en general, al uso o asignación (reparto) de una serie de recursos (dinero, materiales, trabajadores, etc.) de la "forma más eficiente posible", es decir, o bien maximizando las ganancias o bien minimizando los costes de utilización de los recursos.

El primer problema particular de programación lineal data del 1941-1942: el problema del transporte. La primera referencia a este problema se remonta a 1781, cuando el matemático francés Gaspard Monge describe el problema de la construcción y abastecimiento de fortificaciones militares de los ejércitos de Napoleón. Formalmente, este problema aparece en 1941 cuando F.L. Hitchcook publica una solución analítica para el mismo en su estudio "The Distribution of a Product from Several Sources to Numerous Localities" [10], que se considera la primera contribución importante para la solución de este tipo de problemas. El desarrollo del problema se produce ya a finales de los 40, cuando Koopmans presenta su tesis doctoral sobre los problemas de embarque en la marina holandesa. Su estudio, no relacionado con el de Hitchcook, llamado "Optimum Utilization of the Transportation System" [13], junto con la contribución de Hitchcook han ayudado al desarrollo del problema del transporte.

El problema del transporte clásico consiste en lo siguiente: supongamos que tenemos n orígenes que tienen que suministrar a m destinos un cierto producto. Además, existe un coste asociado al envío de cada unidad de producto desde un origen a un destino determinado. El problema se basa en determinar cuántas unidades de producto deben enviarse desde el origen i, $i \in \{1, \ldots, n\}$, al destino j, $j \in \{1, \ldots, m\}$, minimizando el coste de envío y a la vez satisfaciendo la demanda de los diferentes destinos y sin exceder la capacidad de los orígenes.

Por otro lado, otra importante referencia viene de la mano de Stigler [24] que, en 1945, propone el problema del régimen alimenticio óptimo para tratar de satisfacer la preocupación

del ejército americano por hallar la manera más económica de alimentar a sus tropas, asegurando al mismo tiempo unos determinados requerimientos nutricionales. Para resolver este problema se necesitaron un total de 9 trabajadores durante aproximadamente 15 días para realizar los cálculos electrónicos necesarios para resolver el problema.

Todos estos problemas cobran vital importancia gracias a la aparición del método Símplex, publicado por George Dantzig en 1947 con el objetivo de resolver problemas de programación lineal en los que intervenían tres o más variables, (una descripción más detallada del método se puede encontrar en la Sección 1.1.3) y al desarrollo de los ordenadores a partir de la década de los 60.

En este capítulo se introducen formalmente los problemas de producción lineal, ejemplificándolos en la vida real y viendo cómo podemos resolverlos a través de la librería `coopProductGame` desarrollada en `R` y cuyo manual se puede ver en el Apéndice A.

1.1. Los problemas de programación lineal

1.1.1. Formulación

En esta sección se introduce formalmente el problema de programación lineal (que a partir de ahora representaremos por las siglas PPL), describiendo todos los elementos que lo conforman y mostrando un pequeño ejemplo general donde se puede ver su aplicabilidad. A continuación, se detalla la notación básica de un PPL:

- z es el valor de la función objetivo.
- $(x_j)_{j\in N}$ son las variables de decisión del problema.
- $(c_j)_{j\in N}$ son los costes o beneficios unitarios.
- $(b_i)_{i\in M}$ son las constantes a la derecha del problema.
- $(a_{ij})_{i\in M, j\in N}$ es la matriz de coeficientes.

Además, para que un problema de programación se diga lineal, han de cumplirse las dos siguientes características:

- El criterio para seleccionar los mejores valores de las variables involucradas a la hora de construir el problema se puede escribir como una ecuación lineal de las mismas que, según

sea el caso, habrá que maximizar o minimizar. A esta función se le llama **función objetivo**.

$$z = c_1x_1 + c_2x_2 + \ldots + c_nx_n.$$

- Las relaciones existentes entre las variables del problema se pueden escribir como un conjunto de ecuaciones o inecuaciones lineales de las variables. Este conjunto se denomina **conjunto de restricciones**.

$$\begin{array}{rl} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & (\leq, \geq, =) \quad b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & (\leq, \geq, =) \quad b_2, \\ \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & (\leq, \geq, =) \quad b_m. \end{array}$$

A la hora de formular un PPL, se deben seguir una serie de pasos, que se describen a continuación:

1. Definir las variables del problema, que usualmente se denominan **variables de decisión**.
2. Considerar el **objetivo** (la meta) que se trata de optimizar, y que se expresa a partir de una función lineal que habrá que maximizar o minimizar.
3. Definir el conjunto de restricciones que se deben satisfacer. Estas restricciones se definen a través de ecuaciones o inecuaciones lineales.

Para poner de manifiesto toda la notación descrita anteriormente, a continuación se puede ver un ejemplo de problema de programación lineal que, a su vez, se corresponde con un problema de producción, principal objetivo de nuestro trabajo y que definiremos formalmente en capítulos posteriores.

Ejemplo 1.1. *Una empresa fabrica dos productos, que denominaremos P_1 y P_2. Para fabricar dichos productos la empresa utiliza dos recursos o materias primas, que denotaremos por M_1 y M_2. En la Tabla 1.1 se indican las cantidades necesarias de cada recurso para fabricar los productos, así como la cantidad (medida en toneladas) disponible de cada uno de ellos. Además, se impone la restricción de que no se puede fabricar más cantidad de P_2 que de P_1. En este problema, hay que maximizar los beneficios por la venta de los productos (que se expresan en miles de euros).*

	P_1	P_2	Disponible/día
M_1	6	4	24
M_2	1	2	6
Beneficio/tonelada	5	4	

Tabla 1.1: Datos del problema

De esta forma, tenemos que las variables de decisión de nuestro problema serán:

- $x_1 =$ *"Toneladas de* P_1*".*
- $x_2 =$ *"Toneladas de* P_2*".*

Como ya se ha comentado, se trata de maximizar los beneficios, de tal forma que la función objetivo se convierte en

$$Maximizar \quad z = 5x_1 + 4x_2$$

Por último, las restricciones del problema vendrán dadas por el siguiente sistema de inecuaciones:

$$\begin{aligned} 6x_1 + 4x_2 &\leq 24 \quad \textit{(recurso 1)} \\ x_1 + 2x_2 &\leq 6 \quad \textit{(recurso 2)} \\ -x_1 + x_2 &\leq 0 \quad \textit{(imposición)} \\ x_1, x_2 &\geq 0 \quad \textit{(prod. positiva)} \end{aligned}$$

1.1.2. Método gráfico

Una vez que se dispone del modelo de programación lineal, hay que resolverlo. Para ello existen diferentes técnicas. Una de las más sencillas, pero que solo se puede utilizar para el caso en el que se tienen dos o tres variables de decisión, es la *resolución gráfica*.

Supongamos que queremos resolver un PPL con dos variables de decisión, x_1 y x_2. Se puede interpretar el par (x_1, x_2) como las coordenadas de un punto en el plano.

Para resolver el problema de forma gráfica, el primer paso consiste en representar las restricciones del problema. Cada ecuación da lugar a una línea recta y cada inecuación a un semiplano. La región resultante de intersecar todas las condiciones es lo que se conoce como **región factible**.

Una vez se tiene la región factible del problema, el siguiente paso consiste en dibujar una recta arbitraria de la función objetivo, determinando de esta forma la dirección de la misma, es

decir, aumentando el valor del objetivo si maximizamos y disminuyéndolo si minimizamos. Las rectas que representan valores del objetivo se denominan **rectas de nivel**.

En función de las rectas de nivel y la región factible ya seremos capaces de encontrar las soluciones óptimas del PPL, si existen.

Ejemplo 1.1 (continuación)**.** *Retomamos el problema de producción donde*

- $x_1 =$ *"Toneladas de* P_1 *que se producen".*
- $x_2 =$ *"Toneladas de* P_2 *que se producen".*

El problema de programación lineal asociado viene dado por:

$$\begin{array}{lrcl} \textbf{\textit{Maximizar}} & 5x_1 + 4x_2 & & \\ \textbf{\textit{sujeto a}} & 6x_1 + 4x_2 & \leq & 24 \\ & x_1 + 2x_2 & \leq & 6 \\ & -x_1 + x_2 & \leq & 0 \\ & x_1, x_2 & \geq & 0 \end{array}$$

Siguiendo los pasos descritos previamente, en primer lugar representamos las restricciones que, en este caso, por ser todas inecuaciones se corresponden con semiplanos (Figura 1.1). De esta forma, a partir de la intersección de todos los semiplanos obtenemos la región factible, tal como se puede ver en la Figura 1.2.

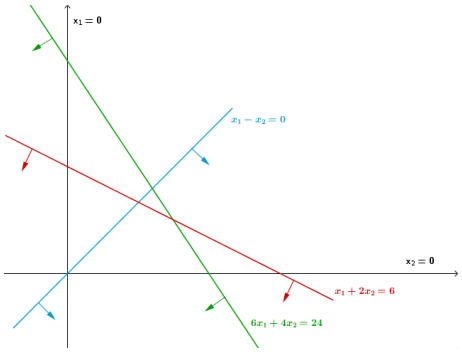


Figura 1.1: Representación de las restricciones.

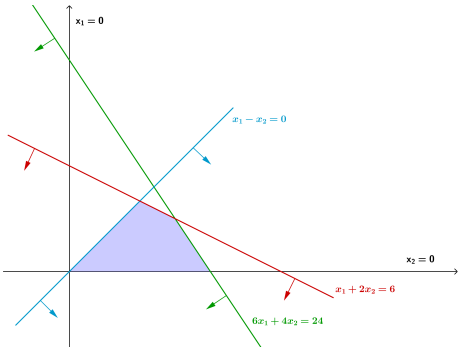


Figura 1.2: Obtención de la región factible.

Por último, dibujamos las rectas de nivel y encontramos la solución óptima que, en este caso, se alcanza en el punto $(3, 3/2)$*. Además, la función objetivo alcanza el valor de* $z = 5x_1 + 4x_2 = 21$ *(Figura 1.3).*

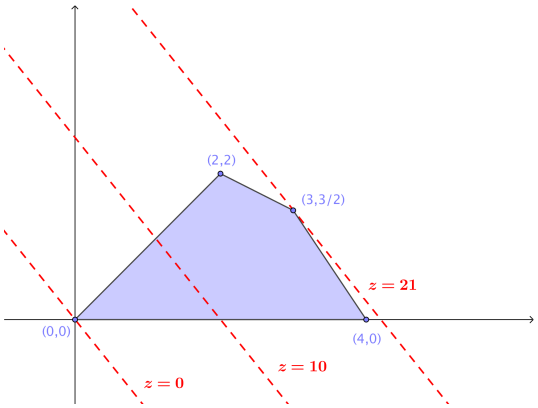


Figura 1.3: Obtención de la solución óptima.

A la hora de intentar resolver un problema de programación lineal, pueden darse varias posibilidades:

- Que haya una única solución óptima (ejemplo anterior).
- Que haya múltiples soluciones óptimas.
- Que la solución no sea acotada.
- Que el problema no tenga solución.

A continuación, se muestran 3 ejemplos donde se ponen de manifiesto estas tres últimas casuísticas.

Ejemplo 1.2. *(Múltiples soluciones óptimas). Supongamos el siguiente PPL:*

$$\begin{array}{lrcl} \textbf{\textit{Maximizar}} & x_1 + 2x_2 & & \\ \textbf{\textit{sujeto a}} & 6x_1 + 4x_2 & \leq & 24 \\ & x_1 + 2x_2 & \leq & 6 \\ & -x_1 + x_2 & \leq & 0 \\ & x_1, x_2 & \geq & 0 \end{array}$$

Tal como se puede ver en la Figura 1.4, todos los puntos de la recta que une $(2, 2)$ *y* $(3, 3/2)$ *son soluciones óptimas.*

Ejemplo 1.3. *(Solución no acotada). Supongamos el siguiente PPL:*

$$\begin{array}{lrcl} \textbf{\textit{Maximizar}} & 5x_1 + 4x_2 & & \\ \textbf{\textit{sujeto a}} & 6x_1 + 4x_2 & \geq & 24 \\ & -x_1 + x_2 & \leq & 0 \\ & x_1, x_2 & \geq & 0 \end{array}$$

En la Figura 1.5 se puede ver que, aunque tiene soluciones factibles, no existe óptimo finito ya que $z_{\acute{o}ptimo} \longrightarrow \infty$.

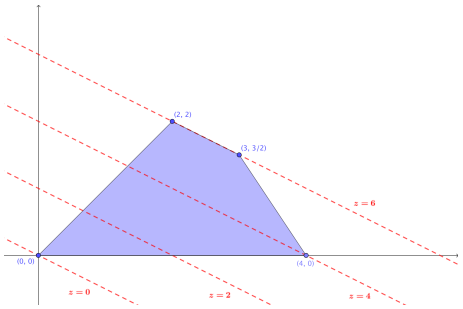


Figura 1.4: Ejemplo de PPL con múltiples soluciones.

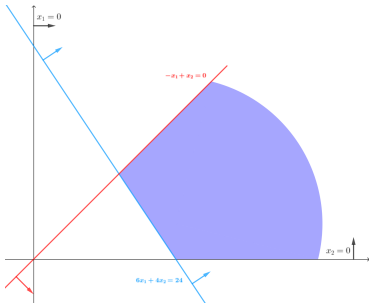


Figura 1.5: Ejemplo de PPL no acotado.

Nótese que un problema de programación lineal no acotado implica que la región factible del problema es no acotada. Sin embargo, el recíproco no tiene por qué ser cierto. Si consideramos el Ejemplo 1.3 pero intentando minimizar z en lugar de maximizar, se tendría que la solución óptima sería el punto $(4, 0)$, tal y como se puede ver en la Figura 1.6.

Ejemplo 1.4. *(No tiene solución). Supongamos el siguiente PPL:*

$$\begin{array}{lrcl} \textbf{\textit{Maximizar}} & 5x_1 + 4x_2 & & \\ \textbf{\textit{sujeto a}} & 6x_1 + 4x_2 & \geq & 24 \\ & x_1 + 2x_2 & \leq & 2 \\ & -x_1 + x_2 & \geq & 0 \\ & x_1, x_2 & \geq & 0 \end{array}$$

En la Figura 1.7 se puede ver que no hay ninguna solución que cumpla las restricciones. De hecho, la región factible es vacía.

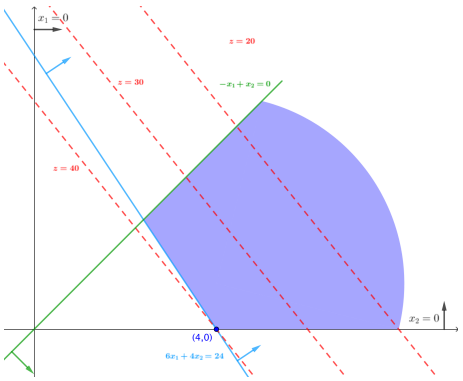


Figura 1.6: PPL con solución óptima y con región factible no acotada.

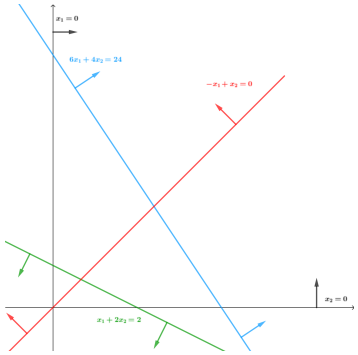


Figura 1.7: Ejemplo de PPL infactible.

1.1.3. Método Símplex

En 1947, el matemático George B. Dantzig, considerado como el "padre de la Programación Lineal", formula la forma general para un PPL e inventa el *método del Símplex* para obtener su resolución.

Antes de ver en qué consiste el método Símplex, veremos cómo se representa un PPL en su forma estándar:

$$\begin{array}{lrcl} \textbf{Max(Min)} & z = c_1x_1 + c_2x_2 + \ldots + c_nx_n & & \\ \textbf{sujeto a} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & = & b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & = & b_2 \\ & & \vdots & \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n & = & b_m \\ & x_1, x_2, \ldots, x_n & \geq & 0, \end{array}$$

siendo $b_1, b_2, \ldots, b_m \geq 0$.

También podemos escribir este problema de forma matricial, resultando:

$$\begin{array}{rl} \textbf{Max(Min)} & z = cx \\ \textbf{sujeto a} & Ax = b \\ & x \geq 0, \end{array} \tag{1.1}$$

donde $b \geq 0$ y

- $A = (a_{ij})_{i\in\{1,\ldots,m\},j\in\{1,\ldots,n\}}$: es la matriz de coeficientes.

- $x = (x_j)_{j\in\{1,\dots,n\}}$: es el vector solución, de actividades o de decisiones.
- $b = (b_i)_{i\in\{1,\dots,m\}}$: es el vector de recursos.
- $c = (c_j)_{j\in\{1,\dots,n\}}$: es el vector de costes o de beneficios, dependiendo de si tenemos que maximizar o minimizar la función objetivo.

Ha de tenerse claro que cualquier PPL se puede expresar como un PPL en forma estándar. Se pueden convertir las desigualdades en igualdades sin más que añadir *variables de holgura* o *variables de exceso*. Por ejemplo, si se tiene $x_1 + x_2 \leq 8$ pasaría a ser $x_1 + x_2 + x_3 = 8$ con $x_3 \geq 0$ o, en el caso de $x_1 + x_2 \geq 8$, sería $x_1 + x_2 - x_3 = 8$ con $x_3 \geq 0$.

A continuación, se enumeran una serie de reglas necesarias para adaptar un problema a su forma estándar según diferentes casuísticas:

- Si algún b_i es negativo se multiplica toda la restricción por -1.
- Si un x_j es ≤ 0, se introduce una variable $x_k \geq 0$, tal que

$$x_j = -x_k.$$

- Si una variable x_j es no restringida (puede tomar cualquier valor), se introducen dos variables positivas, x_k y x_l, tales que

$$x_j = x_k - x_l.$$

Dado el PPL estándar en forma matricial (1.1), se introducen a continuación una serie de definiciones básicas que emplearemos a lo largo del trabajo:

- Una **solución factible** es un vector no negativo x tal que $Ax = b$.
- Una **solución óptima** es una solución factible que, además, maximiza (minimiza) la función del objetivo.
- Una variable se conoce como **básica** si aparece en una única ecuación del sistema con coeficiente 1. Las variables que no cumplen esto, se dicen **no básicas**.
- Un sistema de ecuaciones se llama **canónico** si en cada ecuación aparece una variable básica.

- Un **pivotaje** es una sucesión de operaciones elementales que reduce el sistema de ecuaciones dado a uno equivalente canónico.
- Una solución es **básica** si se obtiene de hacer las variables no básicas cero y se calcula el valor de las básicas en cada ecuación.
- Una solución es **factible básica** si es básica y los valores de las variables básicas son no negativos.
- Dado un PPL con n variables y m restricciones, el número de soluciones básicas está acotado por $\binom{n}{m}$.
- El **beneficio relativo** de una variable no básica es el cambio que se produce en la función objetivo por unidad aumentada en dicha variable no básica. Se denotará por $\bar{c}_j$.

El algoritmo del Símplex es un procedimiento iterativo eficiente de optimización para encontrar soluciones de problemas de programación lineal localizadas en los vértices de la región factible (soluciones básicas factibles).

Partiendo del PPL en forma estándar, este método se basa en conseguir:

- La **optimalidad**. Se construye una solución básica factible y se elige una variable no básica cuya introducción en la base produzca otra solución básica factible que mejore la función objetivo.
- La **factibilidad**. Al introducir en el apartado anterior una variable nueva en la base, se intenta eliminar otra variable de la base de forma que la nueva variable básica sea factible.

En el siguiente cuadro se muestran las fases del algoritmo:

Algoritmo del Símplex para un problema de maximización

Paso 1. Obtener una solución factible básica inicial.
Paso 2. Condición de optimalidad:

- Si todos los beneficios relativos de la solución $\bar{c}_j \leq 0$, la solución es **óptima** y finalizamos.
- En otro caso, entra en la base la variable no básica x_r tal que

$$\bar{c}_r = \text{máx}\{\bar{c}_j : \bar{c}_j > 0\}.$$

Paso 3. Condición de factibibilidad:

- Sea y_{kr} el elemento en la fila k en la columna pivote r. Si no existe ninguna variable básica x_k tal que $y_{kr} > 0$ entonces la solución no es acotada.
- En otro caso, sale de la base la variable básica x_i tal que

$$\frac{x_i}{y_{ir}} = \min_{y_{kr}>0} \left\{ \frac{x_k}{y_{kr}} \right\},$$

siendo y_{kr} el elemento pivote a partir del cual se obtendrá el siguiente sistema en forma canónica.

Paso 4. Resolvemos el nuevo sistema, obteniendo una nueva solución factible. Volvemos al Paso 2.

Para ver de forma clara el funcionamiento del mismo lo haremos a través del siguiente ejemplo.

Ejemplo 1.5. *Consideremos el siguiente PPL en su forma estándar:*

$$\begin{array}{rrcl} \boldsymbol{Max} & z = 3x_1 + 2x_2 & & \\ \boldsymbol{sujeto\ a} & 2x_1 + x_2 + x_3 & = & 18 \\ & 2x_1 + 3x_2 + x_4 & = & 42 \\ & 3x_1 + x_2 + x_5 & = & 24 \\ & x_1, x_2, x_3, x_4, x_5 & \geq & 0 \end{array} \tag{1.2}$$

Una solución factible básica de este PPL es:

$$x_1 = x_2 = 0, x_3 = 18, x_4 = 42, x_5 = 24,$$

resultando $z = 0$.

El siguiente paso es calcular los beneficios relativos de las variables involucradas en el problema:

$$\textit{Beneficio relativo de } x_1 : x_1 = 1 \Longrightarrow \begin{cases} x_3 = 16 & z = 3 \\ x_4 = 40 & \\ x_5 = 21 & \end{cases}$$

$$\textit{Beneficio relativo de } x_2 : x_2 = 1 \Longrightarrow \begin{cases} x_3 = 17 & z = 2 \\ x_4 = 39 & \\ x_5 = 23 & \end{cases}$$

Entonces tenemos que $\bar{c}_1 = 3$ *y* $\bar{c}_2 = 2$ *y, además, sabemos que la solución anterior no es óptima.*

Por otro lado, una solución básica ***adyacente*** *a otra solución básica difiere de esta última solución en una única variable básica.*

Sobre nuestro ejemplo, incluimos en la base a la variable x_1*, que era la que tenía mayor beneficio relativo. Para que sea factible la nueva solución básica tendrá que verificar:*

$$\begin{aligned} 2x_1 + x_3 &= 18 \\ 2x_1 + x_4 &= 42 \\ 3x_1 + x_5 &= 24 \end{aligned}$$

Para que $x_3, x_4, x_5 \geq 0$*, se debe verificar que* $x_1 \leq 8$*. De esta forma,* x_5 *pasa a ser no básica y* $x_1 = 8$*. Obtenemos así una solución básica adyacente.*

Entonces, la tabla inicial del método Símplex sería:

Primera tabla						
Base	x_1	x_2	x_3	x_4	x_5	Solución
x_3	2	1	1	0	0	18
x_4	2	3	0	1	0	41
x_5	3	1	0	0	1	24
z	3	2	0	0	0	0

Para calcular los beneficios relativos tendremos en cuenta:

$$\bar{c}_i = c_i - c_B P_i,$$

donde c_B son los coeficientes en el objetivo de las variables básicas y P_i es la columna en la tabla de la variable x_i.

Tras un pivotaje se tiene que:

- *Nueva fila pivote = fila antigua pivote dividida por el pivote.*
- *Nueva fila = fila antigua - (coeficiente columna pivote) × (nueva fila pivote).*

Primera tabla							
Base	x_1	x_2	x_3	x_4	x_5	Solución	Razón
x_3	2	1	1	0	0	18	9
x_4	2	3	0	1	0	42	21
x_5	3	1	0	0	1	24	8
z	3	2	0	0	0	0	0

Segunda tabla							
Base	x_1	x_2	x_3	x_4	x_5	Solución	Razón
x_3	0	1/3	1	0	-2/3	2	6
x_4	0	7/3	0	1	-2/3	26	78/7
x_1	0	1/3	0	0	1/3	8	24
z	0	1	0	0	-1	24	

Tercera tabla							
Base	x_1	x_2	x_3	x_4	x_5	Solución	Razón
x_2	0	1	3	0	-2	6	–
x_4	0	0	-7	1	4	12	3
x_1	1	0	-1	0	1	6	6
z	0	1	-3	0	1	30	

Cuarta tabla							
Base	x_1	x_2	x_3	x_4	x_5	Solución	Razón
x_2	0	1	-1/2	1/2	0	12	
x_5	0	0	-7/4	1/4	1	3	
x_1	1	0	3/4	-1/4	0	3	
z	0	0	5/4	1/4	0	33	

Además, una observación muy importante, de la que haremos especial uso en capítulos posteriores, es que si en la tabla óptima aparece algún beneficio relativo de las variables no básicas igual a 0, entonces hay soluciones óptimas alternativas. En otro caso, la solución es única.

Por otro lado, hemos dado los pasos para el algoritmo del Símplex en el caso de que se trate de un problema de maximización. En caso contrario, si el problema es de minimización, podemos proceder de dos formas diferentes:

- Tenemos en cuenta que $\text{mín}\, z = \text{máx}\, -z$.
- Cambia el criterio de entrada, entrando la variable no básica con menor beneficio relativo (≤ 0) y paramos cuando todos los beneficios sean no negativos.

En el Símplex, se parte de una solución básica factible. En ocasiones, esta solución se obtiene fácilmente; por ejemplo considerando variables de holgura. Sin embargo, en otras tendremos que hacer uso de otros métodos para conseguir nuestro objetivo. A continuación, describimos brevemente dos de ellos, donde se permite que variables artificiales desempeñen el trabajo de las holguras en la primera iteración, para después, en alguna iteración posterior, desecharlas de forma legítima.

Método de las penalizaciones.

Este método se basa en introducir variables artificiales con el objetivo de obtener una solución básica factible inicial. Además, se modifica la función objetivo de forma que se imponga una penalización muy grande sobre las variables artificiales para que salgan de la base y entren otras variables.

$$\begin{array}{lrcl} \textbf{Min} & z = cx & & \\ \textbf{s.a.} & Ax & = & b \\ & x & \geq & 0 \end{array} \quad \Longrightarrow \quad \begin{array}{lrcl} \textbf{Min} & z = cx + Mx_a & & \\ \textbf{s.a.} & Ax + Ix_a & = & b \\ & x, x_a & \geq & 0 \end{array}$$

donde x_a es el vector de variables artificiales.

De esta forma, obtenemos dos posibles resultados:

- Obtenemos una solución en la que todas las variables artificiales son 0. Esto indica que el problema inicial tiene solución.
- Obtenemos una solución en la que alguna variable artificial es mayor que 0. Esto indica que el problema inicial no tiene solución.

Método de las Dos Fases.

En este método, se busca la solución básica factible inicial en dos fases.

$$\begin{array}{lrcl} \textbf{Min} & z = cx & & \\ \textbf{s.a.} & Ax & = & b \\ & x & \geq & 0 \end{array} \quad \Longrightarrow \quad \begin{array}{lrcl} \textbf{Min} & z = \sum \text{v.artificiales} & & \\ \textbf{s.a.} & Ax + Ix_a & = & b \\ & x, x_a & \geq & 0 \end{array}$$

donde x_a es el vector de variables artificiales.

