UNIVERSIDAD DE MURCIA

FACULTAD DE MATEMÁTICAS

TRABAJO FIN DE MÁSTER

REPARTO DE COSTES EN PROBLEMAS DE ÁRBOL GENERADOR DE COSTE MÍNIMO

David Gómez Saura

Curso 2015-2016

Declaración de origininalidad:

DAVID GÓMEZ SAURA, autor del TFM *Reparto de costes en problemas de árbol generador de coste mínimo*, bajo la tutela del profesor MANUEL ANDRÉS PULIDO CAYUELA, declara que el trabajo que presenta es original, en el sentido de que ha puesto el mayor empeño en citar debidamente todas las fuentes utilizadas.

En Murcia, a 6 de julio de 2016

(Nota: En la Secretaría de la Facultad de Matemáticas se ha presentado una copia firmada de esta declaración.)

Reparto de costes en problemas de árbol generador de coste mínimo

David Gómez Saura

dirigido por

Manuel Andrés Pulido Cayuela

Índice general

Resumen

En el presente trabajo se llevará a cabo un estudio en profundidad del problema del árbol generador de coste mínimo y el reparto de costes en dicho problema. El problema surge al considerar una serie de usuarios o agentes, que quieren conectarse a un determinado servicio que es provisto desde un único servidor y están dispuestos a colaborar para hacerlo. Los usuarios no tienen que conectarse al servicio de forma directa necesariamente, sino que lo pueden hacer indirectamente a través de otros usuarios ya conectados. Así, puede considerarse un grafo completo en el que un nodo específico representa el servidor que provee el servicio y el resto de nodos representan los usuarios interesados en conectarse. Cada arista tiene asociado un coste determinado. El problema consta pues de dos fases: la primera de ellas será determinar la estructura que conecte a todos los usuarios con la fuente a menor coste, es decir, encontrar el árbol generador de coste mínimo de la red considerada. Este problema es un problema sobradamente conocido dentro de la investigación operativa y para el que existen algoritmos eficaces como son los algoritmos de *Kruskal* y *Prim*, para los que dado una red conexa arbitraria, determinar un árbol generador de mínimo coste, que no necesariamente debe ser único. Una vez determinado dicho árbol y, por tanto, el mínimo coste por el cual los agentes pueden conectarse al servicio, comienza la segunda fase del problema: determinar la cantidad que debe pagar cada usuario. Es entonces cuando aparece en escena la teoría de juegos. La teoría de juegos es el área de las matemáticas que se encarga del estudio de situaciones en las que interviene un un conjunto de individuos que tienen una serie de incentivos y que quieren satisfacer, ya sea colaborando (modelos de colaboración o cooperación), o compitiendo (modelos de competencia) entre ellos. Los modelos de cooperación se encargan fundamentalmente de estudiar cómo distribuir el beneficio obtenido de la colaboración de los individuos. Por otro lado, los modelos de competencia se encargan de estudiar y analizar el comportamiento estratégico de los individuos. El problema que se trata en este trabajo se encuadra dentro de los modelos de cooperación.

Un juego cooperativo es un par compuesto por un conjunto de jugadores y una función de característica que a cada coalición de jugadores le asigna una cantidad que representa el beneficio (o el coste) asociado a dicha coalición. De este modo, a partir de todo problema de árbol generador de coste mínimo, surge un juego cooperativo, denominado juego de árbol generador de coste mínimo, fruto de considerar para cada coalición el coste que le supone a los integrantes

de dicha coalición conectarse al servicio sin utilizar las conexiones del resto de jugadores.

Este problema matemático aparece en diferentes contextos de la vida real, como por ejemplo, en el caso en el que quiere construirse una instalación para proveer de un determinado bien a zonas geográficamente separadas, como pudiera ser el emplazamiento de tuberías para el suministro de agua a diferentes ciudades o la instalación de cableado para la red eléctrica. Otra situación en la que surge un problema de árbol generador de mínimo coste es el caso en el que una serie de vecinos quieren establecer una red que los conecte conjuntamente a servicios como internet o televisión por cable. Por tanto, las propiedades matemáticas estudiadas para problemas de árbol generador de coste mínimo pueden servir para establecer repartos de costes justos en estas situaciones de la vida real en las que hay que dividir un determinado coste entre distintos individuos o entidades.

El presente texto consta de dos capítulos que componen el cuerpo del trabajo, bibliografía y por último, un índice terminológico para facilitar la búsqueda de definiciones.

El capítulo 1 comienza con una sección dedicada a la introducción a la teoría de grafos. En ella se definirán conceptos que aparecerán durante todo el trabajo y que resultan claves para su buen entendimiento. Se mostrarán también los ya comentados algoritmos de Kruskal y Prim. La segunda sección del capítulo estará dedicada a la introducción de conceptos y resultados, esta vez relacionados con la teoría de juegos. Aparecerá la definición de *juegos de utilidad transferible*, así como otros conceptos que serán frecuentes a lo largo de todo el texto. Habitualmente, los textos dedicados a juegos cooperativos se refieren a juegos de beneficios, sin embargo, en este texto, por la naturaleza de los juegos de árbol generador de coste mínimo, se hablará de juegos de coste y las definiciones se referirán siempre a este tipo de juegos. Uno de los conceptos introducidos de mayor importancia será el de *núcleo* de un juego. El núcleo de un juego es el conjunto formado por los vectores de coste que reparten el coste total de todos los jugadores y, además, ninguna coalición de jugadores tiene razones para rechazar, pues dicha asignación les es más ventajosa que dejar de colaborar con el resto de jugadores para satisfacer por ellos mismos sus intereses. De gran importancia será pues encontrar vectores de asignación de costes que pertenezcan al núcleo de un juego, aunque en general no se tienen garantías de que dicho conjunto sea no vacío.

La tercerá sección del capítulo estará dedicada a la definición formal del problema de árbol generador de coste mínimo y el juego cooperativo asociado. Se verán algunos ejemplos de problemas y se mostrará el resultado central del capítulo: el núcleo de un juego de árbol generador de coste mínimo es distinto de vacío. Es más, se verá que pueden encontrarse vectores en el núcleo a partir de un árbol generador de coste mínimo asociado al problema, sin tener que obtener la función característica del juego. Además, se proporcionará un método que permitirá, a partir de vectores conocidos pertenecientes al núcleo, generar nuevos vectores dentro del mismo.

La última sección del capítulo estará dedicada al estudio de *formas irreducibles*. Una forma irreducible es un problema de árbol generador de coste mínimo en el que si se reduce el coste de una arista cualquiera, el coste del árbol generador de coste mínimo asociado al problema se reduce estrictamente. Se estudiarán las propiedades de este tipo de problemas y la estructura del núcleo del juego asociado. Los problemas en forma irreducible serán de gran importancia en el segundo capítulo.

La cuestión central dentro de los problemas de árbol generador de coste mínimo es la de cómo repartir de forma justa el coste total entre los usuarios. Para ello, en la literatura relacionada con el tema, se han propuesto diversas *reglas de reparto* de costes. Una regla de reparto es una aplicación que, dado un problema de árbol generador de coste mínimo determinado, proporciona un único vector de asignación de costes entre los usuarios. Además, en los textos relacionados con el problema, se han introducido también diversas propiedades matemáticas y de sentido económico, que permiten comparar las características de las reglas. En este texto se considerarán las siguientes: *selección del núcleo, monotonicidad del coste, solidaridad, monotonicidad de la población, positividad, separabilidad, simetría, independencia de otros costes, independencia de bajos costes, independencia de grandes costes, igualdad de reparto de costes extra* e *independencia de árboles irrelevantes*. En la primera sección del segundo capítulo, además de dar la correspondiente definición formal de cada propiedad, se llevará a cabo una interpretación de las virtudes que confiere a una regla de reparto el cumplimiento de estas propiedades.

Cuatro de las reglas de reparto más conocidas, y que son estudiadas en este texto, son la *regla de Bird*, introducida por C.G. Bird [7] en 1976, la regla de *Kar*, cuya axiomatización fue llevada a cabo por A. Kar [18] en 2002, la regla de *Dutta-Kar*, introducida por B. Dutta y A. Kar [12] en 2004 y la regla de *Feltkamp-Tijs-Muto*, desarrollada por V. Feltkamp, S. Tijs y S. Muto [13] en 1994 y ampliamente estudiada posteriormente por autores como G. Bergantiños y J. Vidal-Puga [1],[2],[3],[4],[5],[6]. En el segundo capítulo se mostrará la definición de cada una de estas reglas y un ejemplo de aplicación a un problema concreto. Se demostrará también cuales de las propiedades anteriores verifica cada regla y se mostrará cómo han sido caracterizadas en la literatura relacionada con el problema del árbol generador de coste mínimo.

Capítulo 1

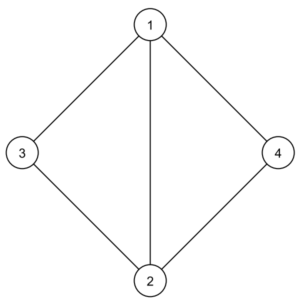
El problema del árbol generador de coste mínimo

1.1. Introducción a la teoría de grafos

Esta primera sección servirá para realizar una revisión de conceptos relacionados con la teoría de grafos que estarán presentes durante toda la memoria. Definiremos, entre otros, el concepto de *árbol generador de coste mínimo* y se verán resultados relacionados que serán utilizados en numerosas demostraciones posteriores. La sección concluirá con la presentación de los algortimos de *Kruskal* y *Prim*.

Definición 1.1.1. *Un **grafo** es un par (V, E) formado por un conjunto finito $V \neq \emptyset$, a cuyos elementos denominaremos **vértices o nodos** y un conjunto de pares no ordenados E, formado por distintos pares de elementos de V a los que denominaremos **aristas**.*
*A dos nodos que forman parte de una misma arista se les llama nodos **adyacentes**.*
*Se dice que una arista es **incidente** con un nodo v cuando v es uno de sus extremos.*
*El **grado de incidencia** de un vértice v es el número de aristas que inciden en él.*

Ejemplo 1.1.2. *Representación gráfica de grafos.*
Consideremos el grafo $G = (V, E)$, donde $V = \{1, 2, 3, 4\}$ y $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4)\}$.
Entonces podemos representar G de la siguiente forma:



Definición 1.1.3. *Dado un grafo $G = (V, E)$, se dice que otro grafo $G' = (V', E')$ es un* ***subgrafo*** *de G si $V' \subseteq V$ y $E' \subseteq E$.*
Llamaremos ***subgrafo parcial*** *de G a un subgrafo $G' = (V, E')$ tal que $E' \subseteq E$.*

Definición 1.1.4. *Un grafo $G = (V, E)$ se dice* ***completo*** *si para cada $v_i, v_j \in V$ con $v_i \neq v_j$ se tiene que $(v_i, v_j) \in E$.*

Definición 1.1.5. *Una* ***cadena*** *o* ***camino*** *es un grafo $P = (V, E)$ donde $V = \{v_1, \cdots, v_n\}$ y $E = \{(v_1, v_2), (v_2, v_3), \cdots, (v_{n-1}, v_n)\}$. En este caso se dice que P* ***conecta*** *v_1 con v_n. Se llama* ***longitud*** *de una cadena al número de aristas que la componen. Un subgrafo de un grafo G que sea una cadena se denomina cadena en el grafo G.*
Un ***ciclo*** *es un grafo $C = (V, E)$, donde el conjunto de nodos es $V = \{v_1, \cdots, v_n\}$ con $n \geq 3$, y el conjunto de aristas $E = \{(v_1, v_2), \cdots, (v_{n-1}, v_n), (v_n, v_1)\}$, o lo que es igual, una cadena más una arista entre los dos vértices unidos por la cadena.*

Definición 1.1.6. *Dado un grafo $G = (V, E)$ y un conjunto $B \subseteq V$, se llama* ***cociclo*** *asociado a B, al conjunto de aristas que inciden en B y en $V \backslash B$, es decir*

$$\omega(B) = \{(v_1, v_2) \in E : v_1 \in B, v_2 \notin B \text{ o } v_1 \notin B, v_2 \in B\}$$

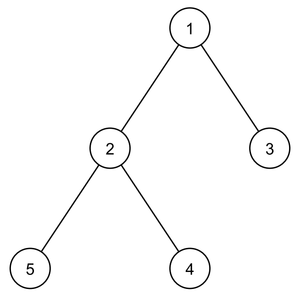
Definición 1.1.7. *Un grafo se dice* ***conexo*** *si cualquier par de vértices distintos están conectados por una cadena. En caso contrario se dice que el grafo es* ***no conexo*** *o* ***disconexo****.*
Llamaremos ***componente conexa*** *de un grafo G a todo subgrafo conexo maximal de G.*

Es claro que todo grafo conexo tiene una única componente conexa. A continuación se introducirán los conceptos de *árbol* y *árbol generador*, así como otros términos relacionados:

Definición 1.1.8. *Un grafo $G = (V, E)$ se dice que es un* ***árbol*** *si es conexo y no contiene ciclos. Un* ***árbol generador*** *de un grafo $G = (V, E)$ es un subgrafo parcial de G conexo y sin ciclos.*

Definición 1.1.9. *En un árbol podemos distinguir un nodo cualquiera que se denominará* ***raíz*** *r . El par compuesto por árbol y raíz se denomina* ***árbol enraizado****.*
Cada nodo $v \neq r$ del árbol enraizado estará conectado a r mediante un único camino $r, v_1, ..., v_k, v$. Diremos entonces que los nodos $r, v_1, ..., v_k$ son ***predecesores*** *de v en el árbol y que v es* ***sucesor*** *de estos nodos.*
En un árbol, los vértices de grado de incidencia 1 reciben el nombre de ***hojas****.*

Ejemplo 1.1.10. *El siguiente grafo es un árbol en el que los nodos 3, 4 y 5 son hojas.*



Teorema 1.1.11 (Caracterización de árboles)**.** *Sea $G = (V, E)$ un grafo tal que $|V| = n$. Las siguientes afirmaciones son equivalentes:*

1. *G es un árbol.*
2. *Entre cada par de vértices de V existe una única cadena.*
3. *G es conexo y el número de aristas es igual a $n - 1$.*
4. *G no contiene ciclos y el número de aristas es igual a $n - 1$.*
5. *G está minimalmente conectado.*
6. *G no contiene ciclos y si añadimos una arista entre dos vértices no adyacentes cualesquiera de V, el grafo que se obtiene contiene un único ciclo.*

Demostración. Ver [9], teorema 3.1 (página 63). □

Definición 1.1.12. *Una **red** es una terna (V, E, c) formada por un grafo $G = (V, E)$ y una función, $c : E \longrightarrow \mathbb{R}$, llamada función de coste.*
*Dada una red (V, E, c) y un árbol generador $T = (V, E')$ se define el **coste** del árbol como*

$$c(T) = \sum_{e \in E'} c(e)$$

Definición 1.1.13. *Diremos que un árbol generador T de un grafo G es un **árbol generador de coste mínimo** si verifica que $c(T) \leq c(T')$ para cualquier árbol generador T'.*

Por tanto, el problema de buscar el árbol generador de coste mínimo de un grafo G es equivalente a resolver el siguiente problema:

$$\begin{array}{rl} min & c(T) \\ s.a & \text{T es un árbol generador de G} \end{array}$$

Veamos ahora el siguiente resultado:

Proposición 1.1.14. *Sea una red $G = (V, E, c)$ y un árbol generador de G, $T = (V, E')$. Las siguientes afirmaciones son equivalentes:*

1. *T es un árbol generador de coste mínimo.*

2. *Dada cualquier arista $a \in E \backslash E'$ y cualquier otra arista $e \in E'$ perteneciente al único ciclo del grafo $(V, E' \cup \{a\})$, se verifica que $c(e) \leq c(a)$.*

3. *Para cualquier arista $e \in E'$, $c(e) \leq c(a)$ para cada $a \in \omega(V_1^e)$ (donde este conjunto es el cociclo en el grafo G asociado a cualquier componente conexa de $(V, E' \backslash \{e\})$).*

Demostración. 1. $\Rightarrow$ 2. Supongamos que $T = (V, E')$ es un árbol generador de coste mínimo y supongamos que existe $a \in E \backslash E'$ tal que $c(a) < c(e)$ donde e pertenece al único ciclo de $(V, E' \cup \{a\})$. Pero entonces se podría retirar la arista e de este último grafo y obtener así un árbol generador de menor coste que T, de forma que se tiene una contradicción.