1. Si el problema original tiene una solución básica factible, el mínimo se alcanza cuando $x_a = 0$ y $z = 0$.
2. Proseguimos con el método Símplex salvo que en la primera fase hayamos concluido que nuestro problema no tiene soluciones factibles.

1.1.4. Teoría de la Dualidad

Dado un problema de programación lineal, denominado **problema primal**, existe otro problema de programación lineal, denominado **problema dual**, relacionado con él. Se dice que ambos problemas son mutuamente duales.

Bajo ciertas hipótesis, los problemas primal y dual dan lugar al mismo valor óptimo de la función objetivo (por el teorema fundamental de la dualidad) y, por tanto, se puede resolver indirectamente el problema primal resolviendo el problema dual, lo que puede suponer una ventaja computacional relevante.

A continuación, veremos de que manera definir el problema dual a partir del problema primal, tanto para el caso de problemas *simétricos* (todas las variables son no negativas, y todas

las restricciones son desigualdades; si el problema es de maximizar, con $\leq$, y si es de minimizar, con $\geq$), como para el caso de los problemas asimétricos.

Reglas de construcción del problema dual:

1. Se define una variable dual por cada restricción del primal.
2. Se define una restricción dual por cada variable primal.
3. La traspuesta de la matriz de coeficientes del primal es la matriz de coeficientes del dual.
4. El vector de costes del problema primal será el vector de constantes de la derecha del dual.
5. El vector de constantes de la derecha del primal será el vector de costes del problema primal.
6. Se invierte maximizar por minimizar (y viceversa), al igual que el sentido de las desigualdades de las restricciones.

Formulación de los problemas primal-dual simétricos:

Problema PRIMAL				**Problema DUAL**			
Max	z	$=$	cx	**Min**	w	$=$	yb
s.a.	Ax	$\leq$	b	**s.a.**	yA	$\geq$	c
	x	$\geq$	0		y	$\geq$	0

siendo

- c un vector $1 \times n$.
- b un vector $m \times 1$.
- A una matriz $m \times n$.
- x el vector de variables de dimensión $n \times 1$.
- y el vector de variables duales de dimensión $1 \times m$.

A continuación, se puede ver un ejemplo numérico que pone de manifiesto estas condiciones:

$$\begin{array}{rrcl} \textbf{Maximizar} & 5x_1 + 5x_2 & & \\ \textbf{s.a.} & 12x_1 + 8x_2 & \leq & 96 \\ & 6x_1 + 12x_2 & \leq & 72 \\ & x_1, x_2 & \geq & 0 \end{array} \qquad \begin{array}{rrcl} \textbf{Minimizar} & 96y_1 + 72y_2 & & \\ \textbf{s.a.} & 12y_1 + 6y_2 & \geq & 5 \\ & 8y_1 + 12y_2 & \geq & 5 \\ & y_1, y_2 & \geq & 0 \end{array}$$

Teorema 1.1. *Teorema débil de dualidad. Dados dos problemas primal-dual simétricos se tiene que*

$$x_0 \leq y_0 b,$$

donde x_0 e y_0 son soluciones factibles del problema primal y dual, respectivamente.

A partir del teorema anterior, surgen una serie de consecuencias naturales que enumeramos a continuación:

- Cualquier solución factible del dual proporciona una cota superior para z.
- Cualquier solución factible del primal proporciona una cota inferior para w.
- Si el problema primal tiene solución no acotada, el dual no tiene solución.
- Si el dual tiene solución no acotada, el primal no tiene solución.
- Si el primal no tiene solución, el dual la tiene no acotada o no la tiene.
- Si el dual no tiene solución, el primal la tiene no acotada o no la tiene.

Teorema 1.2. *Teorema fundamental de la dualidad. Si el primal y el dual tienen soluciones factibles, entonces tienen soluciones óptimas cuyos valores en el objetivo coinciden.*

Teorema 1.3. *Teorema de las holguras complementarias. Sean x_0 e y_0 soluciones factibles del problema primal y dual, respectivamente, se tiene que*

$$x_0 \text{ e } y_0 \text{ son soluciones óptimas} \iff (y_0 A - c)x_0 + y_0(b - Ax_0) = 0.$$

1.2. El problema de producción lineal

Los problemas de producción lineal son un caso particular de los problemas de programación lineal. En esta sección, además de introducir estos problemas, también se describen las primeras funciones de la librería `coopProductGame`, principal objetivo de este trabajo, que utiliza todos

los puntos descritos en la sección anterior, para conseguir todas las funcionalidades que se pretenden con ella. Para el caso de problemas en dos dimensiones, esta librería permitirá resolver el problema de forma gráfica y, tanto para estos problemas, como para aquellos de una dimensión mayor, se hará uso del método Símplex, tomando como base la librería `lpSolveAPI` [14]. A lo largo del trabajo, a la vez que se introducen los conceptos correspondientes, se irán mostrando todas las funcionalidades de la misma a través de diferentes ejemplos que intentan poner de manifiesto diferentes casuísticas asociadas a la misma.

1.2.1. Definición

En un problema de producción lineal se dispone de una serie de recursos (mano de obra, capital, materiales, maquinaria, espacio disponible, etc) con los que se quieren obtener diferentes productos, a partir de los que se conseguirán beneficios. El objetivo de estos problemas consiste en maximizar los beneficios, teniendo en cuenta las restricciones asociadas a los recursos de los que se dispone. De esta forma, en un problema de producción lineal se tienen los siguientes elementos:

- Hay r recursos.
  El vector $b = (b_k)_{k\in\{1,\dots,r\}} \geq 0$ representa las cantidades que tenemos de los diferentes recursos, es decir, b_k es la cantidad de recurso k disponible.
- Hay p productos.
  Denotaremos el vector de producción por $x = (x_j)_{j\in\{1,\dots,p\}} \geq 0$, donde x_j es la cantidad de producto j que se obtiene.
- La producción de una unidad del producto j requiere a_{kj} unidades del recurso k. De esta forma, se tiene la denominada matriz de producción $A = (a_{kj})_{k\in\{1,\dots,r\},\ j\in\{1,\dots,p\}} \geq 0$.
- Por cada unidad del producto j tenemos un beneficio de c_j unidades. De esta forma, el vector $(c_j)_{j\in\{1,\dots,p\}} \geq 0$ nos indica los beneficios unitarios de los diferentes productos.

Así pues, el objetivo del problema de producción lineal es maximizar los beneficios derivados del proceso de producción teniendo en cuenta los recursos disponibles.

Un problema de producción lineal se puede expresar como el siguiente PPL:

$$\begin{aligned} \textbf{Maximizar} \quad & z = \sum_{j=1}^{p} c_j x_j \\ \textbf{sujeto a} \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{1.3}$$

Veamos un pequeño ejemplo real de un problema de producción lineal para ver de una forma más clara cada uno de los elementos descritos anteriormente, así como la aplicación de una de las funciones de la librería `R` programada.

Ejemplo 1.6. *Una pastelería produce cada día dos tipos de pasteles diferentes que denotaremos por A y B, respectivamente.*

Para elaborar una docena de pasteles de tipo A necesita 4 kg de harina y 6 kg de azúcar. En el caso de los pasteles de tipo B, es necesario disponer de 5 kg de harina y 2 kg de azúcar.

El beneficio que se obtiene por cada docena de pasteles del tipo A es de 68 euros, mientras que cada docena del tipo B que se venda supone un beneficio de 52 euros. Además, diariamente, la empresa dispone de 70 kg de harina y 72 kg de azúcar. El objetivo es calcular el número de docenas de cada tipo que habrá que producir para que el beneficio de la pastelería sea máximo.

Solución:

Para este problema tendríamos:

$$x_1 = \text{“Número de docenas de pasteles del tipo A”}$$

$$x_2 = \text{“Número de docenas de pasteles del tipo B”}$$

El vector de beneficios será $c = (68, 52)^t$*, el vector de recursos* $b = (70, 72)^t$ *y la matriz de producción:*

$$A = \begin{pmatrix} 4 & 5 \\ 6 & 2 \end{pmatrix}$$

El PPL a resolver entonces sería:

$$\begin{array}{rrcl} \textbf{\textit{Maximizar}} & z = 68x_1 + 52x_2 & & \\ \textbf{\textit{sujeto a}} & 4x_1 + 5x_2 & \leq & 70 \\ & 6x_1 + 2x_2 & \leq & 72 \\ & x_1, x_2 & \geq & 0 \end{array} \tag{1.4}$$

Resolveremos este problema a partir de la librería `coopProductGame`*. Aunque el principal objetivo de esta librería se centra en el juego de producción lineal, también podemos utilizarla para resolver un problema de producción lineal. Haremos uso de la función* `productLinearProblem`*, que recibe como argumentos todos los elementos del PPL asociado: vector de beneficios, matriz*

de producción y vector de recursos. Además, esta función permite habilitar la obtención de la solución gráfica (`plot = TRUE`), siempre que el problema sea en dos dimensiones y la salida por consola de los resultados (`show.data = TRUE`).

A continuación, se detalla la sintaxis necesaria para resolver el problema junto con la salida de la misma. De esta forma, se puede ver tanto la solución analítica como gráfica del problema en las que se obtiene como solución el punto $(10, 6)$*, que da lugar a un beneficio de* $z = 992$*, resultado que se puede ver observar en la Figura 1.8.*

```
# Matriz de producción:
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(68, 52)
# Matriz de recursos:
b <- c(70, 72)
# Resolución del problema de producción asociado:
productLinearProblem(c, A, b, plot = TRUE)
```

```
------------------------------------------------------------------------
 Objetive value:
------------------------------------------------------------------------
 [1] "Z = 992"
------------------------------------------------------------------------
 Optimal solution:
------------------------------------------------------------------------
 [1] 10  6
------------------------------------------------------------------------
```

Figura 1.8: Solución gráfica del problema (1.4).

1.2.2. Aplicaciones del problema de producción lineal

Existen múltiples estudios que ponen de manifiesto la aplicabilidad de la optimización lineal en lo que a problemas de producción se refiere. En esta sección se muestran algunas de estas aplicaciones, demostrando así el gran potencial de este tipo de problemas en el mundo real.

Todos los casos que se muestran a continuación persiguen una serie de objetivos comunes, tales como pueden ser la maximización del beneficio que puede obtener la empresa o hacer un mejor uso de los recursos disponibles. De esta forma, se muestra la gran importancia que tiene este tipo de modelos en la industria, permitiendo a las empresas hacer un uso óptimo de sus existencias e intentando prevenir lo que pueda ocurrir en un futuro relativamente cercano.

1.2.2.1. Caso de estudio en una pequeña granja

Es habitual que los agricultores se enfrenten al problema de cómo asignar sus limitados recursos de producción entre las diferentes actividades agrícolas y ganaderas. Éstos siempre intentan buscar la combinación óptima de actividades que maximice sus beneficios. Para poder conseguir esta combinación óptima, lo habitual es que hagan uso tan solo de su instinto y experiencia. Sin embargo, este método no siempre proporciona la solución óptima. Para resolver este tipo de problemas, es de gran utilidad hacer uso de las técnicas de programación lineal.

En [7] se desarrolla un modelo de optimización que intenta calcular cuál sería el patrón de cultivo óptimo para un agricultor que produce principalmente para la subsistencia y vende sus excedentes de producción. Para resolver el problema, los autores hacen uso del software MS Office Excel pero nosotros mostraremos el código necesario para resolverlo a través de nuestra librería.

Este estudio se realizó en el distrito de Bindura, situado en la provincia de Mashonaland Central de Zimbabwe. Se trata de una provincia principalmente rural en la que la agricultura constituye la principal base económica. El agricultor para el que se ha hecho el estudio disponía de un total de 5 hectáreas de terreno destinadas a la producción de maíz, soja, algodón y tabaco. El ingreso bruto esperado era de 285$ por tonelada de maíz, 1325$ por cada hectárea de soja, 525$ por cada hectárea de algodón y 5250$ por cada hectárea de tabaco.

Antes de construirse el modelo de optimización, el plan existente del hogar consistía en asignar 1.5 ha para el maíz, 0.5 ha para la soja, 0.5 ha para el algodón y 0.9 ha para el tabaco. Veremos posteriormente en qué medida esta solución difiere de la proporcionada a través de las técnicas de programación lineal.

En primer lugar, se introduce la notación necesaria para la formulación del problema que, posteriormente, se resolverá con ayuda de la librería `coopProductGame` y se complementará con un análisis de sensibilidad que permitirá obtener conclusiones sobre los resultados obtenidos.

Las variables de decisión del modelo son:

- x_1: hectáreas asignadas a la producción de maíz.
- x_2: toneladas de maíz producidas para la venta.
- x_3: toneladas de maíz reservadas para el consumo.
- x_4: hectáreas asignadas a la producción de soja.
- x_5: hectáreas asignadas a la producción de algodón.
- x_6: hectáreas asignadas a la producción de tabaco.

Los parámetros del modelo son:

- Se dispone de 6 productos. En este caso se combinan tanto hectáreas de terreno reservadas como toneladas para el consumo y la venta. Entonces $\{1, \ldots, 6\}$ es el conjunto de productos y se tendrá que $j \in \{1, \ldots, 6\}$.

- c_j : beneficio unitario en dólares para el producto j.
- a_{1j}: superficie (en hectáreas) necesaria por cada unidad de producto j.
- a_{2j}: tiempo necesario (en días) por cada unidad de producto j.
- a_{3j}: producción de maíz necesaria (en toneladas) por cada unidad de producto j.
- a_{4j}: consumo de maíz necesario (en toneladas) por cada unidad de producto j.
- a_{5j}: dinero necesario (en dolares) por cada unidad de producto j.
- b_1: superficie cultivable (en hectáreas).
- b_2: días de trabajo disponible.
- b_3: proporción de producción del maíz (en toneladas).
- b_4: consumo de maíz de la familia (en toneladas).
- b_5: capital disponible (en dolares).

De esta forma, el PPL se formula de la siguiente manera:

$$
\begin{aligned}
\textbf{Maximizar} \quad & 285x_2 + 1325x_4 + 525x_5 + 5260x_6 \\
\textbf{sujeto a} \quad & x_1 + x_4 + x_5 + x_6 \leq 5 && (1.5)\\
& 30x_1 + 30x_4 + 40x_5 + 40x_6 \leq 312 && (1.6)\\
& x_2 + x_3 \leq 8x_1 && (1.7)\\
& x_3 \geq 2 && (1.8)\\
& 918x_1 + 730x_4 + 365x_5 + 1183x_6 \leq 3000 && (1.9)\\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

En (1.5) se refleja la restricción de hectáreas de terreno disponibles, en (1.6) la restricción por días de trabajo, en (1.7) se recoge la restricción asociada a la producción de maíz (se puede ver que por cada hectárea se producen 8 toneladas de maíz), en (1.8) se tiene la limitación asociada al consumo propio de maíz y (1.8) representa la limitación del dinero que habría que invertir en la materia prima.

Este PPL, se puede traducir a un sencillo problema de producción, donde las variables de decisión serían:

- y_1: hectáreas dedicadas al cultivo de maíz para vender.
- y_2: hectáreas dedicadas al cultivo de soja para vender.
- y_3: hectáreas dedicadas al cultivo de algodón para vender.
- y_4: hectáreas dedicadas al cultivo de tabaco para vender.

Dado que el consumo propio de maíz no reporta beneficios en la venta y únicamente se exige que se produzcan 2 toneladas, lo óptimo será que, en principio, fijemos un consumo propio de 2 Tn o, lo que es lo mismo, dedicar $1/4$ de ha al consumo propio de maíz. Teniendo esto en cuenta tenemos la siguiente disponibilidad de recursos:

- Héctareas: $5 - 0.25 = 4.75\ ha$.
- Días: $312 - 30 \cdot 0.25 = 312 - 7.5 = 304.5$ días.
- Inversión: $3000 - 918 \cdot 0.25 = 3000 - 229.5 = 2770.5\$$.

De esta forma, teniendo en cuenta que el beneficio por hectárea dedicada a la producción de maíz para la venta es de $285 \cdot 4 = 2280\$$ (ya que por cada hectárea se producen 8 toneladas), el problema de producción vendrá dado por:

$$
\begin{array}{rrcl}
\textbf{Maximizar} & 2280y_1 + 1325y_2 + 525y_3 + 5260y_4 & & \\
\textbf{sujeto a} & y_1 + y_2 + y_3 + y_4 & \leq & 4.75 \\
& 30y_1 + 30y_2 + 40y_3 + 40y_4 & \leq & 304.5 \\
& 918y_1 + 730y_2 + 365y_3 + 1183y_4 & \leq & 2770.5 \\
& y_1, y_2, y_3, y_4 & \geq & 0
\end{array}
$$

Para resolver el problema, haremos uso de nuestro paquete, mostraremos también un pequeño análisis de sensibilidad de forma complementaria que permitirá analizar los resultados en detalles y compararemos estos resultados con los planteados previamente por el agricultor en base a su experiencia e intuición.

Así pues, podemos hacer uso de la función `productLinearProblem` de nuestra librería con el código que se recoge a continuación y cuya solución aparece reflejada en la Tabla 1.2. En ella, se han añadido también la sensibilidad de cada una de las variables, es decir, el intervalo en el que podría variar el beneficio asociado a cada una de ellas, manteniéndose la solución óptima que hemos obtenido.

```
# Matriz de producción
A <- matrix(c(1, 1, 1, 1,
              30, 30, 40, 40,
              918, 730, 365, 1183), ncol = 4, byrow = TRUE)

# Vector de beneficios:
c <- c(2280, 1325, 525, 5250)

# Vector de recursos:
b <- c(4.75, 304.5, 2770.5)

# Resolución del problema de producción asociado:
productLinearProblem(c, A, b, show.data = TRUE)
```

Variable	Valor según el modelo LP	Coste Reducido	Valor según el agricultor
y_1	0	$[-\infty, 4073.964)$	1.25
y_2	0	$[-\infty, 3239.645)$	0.5
y_3	0	$[-\infty, 1619.822)$	0.5
y_4	2.34	$[2938.17, \infty)$	0.9

Tabla 1.2: Solución del modelo

La solución óptima de este problema se basa en producir 2.34 *ha* de tabaco, obteniendo así un total de $12,295.1$\$. Podemos observar que el beneficio obtenido procede de la producción de tabaco. Si analizamos la solución propuesta por el agricultor, vemos que con su propuesta el beneficio sería de $8,500$\$ que, si comparamos con la generada gracias a la ayuda de las técnicas de programación lineal, se puede mejorar en casi un 45 %. Vemos de esta forma que los métodos tradicionales no obtienen una solución óptima y que las técnicas descritas en las secciones anteriores pueden ayudar a aumentar el benificio obtenido de forma considerable.

Por otro lado, a partir de los costes reducidos que aparecen recogidos también en la Tabla 1.2, podemos observar que para que varíe la solución óptima y empiece a compensar producir cualquier otro tipo de producto que no sea tabaco, los beneficios asociados deberían ser mucho mayores; así, por ejemplo, el precio por *ha* de soja debería ser mayor de 3239\$ en lugar de 1325\$ o que, en el caso del algodón, debería pasar de 525\$ a más de 1619\$. De esta forma, vemos que lo más beneficioso va a ser producir la mayor cantidad de *ha* de tabaco posibles, siempre

respetando las restricciones que aseguren los consumos propios.

Además, en la Tabla 1.3 podemos ver cuáles son las restricciones que están saturadas, es decir, aquellas de las que se está consumiendo todo el recurso disponible y en cuáles tenemos sobrante. De esta forma, vemos que la restricción del capital disponible es la que limita el beneficio obtenido, sobrando tanto días de trabajo como parte de las hectáreas disponibles por la familia.

Restricción	RHS actual	Sobrante	Precio dual
1	4.5	2.408	0
2	304.5	210.822	0
3	2770.5	0	4.43787

Tabla 1.3: Sobrante y precios duales

1.2.2.2. Caso de estudio en la Industria del Té

En esta sección se analizará un ejemplo real, relacionado con la industria del té, que se encuentra recogido en [9] y que detallaremos a continuación. En primer lugar, veremos de que manera se puede plantear la formulación del mismo como un problema de producción lineal. A continuación, pasaremos a resolverlo a través de las funciones programadas en nuestra librería y, finalmente, haremos un análisis de sensibilidad en detalle para poder ver cuáles podrían ser algunas posibles modificaciones sobre el problema original que puedan producir cambios sobre la solución óptima, viendo de nueavo el gran potencial de la modelización matemática a la hora de resolver este tipo de problemas.

Sri Lanka es el tercer país con mayor producción de té a nivel mundial, con una cuota de producción del 9 % en el mercado internacional, y uno de los principales exportadores mundiales con una cuota aproximada del 19 % de la demanda mundial.

En este estudio, seleccionaron a la empresa Dilmah Tea Company, creada en 1974, y considerada una empresa de exportación líder en Sri Lanka. Usan sus propias hojas de té y fabrican más de 25 productos diferentes. El presente estudio se focaliza en dos objetivos principales: en primer lugar, formular un modelo matemático que proponga una mezcla de productos viable para asegurar el máximo beneficio y, por otro lado, poner de manifiesto cómo la aplicación de las técnicas de programación lineal permite ser más rentable a la compañía.

A continuación se define la notación necesaria para el planteamiento del modelo, que posteriormente resolveremos a través de nuestras funciones programadas en `R`.

Parámetros del modelo:

- Se consideran p tipos de té. De esta forma, $\{1, \ldots, p\}$ es el conjunto de productos (o tipos de té) y tomaremos el índice $j \in \{1, \ldots, p\}$.
- c_j: beneficio unitario en dólares para el producto j.
- a_{1j}: materia prima (en kg) necesaria por cada unidad de producto j.
- a_{2j}: tiempo de maquinaria (en minutos) necesario para cada unidad del producto j.
- a_{3j}: tiempo de trabajo (en minutos) necesario para cada unidad de producto j.
- b_1: cantidad total de materia prima disponible (en kg).
- b_2: capacidad total de tiempo de la maquinaria en minutos.
- b_3: tiempo de trabajo disponible (en minutos).
- U_j: el límite superior de la demanda (en kg) para el producto j.

Variables de decisión:

- x_j, número de unidades del producto j a ser fabricadas.

El PPL se formula como:

$$\text{máx} \quad \sum_{j=1}^{p} c_j x_j \tag{1.10}$$

$$\text{s.a.} \quad \sum_{j=1}^{p} a_{1j} x_j \leq b_1 \tag{1.11}$$

$$\sum_{j=1}^{p} a_{2j} x_j \leq b_2 \tag{1.12}$$

$$\sum_{j=1}^{p} a_{3j} x_j \leq b_3 \tag{1.13}$$

$$x_j \leq U_j \tag{1.14}$$

$$x_j \geq 0$$

En (1.11) se limita la disponibilidad de las materias primas existentes, en (1.12) se limita la capacidad total de la maquinaria y en (1.13) la capacidad laboral de las personas propiamente dicha. Además, se tiene la restricción (1.14), que tiene en cuenta las limitaciones de la demanda.

Los datos concretos para la formulación del problema aparecen recogidos en la Tabla 1.4:

Producto	Materia Prima(kg)	Capacidad de las máquinas(min)	Capacidad laboral(min)	Demanda media máxima	Beneficio por unidad($)
1	1.76	0.4	0.46	4000	3.8
2	1.2	0.66	0.75	15000	2.91
3	3	0.24	0.27	2100	5.73
4	4.8	0.88	1	1000	11.17
5	0.6	0.22	0.25	20000	1.57
6	2.4	1.33	1.52	1200	5.67
7	1.92	0.88	1	5000	4.72
8	1.44	0.3	0.34	12000	3.37
9	1.44	0.48	0.55	6200	3.39
10	2.4	6.6	7.52	3000	7.02
11	0.6	0.34	0.39	14000	2.24
12	0.6	0.83	0.95	8200	3.54
13	0.6	0.45	0.51	16000	1.95
14	1.5	0.48	0.55	5000	4.6
15	0.75	0.66	0.75	12000	2.52
16	0.24	0.24	0.27	22000	3.27
17	0.75	0.9	1.03	6500	3.66
18	0.3	0.3	0.34	18000	1.88
19	0.9	0.9	1.03	4000	3.71
20	0.6	0.83	0.95	7500	3.4
21	0.3	0.3	0.34	9000	3.32
22	1.2	0.4	0.46	7500	1.7
23	2	1.14	1.3	3500	4.19
24	1.08	0.48	0.55	4500	2.65
25	0.45	0.45	0.51	10000	2.38
Disponibilidad de recursos	175000	168000	192000		

Tabla 1.4: Datos de entrada

En [9] se resuelve el problema a través del entorno "LINGO"; sin embargo, al igual que en el ejemplo anterior, haremos uso de nuestro paquete, obteniendo la solución reflejada en la Tabla 1.5. En ella, se han añadido también el intervalo en el que podría variar el beneficio asociado a cada una de ellas, manteniéndose la solución óptima actual.

```
# Materia prima necesaria por cada unidad de producto
a1 <- c(1.76, 1.2, 3, 4.8, 0.6, 2.4, 1.92, 1.44,
        1.44, 2.4, 0.6, 0.6, 0.6, 1.5, 0.75, 0.24,
        0.75, 0.3, 0.9, 0.6, 0.3, 1.2, 2, 1.08, 0.45)

# Tiempo de maquinaria necesario para cada unidad de producto
a2 <- c(0.4, 0.66, 0.24, 0.88, 0.22, 1.33, 0.88, 0.3,
        0.48, 6.6, 0.34, 0.83, 0.45, 0.48, 0.66, 0.24,
        0.9, 0.3, 0.9, 0.83, 0.3, 0.4, 1.14, 0.48, 0.45)

# Tiempo de trabajo necesario para cada unidad de produco
a3 <- c(0.46, 0.75, 0.27, 1, 0.25, 1.52, 1, 0.34, 0.55,
        7.52, 0.39, 0.95, 0.51, 0.55, 0.75, 0.27, 1.03,
        0.34, 1.03, 0.95, 0.34, 0.46, 1.3, 0.55, 0.51)

# Demanda media máxima de cada producto
x <- diag(25)

# Matriz de producción
A <- rbind(a1, a2, a3, x)

# Vector de beneficios:
c <- c(3.8, 2.91, 5.73, 11.17, 1.57, 5.67, 4.72, 3.37,
       3.39, 7.02, 2.24, 3.54, 1.95, 4.6, 2.52, 3.27, 3.66,
       1.88, 3.71, 3.4, 3.32, 1.7, 4.19, 2.65, 2.38)

# Vector de recursos:
b <- c(175000, 168000, 192000, 4000, 15000, 2100, 1000, 20000,
       1200, 5000, 12000, 6200, 3000, 14000, 8200, 16000, 5000,
       12000, 22000, 6500, 18000, 4000, 7500, 9000, 7500, 3500,
       4500, 10000)

# Resolución del problema de producción asociado:
productLinearProblem(c, A, b, show.data = TRUE)
```

Los resultados nos muestran que todas las variables de decisión contribuyen a la función objetivo excepto x_{22}, produciendo un beneficio total de 619161.7$. Además, se puede ver que

la variable x_{16} con una contribución de 22000 fue la variable que más aportó, seguida de las variables x_5 y x_{18} con 20000 y 18000, respectivamente.