2. $\Rightarrow$ 3. Sea $e \in E'$. Retirando e desconectamos el árbol obteniendo dos componentes conexas. Sea ahora $a \in \omega(V_1^e)$. Añadiendo a obtenemos un árbol generador de G (en particular se tiene una cadena entre los extremos de e). Añadiendo de nuevo la arista e se forma un ciclo en el que está contenido la arista a, luego, por hipótesis, $c(e) \leq c(a)$.

3. $\Rightarrow$ 1. Sea $T^* = (V, E^*)$ un árbol generador de coste mínimo de G y supongamos que T es un árbol cumpliendo las condiciones de 3. Si $T = T^*$ entonces el resultado es obvio. Supongamos entonces que $T \neq T^*$. Entonces existe $e \in E' \backslash E^*$. Retirando dicha arista del árbol T se obtienen dos componentes conexas, que denotaremos V_1^e y $V \backslash V_1^e$. Añadiendo la arista e al árbol T^* se forma un ciclo y además existirá una cadena, que no contiene a e, uniendo un nodo de V_1^e y otro de $V \backslash V_1^e$, por lo que existirá una arista $a \in \omega(V_1^e)$. Pero entonces por hipótesis $c(a) \geq c(e)$. Ahora bien, T^* es un árbol generador de mínimo coste, por lo que $c(a) \leq c(e)$, de lo contrario eliminando a y añadiendo e obtendríamos un árbol generador de menor peso. De las desigualdades obtenidas se deduce que $c(a) = c(e)$. Ahora sustituyendo la arista a por la arista e en T obtenemos un árbol generador con el mismo peso que T y una arista más en común con T^*. Repitiendo el proceso, obtenemos una sucesión de árboles generadores del mismo coste hasta obtener T^*, por lo que $c(T) = c(T^*)$. □

El teorema anterior justifica la eficacia de los siguientes algoritmos para obtener un árbol generador de coste mínimo asociado a una red (V, E, c), con $|V| = n$:

Algoritmo de Kruskal

Etapa 1:

Ordenar las aristas de E en orden creciente atendiendo a su coste, es decir, tomar $E := (e_1, \cdots, e_m)$ de forma que $c(e_i) \leq c(e_m)$ $\forall i < m$ y considerar $T^* = (V, \emptyset)$.

Etapa 2:

Añadir aristas a T^* sucesivamente en orden creciente de coste sin formar ciclos. Parar al añadir $n-1$ aristas.

Algoritmo de Prim

Etapa 1:

Elegir un vértice $r \in V$ y hacer $V_1 = \{r\}$, $V_2 = V \backslash \{r\}$. Considerar $T^* = (V, \emptyset)$.

Etapa 2:

Añadir a T^* la arista de menor coste de $\omega(V_1)$, que denotaremos (v_1, v_2) con $v_1 \in V_1$ y $v_2 \in V_2$. Hacer $V_1 = V_1 \cup v_2$ y $V_2 = V_2 \backslash v_2$.

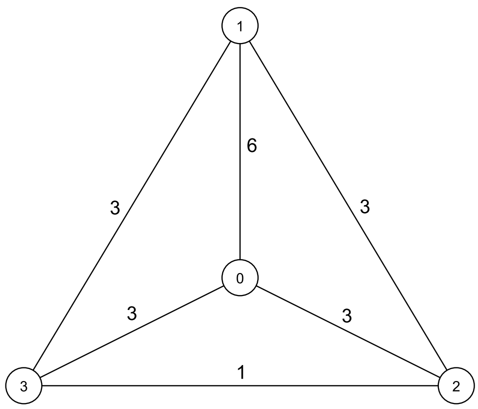
Etapa 3:

Si $|V_1| = n$ parar. Si no, volver a la Etapa 2.

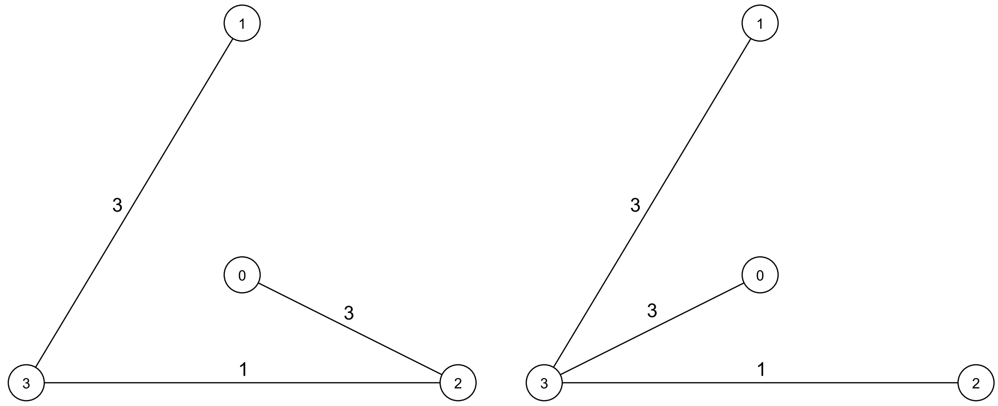
De este modo, dado una red (V, E, c), mediante la aplicación de cualquiera de estos algoritmos podemos obtener un árbol generador de mínimo coste asociado y así, el valor del coste de dicho de árbol.

Observación 1.1.15. *En general, una red $G = (V, E, c)$ no tiene un único árbol generador de coste mínimo.*

Ejemplo 1.1.16. *Para la red*



Los siguientes, son ambos árboles generadores de coste mínimo ($c = 7$):



1.2. Introducción a la teoría de juegos

En esta sección se llevará a cabo una introducción a la teoría de juegos, y más en particular, a los *juegos cooperativos de utilidad transferible*, así como una revisión de términos relacionados que estarán presentes a lo largo de toda la memoria.

El concepto de *juego cooperativo* hace referencia a que se consideran situaciones en las que intervienen un conjunto de individuos con capacidad para alcanzar acuerdos vinculantes y con voluntad para hacerlo. Por otro lado, por *utilidad transferible* se entiende que el valor en el que se cuantifica la cooperación de cualquier coalición de jugadores corresponde a una medida infinitamente divisible, que se puede repartir de cualquier manera entre los miembros de la coalición y cuya utilidad para ellos es directamente proporcional a su cantidad.

Definición 1.2.1. *Un* ***juego cooperativo de utilidad transferible*** *(o* ***juego TU****) es un par (N, v), donde:*

- $N = \{1, 2, \cdots n\}$ *es un conjunto de jugadores.*
- $v : 2^N \longrightarrow \mathbb{R}$ *es una función característica verificando $v(\emptyset) = 0$.*

Definición 1.2.2. *Llamaremos* ***coalición*** *a cualquier subconjunto del conjunto de jugadores N.*

Así, para cada coalición $S \subseteq N$, $v(S) \in \mathbb{R}$ representa la *utilidad* o *ganancia* que pueden asegurarse los miembros de S si deciden cooperar, independientemente de las actuaciones que lleven a cabo el resto de jugadores.

A partir de esta definición surge naturalmente otro tipo de juegos: los *juegos de coste*. Consideraremos juegos de coste en aquellas situaciones en las que, en lugar de un beneficio, exista un coste que deba ser repartido entre los jugadores. Así la definición de juego de coste vendrá dada como sigue:

Definición 1.2.3. *Un* ***juego de coste*** *es un par* (N, c)*, donde* N *denota el conjunto de jugadores y* $c : 2^N \longrightarrow \mathbb{R}$ *es una* ***función de coste*** *que asigna a cada coalición* $S \subseteq N$ *el coste mínimo,* $c(S)$*, que supondría para los integrantes de* S *colaborar para conseguir sus propios propósitos, considerando* $c(\emptyset) = 0$*. Denotaremos por* CG^N *al conjunto de juegos de coste en los que* N *es el conjunto de jugadores.*

Observación 1.2.4. *A lo largo de todo el texto consideraremos, salvo que se indique lo contrario, que el conjunto de jugadores vendrá dado por* $N = \{1, ..., n\}$*, por tanto, siempre se cumplirá que* $|N| = n$*.*

Por su naturaleza, los juegos de *árbol generador de coste mínimo* son juegos de coste, por lo que en lo sucesivo consideraremos siempre juegos de coste y la definición de los conceptos relacionados se llevará a cabo en función de este tipo de juegos. Lo que resta de sección será pues, una adaptación a los juegos de costes de la introducción a los juegos de beneficios llevada a cabo en [21].

Definición 1.2.5. *Se dice que un juego de coste* (N, c) *es* ***cóncavo*** *si para cada* $S, T \subseteq N$ *e* $i \in N$*, con* $S \subseteq T$ *e* $i \notin T$ *se verifica que*

$$c(S \cup \{i\}) - c(S) \geq c(T \cup \{i\}) - c(T).$$

A continuación pasaremos a introducir el concepto de *solución* de un juego de coste, aunque para ello será necesaria la definición de algunos conceptos previos:

Definición 1.2.6. *Dado un conjunto finito de jugadores* N*, un* ***vector de costes*** *para* N *es una función* $x : N \longrightarrow \mathbb{R}^N$*. Por lo tanto,* $\mathbb{R}^N$ *es el conjunto de vectores de costes para el conjunto de jugadores* N*. Así, dado* $x \in \mathbb{R}^N$*,* $x(i) = x_i$ *representará la cantidad que deberá pagar el jugador* i *según el vector de costes* x*.*

Cabe aclarar que dados $x \in \mathbb{R}^N$ y una coalición $S \subseteq N$, denotaremos por $x(S) = \sum_{i \in S} x_i$ al pago total realizado por los jugadores que conforman la coalición S y por $x^S \in \mathbb{R}^S$ a la restricción de x a la coalición S, es decir, $x^S = (x_i)_{i \in S}$.

Recordemos que $c(S)$ hace referencia al coste mínimo que supone para los integrantes de la coalición S colaborar para la consecución de sus objetivos sin tener en cuenta al resto de jugadores. Por ello, y dado que $c(N)$ es una cantidad que ha de ser pagada entre todos los jugadores parece lógico exigir a un vector de pagos que obligue a los jugadores en su conjunto a pagar una cantidad no inferior a $c(N)$. Surge así el concepto de vector de costes *factible.*

Definición 1.2.7. *Dado* $c \in CG^N$*, diremos que un vector* $x \in \mathbb{R}^N$ *es un vector de costes* ***factible*** *si verifica que* $c(N) \leq x(N)$*. Además denotaremos al conjunto de vectores de costes factibles como*

$$X^*(c) = \{x \in \mathbb{R}^N : c(N) \leq x(N)\}$$

Podemos, ahora sí, dar la definición de *solución* de un juego:

Definición 1.2.8. *Sea CG_0^N un subconjunto del conjunto de juegos de costes de N jugadores, es decir, $CG_0^N \subseteq CG^N$. Una **solución** sobre CG_0^N es una (multi)función σ que asocia a cada juego $c \in CG_0^N$ un subconjunto $\sigma(c)$ de $X^*(c)$.*

Como se ha comentado anteriormente, los individuos de un juego de coste tienen capacidad de alcanzar acuerdos y voluntad para hacerlo, por tanto, es de esperar que la suma total de las cantidades asignadas a cada jugador no supere el coste total, $c(N)$. Surge así la siguiente definición:

Definición 1.2.9. *Dado un juego (N, c), diremos que un vector de costes $x \in \mathbb{R}^N$ es **eficiente** si verifica que $x(N) = c(N)$. Denotaremos al conjunto de vectores eficientes como*

$$X(c) = \{x \in \mathbb{R}^N : x(N) = c(N)\}$$

Además de esta propiedad, parece lógico que un *buen* vector de costes asigna a cada jugador i una cantidad menor o igual al coste que supondría para el jugador i satisfacer sus necesidades por sí mismo, pues de los contrario no estaría dispuesto a cooperar:

Definición 1.2.10. *Dado un juego (N, c), diremos que un vector de costes $x \in R^N$ satisface el **principio de individualidad racional** si cumple que*

$$x_i \leq c(\{i\}) \text{ para cada } i \in N.$$

Definición 1.2.11. *Llamaremos **imputación** a un vector de costes eficiente que verifica el principio de individualidad racional. Denotaremos al conjunto de imputaciones como*

$$I(c) = \{x \in \mathbb{R}^N : x(N) = c(N) \text{ y } x_i \leq c(\{i\}), \text{ para todo } i \in N\}$$

A continuación se introducirá uno de los conceptos de solución más importante y estudiado la teoría de juegos: el *núcleo*.

Definición 1.2.12. *Sea $c \in CG^N$. Se define el **núcleo** de c como:*

$$C(c) = \{x \in \mathbb{R}^N : x(N) = c(N) \text{ y } x(S) \leq c(S), \text{ para todo } S \subseteq N\}$$

El núcleo está compuesto por los vectores que ninguna coalición de jugadores posible tiene motivos para rechazar, pues el coste asignado a la coalición es menor o igual que el coste que les supondría dejar de colaborar con el resto de jugadores para lograr sus objetivos por ellos mismos. De la definición se deduce que el núcleo es un poliedro acotado de $\mathbb{R}^N$.

Definición 1.2.13. *Un juego $c \in CG^N$ se dice **equilibrado** si $C(c) \neq \emptyset$.*

A continuación introduciremos el concepto de *valor de Shapley* y estudiaremos algunas de sus propiedades. Antes de la definición, necesitaremos unas nociones previas.

Definición 1.2.14. *Un* ***valor*** *sobre un conjunto de juegos* $CG_0^N \subseteq CG^N$ *es una función* $\psi : CG_0^N \longrightarrow \mathbb{R}^N$ *que para cada juego* $c \in CG_0^N$*, describe la cantidad que deberá pagar cada jugador.*

Definición 1.2.15. *Diremos que un valor* ψ *es* ***eficiente*** *si para cada* $c \in CG_0^N \subseteq CG^N$ *se verifica que*

$$\sum_{i\in N} \psi_i(c) = c(N)$$

Por tanto, un valor es eficiente si reparte entre los jugadores exactamente la cantidad a pagar.