Los costes reducidos, que también podemos ver en la Tabla 1.5, nos proporcionan información de cuál es el intervalo en que podríamos variar los coeficientes de la función objetivo sin que varien las variables que intervienen en la solución óptima actual. Por ejemplo, vemos que el intervalo de variación para la variable x_{22} es $(-\infty, 2.292)$; esto quiere decir que si el coeficiente de la función objetivo correspondiente a esta variable fuese superior a 2.292 o, lo que es lo mismo, lo aumentásemos en 0.592, x_{22} pasaría a ser una variable básica y, por tanto, en ese momento, compensaría fabricar ese tipo de té. Este análisis, que también podemos ver de forma más visual en la Figura 1.9, da a la empresa un gran indicador de cara a tomar las correspondientes decisiones de negocio que, por un lado, puede ser dejar de producir dicho tipo de té, o aumentar ligeramente su producción, cambiando de esta forma la solución óptima del problema.

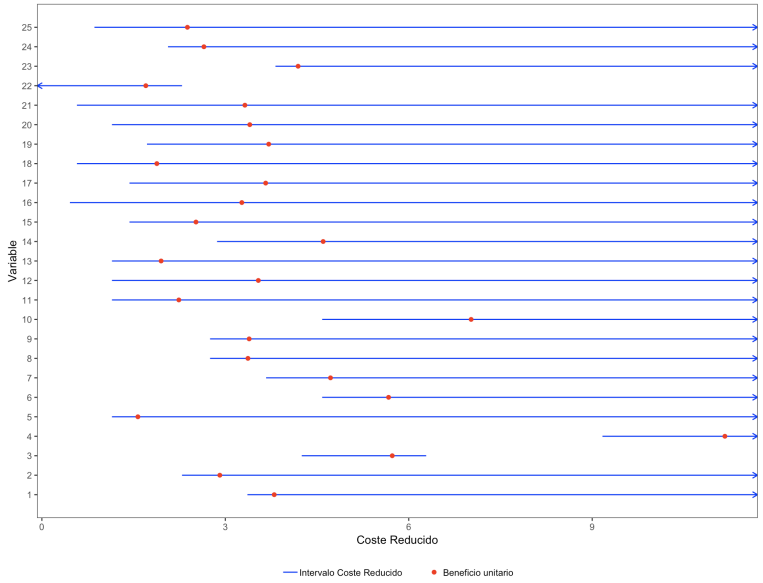


Figura 1.9: Representación gráfica de los costes reducidos y beneficio unitario actual.

Por otro lado, suele resultar muy interesante estudiar cómo un cambio en el valor de los lados derechos de las restricciones puede cambiar el valor óptimo del problema. En este sentido, se pueden emplear los precios sombra de las restricciones, obteniendo el resultado que se puede ver en la Tabla 1.6 y de forma visual en la Figura 1.10. Técnicamente, el precio sombra de una restricción indica cuánto cambia el valor de la función objetivo (óptimo) ante una variación marginal del lado derecho de la restricción, asumiendo que el resto de parámetros del modelo

permanecen constantes. Por ejemplo, si el valor derecho de la restricción correspondiente a la materia prima aumenta en una unidad, es decir, un kg de té, la ganancia aumentará en 1.91$. De esta forma, podemos observar que los productos x_{19} y x_{24} son aquellos en los que la empresa debería invertir más ya que son los que más aumentarían las ganancias, suponiendo 2.8116$ y 2.747$, respectivamente.

Por otro lado, se puede ver que el precio sombra de muchas de las variables es igual a 0, lo cual implica que, aunque aumentásemos la cantidad de recurso correspondiente, la función objetivo no aumentaría. Además, en la misma tabla se puede ver cuál sería el sobrante de los recursos (coincidiendo las restricciones con sobrante mayor que cero con aquellas en las que el precio sombra es cero). Un ejemplo de ello se encuentra en la restricción correspondiente al tiempo de maquinaria necesario para la producción, pudiendo observar que sobran un total de 49647.04 minutos.

Con este análisis vemos el gran potencial que nos pueden dar los problemas de producción lineal. Este tipo de estudios pueden ser de gran utilidad para la toma de decisiones de la empresa, permitiendo analizar rápidamente varios escenarios bajo la pregunta "¿Qué pasaría si ...?" Los responsables de los procesos de producción deben garantizar que los recursos disponibles se utilicen de manera que se mantengan los valores óptimos del modelo para maximizar los beneficios correspondientes.

Tras nuestro análisis de sensibilidad, seríamos capaces de analizar cuáles son los recursos que más escasean y aumentar los mismos, así como ver cuáles son los costes innecesarios que intervienen en el proceso de producción o, lo que es lo mismo, aquellos gastos en recursos que no se están empleando que, en este caso, se corresponden sobre todo al tiempo de maquinaria y al tiempo de trabajo necesario para la producción de los productos.

Variable	Valor	Coste Reducido
x_1	4000	$[3.3616, \infty)$
x_2	15000	$[2.292, \infty]$
x_3	1712.333	$[4.25, 6.285]$
x_4	1000	$[9.168, \infty)$
x_5	20000	$[1.146, \infty)$
x_6	1200	$[4.584, \infty)$
x_7	5000	$[3.6672, \infty)$
x_8	12000	$[2.7504, \infty)$
x_9	6200	$[2.7504, \infty)$
x_{10}	3000	$[4.584, \infty)$
x_{11}	14000	$[1.146, \infty)$
x_{12}	8200	$[1.146, \infty)$
x_{13}	16000	$[1.146, \infty)$
x_{14}	5000	$[2.865, \infty)$
x_{15}	12000	$[1.4325, \infty)$
x_{16}	22000	$[0.4584, \infty)$
x_{17}	6500	$[1.4325, \infty)$
x_{18}	18000	$[0.573, \infty)$
x_{19}	4000	$[1.719, \infty)$
x_{20}	7500	$[1.146, \infty)$
x_{21}	9000	$[0.573, \infty)$
x_{22}	0	$(-\infty, 2.292)$
x_{23}	3500	$[3.82, \infty)$
x_{24}	4500	$[2.0628, \infty)$
x_{25}	10000	$[0.8595, \infty)$

Tabla 1.5: Solución del modelo

Restricción	RHS actual	Sobrante	Precio dual
1	175000	0	1.91
2	168000	49647.04	0
3	192000	57228.67	0
4	4000	0	0.4384
5	15000	0	0.618
6	2100	387.67	0
7	1000	0	2.002
8	20000	0	0.424
9	1200	0	1.086
10	5000	0	1.0528
11	12000	0	0.6196
12	6200	0	0.6396
13	3000	0	2.436
14	14000	0	1.094
15	8200	0	2.394
16	16000	0	0.804
17	5000	0	1.735
18	12000	0	1.0875
19	22000	0	2.8116
20	6500	0	2.2275
21	18000	0	1.307
22	4000	0	1.991
23	7500	0	2.254
24	9000	0	2.747
25	7500	7500	0
26	3500	0	0.37
27	4500	0	0.5872
28	10000	0	0.5205

Tabla 1.6: Sobrante y precios duales

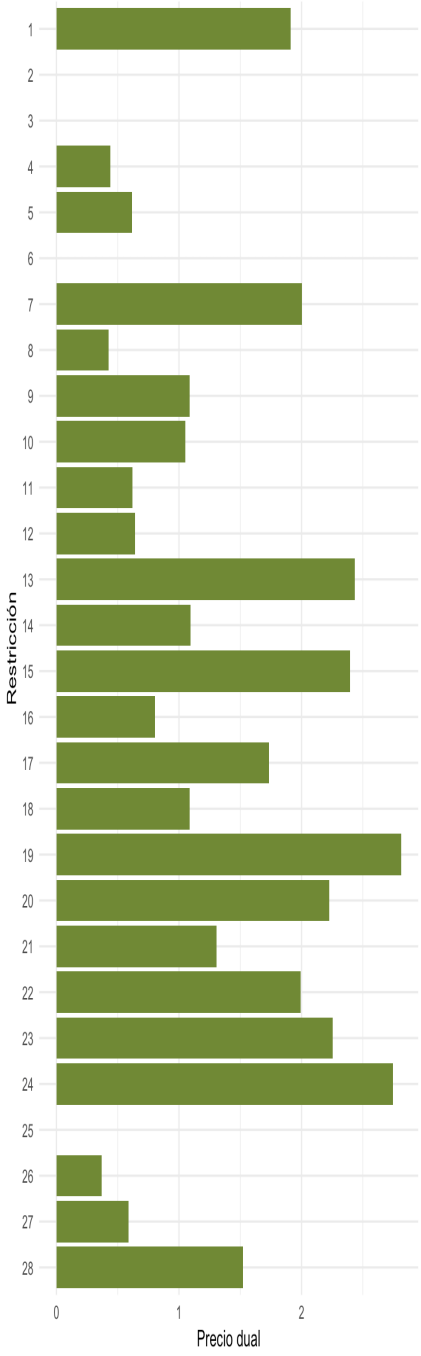


Figura 1.10: Representación gráfica precios duales.

Capítulo 2

Cooperación en el problema de producción lineal

La Teoría de Juegos intenta modelizar ciertas situaciones en las que existen varios agentes o jugadores que buscan diferentes objetivos, habitualmente independientes unos de otros, y con la propiedad de que las decisiones o preferencias de cada jugador afectan a lo que pueden conseguir todos los demás. En este sentido, la Teoría de Juegos puede dividirse en dos grandes bloques: los juegos estratégicos, también llamados juegos no cooperativos, y los juegos coalicionales, también llamados juegos cooperativos.

En contraposición a los juegos estratégicos, en los que están especificadas todas las posibles acciones, en los juegos cooperativos este conjunto de acciones no se indica explícitamente, sino que se consideran todos los posibles resultados de una hipotética cooperación. De esta forma, en un modelo cooperativo, se debe especificar cuáles son los beneficios de cooperación ya no solo de todos los jugadores, sino también de cada posible coalición que pueda formarse.

Aún cuando la cooperación es posible, puede ocurrir que la utilidad pueda repartirse de cualquier modo entre los jugadores, o bien que existan ciertas restricciones que no permitan cualquier reparto. Es por ello que la Teoría de Juegos Cooperativos está dividida en dos partes: juegos cooperativos con utilidad transferible (juegos TU) y juegos cooperativos con utilidad no necesariamente transferible (juegos NTU). En este punto, hay que destacar que el modelo NTU recoge como caso particular al modelo TU.

Este capítulo se centra en el estudio de los juegos de producción lineal, que se engloban dentro de la Teoría de Juegos Cooperativos. Para ello, en primer lugar se introduce cierta notación de los juegos cooperativos, necesaria para comprender el resto del trabajo y, a continuación, se

estudian distintas soluciones que han aparecido a lo largo de los años.

2.1. Juegos cooperativos con utilidad transferible

2.1.1. Introducción

El modelo de utilidad transferible se utiliza para modelizar situaciones en las que la cooperación beneficia a los agentes, ya sea en términos de ganancias o en términos de costes. Las distintas soluciones que se plantean bajo las condiciones de este modelo proponen repartos de los beneficios obtenidos tras la cooperación.

Además, supondremos que los jugadores o agentes negocian con un bien infinitamente divisible o, en caso de que esto no ocurra, podremos asumir que tienen acceso a otro bien compensatorio que sí lo sea (habitualmente dinero). De esta forma, los elementos clave para definir el modelo TU son los siguientes:

- N: conjunto finito de jugadores.
- 2^N: conjunto de todos los subconjuntos de N.

Definición 2.1. *Un juego cooperativo, o coalicional, es un par (N, v), siendo N un conjunto finito de jugadores y v una función*

$$v : 2^N \longrightarrow \mathbb{R}$$

*tal que $v(\emptyset) = 0$. A los elementos de N se les llama **jugadores**, a los subconjuntos de N **coaliciones** y v se denomina **función característica**. De esta forma, una coalición se define como cualquier subconjunto $S \subseteq N$ con $|S|$ elementos.*

Denotaremos por G^N a la clase de todos los juegos con conjunto de jugadores N.

A continuación, se introducen una serie de propiedades básicas de los juegos descritos en la definición anterior.

Definición 2.2. *Sea $(N, v) \in G^N$. Diremos que (N, v) es*

- ***Aditivo** si $v(S \cup T) = v(S) + v(T)$ para todo $S, T \in 2^N$, $S \cap T = \emptyset$.*
- ***Convexo** si $v(S) + v(T) \leq v(S \cup T) + v(S \cap T)$ para todo $S, T \in 2^N$.*
- ***Cóncavo** si $(N, -v)$ es convexo.*

Las condiciones de convexidad también pueden reformularse de un modo más intuitivo, en términos de las contribuciones marginales de los jugadores a las coaliciones. De hecho, la definición anterior de juego convexo es equivalente a la siguiente.

Definición 2.3. *Sea* $(N, v) \in G^N$. *Se tiene que:*

$$(N, v) \text{ es convexo} \Leftrightarrow v(S \cup \{i\}) - v(S) \geq v(T \cup \{i\}) - v(T) \quad \forall S, T \in 2^N \ : \ T \subset S \subset N \setminus \{i\}.$$

Con esta caracterización, un juego es convexo si la contribución de cada jugador a una coalición no decrece cuantos más jugadores tenga la coalición (también conocido como *efecto bola de nieve*).

Los juegos cooperativos TU buscan responder a una serie de preguntas básicas: ¿Qué coalición o coaliciones se van a formar? ¿Cómo se dividen las ganancias entre los jugadores?

El principal objetivo en este tipo de juegos es que se forme la gran coalición N y que los jugadores se repartan entre ellos el beneficio (o el coste). Cuando se intenta conseguir este objetivo surgen varios enfoques diferentes. Por un lado, está el enfoque basado en la idea de estabilidad, en la cual se busca encontrar un conjunto de repartos o asignaciones que sea estable, en el sentido de que el reparto final sea un elemento de dicho conjunto. De este enfoque resultan, entre otras, soluciones en *el núcleo* [8], *los conjuntos estables* [26] y *el conjunto de negociación* [1]. Por otro lado, se plantea también la idea de ecuanimidad, idea que trata de proponer para cada juego un reparto ecuánime que pueda ser aceptado por los jugadores. En este caso se tienen el *valor de Shapley* [20] y el *nucleolo* [19], entre otros.

Formalmente, un reparto es un vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, donde cada componente x_i representa la cantidad asignada al jugador i. Dado un reparto $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, la suma de las cantidades asignadas a los miembros de una coalición $S \subseteq N$ se denotará por $x(S) = \sum_{i \in S} x_i$. De esta forma, surgen dos criterios mínimos que definimos a continuación.

- **Racionalidad individual**. Un reparto $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ satisface la racionalidad individual si cada jugador recibe un pago que no es inferior a lo que pueda garantizarse por sí mismo, es decir, $x_i \geq v(i)$ para todo $i \in N$ (en caso de beneficios).
- **Eficiencia**. Un reparto $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ se dice que es eficiente si el valor de la gran coalición $v(N)$ se distribuye entre los jugadores o, lo que es lo mismo, $x(N) = x_1 + \cdots + x_n = v(N)$.

Además, los repartos que cumplen estos dos criterios se denominan imputaciones, concepto que describimos formalmente en la siguiente definición:

Definición 2.4. *Sea (N, v) un juego TU.*

- *Se define el conjunto de preimputaciones del juego (N, v) como el conjunto de todos los repartos eficientes:*

$$I^*(N, v) = \{x = (x_i)_{i \in N} \in \mathbb{R}^n : \sum_{i \in N} x_i = v(N)\}.$$

- *El conjunto de imputaciones de un juego (N, v) está formado por los repartos eficientes que verifiquen además la propiedad de racionalidad individual:*

$$I(N, v) = \{x = (x_i)_{i \in N} \in I^*(N, v) : x_i \geq v(\{i\}), \text{ para todo } i \in N\}.$$

2.1.2. El núcleo

En esta sección nos centraremos en el concepto de núcleo, introducido por Gillies en 1953 [8] y sobre el cual se centran muchos de los resultados de este trabajo. El núcleo es el conjunto de las imputaciones que satisfacen, además, una condición de racionalidad coalicional.

La **racionalidad coalicional** supone que a cada coalición $S \subseteq N$ debemos darle, al menos, lo que se puede garantizar por sí misma, formalmente, $x(S) = \sum_{i \in S} x_i \geq v(S)$, para cada $S \subseteq N$. Los repartos que cumplen esta característica se conocen como repartos estables, en el sentido de que ninguna coalición tendrá incentivos para desviarse y obtener un reparto mejor (en sentido estricto) que lo que le asegura un reparto en el núcleo.

Definición 2.5. *Dado un juego (N, v), se define el **núcleo** como el conjunto:*

$$C(N, v) = \Big\{x \in I(N, v) : \sum_{i \in S} x_i \geq v(S), \text{ para todo } S \subseteq N, S \neq \emptyset\Big\}.$$

Como bien hemos comentado anteriormente, el núcleo busca la condición de estabilidad. Así pues, a continuación se muestran una definición y un resultado que permiten ver de otra manera la relación entre el núcleo y dicha condición de estabilidad.

Definición 2.6. *Sea $v \in G^N$ un juego TU y sean $S \in 2^N \setminus \emptyset$ $x, y \in I(N, v)$. Se dice que x domina a y a través de S si se cumple que:*

1. *$x_i > y_i$, para todo $i \in S$.*
2. *$\sum_{i \in S} x_i \leq v(S)$.*

Se dice que x domina a y si existe una coalición $T \in 2^N \setminus \emptyset$ tal que x domina a y a través de T. Se dice que x es no dominada si no existe $z \in I(N,v)$ tal que z domina a x.

Nótese que x domina a y a través de S cuando todos los jugadores de S prefieren x a y. Además, la cantidad que propone x para los agentes de S nunca es superior a la cantidad que pueden garantizarse por sí mismos. En resumen, para que una imputación sea realmente estable debe ser no dominada. Esto se pone de manifiesto con el siguiente resultado.

Proposición 2.1. *Sea $v \in G^N$ un juego TU.*

Si $x \in C(N,v)$, entonces x es no dominada.

Existen resultados que, de manera relativamente sencilla, permiten ver aquellas condiciones que los juegos deben verificar para poder garantizar su estabilidad. A continuación, se proporciona una condición necesaria y suficiente para el carácter no vacío del núcleo de un juego. El resultado asociado fue probado de forma independiente por Bondareva (1963) [2] y Shapley (1967) [21].

Definición 2.7. *Una familia de coaliciones $F \subset 2^N \setminus \{\emptyset\}$ se denomina **equilibrada** si existe una familia asociada de números reales positivos, denominados coeficientes de equilibrio, $\{y_S : S \in F\}$ tal que, para todo $i \in N$,*

$$\sum_{S \in F, i \in S} y_S = 1.$$

Definición 2.8. *Un juego $v \in G^N$ se denomina **equilibrado** si, para cualquier familia de coaliciones equilibrada F con coeficientes de equilibrio $\{y_S : S \in F\}$, se tiene que*

$$\sum_{S \in F} y_S v(S) \leq v(N).$$

En otras palabras, que un juego sea equilibrado quiere decir que las coaliciones "intermedias" no tienen demasiado poder. De esta forma, dicha propiedad parece estar relacionada con la estabilidad de la situación de negociación coalicional descrita por el propio juego. Esto es precisamente lo que afirma el teorema de Bondareva-Shapley.

Teorema 2.1. *Sea $v \in G^N$ un juego TU. Entonces $C(N,v) \neq \emptyset$ si y solo si v es equilibrado.*

A continuación se recogen una serie de resultados interesantes de los que, además, haremos uso más adelante.

Proposición 2.2. *Sea (N,v) un juego superaditivo. Entonces*

$$(N,v) \textit{es aditivo} \Longleftrightarrow v(N) = \sum_{k \in N} v(k).$$

Definición 2.9. *(Schemeidler, 1972) [18]. Sea (N, v) un juego cuyo núcleo no es vacío. Diremos que (N, v) es exacto si para cada $S \subset N$, existe $x \in C(N, v)$ tal que $\sum_{i \in S} x_i = v(S)$.*

Definición 2.10. *(Weber, 1988) [27]. Sea (N, v) un juego cooperativo con utilidad transferible y sea $\sigma : N \longrightarrow N$ una permutación de N. El vector marginal m^σ se define como*
$m^\sigma_{\sigma(1)} := v(\sigma(1))$.
$m^\sigma_{\sigma(k)} := v(\sigma(1), \ldots, \sigma(k)) - v(\sigma(1), \ldots, \sigma(k-1))$ *para todo $k \in N \setminus \{1\}$.*

El conjunto de Weber $W(N, v)$ se define como la envoltura convexa de los $n!$ vectores marginales.

Teorema 2.2. *(Shapley, 1971 [22]; Ichiishi, 1981 [11]; Derks, 1992 [5]). Para todo juego cooperativo (N, v) se tiene que:*

$$(N, v) \text{ es convexo } \iff W(N, v) = C(N, v).$$

Proposición 2.3. *Todo juego convexo es exacto. Además, si $n \leq 3$ las propiedades de convexidad y exactitud son equivalentes.*

2.1.3. El valor de Shapley

En esta sección veremos una de las reglas de reparto más importantes cuando hablamos de juegos cooperativos con utilidad transferible: el valor de Shapley [20]. La aproximación de Shapley intenta proponer, para cada juego TU, una asignación que sea un compromiso aceptable para los jugadores siguiendo la idea de ecuanimidad.

Antes de definir esta regla introducimos dos tipos de jugadores especiales, así como el concepto de regla de asignación.

Definición 2.11. *Sea $(N, v) \in G^N$.*

- *Un jugador $i \in N$ se llama **títere** si no aporta beneficio adicional a los demás jugadores, es decir, si*
$$v(S \cup \{i\}) = v(S) + v(\{i\}) \qquad \forall S \subseteq N \setminus \{i\}.$$
*Si un jugador títere $i \in N$ verifica que $v(\{i\}) = 0$, se dice además que es un **jugador nulo**.*

- *Dos jugadores $i, j \in N$ son **simétricos** si*
$$v(S \cup \{i\}) = v(S \cup \{j\}) \qquad \forall S \subseteq N \setminus \{i, j\}.$$

Definición 2.12. *Una regla de asignación es una aplicación $\varphi : G^N \longrightarrow \mathbb{R}^n$ que asocia a cada juego (N, v) un vector $\varphi(N, v) := (\varphi_i(N, v))_{i \in N} \in \mathbb{R}^n$. Para cada jugador $i \in N$, $\varphi_i(N, v)$ no es más que la asignación que φ propone para el agente i en (N, v).*

De esta forma, se puede ver el valor de Shapley como la regla de asignación que se describe a continuación.

Definición 2.13. *Sea $(N, v) \in G^N$. El valor de Shapley para cada jugador $i \in N$ se define como*

$$Sh_i(N, v) := \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)].$$

Una forma de interpretarlo es la siguiente: supongamos que los jugadores van llegando a un lugar de forma aleatoria y, al llegar, cada jugador recibe como utilidad su aportación marginal a los jugadores que ya están presentes. El valor de Shapley se puede ver como el vector de utilidades esperadas de los jugadores bajo este procedimiento.

En 1953, Shapley introdujo esta regla como la única que verificaba el conjunto de propiedades que describimos a continuación:

- **Eficiencia.** La regla de reparto φ es eficiente si para todo $(N, v) \in G^N$,

  $$\sum_{i \in N} \varphi_i(N, v) = v(N).$$

  Esta propiedad nos indica que la suma de las asignaciones que reciben todos los jugadores ha de coincidir con el valor de la gran coalición.

- **Jugador nulo.** La regla de reparto φ verifica jugador nulo si para todo $(N, v) \in G^N$ e i jugador nulo,

  $$\varphi_i(N, v) = 0.$$

  De esta forma, si el jugador no realiza aportación adicional entonces su asignación individual será nula.

- **Simetría.** La regla de reparto φ verifica simetría si para todo $(N, v) \in G^N$ y todo par i, j de jugadores simétricos,

  $$\varphi_i(N, v) = \varphi_j(N, v).$$

  En otras palabras, si dos jugadores son intercambiables en el juego, deben recibir el mismo pago.

- **Aditividad.** La regla de reparto φ verifica aditividad si para todo $(N,v),(N,w) \in G^N$, se verifica que

$$\varphi(N, v+w) = \varphi(N,v) + \varphi(N,w).$$

La propiedad de aditividad establece que el pago que reciben los jugadores en un juego es igual a la suma de los pagos que recibirán si el juego se descompone en suma de dos.

Además, Shapley demostró el siguiente resultado:

Teorema 2.3. *Sea $(N,v) \in G^N$ un juego convexo. Entonces $Sh(N,v) \in C(N,v)$.*

Este hecho se podría demostrar fácilmente si se tiene en cuenta que el núcleo de un juego convexo es la envoltura convexa de los vectores de contribuciones marginales y, justamente, el valor de Shapley es el promedio de ellos.

2.1.4. El nucleolo

El concepto de nucleolo fue introducido por Schmeidler(1969) [19] y se basa en la noción básica de exceso de coalición. El exceso de una coalición S en un vector x se define como la diferencia entre el valor de la coalición y lo que le da la asignación x, (formalmente, $e(S,x) = v(S) - x(S)$). Si este valor es positivo, es decir si $v(S) > x(S)$, los miembros de S no estarán contentos con lo que les asigna x, ya que serían capaces de conseguir más por ellos mismos. En lenguaje coloquial, podríamos decir que el exceso mide la "infelicidad" de la coalición frente a un teórico reparto y que cada coalición estará mejor cuanto menor sea dicho exceso.

Así pues, basándose en esta idea, se tiene que el nucleolo es el reparto que hace que las quejas de las coaliciones sean lo más pequeñas posibles, en el sentido lexicográfico que definiremos a continuación.

Definición 2.14. *Sean $x, y \in \mathbb{R}^n$ para algún $n \in \mathbb{N}$.*

- *Decimos que $x <_L y$ si existe $k \in \mathbb{N}$, $1 \leq k \leq n$, tal que*

$$x_i = y_i, 1 \leq i < k \quad y$$

$$x_k < y_k.$$

- *Decimos que $x \leq_L y$ si $x = y$ ó $x <_L y$.*

En general, no hay forma explícita para el cálculo del nucleolo sino que es necesario recurrir a resolver un número finito de problemas de programación lineal, aunque existen clases específicas de juegos en los que se dispone de fórmulas.

Dado un reparto $x \in \mathbb{R}^n$, sea $\theta(x)$ la 2^n-tupla cuyas componentes son los excesos del valor de todas las coaliciones con respecto a x, ordenados de forma decreciente, es decir,

$$\theta_i(x) \geq \theta_j(x) \text{ si } 1 \leq i \leq j \leq 2^n.$$

De esta forma, definimos el nucleolo de un juego como las asignaciones que minimizan lexicográficamente los vectores de excesos. La idea sería intentar contentar a las coaliciones más inestables.

Definición 2.15. *Sea $(N, v) \in G^N$. Definimos el* ***nucleolo*** *del juego (N, v) como*

$$Nu(N, v) = \{x \in I(N, v) : \theta(x) \leq_L \theta(y), \quad \textit{para todo} \quad y \in I(N, v)\}.$$

Los elementos del núcleo verifican que $e(S, x) \leq 0$ para toda $S \subseteq N$. Entonces, es claro que los vectores de excesos de los elementos del núcleo son lexicográficamente menores que los de cualquier punto fuera del núcleo. A partir de esto, se obtienen los siguientes resultados:

1. Si el núcleo de un juego (N, v) no es vacío, entonces el nucleolo está contenido en el núcleo.
2. Si el conjunto de imputaciones es no vacío, entonces el nucleolo existe y también consiste en una única imputación.

2.2. El proceso de producción lineal y el juego asociado

Supongamos que varios agentes (generalmente empresas), con capacidad para producir por sí mismos con la misma tecnología, deciden asociarse y aportar sus recursos a un proceso productivo común.

Este tipo de problemas cuenta con la intervención de varios decisores, cada uno con intereses diferentes. Como suele ser común en las situaciones cooperativas, es difícil que todos estén de acuerdo con un determinado reparto de los beneficios que resulten del proceso de producción. De esta forma, podemos formular un juego cooperativo de n jugadores (agentes que intervienen en el proceso de producción) y una función característica, que representa el beneficio que obtendría cada coalición si afrontara por sí misma el proceso productivo usando exclusivamente los recursos de sus miembros.

Antes de establecer una formulación formal del problema, se introducirá alguna notación con respecto a vectores y matrices, que permitirá seguir los siguientes resultados de una forma más cómoda y clara.

Sea A una matriz cualquiera, $I \subseteq R$ un subconjunto apropiado de recursos y $J \subseteq P$ un subconjunto apropiado de productos. Se define:

- A_{ij} := elemento de A correspondiente a la fila i y a la columna j.
- $A_{i\bullet}$:= la fila i de la matriz A.
- $A_{\bullet j}$:= la columna j de A.
- $A_{I\bullet}$:= las filas de la matriz A que pertenecen al subconjunto $I \subset R$.
- $A_{\bullet j}$:= las columnas de la matriz A que pertenecen al subconjunto $J \subset P$.
- $A_{-i\bullet}$:= la matriz A excepto la fila i-ésima.
- $A_{\bullet -j}$:= la matriz A excepto la columna j-ésima.
- $A_{-I\bullet}$:= la matriz A excepto el conjunto de filas que pertenecen a I.
- $A_{\bullet -J}$:= la matriz A excepto el conjunto de columnas que pertenecen J.

Para una coalición $S \subseteq N$, se define su vector característico $e_S \in \mathbb{R}^n$ por

$$\begin{aligned} (e_S)_k &= 1, \quad \text{si } k \in S. \\ (e_S)_k &= 0, \quad \text{en otro caso.} \end{aligned}$$

Además, la matriz identidad $N \times N$ se denotará por I_N.

2.2.1. Definiciones

Estamos ya en condiciones de definir formalmente un proceso de producción lineal, así como el juego asociado para así, posteriormente, poder estudiar sus características y propiedades.

Definición 2.16. *Un proceso de producción lineal es una tupla* (N, R, P, A, B, c)[1] *donde:*

- $N = \{1, \ldots, n\}$ *es el conjunto de agentes o jugadores.*

[1] A partir de ahora, pondremos en su lugar (A, B, c) ya que N, R y P están definidos implícitamente.

- $R = \{1, \ldots, r\}$ *es el conjunto de recursos.*
- $P = \{1, \ldots, p\}$ *es el conjunto de productos.*
- $A \in \mathbb{R}_+^{r \times p}$ *es la matriz de producción.*
- $B \in \mathbb{R}_+^{r \times n}$ *es la matriz de recursos. De esta forma, se tiene que* Be_S *sería el vector de recursos de dicha coalición cuando los agentes de* S *deciden colaborar y aportar todos sus recursos para el bien común.*
- $c \in \mathbb{R}_+^p$ *es el vector de beneficios.*

Además, se verifican las siguientes condiciones:

- $R, P, N \neq \emptyset$.
- $Be_N > 0$.
- Existe al menos un producto $j \in P$ con $c_j \geq 0$.
- Si $c_j > 0$, entonces existe al menos un recurso $i \in R$ con $A_{ij} > 0$.

El conjunto de procesos de producción lineal se denotará por $\mathscr{L}$.

En el caso de los procesos de producción lineal, la interpretación es análoga al caso del problema de producción lineal (1.3), teniendo en cuenta ahora la intervención de varios decisores. De esta forma, se tiene que A es la *matriz de producción*, donde A_{ij} nos da la cantidad del recurso i necesaria para producir una unidad del producto j; B contiene los vectores de recursos de todos los decisores, de forma que B_{ik} es la cantidad de recursos i que posee el agente k y c es el *vector de precios*, donde c_j es el precio de mercado para el producto j.

La condición $Be_N > 0$ resume el hecho de que cada recurso es poseído en una cantidad positiva por al menos un decisor. La última condición establece que, si el precio de mercado es estrictamente positivo, entonces se necesitan algunos recursos para producirlo.

Formalmente, se tiene que el vector de recursos de cada coalición $S \subseteq N$ será $b(S) = Be_S$ y el problema de producción que habrá que resolver ahora será:

$$\begin{aligned} \textbf{Maximizar} \quad & z = \sum_{j=1}^{p} c_j x_j \\ \textbf{sujeto a} \quad & Ax \leq Be_S \\ & x \geq 0 \end{aligned} \tag{2.1}$$

Así pues, si se quiere resolver el problema de producción en el que los agentes de la coalición S cooperan se tendrá que resolver el problema (2.1).

Teniendo esto en cuenta, podemos plantear el juego asociado a nuestro problema de producción lineal que se describe formalmente en la siguiente definición.

Definición 2.17. *Dado un proceso de producción (A, B, c), se define el **juego de producción lineal** asociado a este proceso como el juego cooperativo $(N, v^{(A,B,c)})$, siendo $N = \{1, \ldots, n\}$ el conjunto de jugadores y $v^{(A,B,c)}$ la función característica que viene dada de la forma:*

$$v^{(A,B,c)}(S) = \begin{cases} 0 & S = \emptyset \\ \sum_{j=1}^{p} c_j \tilde{x}_j(S) & \text{en otro caso,} \end{cases} \tag{2.2}$$

donde $\tilde{x}(S)$ es la solución del problema de programación lineal (2.1).

A continuación, se muestra un problema de producción lineal para tres empresas únicamente con dos recursos para, de esta forma, poder ver tanto la solución gráfica como analítica del problema. Con este ejemplo, empezaremos a introducir las funciones disponibles en la librería `coopProductGame` para el caso de los juegos asociados a los procesos de producción lineal, presentando todas ellas a partir de ejemplos sobre los que se mostrará tanto el código empleado como la salida del mismo.

Ejemplo 2.1. *Consideremos el proceso de producción lineal (A, B, c) donde el vector de beneficios es $c = (68, 52)^t$, la matriz de producción es*

$$A = \begin{pmatrix} 4 & 5 \\ 6 & 2 \end{pmatrix}$$

y la matriz que incluye los vectores de recursos (como columnas) es:

$$B = \begin{pmatrix} 4 & 6 & 60 \\ 33 & 39 & 0 \end{pmatrix}.$$

En primer lugar, resolveremos el problema coalición a coalición. Al final del ejemplo se muestra el código necesario para obtener los resultados.

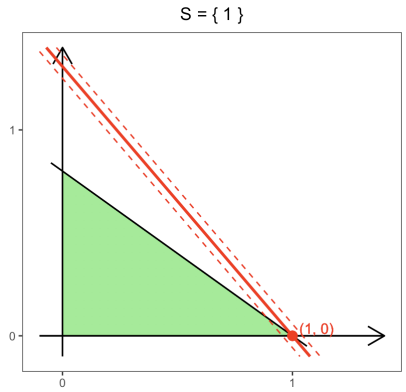
Coalición $S = \{1\}$:

Maximizar $z = 68x_1 + 52x_2$

sujeto a $4x_1 + 5x_2 \leq 4$

$6x_1 + 2x_2 \leq 33$

$x_1, x_2 \geq 0$



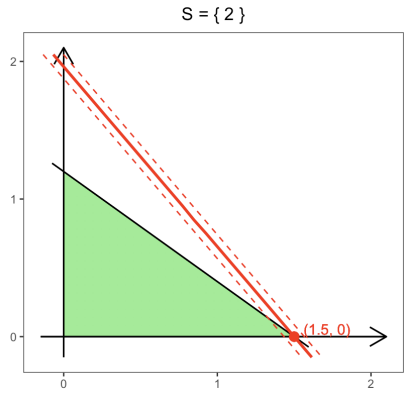
Coalición $S = \{2\}$:

Maximizar $z = 68x_1 + 52x_2$

sujeto a $4x_1 + 5x_2 \leq 6$

$6x_1 + 2x_2 \leq 39$

$x_1, x_2 \geq 0$



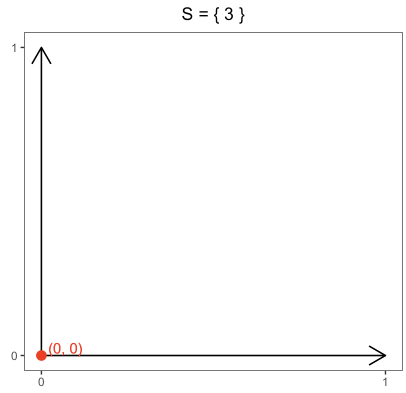
Coalición $S = \{3\}$:

Maximizar $z = 68x_1 + 52x_2$

sujeto a $4x_1 + 5x_2 \leq 60$

$6x_1 + 2x_2 \leq 0$

$x_1, x_2 \geq 0$



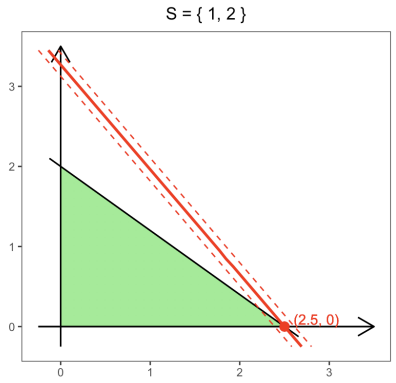
Coalición $S = \{1, 2\}$:

Maximizar $z = 68x_1 + 52x_2$

sujeto a $4x_1 + 5x_2 \leq 10$

$6x_1 + 2x_2 \leq 72$

$x_1, x_2 \geq 0$



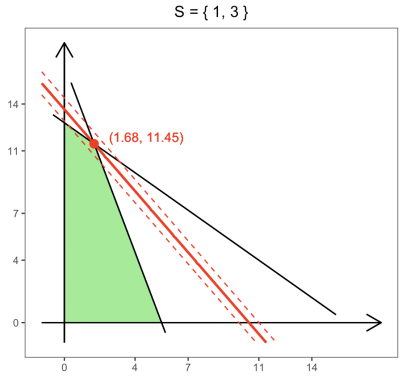
Coalición $S = \{1, 3\}$:

Maximizar $z = 68x_1 + 52x_2$

sujeto a $4x_1 + 5x_2 \leq 64$

$6x_1 + 2x_2 \leq 33$

$x_1, x_2 \geq 0$



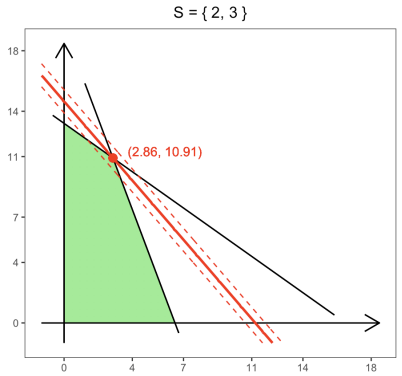
Coalición $S = \{2, 3\}$:

Maximizar $z = 68x_1 + 52x_2$

sujeto a $4x_1 + 5x_2 \leq 66$

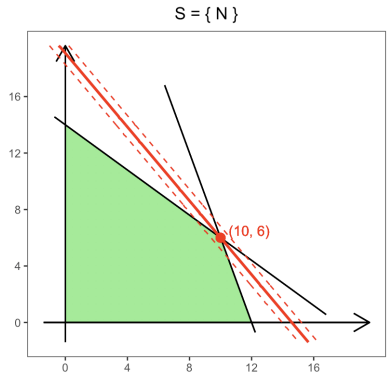
$6x_1 + 2x_2 \leq 39$

$x_1, x_2 \geq 0$



Coalición $S = \{1, 2, 3\} = N$:

$$\begin{aligned} \textbf{Maximizar} \quad & z = 68x_1 + 52x_2 \\ \textbf{sujeto a} \quad & 4x_1 + 5x_2 \leq 70 \\ & 6x_1 + 2x_2 \leq 72 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Aplicación de la librería en R sobre este ejemplo

En primer lugar, mostraremos el código necesario para resolver el problema solo de forma gráfica y coalición a coalición (este es el código que se ha empleado para obtener todas las figuras anteriores). La función `makeLP` nos permite construir el problema de producción lineal a partir del vector de beneficios, la matriz de producción y el vector de recursos correspondiente. Una vez construido el problema, se puede hacer uso de la función `plotlm` para hacer la representación gráfica de los problemas de producción lineal de dos dimensiones. A continuación, a modo de ejemplo, se muestra el código necesario para resolver el problema para la coalición $S = \{1, 2\}$:

```
# Vector de beneficios
c <- c(68, 52)
# Matriz de producción
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Vector de recursos para la coalición
b <- c(10, 72)
# Contrucción problema de producción lineal
prod <- makeLP(c, A, b)
# Solución gráfica del problema de producción lineal asociado
plotlm(prod, A, b, c, title = 12)
```

Por otro lado, si queremos obtener de forma conjunta tanto la solución gráfica como la analítica del problema, se puede recurrir a la función `productLinearProblem`, función que también recibe como parámetros el vector de beneficios, la matriz de producción y el vector de recursos correspondiente a la coalición de estudio pero, además, tiene dos argumentos adicionales que

podemos activar, `show.data` y `plot`; el primero de ellos permite mostrar la salida del código a través de la consola y el segundo muestra la solución gráfica, siempre que se trate de un problema de dos dimensiones. Se muestra a continuación el código necesario para resolver el problema de producción para la coalición N:

```
# Vector de beneficios
c <- c(68, 52)
# Matriz de producción
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Vector de recursos para la coalición
b <- c(70, 72)
# Solución gráfica y analítica del problema de producción lineal asociado
productLinearProblem(c, A, b, plot = TRUE, show.data = TRUE)
```

De esta forma, además de la solución gráfica (igual a la del punto anterior) también obtendríamos la solución analítica de la siguiente forma:

```
------------------------------------------------------------------------
 Objetive value:
------------------------------------------------------------------------
 [1] "Z = 992"
------------------------------------------------------------------------
 Optimal solution:
------------------------------------------------------------------------
 [1] 10  6
------------------------------------------------------------------------
```

Con esta aproximación, necesitaríamos plantear un problema diferente para cada una de las coaliciones. Sin embargo, existe otra posibilidad con la que podríamos obtener, de forma conjunta, la solución gráfica y analítica de todas las coaliciones, obteniendo así también el juego asociado al proceso de producción lineal (A, B, c). Para ello se dispone de la función `linearProductionGame`, la cual, a diferencia de las anteriores, recibe la matriz de recursos compuesta de los vectores de recursos de cada jugador:

```
# Vector de beneficios
c <- c(68, 52)
# Matriz de producción
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Matriz de recursos
B <- matrix(c(4, 6, 60, 33, 39, 0), ncol = 3, byrow = TRUE)
# Solución gráfica de todas las coaliciones y analítica
# del juego de producción lineal
linearProductionGame(c, A, B, show.data = TRUE, plot = TRUE)
```

Con esta solución obtendríamos la solución gráfica representada en la Figura 2.1, así como la solución analítica que se recoge a continuación:

```
------------------------------------------------------------------------
Optimal solution of the problem for each coalition:
------------------------------------------------------------------------
 S={1}      1.00  0.00
 S={2}      1.50  0.00
 S={3}      0.00  0.00
 S={1,2}    2.50  0.00
 S={1,3}    1.68 11.45
 S={2,3}    2.86 10.91
 S={1,2,3} 10.00  6.00
------------------------------------------------------------------------
Cooperative production game:
------------------------------------------------------------------------
                  S={0} S={1} S={2} S={3} S={1,2} S={1,3} S={2,3} S={1,2,3}
 Associated game      0    68   102     0     170     710     762       992
------------------------------------------------------------------------
```

En primer lugar, se tienen las coordenadas de los puntos que generan la solución óptima para cada una de las coaliciones o, lo que es lo mismo, la cantidad de cada producto que se debe fabricar en cada caso para que el beneficio sea óptimo. A continuación, se muestra cuál es el valor objetivo o beneficio óptimo asociado a cada coalición, consiguiendo de esta forma el juego cooperativo asociado. Por ejemplo, para la coalición N, se obtendría como solución $v(N) = 992$ y $\tilde{x}(N) = (10, 6)^t$.

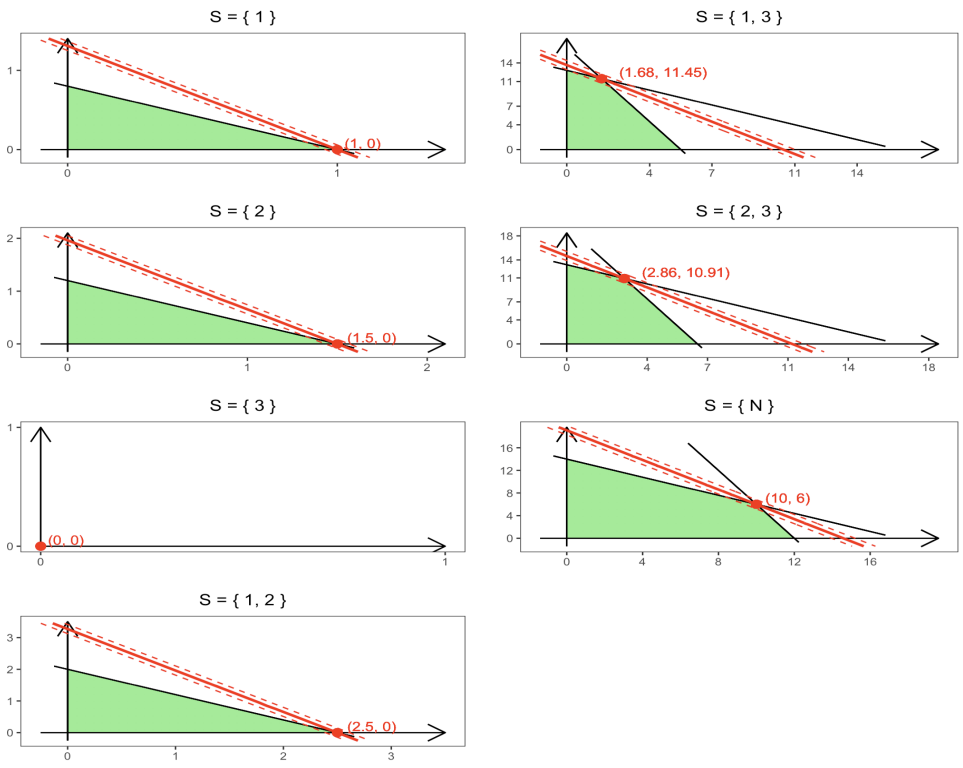


Figura 2.1: Solución gráfica para todas las coaliciones

2.2.2. Propiedades del juego de producción lineal

Sea el proceso de producción lineal (A, B, c). El juego de producción asociado a este proceso, $(N, v^{(A,B,c)})$, verifica una serie de propiedades que describimos a continuación.

- Es superaditivo:

$$v^{(A,B,c)}(S \cup T) \geq v^{(A,B,c)}(S) + v^{(A,B,c)}(T) \quad \forall S, T \subseteq N, S \cap T = \emptyset.$$

- Es equilibrado:

$$C(N, v^{(A,B,c)}) \neq \emptyset.$$

  Esta propiedad se demostrará en la Sección 2.3.3, donde veremos que el conjunto de Owen pertenece al núcleo.

- Es totalmente equilibrado. Owen [17] demostró que el juego de producción lineal es totalmente equilibrado, ya que cada subjuego es equilibrado.

- Es no negativo:

$$v^{(A,B,c)} \geq 0 \quad \forall S \subseteq N.$$

Sin embargo, los juegos de producción lineal no tienen por qué ser convexos ni aditivos. Una muestra de ello es el juego del Ejemplo 2.1, que se muestra a continuación:

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	N
$v(S)$	0	68	102	0	170	710	762	992

Tabla 2.1: Juego cooperativo asociado al proceso de producción lineal (A, B, c) del Ejemplo 2.1.

Este juego no es aditivo, ya que si consideramos las coaliciones $S = \{2\}$ y $T = \{3\}$, es claro que $S \cap T = \emptyset$ y

$$v^{(A,B,c)}(S \cup T) = v^{(A,B,c)}(\{23\}) = 762 \neq 102 = v^{(A,B,c)}(\{2\}) + v^{(A,B,c)}(\{3\}) = v^{(A,B,c)}(S) + v^{(A,B,c)}(T).$$

Además, tampoco es convexo, ya que si consideramos $T = \{3\}$, $S = \{23\}$ e $\{i\} = \{1\}$, se tiene que $T \subseteq S \subseteq N \setminus \{i\}$ y

$$\begin{aligned} v^{(A,B,c)}(S \cup \{i\}) - v^{(A,B,c)}(S) &= v^{(A,B,c)}(\{123\}) - v^{(A,B,c)}(\{23\}) = \\ &= 992 - 762 = 230 \not\geq 710 = \\ &= v^{(A,B,c)}(\{23\}) - v^{(A,B,c)}(\{3\}) = \\ &= v^{(A,B,c)}(T \cup \{i\}) - v^{(A,B,c)}(T). \end{aligned}$$

Sin embargo, Gellekom et al. [25] nos demuestran bajo qué condiciones sí lo son. Aunque no reflejaremos en este documento todas sus demostraciones, sí recogeremos un pequeño resumen de los pasos que siguen los autores.

Teorema 2.4. *Sea un proceso de producción lineal $(A, B, c) \in \mathscr{L}$. Entonces:*

$$C(A, BX, c) = C(A, B, c)X \text{ para todo } X \in \mathbb{R}_+^{n \times m} \text{ con } Xe_M = e_N.$$

$$\Updownarrow$$

$$(N, v^{(A,B,c)}) \text{ es aditivo.}$$

Demostración. "⇒". En este caso es suficiente probar que $v(S \cup \{k\}) = v(S) + v(\{k\})$ para todo $k \in N$ y todo $S \subseteq N \setminus \{k\}$. Gellekom et al. lo demuestran para un agente, ya que los demás casos serían análogos.

"⇐". Dado que, como ya se ha comentado previamente, el juego de producción lineal es superaditivo, los autores hacen uso de la Proposición 2.2 de forma que, para ver que el juego es aditivo demuestran que $v(N) = \sum_{k \in N} v(k)$. □

Teorema 2.5. *Sea un proceso de producción lineal* $(A, B, c) \in \mathscr{L}$*. Entonces:*

$$C(A, BX, c) = C(A, B, c)X \text{ para todo } X \in \mathbb{R}^{n \times m}(\{0,1\}) \text{ con } Xe_M = e_N.$$

$$\Updownarrow$$

$$(N, v^{(A,B,c)}) \text{ es convexo.}$$

Demostración. "⇒". Para probar este resultado, en primer lugar se demuestra que el juego $(N, v^{(A,B,c)})$ es exacto, probando que existe un elemento en el núcleo $y \in C(A, B, c)$ tal que $y(S) = v^{(A,B,c)}(S)$. Dado que las condiciones de exactitud y convexidad son equivalentes para $n \leq 3$, tal y como se recoge en la Proposición 2.3, se asume que $n \geq 4$ y se demuestra que

$$v^{(A,B,c)}(S) + v^{(A,B,c)}(T) \leq v^{(A,B,c)}(S \cap T) + v^{(A,B,c)}(S \cup T) \quad \forall S, T \subseteq N.$$

"⇐". Dada la matriz $X \in \mathbb{R}^{n \times m}(\{0,1\})$ con $Xe_M = e_N$, se pueden ver las columnas de X como los vectores característicos de los diferentes subconjuntos de N. Además, dado que $Xe_M = e_N$, estos subconjuntos forman una "partición" de N. De esta forma, el juego de producción lineal $v^{(A,BX,c)}$ es una restricción del juego $v^{(A,B,c)}$ sobre sus subconjuntos y uniones correspondientes. Con esto, claramente se tiene que para todas las matrices $X \in \mathbb{R}^{n \times m}(\{0,1\})$ con $Xe_M = e_N$ y para todos los procesos de producción lineal $(A, B, c) \in \mathscr{L}$:

$$C(A, B, c)X \subseteq C(A, BX, c).$$

Ahora, tan solo sería necesario probar que $(A, BX, c) \subseteq C(A, B, c)X$. Para demostrar esto, se basan en el Teorema 2.2, viendo que el conjunto de Weber es igual a $C(A, BX, c)$ y, a continuación, demuestran que todos los vectores de contribuciones marginales $\bar{m}^{\sigma}$, correspondientes al juego cooperativo asociado al proceso de producción lineal (A, BX, c), son elementos del núcleo $C(A, B, c)X$. □

Otro resultado interesante sobre los juegos de producción lineal, demostrado por Curiel, se puede ver en el siguiente teorema.

Teorema 2.6. *(Curiel, 1977. [4]) Todo juego no negativo totalmente equilibrado es un juego de producción lineal.*

A continuación se muestra un cuadro resumen que nos permite ver qué propiedades cumple el juego de producción lineal.

	Juego de producción lineal
Superaditivo	✓
Equilibrado	✓
Totalmente equilibrado	✓
No negativo	✓
Convexo	✗
Aditivo	✗

Tabla 2.2: Propiedades del juego de producción lineal.

2.3. Reglas de repartos para los procesos de producción lineal

2.3.1. Definiciones

Definición 2.18. *Una regla de repartos φ en $\mathscr{L}$ es una función que asigna a cada proceso de producción lineal $(A, B, c) \in \mathscr{L}$ un subconjunto de $\mathbb{R}^n$.*

Nótese que puede darse el hecho de que $\varphi(A, B, c) = \emptyset$ para alguna o para todas las tuplas $(A, B, c) \in \mathscr{L}$.