Definición 1.2.16. *Se dice que un valor* ψ *es* ***aditivo*** *si para todo* $c_1, c_2 \in CG^N$ *se tiene que*

$$\psi(c_1 + c_2) = \psi(c_1) + \psi(c_2)$$

Definición 1.2.17. *Se dice que un valor* ψ *posee la propiedad del* ***jugador títere*** *si* $\psi_i(c) = c(\{i\})$ *para todo* $c \in CG_0^N \subseteq CG^N$ *y todo* $i \in N$ *que verifique que*

$$c\,(S \cup \{i\}) - c\,(S) = c(\{i\}) \text{ para cada } S \subseteq N \backslash \{i\}.$$

Así, un valor satisface la propiedad del jugador títere si, en el caso de que exista algún jugador i cuya adhesión a cualquier coalición mantiene el coste de la coalición, el valor asocia a dicho jugador la cantidad $c(\{i\})$.

Definición 1.2.18. *Se dice que un valor* ψ *posee la propiedad del* ***anonimato*** *si para cada juego* $c \in CG_0^N \subseteq CG^N$ *y cada permutación* $\theta : N \longrightarrow N$*, con* $\theta c \in CG_0^N$*, se tiene que* $\psi_{\theta(i)}(\theta c) = \psi_i(c)$ *para cada* $i \in N$*, donde* θc *es la función de coste dada por*

$$(\theta c)(\theta S) = c(S) \text{ para todo } S \subseteq N.$$

Así, un valor satisface la propiedad del anonimato si el coste asignado a los jugadores no depende del nombre de estos. Es decir, si los jugadores tienen el mismo papel en los juegos, pese a aparecer numerados de forma distinta, entonces la cantidad asignada por el valor para los jugadores con rol análogo es la misma.

Teorema 1.2.19 (Shapley, 1953)**.** *Existe un único valor* ψ *sobre* CG^N *que es eficiente, aditivo y posee las propiedades del jugador títere y del anonimato.*

Demostración. Ver [11], sección 2 y [23]. □

Definición 1.2.20. *Llamaremos* ***valor de Shapley*** *y lo denotaremos como* Sh*, al valor caracterizado por el teorema anterior.*

Para cada juego $c \in CG^N$, podemos expresar el valor de Shapley de la siguiente forma:

$$Sh_i(c) = \sum_{T \subseteq N \backslash \{i\}} \frac{|T|!(|N| - |T| - 1)!}{|N|!} \left(c(T \cup \{i\}) - c(T)\right), \text{ para cada } i \in N.$$

Veamos ahora el siguiente concepto:

Definición 1.2.21. *Dado un juego $c \in CG^N$, $i \in N$ y $S \subseteq N \backslash \{i\}$, se llama* ***contribución marginal*** *del jugador i a la coalición S a la cantidad $c(S \cup \{i\}) - c(S)$.*

Planteemos una situación en la que los participantes del juego acceden a un centro de asignación de costes en un determinado orden. El coste asignado a un individuo, con respecto a este orden, es su contribución marginal a la coalición formada por aquellos jugadores que llegaron antes que él, es decir, sus predecesores. Una opción para llegar a un reparto considerado como justo mediante este proceso, es considerar todos los órdenes de llegada con la misma probabilidad. En este caso, el valor esperado de esta asignación de costes es precisamente el valor de Shapley. Observemos que en la expresión para el valor de Shapley dada anteriormente, fijo un $i \in N$, cada sumando consiste en la contribución marginal de dicho jugador sobre una coalición T, multiplicada por un factor que indica la proporción de permutaciones, entre las $|N|!$ posibles formas de ordenar los individuos de N, en las cuales los miembros de T llegan justo antes que el jugador i.

Definición 1.2.22. *Sea $c \in CG^N$ y $\pi \in \Pi_N$, siendo Π_N el conjunto de todas las permutaciones del conjunto de jugadores. El* ***vector de costes marginales*** *de c asociado a π, m_c^π, se define como:*

$$m_c^\pi(i) = c(P_\pi(i) \cup \{i\}) - c(P_\pi(i)) \text{ para todo } i \in N,$$

donde $P_\pi(i) = \{j \in N : \pi(j) < \pi(i)\}$ denota el conjunto de predecesores de i respecto de la permutación π.

Así, el valor de Shapley será el valor promedio de estos vectores de contribuciones marginales, es decir:

$$Sh(c) = \frac{1}{|N|!} \sum_{\pi \in \Pi_N} m_c^\pi$$

Dedicaremos esta última parte de la revisión de la teoría de juegos a la introducción del *conjunto de Weber*.

Definición 1.2.23. *Sea $c \in CG^N$. Se define el* ***conjunto de Weber*** *de c como la envolvente convexa de los vectores de coste marginales:*

$$W(c) = Conv\{m_c^\pi : \pi \in \Pi_N\}$$

Es claro que el valor de Shapley pertenece al conjunto de Weber, pues es una combinación convexa de vectores de costes marginales. Un importante resultado es que, además, el conjunto de Weber de un juego de coste contiene al núcleo de dicho juego.

Teorema 1.2.24. *(Weber, 1978) Para todo* $c \in CG^N$*,* $C(c) \subseteq W(c)$*.*

Demostración. Ver [25], teorema 14, página 29. □

En general, la inclusión recíproca no se verfica para juegos de coste arbitrarios, sin embargo, sí que que se verifica para la clase de los juegos cóncavos. De hecho, esta propiedad caracteriza esta clase de juegos.

Teorema 1.2.25. *(Shapley, 1971; Ichiishi, 1981) Un juego de coste* $c \in CG^N$ *es cóncavo si y solo si* $C(c) = W(c)$*.*

La demostración de la implicación directa fue realizada por L.S. Shapley y puede verse en [24] (teoremas 3 y 5). Por su parte, la implicación recíproca fue realizada por T. Ichiisi y se puede encontrar en [17] (secciones 2 y 3).

1.3. Juegos de árbol generador de coste mínimo

Consideremos una situación en la que una serie de usuarios o agentes quieren conectarse a un determinado servicio, que es provisto por una única fuente. Para ello deben establacer enlaces que conecten, directa o indirectamente, a cada uno de los usuarios con el servicio. El establecimiento de estas conexiones acarreará un coste, que consideraremos siempre mayor que 0, por lo que los usuarios estarán interesados en determinar la estructura de menor coste que los conecte a todos con el servicio, es decir, el árbol generador de mínimo coste asociado a la red. A un problema de este tipo se le denomina **problema de árbol generador de coste mínimo**. En lo sucesivo nos referiremos a este tipo de problemas como **problemas *mcst*** (*Minimum Cost Spanning Tree*) y serán denotados como (N_0, C). Además, denotaremos como 0 al nodo que proporciona el servicio, al que denominaremos **fuente**, $N = \{1, ..., n\}$ al conjunto de usuarios, $N_0 := N \cup \{0\}$ y C a la matriz que contiene los costes de establecer las conexiones entre los distintos jugadores y de estos con la fuente. Convendremos siempre que $c_{ij} = c_{ji}$ para cualesquiera $i, j \in N_0$, donde c_{ij} es el coste de construir la línea (i, j) y se considera $c_{ii} = 0$ para cada $i \in N_0$. Por último, si consideramos un problema *mcst* (N_0, C), el coste de un árbol generador de mínimo coste de la red considerada será denotado como $c_{mcst}(N_0, C)$, o bien, siempre que no cause confusión, $c_{mcst}(N_0)$.

A partir de un problema de árbol generador de coste mínimo surge naturalmente un juego de coste. Dado un problema *mcst* (N_0, C), consideremos la función de coste c dada por

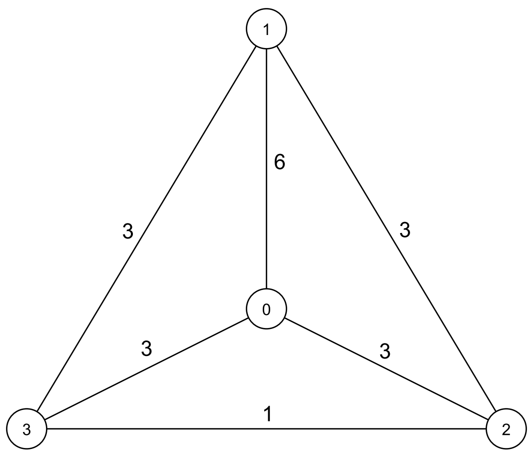
$$c(S) = \sum_{(i,j) \in E_{T_S}} c_{ij} \text{ para cada } S \subseteq N$$

donde E_{T_S} es el conjunto de aristas de un árbol generador de coste mínimo en el grafo completo $G_S = (S_0, E_S)$, siendo $S_0 := S \cup \{0\}$, es decir, $c(S)$ es el coste de conectar a los agentes de S a la fuente sin hacer uso de las conexiones de los agentes de $N \backslash S$. De este modo, a partir del problema (N_0, C), considerando la función de coste c, se obtiene un juego de coste (N, c). A este tipo de juegos se les denomina **juegos de árbol generador de coste mínimo**, o bien, ***juegos mcst***.

Dado un problema *mcst* (N_0, C) y una coalición $S \subseteq N$, nos referiremos al problema restringido a la coalición S como (S_0, C) y al juego de coste que se obtiene de dicho problema como $(S, c_{|S})$, o directamente (S, c) si no hay confusión. Por último, consideraremos siempre, salvo que se específique lo contrario, que el nodo 0 es la raíz de todos los árboles generadores de coste mínimo considerados.

Mostremos un primer ejemplo introductorio:

Ejemplo 1.3.1. *Juegos mcst.*
Supongamos que tres usuarios quieren conectarse a un servicio que es provisto desde una única fuente. Las posibles conexiones que pueden establecerse, y sus costes, son las siguientes:



Por tanto, los costes asociados a cada coalición vendrán dados por:

S	$c(S)$	S	$c(S)$	S	$c(S)$
$\{1\}$	6	$\{1,2\}$	6	$\{1,2,3\}$	7
$\{2\}$	3	$\{1,3\}$	6	$\emptyset$	0
$\{3\}$	3	$\{2,3\}$	4		

El objetivo de este texto de aquí en adelante será mostrar las propiedades generales de los juegos *mcst*, así como presentar reglas de reparto de costes entre los usuarios, estudiar las propiedades de estas reglas y establecer una comparación entre las mismas.

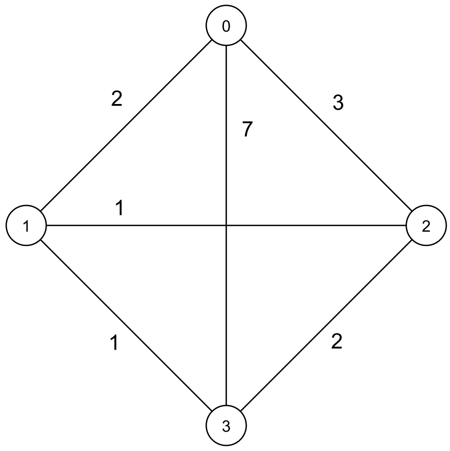
Comenzaremos el estudio de las propiedades de los juegos *mcst* con el estudio de la estructura del núcleo de este tipo de juegos. Como se ha comentado en la sección anterior, los vectores de costes pertenecientes al núcleo son vectores eficientes y ninguna coalición posible se encontraría en una mejor posición si decidiese dejar de colaborar con el resto de usuarios para conectarse al servicio. El núcleo de un juego arbitrario puede ser vacío, sin embargo, se demostrará que, en el caso de juegos *mcst*, existen siempre vectores de costes pertenecientes al núcleo. Varios autores han demostrado la no vacuidad del núcleo para juegos *mcst*. A continuación se mostrará la prueba realizada por Granot y Huberman [15]. Para ello será necesario introducir algunos conceptos previos y mostrar algunos resultados auxiliares:

Definición 1.3.2. *Sea (N, c) un juego mcst. Definiremos el* ***juego de árbol generador de coste mínimo monótono****, o juego* ***mmcst****, derivado de (N, c), al juego dado por $(N, \overline{c})$ donde:*

$$\overline{c}(S) = min\{c(T) : S \subseteq T \subseteq N\} \text{ para cada } S \subseteq N.$$

Así, dado un juego *mcst* (N, c), $\overline{c}(S)$ se obtiene al considerar el coste mínimo de entre todos los costes asociados a cualquier coalición en la que se encuetre contenida S. Directamente de la definición se tiene que $\overline{c}(S) \leq c(S)$ para todo $S \subseteq N$. Además, se verifica que $\overline{c}(S) \leq \overline{c}(T)$ para todo $S \subseteq T \subseteq N$, es decir, $\overline{c}$ es una función de conjuntos monótona creciente.

Ejemplo 1.3.3. *Consideremos el siguiente problema:*



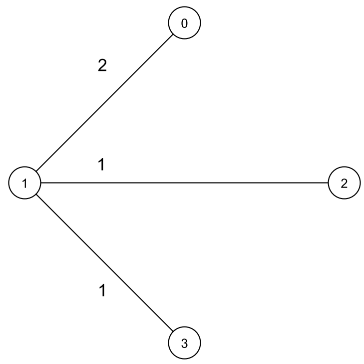
Entonces, las funciones de coste para el juego mcst (N, c) y el juego mmcst $(N, \overline{c})$ vendrán dadas por:

S	$c(S)$	$\overline{c}(S)$	S	$c(S)$	$\overline{c}(S)$	S	$c(S)$	$\overline{c}(S)$
$\{1\}$	2	2	$\{1,2\}$	3	3	$\{1,2,3\}$	4	4
$\{2\}$	3	3	$\{1,3\}$	3	3	$\emptyset$	0	0
$\{3\}$	7	3	$\{2,3\}$	5	4			

Definición 1.3.4. *Diremos que un vector $(x_1, ..., x_n)$ es una* ***solución de árbol de coste mínimo*** *del problema mcst (N_0, C), o bien, del juego mcst (N, c), si existe algún árbol generador*

de coste mínimo $T = (V, E_T)$ *asociado a* (N_0, C) *para el cual, denotando por* $e^i \in E_T$ *a la arista incidente en el nodo* i *en el único camino que conecta* i *con* 0 *en* T*, se verifica que* $(x_1, ..., x_n) = (e^1, ..., e^n)$.

Ejemplo 1.3.5. *Para el problema dado en el ejemplo 1.3.3, el árbol generador de coste mínimo vendrá dado por*



Por tanto, el vector $x = (2, 1, 1)$ *será una solución de árbol de coste mínimo asociada al problema.*

A continuación, llevaremos a cabo la definición de dos conjuntos que serán considerados en los resultados posteriores. Sea (N_0, C) un problema *mcst*, (N, c) el juego derivado de (N_0, C) y sea $T = (N_0, E_T)$ un árbol generador de coste mínimo asociado. Denotemos por $e^i \in E_T$, con $i = 1, ..., n$, a la arista incidente en el nodo i en el único camino que conecta i con 0 en T. Sea $S \subseteq N$, tomaremos:

- $X_S := \{e^i : i \in S\}$.
- $\overline{X}_S := E_T \backslash X_S$.