2.3.2. Conjuntos de repartos basados en soluciones de juegos cooperativos

Definición 2.19. *Definimos el núcleo de un proceso de producción lineal $(A, B, c) \in \mathscr{L}$ como el núcleo del juego asociado correspondiente, es decir,*

$$C(A, B, c) := C(N, v^{(A,B,c)}).$$

De la misma forma, definimos el valor de Shapley y el nucleolo del proceso de producción lineal como los conjuntos formados por el valor de Shapley y el nucleolo del juego asociado, es decir,

$$Sh(A, B, c) := \{Sh(N, v^{(A,B,c)})\}$$

$$Nu(A, B, c) := \{Nu(N, v^{(A,B,c)})\}.$$

2.3.3. Otros conjuntos de repartos: el conjunto de Owen

El problema (2.1) es factible, ya que $x = 0$ es un vector factible y las condiciones establecidas anteriormente sobre la tupla (A, B, c) aseguran que se trata de un problema acotado, lo cual se puede demostrar con la ayuda del problema de programación dual asociado:

$$\begin{array}{ll} \textbf{Minimizar} & w = Be_S^t y \\ \textbf{sujeto a} & A^t y \geq c \\ & y \geq 0 \end{array} \tag{2.3}$$

El siguiente vector es un punto factible del problema (2.3):

$$y_i := \begin{cases} 0 & \text{si } A_{ij} = 0 \ \ \forall j \in P \\ \text{máx}_{j \in P: A_{ij} > 0} \{\frac{c_j}{A_{ij}}, 0\} & \text{en otro caso} \end{cases}$$

Si nos fijamos en el problema (2.3), la región factible no depende de la coalición que se forma; sin embargo, la solución del problema y el valor de la función objetivo sí que dependen de cada coalición $S \subseteq N$. Denotaremos entonces por $O_{\text{dual}}(A, B, c)$ al conjunto de soluciones óptimas del problema (2.3).

Si los recursos que aporta cada jugador se valoran en función del vector de precios sombra asociado al problema cuando se forma la coalición N, el vector resultante es un reparto del núcleo del juego. El reparto o conjunto de repartos obtenidos de esta forma se conoce como **conjunto de Owen** y, claramente, ya vemos que se trata de un conjunto no vacío.

Definición 2.20. *Sea $(A, B, c) \in \mathscr{L}$. Definiremos el **conjunto de Owen** del proceso de producción lineal como:*

$$Owen(A, B, c) = \{B^t y \in \mathbb{R}^n : y \in O_{dual}(A, B, c)\}.$$

Puede haber más de una solución óptima del problema dual, por lo que el conjunto de Owen puede dar lugar a más de un reparto. De hecho, todas las valoraciones duales que dichas soluciones representan son válidas para obtener elementos del conjunto de Owen.

Previamente, hemos visto cuáles son las funciones específicas de la librería programada para calcular el juego asociado a un proceso de producción lineal, tanto de forma gráfica como analítica, y ahora veremos cuáles son las propias para calcular las reglas de reparto que se han definido previamente.

Para calcular las reglas de reparto se dispone de las funciones `shapleyValue` y `nucleolus` que, a partir de un juego en su forma estándar, nos proporcionan el valor de Shapley y el nucleolo respectivamente. Estas funciones hacen uso de otras ya implementadas en el paquete "GameTheory" [3]. Por otro lado, la función `owenSet` nos proporciona el conjunto de Owen a partir de un problema de producción lineal. Dado que éste puede tener múltiples repartos, en el caso de tener exactamente dos recursos o, lo que es lo mismo, que el problema dual asociado sea en dos dimensiones, la librería proporcionará todos los repartos de Owen, mostrando cuáles son los puntos extremos del segmento que los contiene. En otro caso, se indicará que existen múltiple repartos y se proporcionará uno de ellos.

Además, también se podría hacer uso de la función `coopProductGame`, función principal de la librería, que agrupa todas las comentadas hasta el momento para, a partir de una única función, poder obtener tanto el juego asociado a un proceso de producción lineal como todas las reglas de reparto descritas.

A continuación, se muestra un ejemplo con dos productos en el que se puede ver que el conjunto de Owen está formado por más de un punto.

Ejemplo 2.2. *Consideremos el juego de producción lineal* $(A, B, c) \in \mathscr{L}$ *donde:*

$$A = \begin{pmatrix} 3 & 2 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 30 & 20 & 50 \\ 6 & 40 & 54 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 10 \\ 12 \end{pmatrix}.$$

Si solo se quiere calcular el conjunto de Owen, se puede hacer uso de la función `owenSet`*, habilitando el argumento* `show.data`*, para ver el resultado por consola:*

```
# Matriz de producción:
A <- matrix(c(3, 2, 1, 2), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(10, 12)
# Matriz de recursos:
B <- matrix(c(30, 20, 50, 6, 40, 54), ncol = 3, byrow = TRUE)
# Resolución del problema de producción asociado:
owenSet(c, A, B, show.data = TRUE)
```

```
------------------------------------------------------------------------
 The linear production problem has multiple Owen's allocations
 All the points between the following are allocations of the Owen Set:
------------------------------------------------------------------------
 [1] "(84, 200, 316)"
 [1] "(180, 120, 300)"
------------------------------------------------------------------------
```

La otra opción sería hacer uso de la función principal del paquete, activando las opciones `plot` *y* `show.data`*; de esta forma, obtenemos tanto la solución gráfica para cada coalición (que se puede ver en la Figura 2.2) como la analítica, cuyo resultado se muestra a continuación y que, además del juego asociado al problema, también nos proporciona el conjunto de Owen asociado al problema de producción, junto con el valor de Shapley y el nucleolo del juego asociado para la coalición N.*

```
# Resolución del problema de producción asociado:
coopProductGame(c, A, B, plot = TRUE, show.data = TRUE)
```

```
--------------------------------------------------------------------------
Optimal solution of the problem for each coalition:
--------------------------------------------------------------------------
 S={1}      6  0
 S={2}      0 10
 S={3}      0 25
 S={1,2}    2 22
 S={1,3}   10 25
 S={2,3}    0 35
 S={1,2,3}  0 50
--------------------------------------------------------------------------
Cooperative production game:
--------------------------------------------------------------------------
                  S={0} S={1} S={2} S={3} S={1,2} S={1,3} S={2,3} S={1,2,3}
 Associated game     0    60   120   300     284     400     420       600
--------------------------------------------------------------------------
 The linear production problem has multiple Owen's allocations
 All the points between the following are allocations of the Owen Set:
--------------------------------------------------------------------------
 [1] "(84, 200, 316)"
 [1] "(180, 120, 300)"
--------------------------------------------------------------------------
Shapley:
--------------------------------------------------------------------------
 [1] "(124, 164, 312)"
--------------------------------------------------------------------------
Nucleolus:
--------------------------------------------------------------------------
 [1] "(164, 128, 308)"
--------------------------------------------------------------------------
```

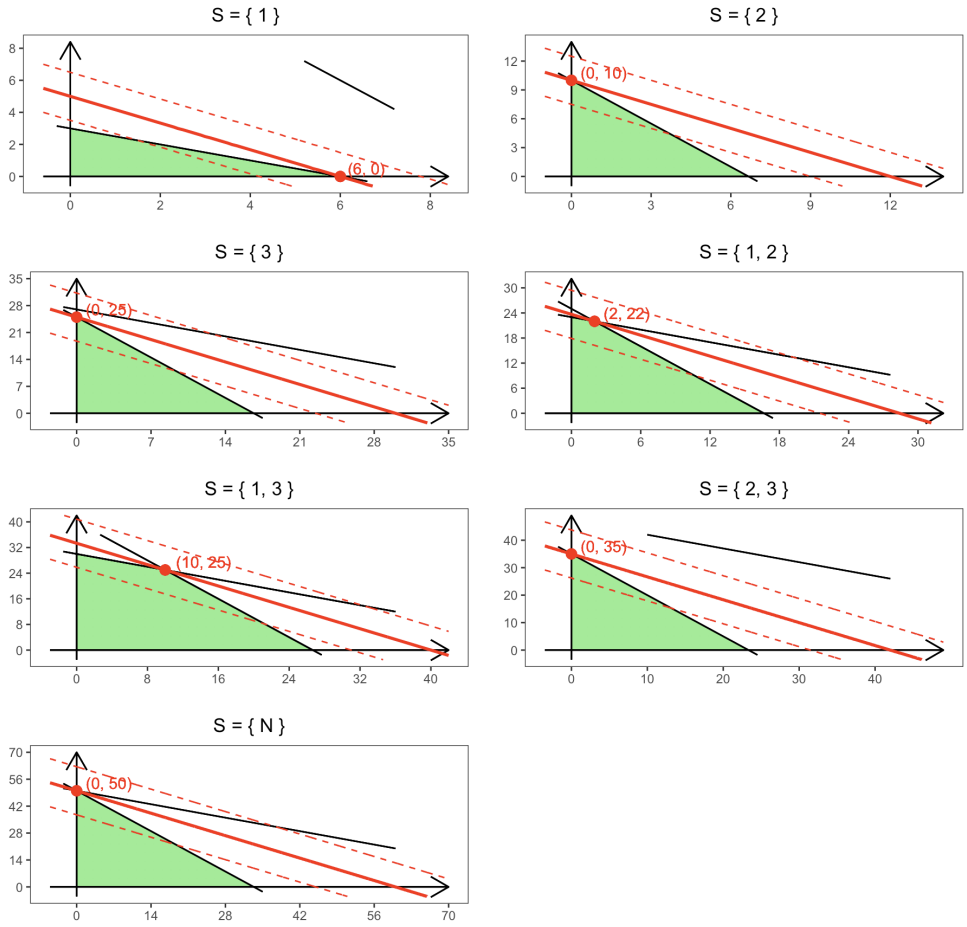


Figura 2.2: Solución gráfica para el problema.

Vemos que, efectivamente, exiten múltiples repartos de Owen y, dado que el problema dual es un problema en dos dimensiones, todos los repartos de Owen vendrán dados por el segmento que une estos puntos.

Además, cuando se trata de un problema con tres jugadores, la librería también dispone de la función `plotCoreSet`*, que permite representar gráficamente el núcleo junto con las reglas de reparto que venimos comentando. Esta función recibe como argumentos los parámetros asociados a un problema de producción lineal. A continuación, se muestra el código necesario para generar la gráfica con todos estos componentes, que aparece recogida en la Figura 2.3.*

```r
# Matriz de producción:
A <- matrix(c(3, 2, 1, 2), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(10, 12)
# Matriz de recursos:
B <- matrix(c(30, 20, 50, 6, 40, 54), ncol = 3, byrow = TRUE)
# Solución gráfica del núcleo y las reglas de reparto
plotCoreSet(c, A, B)
```

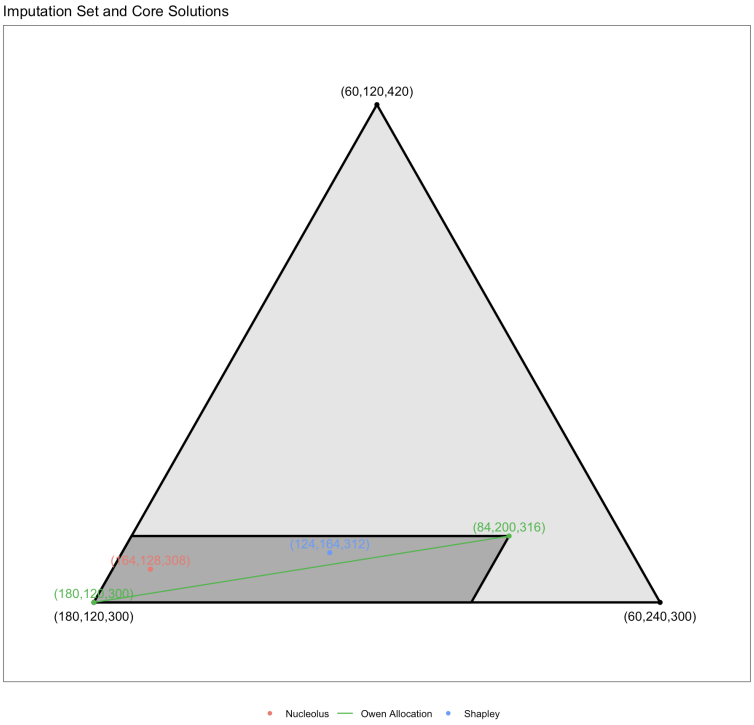


Figura 2.3: Solución gráfica para el problema.

Hemos visto un ejemplo con dos recursos en el que el conjunto de Owen está formado por múltiples repartos, que se pueden calcular con ayuda del paquete de `R` que hemos creado. A continuación, se muestra un ejemplo con más de dos recursos en el que el conjunto de Owen también está formado por más de un reparto.

Ejemplo 2.3. *Consideremos el proceso de producción* (A, B, c) *donde*

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 4 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 9 & 3 \\ 1 & 2 & 3 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

Al igual que en el caso anterior, haremos uso de la función `owenSet`*:*

```
# Matriz de producción:
A <- matrix(c(1, 1, 1, 4, 1, 2), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(3, 9)
# Matriz de recursos:
B <- matrix(c(1, 1, 1, 3, 9, 3, 1, 2, 3), ncol = 3, byrow = TRUE)
# Resolución del problema de producción asociado:
owenSet(c, A, B, show.data = TRUE)
```

```
------------------------------------------------------------------------
 The linear production problem has multiple Owen's allocations
 One of them is the following
------------------------------------------------------------------------
 [1] "(4.5, 9, 13.5)"
------------------------------------------------------------------------
```

En este caso, la librería nos dice que existen múltiples repartos de Owen y nos proporciona uno de ellos.

Por último, se muestra otro ejemplo con tres recursos en el que, a diferencia de los anteriores, el conjunto de Owen está formado por un único punto.

Ejemplo 2.4. *Consideremos el proceso de producción* (A, B, c) *con tres jugadores donde la matriz de recursos viene dada por:*

$$B = \begin{pmatrix} 139 & 181 & 110 \\ 140 & 87 & 183 \\ 130 & 225 & 215 \end{pmatrix}$$

Se producen tres bienes con unos beneficios dados por el vector $c^t = (2.5, 5, 4)$*. La matriz de producción del problema es:*

$$A = \begin{pmatrix} 2 & 9 & 3.5 \\ 6 & 4 & 9 \\ 8 & 9 & 7 \end{pmatrix}$$

Al igual que en los ejemplos anteriores, se hará uso de la función `owenSet`*. En este caso, se puede ver que el conjunto de Owen está formado por un único punto.*

```
# Matriz de producción:
A <- matrix(c(2, 9, 3.5,
              6, 4, 9,
              8, 9, 7), ncol = 3, byrow = TRUE)
# Vector de beneficios:
c <- c(2.5, 5, 4)
# Matriz de recursos:
B <- matrix(c(139, 181, 110,
              140, 87, 183,
              130, 225, 215), ncol = 3, byrow = TRUE)
# Resolución del problema de producción asociado:
owenSet(c, A, B, show.data = TRUE)
```

```
-------------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
-------------------------------------------------------------------------------
 [1] "(98.821, 102.366, 98.142)"
-------------------------------------------------------------------------------
```

Por otro lado, hacemos también la representación gráfica del conjunto de imputaciones (con ayuda de la función `plotCoreSet`*, al igual que en los casos anteriores) y las soluciones asociadas, obteniendo el gráfico que aparece en la Figura 2.4. Además, en este caso, el valor de Shapley no pertenece al núcleo del juego asociado.*

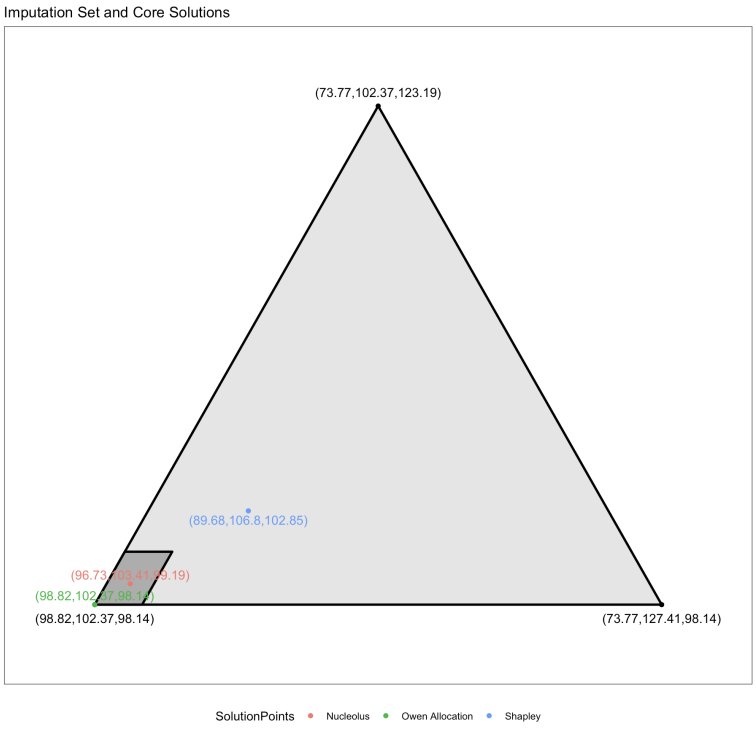


Figura 2.4: Solución gráfica para el problema.

A diferencia del valor de Shapley y del nucleolo, que dependen de la definición del juego considerado, en el caso del conjunto de Owen, este conjunto depende del proceso de producción lineal. Esto significa que el conjunto de Owen es una regla de solución sobre el conjunto de procesos de producción lineales, no sobre el conjunto de los juegos de producción lineal. Una demostración simple de esta casuística se puede ver a través del siguiente ejemplo:

Ejemplo 2.5. *Considérese el siguiente proceso de producción lineal* $(A, B, c) \in \mathscr{L}$*:*

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 4 \\ 8 & 3 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 5 \\ 6 \\ 8 \end{pmatrix}.$$

Resolveremos este problema con la ayuda de la función `coopProductGame` *que, tal como hemos visto, nos devuelve el juego asociado al proceso de producción lineal y el conjunto de Owen asociado al proceso, entre otras cosas. Para ello, empleamos el siguiente código.*

```
# Matriz de producción:
A <- matrix(c(1, 2, 1, 1, 1, 4), ncol = 3, byrow = TRUE)
# Vector de beneficios:
c <- c(5, 6, 8)
# Matriz de recursos:
B <- matrix(c(0, 4, 8, 3), ncol = 2, byrow = TRUE)
# Resolución del problema de producción asociado:
coopProductGame(c, A, B, show.data = TRUE)
```

Parte del resultado a este código es:

```
....
....
------------------------------------------------------------------------
Cooperative production game:
------------------------------------------------------------------------
                  S={0} S={1} S={2} S={1,2}
 Associated game     0     0    16      27
------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
------------------------------------------------------------------------
 [1] "(8, 19)"
------------------------------------------------------------------------
Shapley:
------------------------------------------------------------------------
 [1] "(5.5, 21.5)"
------------------------------------------------------------------------
Nucleolus:
------------------------------------------------------------------------
 [1] "(5.5, 21.5)"
------------------------------------------------------------------------
```

De esta forma podemos ver que el juego asociado a este problema viene dado por $v(\{1\}) = 0$, $v(\{2\}) = 16$ *y* $v(\{12\}) = 27$. *Además, el reparto de Owen viene dado por el punto* $(8, 19)$ *y el valor de Shapley y el nucleolo coinciden en el punto* $(5.5, 21.5)$.

Sea ahora el proceso de producción lineal $(A', B', c') \in \mathscr{L}$, *donde:*

$$A' = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad B' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad y \quad c' = \begin{pmatrix} 16 \\ 27 \end{pmatrix}.$$

De nuevo, haciendo uso de la librería `coopProductGame`*:*

```
# Matriz de producción:
A <- matrix(c(0, 1, 1, 1), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(16, 27)
# Matriz de recursos:
B <- matrix(c(1, 0, 0, 1), ncol = 2, byrow = TRUE)
# Resolución del problema de producción asociado:
coopProductGame(c, A, B, show.data = TRUE)
```

```
....
....
------------------------------------------------------------------------
Cooperative production game:
------------------------------------------------------------------------
                 S={0} S={1} S={2} S={1,2}
 Associated game     0     0    16      27
------------------------------------------------------------------------
 The linear production problem has multiple Owen's allocations
 All the points between the following are allocations of the Owen Set:
------------------------------------------------------------------------
 [1] "(11, 16)"
 [1] "(0, 27)"
------------------------------------------------------------------------
Shapley:
------------------------------------------------------------------------
 [1] "(5.5, 21.5)"
------------------------------------------------------------------------
Nucleolus:
------------------------------------------------------------------------
 [1] "(5.5, 21.5)"
------------------------------------------------------------------------
```

Vemos que el juego asociado coincide con el anterior y, por eso, tanto el valor de Shapley como el nucleolo coinciden (dado que dependen del juego). Sin embargo, el conjunto de Owen es diferente ya que, en este caso, está formado por todos los repartos comprendidos en el segmento que une los puntos $(11, 16)$ *y* $(0, 27)$ *o, lo que es lo mismo, el conjunto:*

$$Owen(A', B', c') = \{(y, 27 - y) \mid 0 \leq y \leq 11\}.$$

2.3.4. Propiedades. Caracterización del conjunto de Owen

Cuando tenemos una regla de repartos es lógico desear que se cumplan ciertas condiciones sobre la misma. A continuación se describen ciertas propiedades deseables para una regla de reparto φ en $\mathscr{L}$ que, posteriormente, analizaremos en detalle sobre las diferentes reglas que comentamos previamente.

Definición 2.21.

- ***Eficiencia unipersonal.*** *Sea $(A, B, c) \in \mathscr{L}$ tal que $B = e_R$. En ese caso $\varphi(A, e_R, c) = \{v^{(A,e_R,c)}(N)\}$.*

  En otras palabras, si solo hay un agente que posee una unidad de todos los recursos, entonces φ le asigna el beneficio máximo que puede obtenerse a partir de sus recursos.

- ***Reescalado.*** *$\varphi(DA, DB, c) = \varphi(A, B, c)$ para todas las matrices diagonales $D \in \mathbb{R}_+^{r\times r}$ con elementos positivos en la diagonal, para todo $(A, B, c) \in \mathscr{L}$.*

  El axioma de reescalado nos asegura que la regla de solución es independiente de las unidades en las que se miden los recursos.

- ***Repartos en el núcleo.*** *$\varphi(A, B, c) \subseteq C(A, B, c) = C(N, v^{(A,B,c)})$.*

- ***No manipulabilidad.*** *$\varphi(A, BX, c) = \varphi(A, B, c)X$ para todas las matrices $X \in \mathbb{R}_+^{n\times m}$ con $Xe_M = e_N$, para todo $(A, B, c) \in \mathscr{L}$, donde $\varphi(A, B, c)X := \{y^tX : y \in \varphi(A, B, c)\}$.*

  Esta propiedad dice que si los recursos se mezclan entre los agentes, la regla de solución cambia de la misma manera.

- ***Consistencia.*** *Para todo $(A, I_N, c) \in \mathscr{L}$ con $n \geq 2$ y para todo $y \in \varphi(A, I_N, c)$ se tiene que $(A_{-i\bullet}, I_{N\setminus i,\tilde{c}}) \in \mathscr{L}$ e $y_{-i} \in \varphi(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})$ para todo $i \in N$, donde $\tilde{c}_j := c_j - y_iA_{ij}$ para todo $j \in P$.*

  Esta propiedad tiene que ver con el caso especial en el que cada agente posea exactamente una unidad de exactamente un recurso y diferentes agentes posean recursos diferentes. En este caso, se tiene que $R = N$, por lo que se tiene la matriz $I_{N\times N}$ (equivalentemente, $I_{R\times R}$) por lo que denotaremos a esta matriz por I_N.

  Supongamos que los agentes están de acuerdo en que la ganancia obtenida se divida de acuerdo con un vector $y \in \varphi(A, I_N, c)$. El agente i coge y_i y se va. Su recurso puede ser utilizado por los demás agentes por un precio de y_i por unidad. Esto es lo mismo que decir que el beneficio de un producto disminuye con y_i para cada unidad necesaria de este recurso. Una regla de solución verifica el axioma de consistencia si la restricción de y a los agentes restantes es una solución al proceso de producción lineal reducido.

- ***Eliminación.*** *Para todo $(A, I_N, c) \in \mathscr{L}$ y para todo $J \subset P$ tal que si $v(A_{\bullet -J}, I_N, c_{-J})(N) = v(A, I_N, c)(N)$, entonces $\varphi(A, I_N, c) \subseteq \varphi(A_{\bullet -J}, I_N, c_{-J})$.*

  Este axioma refleja el hecho de que si no se necesita un producto para obtener el máximo beneficio, entonces podrá eliminarse.

Además, conviene remarcar los siguientes dos axiomas, aunque sean implicaciones de los cinco anteriores.

- *No vacío.* $\varphi(A, B, c) \neq \emptyset$ para todo $(A, B, c) \in \mathscr{L}$.
- *Eficiencia.* $(e_N)^t y = v(A, B, c)(N)$ para todo $y \in \varphi(A, B, c)$ y para todo $(A, B, c) \in \mathscr{L}$.

En esta sección se verá cuáles son las propiedades que caracterizan al conjunto de Owen. El propio Owen demostró que el conjunto de repartos que lleva su nombre está contenido en el núcleo, hecho que se recoge en el siguiente teorema.

Teorema 2.7. *(Owen, 1975) [17]. Sea el proceso de producción lineal $(A, B, c) \in \mathscr{L}$. Entonces se tiene que*

$$Owen(A, B, c) \subseteq C(A, B, c).$$

Demostración. Consideremos el dual del problema de programación lineal (2.1), que aparece reflejado en (2.3), y sea $y^* = (y_1^*, \ldots, y_r^*)$ la solución para (2.3) cuando $S = N$. Entonces, es claro que

$$v^{(A,B,c)}(N) = b_1(N)y_1^* + \ldots + b_r(N)y_r^*, \tag{2.4}$$

mientras que, para cualquier coalición S,

$$v^{(A,B,c)}(S) \leq b_1(S)y_1^* + \ldots + b_r(S)y_r^*, \tag{2.5}$$

ya que $v^{(A,B,c)}(S)$ es el mínimo sobre todos los posibles vectores y.

Consideremos ahora el vector de pagos $u = (u_1, \ldots, u_n)$, definido de la forma

$$u_i = b_1^i y_1^* + b_2^i y_2^* + \ldots + b_r^i y_r^*. \tag{2.6}$$

Para todo S, tenemos

$$\sum_{i \in S} u_i = \sum_{i \in S} \sum_{k=1}^{r} b_k^i y_k^* = \sum_{k=1}^{r} \sum_{i \in S} b_k^i y_k^* = b_1(S)y_1^* + \ldots + b_r(S)y_r^*.$$

Tenemos entonces por (2.4),

$$\sum_{i \in N} u_i = v^{(A,B,c)}(N)$$

y, por (2.5),

$$\sum_{i \in S} u_i \geq v^{(A,B,c)}(S)$$

para todo $S \subseteq N$. Por tanto, u es una imputación del núcleo.

□

De esta forma, también se puede deducir que el juego cooperativo asociado al problema de producción lineal es equilibrado, obteniendo así un método para obtener un punto en el núcleo. Para ello, tan solo se tiene que calcular el vector y^*, resolviendo un problema de programación lineal de un tamaño razonable, a partir del cual obtendremos una imputación por (2.6).

Heurísticamente, los componentes $y_1^*, \ldots, y_r^*$ se pueden ver como un vector de precios sombra para los r recursos. Cada jugador recibirá un pago por sus recursos acorde al vector de precios equilibrado y^*, de forma que los pagos siempre serían un vector que pertenece al núcleo.

Hemos visto que un vector de precios equilibrado y^* dará lugar a un punto en el núcleo y, como es lógico, puede existir más de un vector con estas características y, por tanto, más de un punto en el núcleo.