Por otro lado, dado un conjunto de aristas $B \subseteq E_T$, definiremos $m(B) := \sum_{(i,j) \in B} c_{ij}$. Observemos que de las definiciones anteriores se obtiene claramente que

$$c(N) = m(X_S) + m(\overline{X}_S).$$

Lema 1.3.6. *Consideremos un problema mcst* (N_0, C)*. Tomemos* $S \subseteq N$ *y el problema mcst* (S_0, C)*. Sea* $T_S = (S_0, E_{T_S})$ *un árbol generador de coste mínimo asociado a dicho problema. Entonces los conjuntos* E_{T_S} *y* $\overline{X}_S$ *son disjuntos.*

Demostración. El resultado se deduce de la definición de ambos conjuntos pues si $(i, j) \in \overline{X}_S$, entonces o bien $i \in N \backslash S$, o bien $j \in N \backslash S$, mientras que si $(i, j) \in E_{T_S}$, entonces $i \in S \cup \{0\}$ y $j \in S \cup \{0\}$. □

Consideremos ahora el grafo $Y_S = (N_0, E(S))$, donde $E(S) = E_{T_S} \cup \overline{X}_S$. Observando que $|X_S| = |E_{T_S}|$ y aplicando el lema anterior se tiene que $|E(S)| = |N_0| - 1 = n$, si vemos que Y_S es conexo, aplicando el teorema 1.1.14 tendremos que es un árbol.

Lema 1.3.7. *Para cada $S \subseteq N$ el grafo $Y_S = (N_0, E(S))$ es un árbol.*

Demostración. Por la definición de $E(S)$, para ver que Y_S es conexo es suficiente ver que para cada $v_1 \in N_0$ existe una cadena en $\overline{X}_S$ que conecta el nodo v_1 con algún usuario de S_0. Si $v_1 \in S_0$, entonces el resultado es obvio. En caso contrario, consideremos $A_1 = \{e^{v_1}, ...e^{v_r}\}$ la única cadena en E_T que conecta v_1 con el nodo 0, donde e^{v_i} denota la arista (v_i, v_{i+1}). Si $A_1 \subseteq \overline{X}_S$, entonces se tiene el resultado pues $0 \in S_0$. En caso contrario sea e^{v_k} la primera arista en A_1 no perteneciente a $\overline{X}_S$. Entonces $e^{v_k} \in E_T \backslash \overline{X}_S$, por lo que se tiene que $v_k \in S \subseteq S_0$. Así, la cadena $\{e^{v_1}, ...e^{v_{k-1}}\} \subseteq \overline{X}_S$ conecta v_1 con S_0. $\square$

Estamos, ahora sí, en condiciones de demostrar el resultado que buscábamos:

Teorema 1.3.8. *Dado un problema mcst (N_0, C), toda solución de árbol de coste mínimo pertenece al núcleo del juego mcst (N, c) y al núcleo del juego mmcst $(N, \overline{c})$.*

Demostración. Sea $x = (x_1, ..., x_n)$ la solución de árbol de coste mínimo correspondiente a un árbol generador de coste mínimo T asociado a (N_0, C). Como se verifica la desigualdad $\overline{c}(S) \leq c(S)$ para todo $S \subseteq N$, el núcleo del juego *mcst* (N, c) contiene al núcleo del juego *mmcst* $(N, \overline{c})$, por lo que es suficiente ver que x pertenece al núcleo de $(N, \overline{c})$. Por construcción del vector x se tiene que $\sum_{i \in N} x_i = \overline{c}(N)$.

Supongamos ahora por reducción al absurdo que existe una coalición $R \subseteq N$ tal que

$$\sum_{i \in R} x_i > \overline{c}(R).$$

Sea $S \subseteq N$ y $R \subseteq S$ tal que

$$\overline{c}(R) = c(S)$$

Como $x_i \geq 0$, se tiene que

$$\sum_{i \in S} x_i > c(S)$$

o equivalentemente que

$$m(X_S) > c(S).$$

Recordemos ahora que se tiene que $c(N) = m(X_S) + m(\overline{X}_S)$ y que habíamos definido $E(S) = E_{T_S} \cup \overline{X}_S$, donde E_{T_S} es el conjunto de aristas de un árbol generador de coste mínimo asociado al problema (S_0, C). Por tanto,

$$c(N) = m(X_S) + m(\overline{X}_S) > c(S) + m(\overline{X}_S) = m(E(S))$$

Ahora bien, aplicando el lema 1.3.7, $Y_S = (V_N, E(S))$ es un árbol, lo que contradice que T sea un árbol generador de coste mínimo de la red, por lo que

$$\sum_{i \in S} x_i \leq \overline{c}(S) \text{ para todo } S \subseteq N$$

luego x pertenece al núcleo de $(N, \overline{c})$ y en consecuencia al de (N, c). □

Obserevemos que del teorema anterior se deduce que para obtener un vector de costes en el núcleo de un juego (N, c), no se necesita calcular la función de coste c, sino que basta con obtener un árbol generador de coste mínimo.

Ya vimos en la sección anterior que el núcleo de un juego arbitrario (N, c) es un poliedro en $\mathbb{R}^N$. Con el resultado siguiente iremos más allá y veremos que una solución de árbol de coste mínimo no sólo pertenece al núcleo, sino que además, es un vértice del mismo.

Teorema 1.3.9. *Dado un problema mcst (N_0, C), toda solución de árbol de coste mínimo es un vértice del núcleo del juego mcst (N, c) y del núcleo del juego mmcst $(N, \overline{c})$.*

Demostración. Por definición de $\overline{c}$, el núcleo del juego *mcst* (N, c) contiene al núcleo del juego *mmcst* $(N, \overline{c})$, por lo que para probar el resultado bastará hacerlo para el juego (N, c). Sea $x = (x_1, ..., x_n)$ la solución de árbol de coste mínimo correspondiente a un árbol generador de coste mínimo T asociado a (N_0, C). Etiquetemos los nodos $\{1, ..., n\}$ de T de forma que si $i > j$, entonces i no está en el único camino en T que conecta el nodo j con el nodo 0. Tenemos que ver que entre las desigualdades

$$\sum_{i \in S} x_i \leq c(S) \text{ para todo } S \subseteq N$$

existen n desigualdades linealmente independientes que son satisfechas en igualdad por el vector x. Pero entonces, las desigualdades correspondientes a los conjuntos $\{0, ..., i\}$ con $i = 1, ..., n$, son desigualdades que satisfacen esta condición.

□

Pese a ser vectores de costes pertenecientes al núcleo, uno de los inconvenientes de las soluciones de árbol de coste mínimo que señalan Granot y Huberman [16], es que perjudica a los jugadores que se encuentran más próximos al nodo que proporciona el servicio, entendiendo la proximidad en función del número de conexiones intermedias entre la fuente y el usuario. Ello se debe a que estos pagan el coste total de las aristas que les conectan al servicio, cuando existen jugadores más alejados a la fuente que utilizan dichas dichas conexiones para obtener el servicio. Veamos un ejemplo que muestre este hecho:

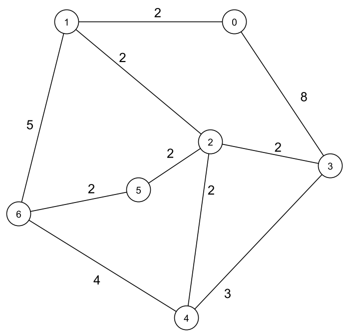
Ejemplo 1.3.10. *Consideremos el problema mcst (N_0, C), donde $N = \{1, 2, 3\}$ y la matriz de costes es la siguiente:*

$$C = \begin{pmatrix} 0 & 100 & 200 & 200 \\ 100 & 0 & 1 & 30 \\ 200 & 1 & 0 & 2 \\ 200 & 30 & 2 & 0 \end{pmatrix}$$

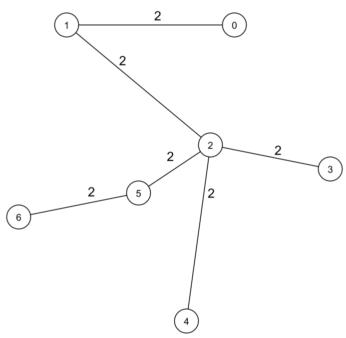
El árbol generador de coste mínimo asociado al problema vendrá dado por las aristas $E_T = \{(0,1),(1,2),(2,3)\}$*, por tanto una solución de árbol de coste mínimo será* $(100,1,2)$*. Como vemos, el jugador 1 deberá pagar el coste completo de la conexión* $(0,1)$ *cuando los jugadores 2 y 3 están utilizando también dicha conexión, siendo además el jugador que tiene la conexión directa con el nodo fuente más barata.*