La pregunta natural que se planteó Owen ante esta casuística, es saber si todos los puntos del núcleo se podrían obtener de esta forma. Aunque existen ciertas clases particulares de problemas de producción lineal que verifican que el conjunto de Owen es igual al núcleo (ejemplos de ello se pueden encontrar en Shapley y Shubik, [23] ó Kalai y Zemel, [12]), es claro que esto no es siempre así, tal y como se puede ver en los Ejemplos (2.1) y (2.2).

Además de las propiedades comentadas previamente, van Gellekom et al. [25] caracterizan al conjunto de Owen con los axiomas que se muestran a continuación.

Proposición 2.4. *El conjunto de Owen satisface:*

(a) Eficiencia unipersonal

(b) Reescalado

(c) No manipulabilidad

(d) Consistencia

(e) Eliminación

Demostración.

(a) *Eficiencia unipersonal.* Sea $(A, B, c) \in \mathscr{L}$ tal que $B = e_R$. En ese caso $\varphi(A, e_R, c) = \{v^{(A,e_R,c)}(N)\}$. Entonces

$$Owen(A, B, c) = \{e_R^t y : y \in O_{\text{dual}}(A, B, c)\} = \{v^{(A,B,c)}(N)\}.$$

(b) *Reescalado.*

Sea $(A, B, c) \in \mathscr{L}$ y sea $D \in \mathbb{R}_+^{r\times r}$ una matriz diagonal con elementos positivos en su diagonal. En primer lugar, nótese que:

$$v^{(DA,DB,c)}(N) = \begin{array}{ll}\textbf{min} & (DBe_N)^t y \\ \textbf{s.a.} & (DA)^t y \geq c \\ & y \geq 0\end{array} \Leftrightarrow \begin{array}{ll}\textbf{min} & (Be_N)^t D^t y \\ \textbf{s.a.} & A^t D^t y \geq c \\ & D^t y \geq 0\end{array} \Leftrightarrow \begin{array}{ll}\textbf{min} & (Be_N)^t z \\ \textbf{s.a.} & A^t z \geq c \\ & z \geq 0\end{array} = v^{(A,B,c)}(N)$$

donde la segunda equivalencia se verifica porque D es una matriz diagonal invertible y no negativa. De esta forma:

$$\begin{aligned}
Owen(DA, DB, c) &= \{(DB)^t y \mid y \in O_{\text{dual}}(DA, DB, c)\} \\
&= \left\{ (DB)^t y \;\middle|\; \begin{array}{ll} v^{(DA,DB,c)}(N) = & \textbf{min} \quad (DBe_N)^t y \\ & \textbf{s.a.} \quad (DA)^t y \geq c \\ & \qquad y \geq 0 \end{array} \right\} \\
&= \left\{ B^t D^t y \;\middle|\; \begin{array}{ll} v^{(DA,DB,c)}(N) = & \textbf{min} \quad (Be_N)^t D^t y \\ & \textbf{s.a.} \quad A^t D^t y \geq c \\ & \qquad D^t y \geq 0 \end{array} \right\} \\
&= \left\{ B^t z \;\middle|\; \begin{array}{ll} v^{(A,B,c)}(N) = & \textbf{min} \quad (Be_N)^t z \\ & \textbf{s.a.} \quad A^t z \geq c \\ & \qquad z \geq 0 \end{array} \right\} \\
&= \{B^t z : z \in O_{\text{dual}}(A, B, c)\} \\
&= \text{Owen}(A, B, c).
\end{aligned}$$

(c) *No manipulabilidad.* Sea $(A, B, c) \in \mathscr{L}$ y $X \in \mathbb{R}_+^{n\times m}$, $Xe_M = e_N$. La suma de las filas de la matriz X es igual a uno, lo cual implica que $v^{(A,BX,c)}(N) = v^{(A,B,c)}(N)$. Entonces:

$$
\begin{aligned}
\mathrm{Owen}(A, BX, c) &= \{(BX)^t y \mid y \in O_{\mathrm{dual}}(A, BX, c)\} \\
&= \left\{ (BX)^t y \;\middle|\; \begin{array}{lll} v^{(A,BX,c)}(N) = & \textbf{min} & (BXe_M)^t y \\ & \textbf{s.a.} & A^t y \geq c \\ & & y \geq 0 \end{array} \right\} \\
&= \left\{ (BX)^t y \;\middle|\; \begin{array}{lll} v^{(A,B,c)}(N) = & \textbf{min} & (Be_N)^t y \\ & \textbf{s.a.} & A^t y \geq c \\ & & y \geq 0 \end{array} \right\} \\
&= \left\{ B^t y \;\middle|\; \begin{array}{lll} v^{(A,B,c)}(N) = & \textbf{min} & (Be_N)^t y \\ & \textbf{s.a.} & A^t y \geq c \\ & & y \geq 0 \end{array} \right\} X \\
&= \mathrm{Owen}(A, B, c)X.
\end{aligned}
$$

(d) *Consistencia.* Sea $(A, I_N, c) \in \mathscr{L}$, $n \geq 2$, $y \in \mathit{Owen}(A, I_N, c)$, $i \in N$. Tenemos que ver que

$$(A_{-i\bullet}, I_{N\setminus I}, \tilde{c}) \in \mathscr{L} \;\text{ y }\; y_{-i} \in \mathrm{Owen}(A_{-i\bullet}, I_{N\setminus I}, \tilde{c}),$$

donde $\tilde{c}_j := c_j - y_i A_{ij}$ para todo $j \in P$.

En primer lugar, veremos que $(A_{-i\bullet}, I_{N\setminus I}, \tilde{c}) \in \mathscr{L}$. Por lo tanto, tenemos que probar que existe al menos un producto j^* con $\tilde{c}_{j^*} \geq 0$ y, si $\tilde{c_j} > 0$ para algún producto j, entonces hay al menos un recurso l con $A_{lj} > 0$. En el caso de que $y \in \mathrm{Owen}(A, I_N, c)$ sea solución óptima del problema dual asociado, se tiene que

$$
\begin{array}{lll} v^{(A,I_N,c)}(N) = & \textbf{min} & e_N^t y \\ & \textbf{s.a.} & A^t y \geq c \\ & & y \geq 0 \end{array} \tag{2.7}
$$

Supongamos que $\tilde{c}_j < 0$ para todo producto j y veamos cómo esto no es posible. En este caso $c_j < y_i A_{ij}$ para todo $j \in P$. Si $(A, I_N, c) \in \mathscr{L}$, implica que hay al menos un producto j^* con $c_{j^*} \geq 0$. Entonces

$$0 \leq c_{j^*} < y_i A_{ij^*},$$

lo que implica que $y_i > 0$. Para un $\varepsilon > 0$ lo suficientemente pequeño tenemos que

$$(1 - \varepsilon) y_i A_{ij} > c_j \qquad \forall j \in P$$

o, lo que es lo mismo, $(1 - \varepsilon) y_i e_i$ pertenece a la región factible del problema (A, I_N, c). Entonces,

$$v^{(A,I_N,c)}(N) \leq (1 - \varepsilon) y_i < y_i \leq \sum_{i=1}^{r} y_i e_N = v^{(A,I_N,c)}(N),$$

lo cual es una contradicción con la suposición previa. Entonces existe al menos un producto j^* con $\tilde{c}_{j^*} \geq 0$. Supongamos que $\tilde{c}_j > 0$ para algún producto j. Entonces

$$\sum_{l=1}^{r} y_j A_{lj} \geq c_j > y_i A_{ij}.$$

De esta forma, existe al menos un recurso $l \in R \setminus \{i\}$ con $A_{lj} > 0$. Por tanto, se tiene que $(A_{-i\bullet})_{lj} = A_{lj} > 0$, lo cual demuestra que $(A_{-i\bullet}, I_{N\setminus\{i\}}, \tilde{c}) \in \mathscr{L}$.

Ahora, solo faltaría demostrar que $y_{-i} \in \text{Owen}(A_{-i\bullet}, I_N, \tilde{c})$ o, lo que es lo mismo,

$$\begin{array}{rlll} v^{(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})}(N) = & \textbf{min} & e_{N\setminus i}^t \; y & \\ & \textbf{s.a.} & A_{-i\bullet}^t y_{-i} \geq c & (2.8) \\ & & y_{-i} \in \mathbb{R}_+^{R\setminus i} & \end{array}$$

En primer lugar, de $y \in \mathbb{R}_+^r$ se sigue claramente que $y_{-i} \in \mathbb{R}_+^{r\setminus i}$.

Por otro lado, para todo $j \in P$, se tiene que $\sum_{l=1}^{r} y_l A_{lj} \geq c_j$, lo que implica que

$$y_{-i} A_{-i\bullet} e_j = \sum_{l \neq i} y_l A_{lj} \geq c_j - y_i A_{ij} = \tilde{c}_j,$$

es decir, $y_{-i} A_{-i\bullet} \geq \tilde{c}$.

Por último, dado que y_{-i} pertenece a la zona factible del problema $(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})$ se sigue que

$$v^{(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})}(N) \leq \textbf{min } e_{N\setminus i}^t \; y_{-i}.$$

Supongamos que $v^{(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})}(N) < \textbf{min } e_{N\setminus i}^t \; y_{-i}$ y z se encuentra en la región factible del problema $(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})$ tal que $v^{(A_{i\bullet}, I_{N\setminus i}, \tilde{c})}(N) = \textbf{min } e_{N\setminus i}^t \; z$.

Entonces $z \geq 0$ y

$$\sum_{l \neq i} z_l A_{lj} \geq c_j - y_i A_{ij},$$

es decir, (z, y_i) está en la región factible del problema (A, I_N, c) y $\textbf{min } e_N^t(z, y_i) < \textbf{min } e_N^t \; y$. Esto contradice el hecho de que y sea solución óptima del problema (A, I_N, c). Entonces tenemos que

$$v^{(A_{-i\bullet}, I_{N\setminus i}, \tilde{c})}(N) = \textbf{min } e_{N\setminus i}^t \; y_{-i}.$$

(e) *Eliminación.*

Sea $(A, I_N, c) \in \mathscr{L}$ y supongamos que existe un subconjunto apropiado de productos $J \subset P$ tal que $v^{(A_{\bullet -J}, I_N, c_{-J})}(N) = v^{(A, I_N, c)}(N)$. En este caso se tiene que ver que

$$\text{Owen}(A, I_N, c) \subseteq \text{Owen}(A_{\bullet -J}, I_N, c_{-J}).$$

Considérese $y \in \text{Owen}(A, I_N, c)$. Entonces:

$$\begin{array}{rll} v^{(A,I_N,c)}(N) = & \textbf{min} & e_N^t\ y \\ & \textbf{s.a.} & A^t y \geq c \\ & & y \geq 0 \end{array} \tag{2.9}$$

lo cual implica que y también satisface

$$\begin{array}{rll} v^{(A_{\bullet -J}, I_N, c_{-J})}(N) = & \textbf{min} & e_N^t\ y \\ & \textbf{s.a.} & A_{\bullet -J}^t\ y \geq c_{-J} \\ & & y \geq 0 \end{array} \tag{2.10}$$

es decir, $y \in \text{Owen}(A_{\bullet -J}, I_N, c_{-J})$.

□

Para las siguientes demostraciones de esta sección, incluidas en van Gellekom et al. [25], se usará con bastante asiduidad la matriz diagonal $r \times r$ que denotaremos por $\hat{D}$ y que se define de la siguiente forma:

$$\begin{array}{ll} \hat{D}_{ii} = (e_i B e_N)^{-1} & \forall i \in R \\ \hat{D}_{ij} = 0 & \text{en otro caso} \end{array}$$

Nótese que $\hat{D}B$ es una matriz cuya suma de filas es igual a 1 o, lo que es lo mismo, $\hat{D}Be_N = e_R$.

Lema 2.1. *Si φ satisface eficiencia unipersonal, reescalado y no manipulabilidad entonces φ es eficiente y no vacía.*

Demostración. Supongamos que φ verifica eficiencia unipersonal, reescalado y no manipulabilidad y tómese $(A, B, c) \in \mathscr{L}$. Por la propiedad de reescalado se tiene que

$$\varphi(\hat{D}A, \hat{D}B, c) = \varphi(A, B, c).$$

Aplicando ahora la no manipulabilidad con $X := e_N$

$$\varphi(\hat{D}A, \hat{D}Be_N, c) = \varphi(\hat{D}A, \hat{D}B, c)e_N.$$

Nótese que $(\hat{D}A, \hat{D}Be_N, c) \in \mathscr{L}$ es un proceso de producción lineal con un agente, el cual tiene en propiedad exactamente una unidad de cada recurso, $\hat{D}Be_N = e_R$.

Teniendo en cuenta ahora la propiedad de eficiencia unipersonal se sigue que

$$\varphi(A,B,c)e_N = \varphi(\hat{D}A,\hat{D}B,c)e_N = \varphi(\hat{D}A,\hat{D}Be_N,c) = \varphi(\hat{D}A,e_R,c) = \{v(\hat{D}A,e_R,c)(N)\},$$

de donde se obtiene que $\varphi(A,B,c) \neq \emptyset$.

Para demostrar que φ es eficiente será suficiente ver que $v^{(\hat{D}A,e_R,c)}(N) = v^{(A,B,c)}(N)$. Esto se sigue de

$$v^{(\hat{D}A,e_R,c)}(N) = v^{(A,(\hat{D})^{-1}e_R,c)}(N) = v^{(A,Be_N,c)}(N) = v^{(A,B,c)}(N),$$

donde la primera igualdad se cumple porque la región factible de un problema de producción lineal no cambia por el reescalado, por lo que el valor óptimo de su dual que, por la teoría de dualidad, es igual al óptimo del problema original, tampoco cambia.

□

Lema 2.2. *Si φ satisface eficiencia unipersonal, reescalado, no manipulabilidad y consistencia entonces $\varphi(A,I_N,c) \subseteq Owen(A,I_N,c)$ para todo $(A,I_N,c) \in \mathscr{L}$.*

Demostración. La demostración se realiza por inducción considerando como n el número de agentes. El caso $n = 1$ se puede ver claramente a partir del Lema 2.1:

$$\varphi(A,I_{\{1\}},c) = \{v^{(A,I_{\{1\}},c)}(N)\} = \text{Owen}(A,I_{\{1\}},c).$$

Supongamos que el lema ha sido demostrado si el número de agentes es menor que $n \geq 2$. Consideremos $(A,I_N,c) \in \mathscr{L}$ e $y \in \varphi(A,I_N,c) \neq \emptyset$ por el Lema 2.1. Si aplicamos nuevamente dicho lema se tiene que

$$\textbf{min } e_N^t\ y = v^{(A,I_N,c)}(N) \geq 0.$$

Así pues, existe un agente i con $y_i \geq 0$. De esta forma, a partir de la propiedad de consistencia respecto a este agente y de la hipótesis de inducción se sigue que

$$y_{-i} \in \varphi(A_{-i\bullet},I_{N\setminus i},\tilde{c}) \subseteq \text{Owen}(A_{-i\bullet},I_{N\setminus i},\tilde{c}),$$

donde $\tilde{c}_j = c_j - y_iA_{ij}$ para todo j. En particular $y_{-i} \geq 0$, por lo que $y \geq 0$. Además, $y_{-i}A_{-ij} \geq \tilde{c}_j = c_j - y_iA_{ij}$, es decir, $yA_{\bullet j} \geq c_j$ para todo j. Resumiendo,

$$\begin{array}{rlll} v^{(A,I_N,c)}(N) = & \textbf{min} & e_N^t\ y & \\ & \textbf{s.a.} & A^t\ y \geq c & (2.11) \\ & & y \geq 0 & \end{array}$$

es decir, y es solución óptima del problema dual asociado a (A, I_N, c) y, además, coincide con el reparto de Owen, Owen(A, I_N, c), como queríamos demostrar.

□

Lema 2.3. *Si φ satisface eficiencia unipersonal, reescalado, no manipulabilidad, consistencia y eliminación entonces Owen$(A, I_N, c) \subseteq \varphi(A, I_N, c)$ para todo $(A, I_N, c) \in \mathscr{L}$.*

Demostración. Sea $(A, I_N, c) \in \mathscr{L}$ y tomemos $y \in \text{Owen}(A, I_N, c)$. Se define $\bar{A} := AI_N$, $\bar{c} := c^t y$. Entonces

$$\text{Owen}(\bar{A}, I_N, \bar{c}) = \left\{ \bar{y} \in \mathbb{R}^n_+ \;\middle|\; \begin{array}{lll} v^{(\bar{A}, I_N, \bar{c})}(N) = & \textbf{min} & e_N^t \; \bar{y} \\ & \textbf{s.a.} & A^t \; \bar{y} \geq c \\ & & \bar{y} \geq y \end{array} \right\}$$

Por el Lema 2.2 tenemos que

$$\emptyset \neq \varphi(\bar{A}, I_N, \bar{c}) \subseteq \text{Owen}(\bar{A}, I_N, \bar{c}) = \{y\}.$$

Por tanto, $\varphi(\bar{A}, I_N, \bar{c}) = \{y\}$. Por otro lado, dado que $v^{(\bar{A}, I_N, \bar{c})}(N) = v^{(A, I_N, c)}(N)$, se puede aplicar la propiedad de eliminación y se tiene que $\varphi(\bar{A}, I_N, \bar{c}) \subseteq \varphi(A, I_N, c)$. Entonces $y \in \varphi(A, I_N, c)$.

□

Lema 2.4. *Si φ satisface eficiencia unipersonal, reescalado, no manipulabilidad, consistencia y eliminación entonces $\varphi(A, B, c) =$ Owen(A, B, c) para todo $(A, B, c) \in \mathscr{L}$.*

Demostración. Sea $(A, B, c) \in \mathscr{L}$. Se tiene entonces que:

$$\begin{array}{rcll} \varphi(A, B, c) & = & \varphi(DA, DB, c) & \text{(reescalado)} \\ & = & \varphi(DA, I_N, c)DB & \text{(no manipulabilidad)} \\ & = & \text{Owen}(DA, I_N, c)DB & \text{(Lemas 2.2 y 2.3)} \\ & = & \text{Owen}(DA, DB, c) & \text{(Proposición 2.4c)} \\ & = & \text{Owen}(A, B, c) & \text{(Proposición 2.4b)} \end{array}$$

□

Hemos visto que el conjunto de Owen satisface todas las propiedades descritas en la Definición 2.21 y además, por el Lema 2.4 sabemos que, si una regla de repartos satisface todas ellas,

entonces se trata del conjunto de Owen. En esta situación cabe preguntarse qué propiedades cumplen las demás reglas estudiadas. En esta sección, analizaremos todas estas propiedades para el valor de Shapley y el nucleolo, ejemplificando además todas ellas para así poder tenerlas más claras.

Sea φ una regla de repartos en $\mathscr{L}$.

- **Eficiencia unipersonal**. Sea $(A, B, c) \in \mathscr{L}$ tal que $B = e_R$. En este caso, $\varphi(A, e_R, c) = \{v^{(A,e_R,c)}(N)\}$.

  Podemos ver que el valor de Shapley y el nucleolo también verifican la propiedad de eficiencia unipersonal:

  $$Sh(A, e_R, c) = \{Sh(N, v^{(A,e_R,c)})\} = \{v^{(A,e_R,c)}(N)\},$$

  $$Nu(A, e_R, c) = \{Nu(N, v^{(A,e_R,c)})\} = \{v^{(A,e_R,c)}(N)\},$$

  donde, en ambos casos, la última igualdad se cumple porque tanto el valor de Shapley como el nucleolo son soluciones eficientes en los juegos cooperativos.

  A continuación, se muestra un ejemplo en el que se puede ver que todas las reglas de reparto estudiadas verifican dicha propiedad.

  Ejemplo 2.6. *Sea el proceso de producción lineal $(A, B, c) \in \mathscr{L}$, donde:*

  $$A = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 10 \\ 12 \end{pmatrix}.$$

  En este caso se tiene que:

  $$C(A, e_R, c) = Sh(A, e_R, c) = Nu(A, e_R, c) = Owen(A, e_R, c) = \{5\} = \{v^{(A,e_R,c)}(N)\}.$$

- **Reescalado**. $\varphi(DA, DB, c) = \varphi(A, B, c)$ para todas las matrices diagonales $D \in \mathbb{R}_+^{r\times r}$ con elementos positivos en la diagonal, para todo $(A, B, c) \in \mathscr{L}$.

  Al igual que el conjunto de Owen, el valor de Shapley y el nucleolo verifican la propiedad de reescalado. Dado el proceso de producción lineal $(DA, DB, c) \in \mathscr{L}$, se tiene que:

  $$v^{(DA,DB,c)}(S) = \begin{cases} 0 & S = \emptyset \\ \sum_{j=1}^{p} c_j \tilde{x}_j(S) & \text{en otro caso.} \end{cases} \tag{2.12}$$

  donde $\tilde{x}(S)$ es la solución del PPL:

$$\begin{array}{rrcl} \textbf{Max} & z & = & \sum_{j=1}^{p} c_j x_j \\ \textbf{s.a.} & DAx & \leq & DB \cdot e_S \\ & x & \geq & 0 \end{array}$$

Dado que $DAx \leq DB \cdot e_S \Leftrightarrow Ax \leq B \cdot e_S$, resolver el PPL anterior es equivalente a resolver el PPL:

$$\begin{array}{rrcl} \textbf{Max} & z & = & \sum_{j=1}^{p} c_j x_j \\ \textbf{s.a.} & Ax & \leq & B \cdot e_S \\ & x & \geq & 0 \end{array}$$

Se tiene entonces que $v^{(DA,DB,c)}(S) = v^{(A,B,c)}(S) = \sum_{j=1}^{p} c_j \tilde{x}_j(S)$, para todo $S \subseteq N$. Esto implica que $Sh(N, v^{(DA,DB,c)}) = Sh(N, v^{(A,B,c)})$ y $Nu(N, v^{(DA,DB,c)}) = Nu(N, v^{(A,B,c)})$, por lo que:

$$Sh(DA, DB, c) = Sh(A, B, c)$$

$$Nu(DA, DB, c) = Nu(A, B, c).$$

A continuación, se muestra un ejemplo donde se pone de manifiesto dicha propiedad sobre las diferentes reglas de reparto.

Ejemplo 2.7. *Sea el proceso de producción lineal* $(A, B, c) \in \mathscr{L}$ *donde:*

$$A = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 10 \\ 6 & 8 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 10 \\ 12 \end{pmatrix}.$$

Podemos calcular las reglas de reparto asociadas con ayuda de la función `coopProductGame` *de nuestra librería:*

```
# Matriz de producción:
A <- matrix(c(2, 3, 1, 2), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(10, 12)
# Matriz de recursos:
B <- matrix(c(4, 10, 6, 8), ncol = 2, byrow = TRUE)
# Resolución del problema de producción asociado:
coopProductGame(c, A, B, show.data = TRUE)
```

```
....
....
-----------------------------------------------------------------------
```

```
 The linear production problem has a unique Owen's allocation:
--------------------------------------------------------------------------
 [1] "(20, 50)"
--------------------------------------------------------------------------
--------------------------------------------------------------------------
Shapley:
--------------------------------------------------------------------------
 [1] "(20, 50)"
--------------------------------------------------------------------------
--------------------------------------------------------------------------
Nucleolus:
--------------------------------------------------------------------------
 [1] "(20, 50)"
--------------------------------------------------------------------------
```

Si consideramos la matriz diagonal

$$D = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix},$$

el proceso de producción lineal (DA, DB, c) *vendría dado por:*

$$A = \begin{pmatrix} 1 & 3/2 \\ 1/2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 5 \\ 3 & 4 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 10 \\ 12 \end{pmatrix}.$$

Calculamos ahora las reglas de reparto asociadas a este nuevo problema

```r
# Matriz de producción:
A <- matrix(c(2, 3, 1, 2), ncol = 2, byrow = TRUE)
# Matriz diagonal:
D <- diag(0.5, 2)
# Nueva matriz de producción:
DA <- D%*%A
# Vector de beneficios:
c <- c(10, 12)
# Matriz de recursos:
B <- matrix(c(4, 10, 6, 8), ncol = 2, byrow = TRUE)
# Nueva matriz de recursos
DB <- D%*%B
# Resolución del problema de producción asociado:
coopProductGame(c, DA, DB, show.data = TRUE)
```

....
....

```
------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
------------------------------------------------------------------------
 [1] "(20, 50)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Shapley:
------------------------------------------------------------------------
 [1] "(20, 50)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Nucleolus:
------------------------------------------------------------------------
 [1] "(20, 50)"
------------------------------------------------------------------------
```

Vemos cómo, efectivamente, las reglas de reparto coinciden con las del problema anterior.

- **Repartos en el núcleo**. $\varphi(A,B,c) \subseteq C(A,B,c) = C(N, v^{(A,B,c)})$ para todo $\varphi(A,B,c) \in \mathscr{L}$.

  Ya se ha demostrado que $Owen(A,B,c) \subseteq C(A,B,c)$ y que $Nu(A,B,c) \subseteq C(A,B,c)$, ya que $C(A,B,c) \neq \emptyset$, por lo que el conjunto de Owen y el nucleolo verifican la propiedad de repartos en núcleo. Sin embargo, $\exists (A,B,c) \in \mathscr{L}$ tal que $Sh(A,B,c) \nsubseteq C(A,B,c)$. Un ejemplo de ello ya lo hemos visto en el Ejemplo 2.4, por lo que podemos afirmar que el valor de Shapley no verifica la propiedad de repartos en el núcleo.

- **No manipulabilidad**. $\varphi(A,BX,c) = \varphi(A,B,c)X$ para todas las matrices $X \in \mathbb{R}_+^{n\times m}$ con $Xe_M = e_N$ y para todo $(A,B,c) \in \mathscr{L}$, donde $\varphi(A,B,c)X := \{y^tX : y \in \varphi(A,B,c)\}$.

  El valor de Shapley y el nucleolo no cumplen la propiedad de no manipulabilidad. A continuación, se muestra un contraejemplo en el que se puede comprobar esto, ejemplificando también dicha propiedad para el conjunto de Owen.

  Ejemplo 2.8. *Consideremos de nuevo el problema de producción lineal* $(A,B,c) \in \mathscr{L}$ *dado en el Ejemplo 2.1, en el que:*

$$A = \begin{pmatrix} 4 & 5 \\ 6 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 6 & 60 \\ 33 & 39 & 0 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 68 \\ 52 \end{pmatrix}.$$

  Calculamos las reglas de reparto asociadas:

```r
# Matriz de producción:
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Vector de beneficios:
c <- c(68, 52)
# Matriz de recursos:
B <- matrix(c(4, 6, 60, 33, 39, 0), ncol = 3, byrow = TRUE)
# Resolución del problema de producción asociado:
coopProductGame(c, A, B, show.data = TRUE)
```

```
....
....
------------------------------------------------------------------------
Cooperative production game:
------------------------------------------------------------------------
                 S={0} S={1} S={2} S={3} S={1,2} S={1,3} S={2,3} S={1,2,3}
 Associated game     0    68   102     0     170     710     762       992
------------------------------------------------------------------------
------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
------------------------------------------------------------------------
 [1] "(230, 282, 480)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Shapley:
------------------------------------------------------------------------
 [1] "(229, 272, 491)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Nucleolus:
------------------------------------------------------------------------
 [1] "(149, 192, 651)"
------------------------------------------------------------------------
```

Supongamos ahora, además, que la matriz X viene dada por:

$$X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

En este caso, los agentes 1 y 2 dan lugar a un nuevo agente, es decir, unen sus recursos en el proceso de producción. De esta forma se tiene que:

$$BX = \begin{pmatrix} 10 & 60 \\ 72 & 0 \end{pmatrix}.$$

Calculamos el juego asociado al nuevo proceso de producción lineal, junto con las reglas de repartos:

```
# Matriz de producción:
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Matriz de recursos:
B <- matrix(c(4, 6, 60, 33, 39, 0), ncol = 3, byrow = TRUE)
# Matriz auxiliar:
X <- matrix(c(1, 0, 1, 0, 0, 1), ncol = 2, byrow = TRUE)
# Nueva matriz de recursos
BX <- B%*%X
# Vector de beneficios:
c <- c(68, 52)
# Resolución del nuevo problema de producción:
coopProductGame(c, A, BX, show.data = TRUE)
```

```
....
....
------------------------------------------------------------------------
Cooperative production game:
------------------------------------------------------------------------
                 S={0} S={1} S={2} S={1,2}
 Associated game     0   170     0     992
------------------------------------------------------------------------
------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
------------------------------------------------------------------------
 [1] "(512, 480)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Shapley:
------------------------------------------------------------------------
 [1] "(581, 411)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Nucleolus:
------------------------------------------------------------------------
 [1] "(581, 411)"
------------------------------------------------------------------------
```

Tenemos entonces que:

$$Owen(A, B, c) \cdot X = (230, 282, 480) \cdot \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = (512, 480) = Owen(A, BX, c)$$

$$Nu(A,B,c)\cdot X=(149,192,651)\cdot\begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}=(341,651)\neq(581,411)=Nu(A,BX,c)$$

$$Sh(A,B,c)\cdot X=(229,272,491)\cdot\begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}=(501,491)\neq(581,411)=Sh(A,BX,c)$$

Vemos que el conjunto de Owen asociado a este problema de producción lineal verifica la propiedad de no manipulabilidad. Sin embargo, ni el nucleolo ni el conjunto de Shapley la cumplen.

- **Consistencia**. Para todo $(A,I_N,c)\in\mathscr{L}$ con $n\geq 2$ y para todo $y\in\varphi(A,I_N,c)$: $(A_{-i\bullet},I_{N\setminus i,\tilde{c}})\in\mathscr{L}$ e $y_{-i}\in\varphi(A_{-i\bullet},I_{N\setminus i},\tilde{c})$ para todo $i\in N$, donde $\tilde{c}_j:=c_j-y_iA_{ij}$ para todo $j\in P$.

  Ejemplo 2.9. *Sea el proceso de producción lineal $(A,B,c)\in\mathscr{L}$, donde*

$$A=\begin{pmatrix}2&2&1\\1&2&0\\1&3&0\end{pmatrix},\quad B=I_N=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\quad y\quad c=\begin{pmatrix}8\\2\\3\end{pmatrix}.$$

En primer lugar, con la función `coopProductGame` *calculamos el juego asociado al proceso anterior, además de las reglas de reparto habituales:*

```
# Matriz de producción:
A <- matrix(c(2, 2, 1, 1, 2, 0, 1, 3, 0), ncol = 3, byrow = TRUE)
# Matriz de recursos:
B <- diag(3)
# Vector de beneficios:
c <- c(8, 2, 3)
# Resolución del nuevo problema de producción:
coopProductGame(c, A, B, show.data = TRUE)
```

```
....
....
------------------------------------------------------------------------
Cooperative production game:
------------------------------------------------------------------------
                S={0} S={1} S={2} S={3} S={1,2} S={1,3} S={2,3} S={1,2,3}
 Associated game     0     3     0     0       3       3       0         4
------------------------------------------------------------------------
------------------------------------------------------------------------
```

```
 The linear production problem has a unique Owen's allocation:
--------------------------------------------------------------------------------
 [1] "(4, 0, 0)"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Shapley:
--------------------------------------------------------------------------------
 [1] "(3.333, 0.333, 0.333)"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Nucleolus:
--------------------------------------------------------------------------------
 [1] "(3.333, 0.333, 0.333)"
--------------------------------------------------------------------------------
```

Vemos que $Sh(A,B,c) = Nu(A,B,c) = (10/3, 1/3, 1/3)$. *Tomemos entonces* $i = 2$ *y consideremos el problema* $(A_{-2\bullet}, I_{N\setminus 2}, \tilde{c})$, *asociado al valor de Shapley y nucleolo, donde:*

$$A_{-2\bullet} = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 0 \end{pmatrix}, \quad B = I_{N\setminus 2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad y \quad \tilde{c} = \begin{pmatrix} 8 - 1/3 \cdot 1 \\ 2 - 1/3 \cdot 2 \\ 3 - 1/3 \cdot 0 \end{pmatrix} = \begin{pmatrix} 23/3 \\ 4/3 \\ 3 \end{pmatrix}.$$

Resolvemos ahora el problema con ayuda de nuestra librería a partir del siguiente código:

```r
# Matriz de producción:
A <- matrix(c(2, 2, 1, 1, 3, 0), ncol = 3, byrow = TRUE)
# Matriz de recursos:
B <- diag(2)
# Vector de beneficios:
c <- c(23/3, 4/3, 3)
# Resolución del nuevo problema de producción:
coopProductGame(c, A, B, show.data = TRUE)
```

Obteniendo:

```
....
....
--------------------------------------------------------------------------------
Shapley:
--------------------------------------------------------------------------------
 [1] "(3.417, 0.417)"
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```

```
Nucleolus:
------------------------------------------------------------------------
 [1] "(3.417, 0.417)"
------------------------------------------------------------------------
```

Se tiene entonces que:

$$Sh(A_{-2\bullet}, I_{N\setminus 2}, \tilde{c}) = Nu(A_{-2\bullet}, I_{N\setminus 2}, \tilde{c}) = (3.41\widehat{6}, 0.41\widehat{6}).$$

Vemos de esta forma que el valor de Shapley y el nucleolo no verifican la propiedad de consistencia.

Si consideramos ahora el mismo proceso de producción lineal $(A_{-2\bullet}, I_{N\setminus 2}, \tilde{c})$*, pero asociado al conjunto de Owen se tiene que:*

$$\tilde{c} = \begin{pmatrix} 8 - 0 \cdot 1 \\ 2 - 0 \cdot 2 \\ 3 - 0 \cdot 0 \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \\ 3 \end{pmatrix}.$$

Si calculamos el conjunto de Owen para este problema tendríamos que:

```
------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
------------------------------------------------------------------------
 [1] "(4, 0)"
------------------------------------------------------------------------
```

Así pues,

$$Owen(A_{-2\bullet}, I_{N\setminus 2}, \tilde{c}) = (4, 0),$$

observando así que los pagos correspondientes a los jugadores 1 y 3 coinciden con los repartos proporcionados a partir del problema original (A, B, c)*.*

- **Eliminación**. Para todo $(A, I_N, c) \in \mathscr{L}$ y para todo $J \subset P$ se tiene que si $v(A_{\bullet -J}, I_N, c_{-J})(N) = v(A, I_N, c)(N)$, entonces $\varphi(A, I_N, c) \subseteq \varphi(A_{\bullet -J}, I_N, c_{-J})$.

Ejemplo 2.10. *Consideremos el problema de producción lineal* $(A, B, c) \in \mathscr{L}$ *del ejemplo anterior, donde*

$$A = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 0 \\ 1 & 3 & 0 \end{pmatrix}, \quad B = I_N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 8 \\ 2 \\ 3 \end{pmatrix}.$$

Ya vimos antes que $v^{(A,B,c)}(N) = 4$*, valor que se obtiene a partir del punto* $(0.5, 0, 0)$*. En este caso, los productos 2 y 3 son irrelevantes a la hora de calcular este beneficio. Consideramos, pues,* $J = \{3\}$*. De esta forma, tendremos el nuevo problema de producción lineal* $(A_{\bullet\{3\}}, I_N, c_{-\{3\}})$*, donde*

$$A = \begin{pmatrix} 2 & 2 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}, \quad B = I_N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad y \quad c = \begin{pmatrix} 8 \\ 2 \end{pmatrix}.$$

Resolvemos el nuevo problema con ayuda de la función ***coopProductGame:***

```
# Matriz de producción:
A <- matrix(c(2, 2, 1, 1, 3, 0), ncol = 3, byrow = TRUE)
# Nueva matriz de producción:
A_3 <- A[,-3]
# Matriz de recursos:
B <- diag(3)
# Vector de beneficios:
c <- c(23/3, 4/3, 3)
# Nuevo vector de beneficios:
c_3 <- c[-3]
# Resolución del nuevo problema de producción:
coopProductGame(c_3, A_3, B, show.data = TRUE)
```

```
....
....
------------------------------------------------------------------------
 The linear production problem has a unique Owen's allocation:
------------------------------------------------------------------------
 [1] "(4, 0, 0)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Shapley:
------------------------------------------------------------------------
 [1] "(1.333, 1.333, 1.333)"
------------------------------------------------------------------------
------------------------------------------------------------------------
Nucleolus:
------------------------------------------------------------------------
 [1] "(1.333, 1.333, 1.333)"
------------------------------------------------------------------------
```

El conjunto de Owen del proceso de producción lineal $(A_{\bullet\{3\}}, I_N, c_{\{3\}})$ *coincide con el del*

proceso (A, B, c)*. Sin embargo, vemos que esto no es así para el valor de Shapley y el nucleolo.*

A continuación, se muestra un cuadro que resume y compara todas las propiedades anteriores para cada uno de los conjuntos de solución que venimos tratando a lo largo de todo el trabajo.

	Conjunto de Owen	Shapley	Nucleolo
Eficiencia Unipersonal	✓	✓	✓
Reescalado	✓	✓	✓
Repartos en el núcleo	✓	✗	✓
No manipulabilidad	✓	✗	✗
Consistencia	✓	✗	✗
Eliminación	✓	✗	✗

Tabla 2.3: Comparativa propiedades.

Además, el nucleolo y el conjunto de Shapley tienen un elevado coste computacional a medida que aumenta el número de jugadores. Esto no ocurre en la misma medida con el conjunto de Owen, cuyo coste computacional viene dado por la solución de un PPL. Hoy en día, el software de optimización disponible es más que suficiente para resolver en un tiempo razonable los PPL a los que nos podríamos enfrentar en este contexto.

Capítulo 3

Aplicación web para los juegos de producción lineal

A lo largo de este trabajo se han mostrado varios ejemplos que ponían de manifiesto los juegos cooperativos asociados a los procesos de producción lineal, así como algunas reglas de reparto. Sin embargo, para obtener la solución gráfica de los mismos teníamos que recurrir a introducir líneas de código y conocer la librería para poder representarlas. Como mejora adicional, también se ha contruido una aplicación en Shiny que permite al usuario final interactuar de una forma más visual con la librería y que no requiere ningún conocimiento del software empleado.

Shiny (http://shiny.rstudio.com/), es una herramienta que permite crear fácilmente aplicaciones web interactivas directamente desde R, ayudando así a los usuarios a interactuar con sus datos sin tener que manipular el código. Además, dispone de widgets pre-construidos, haciendo posible la construcción de aplicaciones visualmente atractivas e interactivas. Estas aplicaciones se pueden alojar en páginas web, insertarlas en documentos de R Markdown o crear cuadros de mando. En nuestro caso, hemos optado por publicarla en Shinyapps.io, una plataforma como servicio (PaaS) orientada a alojar aplicaciones web de Shiny.

La aplicación web que se ha contruido permite representar los problemas de producción lineal e ir viendo estos problemas para cada una de las coaliciones del juego. Para ejecutar esta aplicación tan solo tenemos que acceder a la web.

Una vez hecho esto, se abrirá una aplicación web en el navegador y podremos introducir de forma manual los parámetros de nuestro problema: vector de beneficios, matriz de producción y matriz de recursos (en la siguiente sección se recoge un manual de uso de la misma). En el caso de que el problema que se quiere resolver incluya a más de un jugador o, lo que es lo

mismo, consista en un juego de producción lineal, al introducir la matriz de recursos aparecerá un nuevo filtro para seleccionar la coalición en función del número jugadores involucrados. Una vez introducidos todos los parámetros y seleccionada la coalición que se quiera estudiar, haremos click en el botón "Submit" y automáticamente se mostrará la formulación del problema de producción lineal asociado, la solución gráfica y el valor de la función objetivo para la solución óptima del problema. Además, en el caso de que se introduzca un problema con tres jugadores, también se mostrará de forma gráfica el conjunto de imputaciones, junto con el núcleo y otros conjuntos de repartos que se han comentado a lo largo del trabajo.

En la Figura 3.1 se muestra una captura de pantalla para uno de los ejemplos que se estudia en uno de los capítulos anteriores de este trabajo. En cuanto se desee ver el resultado para otra coalición o se cambie alguno de los parámetros del problema, automáticamente la aplicación actualiza los datos y nos muestra los nuevos resultados.

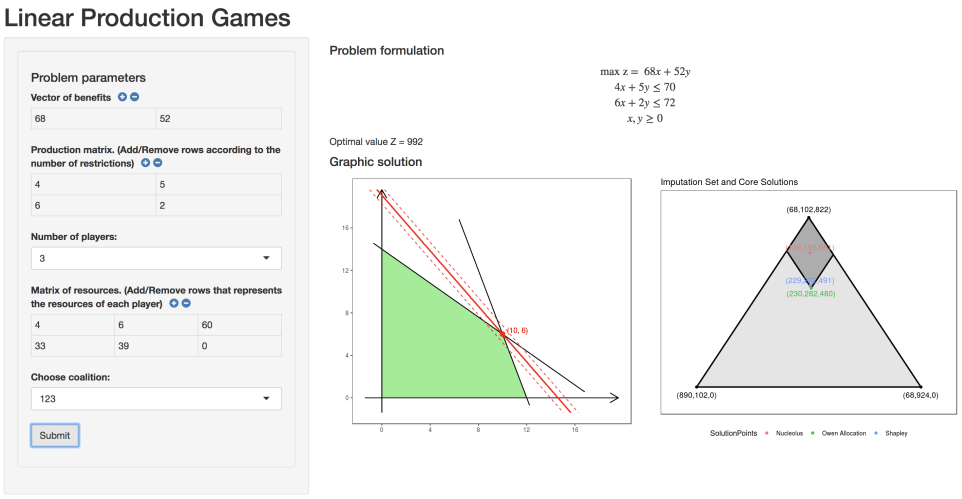


Figura 3.1: Captura de pantalla aplicación Shiny.

3.1. Manual de usuario

En esta sección se detallan los pasos necesarios para resolver un problema de producción lineal. Veremos los pasos a seguir a través del Ejemplo 2.1. Se trata de un problema de producción lineal donde el vector de beneficios viene dado por $c = (68, 52)^t$, la matriz de producción es

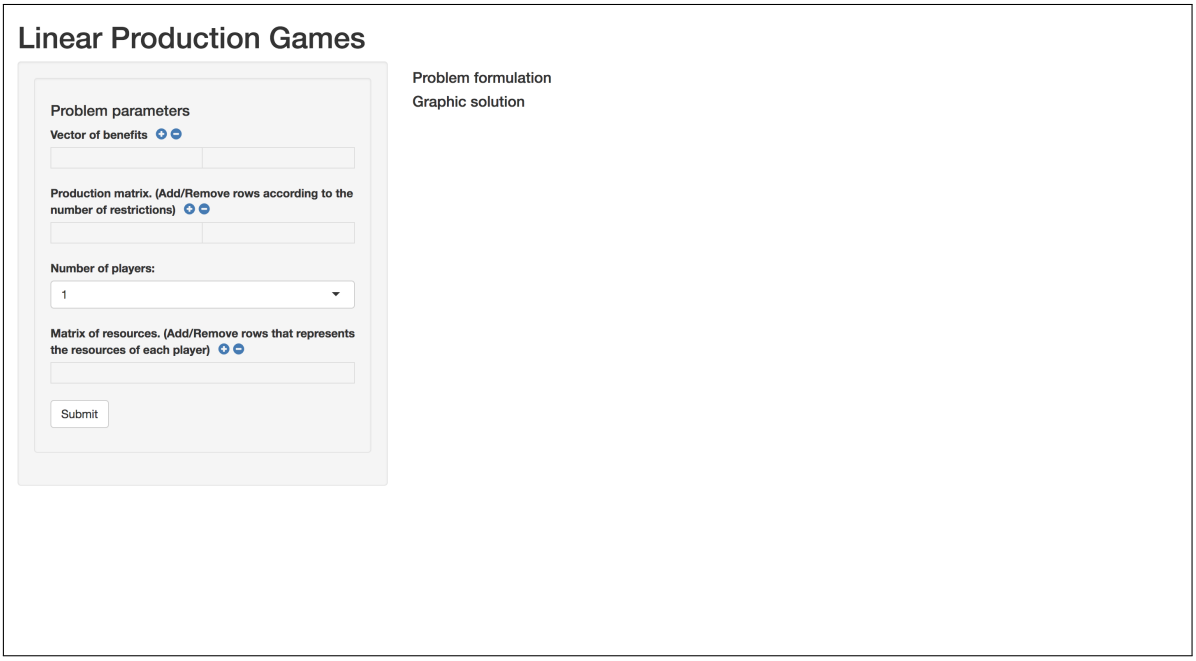
$$A = \begin{pmatrix} 4 & 5 \\ 6 & 2 \end{pmatrix}$$

y la matriz que incluye los vectores de recursos viene dada por:

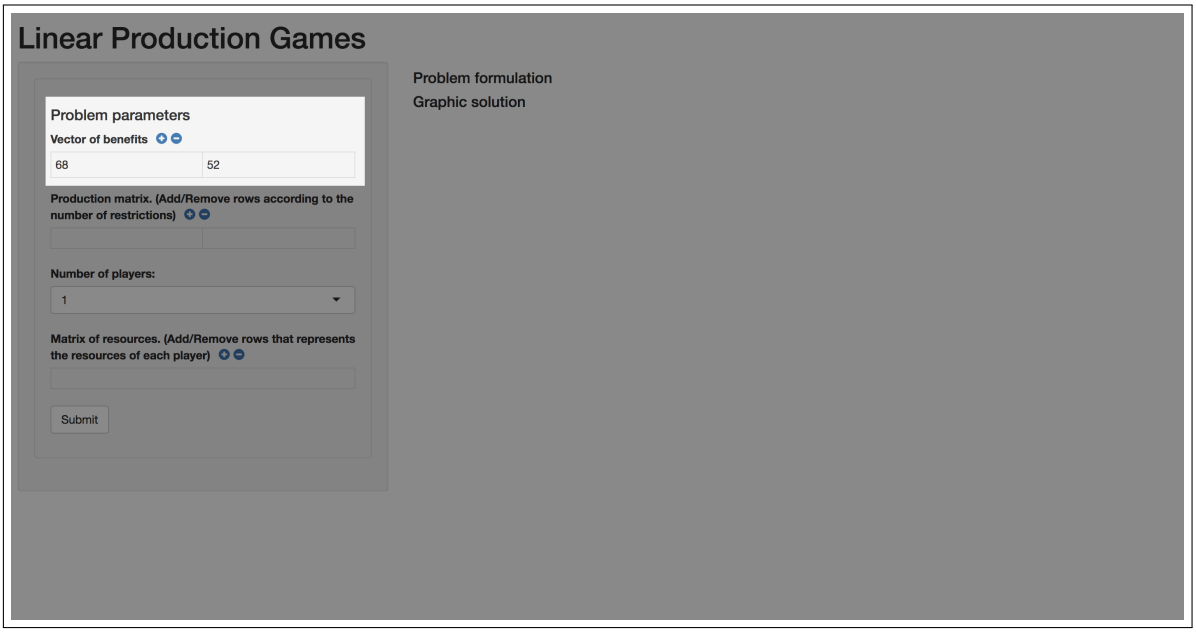
$$B = \begin{pmatrix} 4 & 6 & 60 \\ 33 & 39 & 0 \end{pmatrix}.$$

Para iniciar la aplicación basta con abrir un navegador e introducir la dirección web https://productlineargames.shinyapps.io/myapp/ y seguir los pasos que se describen a continuación:

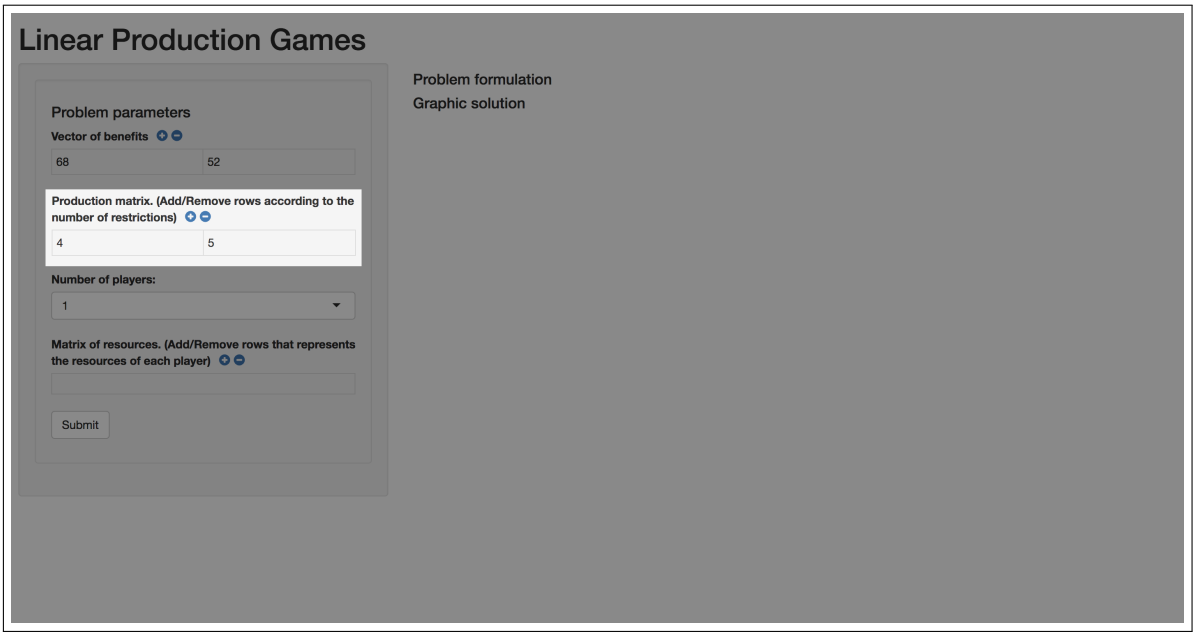
Pantalla inicial de la aplicación.



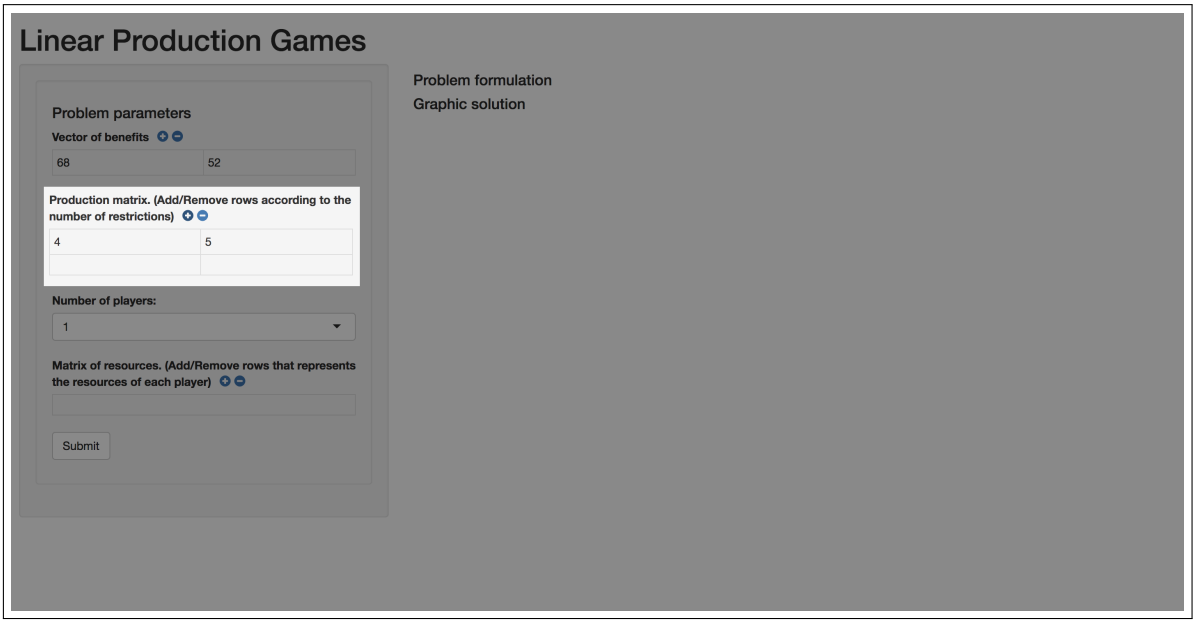
1. Introducir el vector de beneficios.



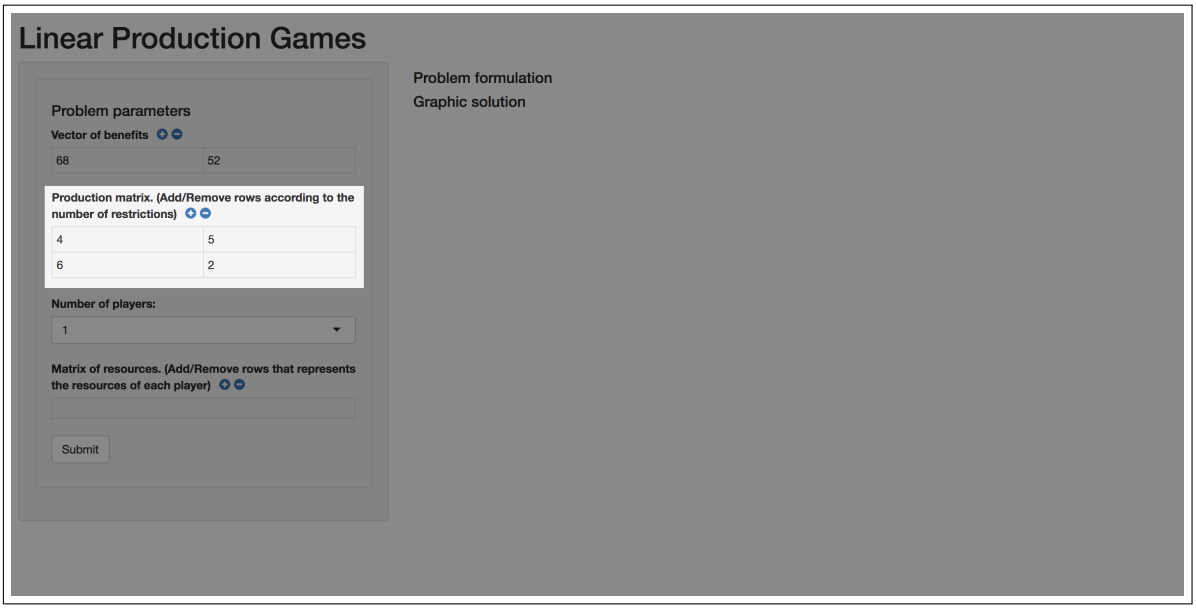
2. Introducir la primera fila de la matriz de producción.



3. Añadimos una fila a la matriz de producción.



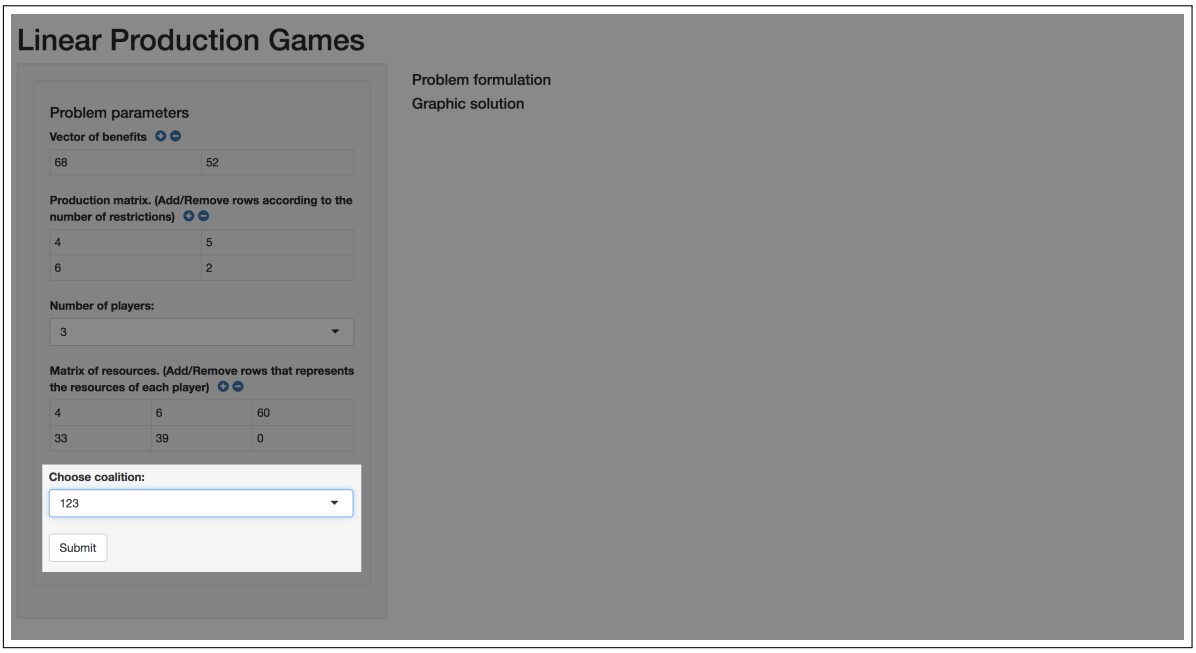
4. Introducimos la segunda fila de la matriz de producción.



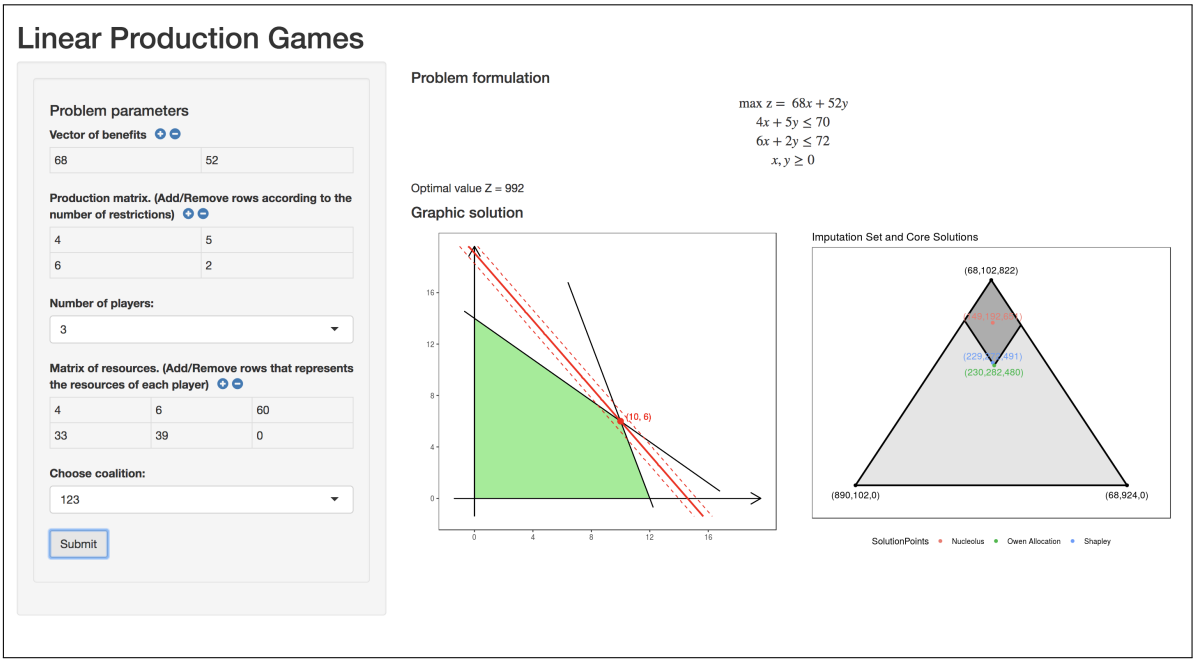
5. Escogemos el número de jugadores (si el número de jugadores es mayor que 1 automáticamente aparece una nueva opción para escoger una coalición).

6. Introducimos los recursos disponibles por los jugadores.

7. Escogemos la coalición para la que queremos resolver el problema y le damos a “Submit”.



8. Resultado final de la aplicación (como se trata de un problema con tres jugadores, obtenemos también el conjunto de imputaciones con el núcleo y algunas soluciones puntuales).



Capítulo 4

Conclusiones y trabajo futuro

A lo largo de este documento se ha intentado poner de manifiesto todo el trabajo desarrollado para la implementación en `R` de los problemas de producción lineal, juego asociado y principales soluciones puntuales asociadas al mismo. Se ha tratado de introducir los problemas a partir de cero, empleando una notación lo más sencilla posible y tratando de ejemplificar todas las funciones implementadas en la librería a través de los ejemplos pertinentes.

Se ha hecho un recorrido por la literatura referente al problema de producción lineal. Sin embargo, existen nueva aproximaciones que no se han contemplado en este trabajo como es el caso multiobjetivo. Hoy en día es muy habitual que, además de maximizar los beneficios en la producción, se persigan otros objetivos complementarios como puede ser minimizar el impacto en el medioambiente. En [28] y [6] se pueden ver dos claros ejemplos de este tipo de casuística.

En [28], se recoge un ejemplo de una industria de acero laminado en la India en el que se intenta encontrar la combinación óptima de producción entre 13 diferentes productos. En este caso, la organización intenta conseguir varios objetivos de forma simultánea: maximización del EBIDTA (indicador financiero que representa el beneficio bruto de explotación calculado antes de la deducibilidad de los gastos), maximizar el uso de la planta de producción, conseguir estabilidad en el mercado a largo plazo, introducción de un nuevo producto en el mercado y expansión de la planta. Todos estos objetivos se pueden modelar para conseguir un problema de producción multiobjetivo.

Por otro lado, en [6] se tiene un ejemplo de un fabricante de componentes metálicos en la industria automotriz. En este artículo, se intenta conseguir, además de maximizar los beneficios, reducir los costes y el impacto medioambiental.

Con todo esto, existen estudios que tratan este tipo de problemas. Un claro referente se puede ver en [16], trabajo en el que se explica cómo obtener un reparto para el conjunto de Owen y el nucleolo. Además, hacen uso de la optimalidad de Pareto para obtener soluciones del problema multiobjetivo. En esta línea, un trabajo muy interesante sería el hecho de implementar las funciones necesarias en `R` que permitiese resolver este tipo de problemas al igual que se hizo para el caso uniobjetivo. De hecho, ya exiten librerías que estudian la optimalidad de Pareto de las que se podría hacer uso como puede ser la librería `mco` [15].

Por otro lado, en cuanto a la librería `coopProductGame` y a la aplicación web se podrían crear, como líneas de trabajo futuras, nuevas versiones que hagan estudio aún más completos. A continuación, se listan algunas de estas líneas de trabajo futuras:

- Actualmente solo se calculan todos los repartos del conjunto de Owen en el caso de que el problema de producción lineal tenga dos recursos. Sería muy interesante mejorar las funciones que realizan estos cálculos para, poder obtener todos los repartos, al menos en el caso de tres recursos, así como la representación correspondiente en tres dimensiones.
- Mejorar la representación del conjunto de imputaciones, núcleo y soluciones descritas en el trabajo para el caso de 4 jugadores.
- Mejorar tanto el diseño como el rendimiento de la aplicación web introduciendo nuevos parámetros de entrada, que permitan seleccionar qué se quiere obtener; por ejemplo, si solo se quiere la formulación del problema, la solución analítica del mismo, la solución gráfica o la representación del conjunto de imputaciones y soluciones puntuales.
- Incorporar la opción de introducir los datos del problema en la aplicación web a través de un archivo; por ejemplo desde un csv.
- Introducir la posibilidad de generar informes a partir de la aplicación que se puedan descargar e incluyan la formulación del problema y su solución tanto gráfica como analítica.

Apéndices

Apéndice A

Manual del paquete "coopProductGame"

Package 'coopProductGame'

August 25, 2018

Type Package

Version 2.0

Date 2018-08-17

Title Cooperative Aspects of Linear Production Programming Problems

Author Daniel Prieto

Maintainer Daniel Prieto <daniel.prieto.rodriguez89@gmail.com>

Depends R (>= 2.7.0)

Imports lpSolveAPI (>= 5.5.2), ggplot2 (>= 2.2.1), grid, GameTheory (>= 2.7), dplyr (>= 0.7.4), kappalab, gtools

Description Computes cooperative games and allocation rules associated with linear production programming problems.

License GPL-3

NeedsCompilation no

RoxygenNote 6.1.0

Repository CRAN

Date/Publication 2018-08-25 16:34:29 UTC

R topics documented:

coopProducGame-package

Cooperative aspects of linear product games

Description

G. Owen (1975, Math. Programming 9, 358-370) assigned to each linear production process a cooperative game, a "linear production game". Further, he introduced a method to find a subset of the core of linear production games that verifies certain properties, which is called the "Owen set." This package computes the linear production games and allocation rules associated.

Details

Package:	coopProductGame
Type:	Package
Version:	2.0
Date:	2018-07-01
License:	GPL-3

The most important function is coopProductGame. Other functions included in the package are auxiliary ones that can be used independently.

Author(s)

Daniel Prieto Rodríguez

Maintainer: Daniel Prieto Rodríguez <daniel.prieto.rodriguez89@gmail.com>

References

S. Cano-Berlanga, J. M. Gimenez-Gomez, and C. Vilella. Enjoying cooperative games: The r package gametheory. Working Paper No. 06; CREIP; Spain, March 2015.

D. B. Gillies. Some Theorems on n-Person Games. PhD thesis, Princeton University, 1953.

G. Owen. On the core of linear production games. Mathematical Programming, 9:358–370, 1975.

D. Schmeidler. The nucleolus of a characteristic function game. SIAM Jounal of Applied Mathematics, 17:1163–1170, 1969.

L. S. Shapley. A value for n-person games. Contributions to the theory games II, 28:124–131, 1953.

J. R. G. van Gellekom et al. Characterization of the owen set of linear production processes. Games and Economic Behavior, 32:139–156, 2000.

`coalitions` *Coalitions for a given numbers of players* `n`.

Description

This functions gives all the coalitions, including the empty coalition, for a number of players `n`.

Usage

```
coalitions(n)
```

Arguments

`n`	Number of players.

Value

A list with the following components:

`Binary`	Matrix where each row is a binary representation of the coalition.
`Usual`	Vector with the usual configurations of the coalitions.

Author(s)

D. Prieto

Examples

```
# Number of players:
n <- 3
# Associated coalitions:
coalitions(n)

# $Binary
#       [,1] [,2] [,3]
# [1,]    0    0    0
# [2,]    1    0    0
# [3,]    0    1    0
# [4,]    0    0    1
# [5,]    1    1    0
# [6,]    1    0    1
# [7,]    0    1    1
# [8,]    1    1    1
#
# $Usual
# [1]   0   1   2   3  12  13  23 123
```

coopProductGame *Cooperative linear production games*

Description

Given a linear production problem `A%*%x <= B`, the `coopProductGame` solves the problem by making use of `lpSolveAPI` where each agent provides his own resources.

Usage

```
coopProductGame(c, A, B, plot = FALSE, show.data = FALSE)
```

Arguments

`c`	vector containing the benefits of the products.
`A`	production matrix.
`B`	matrix containing the amount of resources of the several players where each row is one player.
`plot`	logical value indicating if the function displays graphical solution (`TRUE`) or not (`FALSE`). Note that this option only makes sense when we have a two-dimension problem.
`show.data`	logical value indicating if the function displays the console output (`TRUE`) or not (`FALSE`). By default the value is `TRUE`.

Value

`coopProductGame` returns a list with the solution of the problem, the objective value and a Owen allocation if it exists. If we have a two dimension dual problem, the function returns all the Owen allocations (if there are more than one we obtain the end points of the segment that contains all possible allocations.)

Author(s)

D. Prieto

Examples

```
# Vector of benefits
c <- c(68, 52)
# Production matrix
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Matrix of resources. Each row is the vector of resources of each player
B <- matrix(c(4, 6, 60, 33, 39, 0), ncol = 3, byrow = TRUE)
# Solution of the associated linear production game
coopProductGame(c, A, B, show.data = TRUE)

 # ----------------------------------------------------------------------
```

```
# Optimal solution of the problem for each coalition:
# ------------------------------------------------------------------------
#
# S={1}      1.00  0.00
# S={2}      1.50  0.00
# S={3}      0.00  0.00
# S={1,2}    2.50  0.00
# S={1,3}    1.68 11.45
# S={2,3}    2.86 10.91
# S={1,2,3} 10.00  6.00
#
# ------------------------------------------------------------------------
#   Cooperative production game:
# ------------------------------------------------------------------------
#                S={0} S={1} S={2} S={3} S={1,2} S={1,3} S={2,3} S={1,2,3}
# Associated game    0    68   102     0     170     710     762       992
# ------------------------------------------------------------------------
#
# ------------------------------------------------------------------------
#   The game has a unique Owen's allocation:
# ------------------------------------------------------------------------
# [1] "(230, 282, 480)"
# ------------------------------------------------------------------------
```

`linearProductionGame` *Cooperative linear production games*

Description

Given a linear production problem, the `linearProductionGame` function solves the problem by making use of `lpSolveAPI` where each agent provides his own resources.

Usage

```
linearProductionGame(c, A, B, plot = FALSE, show.data = FALSE)
```

Arguments

`c`	vector containing the benefits of the products.
`A`	production matrix.
`B`	matrix containing the amount of resources of the several players where each row is one player.
`plot`	logical value indicating if the function displays graphical solution (`TRUE`) or not (`FALSE`). Note that this option only makes sense when we have a two-dimension problem.
`show.data`	logical value indicating if the function displays the console output (`TRUE`) or not (`FALSE`). By default the value is `TRUE`.

Value

`linearProductionGame` returns a list with the solutions of the associated problem of each coalition and the objective value for coalition N.

Author(s)

D. Prieto

Examples

```
# Vector of benefits
c <- c(68,52)
# Production matrix
A <- matrix(c(4,5,6,2),ncol=2, byrow = TRUE)
# Matrix of resources. Each column is the vector of resources of each player
B <- matrix(c(4, 6, 60, 33, 39, 0),ncol = 3, byrow = TRUE)
# Solution of the associated linear production game
linearProductionGame(c, A, B, show.data = TRUE)
```

```
 # ------------------------------------------------------------------------
 # Optimal solution of the problem for each coalition:
 # ------------------------------------------------------------------------
 #
 # S={1}      1.00  0.00
 # S={2}      1.50  0.00
 # S={3}      0.00  0.00
 # S={1,2}    2.50  0.00
 # S={1,3}    1.68 11.45
 # S={2,3}    2.86 10.91
 # S={1,2,3} 10.00  6.00
 #
 # ------------------------------------------------------------------------
 #   Cooperative production game:
 # ------------------------------------------------------------------------
 #                S={0} S={1} S={2} S={3} S={1,2} S={1,3} S={2,3} S={1,2,3}
 # Associated game    0    68   102     0     170     710     762       992
 # ------------------------------------------------------------------------
```

`makeLP` *Make a linear production programming problem*

Description

Given a linear production problem `A %*% x <= b`, the `makeLP` function creates a new `lpSolve` linear program model object.

Usage

```
makeLP(c, A, b)
```

Arguments

c	vector of benefits.
A	production matrix.
b	vector of resources.

Value

makeLP returns a lpSolve linear program model object. Specifically an R external pointer with class lpExtPtr.

Author(s)

D. Prieto

Examples

```
# Vector of benefits
c <- c(68,52)
# Production matrix
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Vector of resources
b <- c(4,33)
# Make the associated linear production problem
prod <- makeLP(c, A, b)
```

---

nucleolus *Nucleolus solution*

Description

This function computes the nucleolus solution of a game with a maximum of 4 agents.

Usage

```
nucleolus(game, show.data = FALSE)
```

Arguments

game	a vector that represents the cooperative game.
show.data	logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default the value is FALSE.

Value

`nucleolus` returns and prints the Nucleolus Solution of associated cooperative game.

Author(s)

D. Prieto

Examples

```
# Cooperative game
game <- c(68, 102, 0, 170, 710, 762, 992)
# Nucleolus solution
nucleolus(game, show.data = TRUE)

 # ------------------------
 # Nucleolus Solution
 # ------------------------
 # [1] "(149, 192, 651)"
```

`owenSet` *Owen Set*

Description

This function computes the Owen Set of a linear production game

Usage

```
owenSet(c, A, B, show.data = FALSE)
```

Arguments

`c`	vector containing the benefits of the products.
`A`	production matrix.
`B`	matrix containing the amount of resources of the several players where each row is one player.
`show.data`	logical value indicating if the function displays the console output (`TRUE`) or not (`FALSE`). By default the value is `FALSE`.

Value

`owenSet` returns and prints the owen Set of associated linear production problem.

Author(s)

D. Prieto

Examples

```
# Vector of benefits
c <- c(68, 52)
# Production matrix
A <- matrix(c(4, 5, 6, 2), ncol=2, byrow = TRUE)
# Matrix of resources. Each row is the vector of resources of each player
B <- matrix(c(4, 6, 60, 33, 39, 0),ncol = 3, byrow = TRUE)
# Solution of the associated linear production game
owenSet(c, A, B, show.data = TRUE)

 # ------------------------------------------------------------------------
 # The linear production problem has a unique Owen's allocation:
 # ------------------------------------------------------------------------
 # [1] "(230, 282, 480)"
```

`plotCoreSet` *Plot Core Set for cooperative production linear games.*

Description

Given a linear production game, the `plotCoreSet` function plots the imputation Set, Core Set and the most common solutions (Nucleolus, Shapley Value and allocations of the Owen Set).

Usage

```
plotCoreSet(c, A, B)
```

Arguments

`c`	vector containing the benefits of the products.
`A`	production matrix.
`B`	matrix containing the amount of resources of the several players where each row is one player.

Details

In most cases the Owen Set consists of a single allocation, but in some cases there are infinities. In the case that there are infinite allocations, if the problem has two dimensions, they will be given by a line, which we will represent graphically. If the problem has more than two dimensions, an allocation of all possible ones will be represented.

Value

`plotCoreSet` returns a `ggplot` object with the imputation set of the game, the core and the most common solutions.

Author(s)

D. Prieto

See Also

coopProductGame

Examples

```
# Vector of benefits
c <- c(68, 52)
# Production matrix
A <- matrix(c(4, 5, 6, 2), ncol = 2, byrow = TRUE)
# Matrix of resources. Each row is the vector of resources of each player
B <- matrix(c(4, 6, 60, 33, 39, 0), ncol = 3, byrow = TRUE)
# Solution of the associated linear production game
plotCoreSet(c, A, B)
```

---

plotlm *Plot method for linear production programming problems*

Description

This function plots the graphical solution of simple linear production programming problems with two decision variables. The decision variables must be real, nonnegative and cannot have a finite upper bound. Only inequality constraints are supported.

Usage

```
plotlm(prod, A, b, c, title = NULL)
```

Arguments

prod	a linear production programming problem of class lpExtPtr.
A	production matrix.
b	vector of resources.
c	vector of benefits.
title	title of the plot. By default is NULL, so it returns a plot without title.

Value

Returns and plot a ggplot object with graphical solution of the problem.

Author(s)

D. Prieto

See Also

makeLP.

Examples

```
# Vector of benefits
c <- c(68,52)
# Matrix of coefficients
A <- matrix(c(4,5,6,2), ncol = 2, byrow = TRUE)
# Vector of resources
b <- c(4,33)
# Make the associated linear program
prod <- makeLP(c, A, b)
plotlm(prod, A, b, c)
```

---

productLinearProblem *Linear production programming problems*

Description

Given a linear production programming problem `A %*% x <= b`, the `productLinearProblem` solves the problem by making use of `lpSolveAPI`.

Usage

```
productLinearProblem(c, A, b, plot = FALSE, show.data = FALSE)
```

Arguments

c	vector of benefits.
A	production matrix.
b	vector of resources.
plot	logical value indicating if the function displays graphical solution (TRUE) or not (FALSE). Note that this option only makes sense when we have a two-dimension problem.
show.data	logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default the value is TRUE.

Value

`productLinearProblem` returns and prints a list with the following components:

`ObjetiveValue` Value of the objetive function from a successfully solved linear production programming problem.

`OptimalSolution` Values of the variables from a successfully solved linear production programming problem.

Author(s)

D. Prieto

Examples

```
# Vector of benefits
c <- c(68,52)
# Production matrix
A <- matrix(c(4,5,6,2),ncol=2, byrow = TRUE)
# Matrix of resources. Each row is the vector of resources of each player
b <- c(4,33)
# Solution of the associated linear production game
productLinearProblem(c,A,b, show.data = TRUE)

# ------------------------------------------------------------------------
# Objetive value:
# ------------------------------------------------------------------------
#   [1] "Z = 68"
#
# ------------------------------------------------------------------------
# Optimal solution:
# ------------------------------------------------------------------------
#   [1] 1 0
# ------------------------------------------------------------------------
```

---

shapleyValue *Shapley Value Solution*

Description

Calculates the Shapley Value for a N-agent cooperative game.

Usage

```
shapleyValue(game, show.data = FALSE)
```

Arguments

game	a vector that represents the cooperative game.
show.data	logical value indicating if the function displays the console output (TRUE) or not (FALSE). By default the value is FALSE.

Value

shapleyValue returns and prints the Shapley Value of associated cooperative game.

Author(s)

D. Prieto

Examples

```
# Cooperative game
game <- c(68, 102, 0, 170, 710, 762, 992)
# Shapley Value
shapleyValue(game, show.data = TRUE)

 # ----------------------------
 # Shapley Value Solution:
 # ----------------------------
 # [1] "(229, 272, 491)"
```

Index

Bibliografía

[1] R. J. Aumann and M. Maschler. The bargaining set for cooperative games. *Advanced in Game Theory*, 52:443–476, 1964.

[2] O. Bondareva. Some applications of linear programming methods to the theory of cooperative games. *Problemy Kibernet*, 10:119–139, 1963.

[3] S. Cano-Berlanga, J. M. Gimenez-Gomez, and C. Vilella. Enjoying cooperative games: The r package gametheory. *Working Paper No. 06; CREIP; Spain*, March 2015.

[4] I. Curiel. *Cooperative Game Theory and Applications*. Springer, 1997.

[5] J. J. M. Derks. A short proof of the inclusion of the core in the weber set. *International Journal of Game Theory*, 21:149–150, 1992.

[6] T. Wen Hsien et al. A product mix decision model using green manufacturing technologies under activity based costing. *Journal of Cleaner Production*, 57:178–187, 2013.

[7] J. Mufandaedza F. Majeke, J. Makeke and M. Shoko. Modelling a small farm livelihood system using linear programming in bindura, zimbabwe. *Research Journal of Management Sciences*, 2(5):20–23, 2013.

[8] D. B. Gillies. *Some Theorems on n-Person Games*. PhD thesis, Princeton University, 1953.

[9] Hemendra Lal Gunasekara, Suhaiza Zainali, and Ali Haj Aghapour. The optimization problem of product-mix and linear programming applications; a single-case study in tea industry. *Autralian Journal of Basic and Applied Sciences*, 9(3):7–18, 2015.

[10] F. L. Hitchcook. The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20:224–230, 1941.

[11] T. Ichiishi. Super-modularity: Applications to convex games and to the greedy algorithm for lp. *Journal of Economic Theory*, 25:283–286, 1981.

[12] E. Kalai and E. Zemel. On the order of eliminating dominated strategies. *Operations Research Letters*, 9(2):85–89, 1988.

[13] T. C. Koopmans. Optimum utilization of the transportation system. *Econometrica*, 17:136–146, 1949.

[14] lp_solve and Kjell Konis. *lpSolveAPI: R Interface to 'lp_solve' Version 5.5.2.0.*, 2016. R package version 5.5.2.0-17.

[15] O. Mersmann. *mco: Multiple Criteria Optimization Algorithms and Related Functions*, 2014. R package version 1.0-15.1.

[16] I. Nishizaki and M. Sakawa. On computation methods for solutions of multiobjective linear producion programming games. *European Journal of Operational Research*, 129:386–413, 2001.

[17] G. Owen. On the core of linear production games. *Mathematical Programming*, 9:358–370, 1975.

[18] D. Schemeidler. Cores of exact games. *Journal of Mathematical Analysis and Applications*, 40:214–225, October 1972.

[19] D. Schmeidler. The nucleolus of a characteristic function game. *SIAM Jounal of Applied Mathematics*, 17:1163–1170, 1969.

[20] L. S. Shapley. A value for n-person games. *Contributions to the theory games II*, 28:124–131, 1953.

[21] L. S. Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14:453–460, 1967.

[22] L. S. Shapley. Cores of convex games. *International Journal of Game Theory*, 1:11–26, 1971.

[23] L. S. Shapley and M. Shubik. The assignment game i: The core. *International Journal of Game Theory*, 1:111–130, 1972.

[24] G. J. Stigler. The cost of subsistence. *Journal of farm economics.*, 27(2):303–314, 1945.

[25] J. R. G. van Gellekom et al. Characterization of the owen set of linear production processes. *Games and Economic Behavior*, 32:139–156, 2000.

[26] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[27] R.J. Weber. *Probabilistic values for games, in The Shapley Value (A. E. Roth, Ed.)*, volume 26, pages 101–119. Cambridge University Press, 1988.

[28] M. M. Gupta Y. Chauhan and D.R. Zanwar. Optimization of product mix in cold rolling steel industry using product portfolio matrix and multi objective goal programming model. *Industrial Engineering Letters*, 2(6), 2012.