Este hecho sirve como motivación para encontrar otros vectores de costes pertenecientes al núcleo. A continuación mostraremos el método que introducen Granot y Huberman [16] para generar vectores en el núcleo a partir de soluciones de árbol mínimo. Mostremos en primer lugar un ejemplo introductorio:

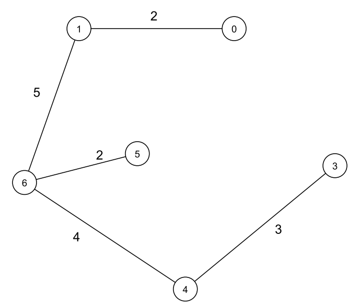
Ejemplo 1.3.11. *Consideremos el siguiente problema mcst:*



donde las aristas no dibujadas pueden considerarse de coste infinito. Así, un árbol generador de coste mínimo será:



por lo que el vector de costes $L = (2,2,2,2,2,2)$ *es una solución de árbol de coste mínimo. Supongamos ahora que no existiese el usuario 1. Entonces, el resto de usuarios tendrían que hacer frente a un coste de 8 unidades, en lugar de 2, para conectarse al nodo 0. Supongamos que el usuario 1 exige que este aumento en el coste de 6 unidades para la coalición* $S = \{2,3,4,5,6\}$ *debe abonársele a él y debe hacerlo su inmediato sucesor, es decir, el usuario 2. Así, el vector de asignación de costes obtenido sería* $(-4,8,2,2,2,2)$*. Supongamos que el usuario 2 sigue la* *misma actuación y observa que las aristas* $(2,3)$, $(2,4)$ *y* $(2,5)$ *son usadas por sus sucesores para conectarse al servicio. Si dicho usuario deja de cooperar, se tendría el siguiente árbol generador de mínimo coste*



Entonces los usuarios del subárbol enraizado en 3 (que contiene únicamente al usuario 3), deberán pagar 3 unidades de coste, en lugar de 2, para conectarse al servicio. Los usuarios del subárbol enraizado en 4 (que contiene solamente al usuario 4), deberán pagar 4 unidades en lugar de 2. Por último, los usuarios del subárbol enraizado en 5 (que contiene a los usuarios 5 y 6), deberán pagar 5 unidades de coste, en lugar de 2, por la arista que les conecta como conjunto al servicio. Así, la no colaboración del usuario 2 supondría un aumento del coste para sus sucesores de 6 unidades. Si dicho usuario exige que sus inmediatos sucesores le abonen dicha cantidad el vector de costes obtenido será $(-4, 2, 3, 4, 5, 2)$. *Por último, si el usuario 5 actúa de la misma forma con el usuario 6, el vector de costes obtenido será* $(-4, 2, 3, 4, 3, 4)$. *Es sencillo comprobar que los tres vectores obtenidos pertenecen al núcleo.*

A continuación, formalizaremos las ideas del ejemplo anterior:
Consideremos un problema *mcst* (N_0, C) y $T = (N_0, E_T)$ un árbol generador de coste mínimo asociado. Para un usuario $i \in N$, consideremos un árbol generador de coste mínimo $T_{N_0\backslash\{i\}} = (N_0\backslash\{i\}, E_{N_0\backslash\{i\}})$ asociado al problema $(N_0\backslash\{i\}, C)$, en el que los subárboles de T enraizados en k, $T_k = (N_k, E_k)$ verifican $E_k \subseteq E_{N_0\backslash\{i\}}$, donde k es un sucesor inmediato de i en T, N_k el conjunto de usuarios siguiente

$$N_k = \{k\} \cup \{s \in N : k \text{ es predecesor de } s \text{ en } T\}$$

y $E_k = \{(i,j) : i \in N_k, j \in N_k, (i,j) \in E_{N_0\backslash\{i\}}\}$. Denotemos por $F(i)$ a los inmediatos sucesores de i en T. Para cada $k \in F(i)$ existe una arista $(r,q) \in E_{N_0\backslash\{i\}}$ tal que $r \in N_k$ y q pertenece al único camino entre todos los nodos de N_k al nodo 0 en $T_{N_0\backslash\{i\}}$. Denotemos el coste de dicha arista por c_k. Una **operación de demanda débil** (del inglés *weak demand operation*) por el usuario i es una transformación que dado un vector de asignación de costes y proporciona un vector $wd^i(y)$, dado por:

$$wd_r^i(y) = \begin{cases} c_k - \left(\displaystyle\sum_{(u,v)\in E_k} c_{uv} - \sum_{j\in N_k\setminus\{k\}} y_j \right) & \text{si } r = k, k \in F(i), \\ y_i - \displaystyle\sum_{k\in F(i)} (wd_k^i(y) - y_k) & \text{si } r = i, \\ y_r & \text{en otro caso.} \end{cases}$$

Cabe observar que si $F(i) = \emptyset$, entonces $wd^i(y) = y$. La definición de operación de demanda débil puede extenderse a una transformación dada por un subconjunto $Q \subseteq N$. Así, tomando $i \in Q$, podemos definir $wd^Q(y)$, prodediendo de forma recursiva considerando

$$wd^Q(y) = wd^i(wd^{Q\setminus\{i\}}(y)).$$

Si $Q = \emptyset$, entonces $wd^Q(y) = y$.

Veamos ahora el siguiente resultado que establece que, considerando un usuario i, el vector obtenido de aplicar una operación de demanda débil sobre una solución de árbol de coste mínimo pertenece al núcleo.

Proposición 1.3.12. *Sea (N_0, C) un problema mcst y sea (N, c) el juego de coste asociado. Sea $T = (N_0, E_T)$ un árbol generador de coste mínimo y $L = (l_1, ..., l_n)$ la solución de árbol de coste mínimo correspondiente. Entonces para cada $i \in N$ el vector $wd^i(L)$ pertenece al núcleo de (N, c).*

Demostración. Sabemos por el teorema 1.3.8 que L pertenece al núcleo de (N, c). Además, para cada coalición $S \subseteq N$, se tiene que $\sum_{r\in S} wd_r^i(L) \le \sum_{j\in S} l_j$ si $i \in S$, teniendo en cuenta la eficiencia de los vectores y que $l_i - wd_i^i(L) = \sum_{k\in F(i)}(wd_k^i(L) - l_k)$. Por tanto, es suficiente probar que $\sum_{r\in S} wd_r^i(L) \le c(S)$ si $i \notin S$. Supongamos por reducción al absurdo que $\sum_{r\in S} wd_r^i(L) > c(S)$ con $i \notin S$. Entonces

$$\sum_{r\in N\setminus\{i\}} wd_r^i(L) > c(S) + \sum_{r\in N\setminus(S\cup\{i\})} wd_r^i(L)$$

Por la definición de L y $wd^i(L)$, se tiene que $\sum_{r\in N\setminus\{i\}} wd_r^i(L)$ es el coste de un árbol generador de coste mínimo $T' = (N_0\setminus\{i\}, E_{N\setminus\{i\}})$ asociado al problema $(N_0\setminus\{i\}, C)$. Pero además, $c(S) + \displaystyle\sum_{r\in N\setminus(S\cup\{i\})} wd_r^i(L)$ es el coste de un árbol generador asociado al mismo problema. Pero entonces

$$\sum_{r\in N\setminus\{i\}} wd_r^i(L) > c(S) + \sum_{r\in N\setminus(S\cup\{i\})} wd_r^i(L)$$

contradice que T' sea un árbol generador de coste mínimo, con lo que se obtiene el resultado.

$\square$

El siguiente teorema muestra que el vector resultante de la aplicación de una operación de demanda débil sobre una solución de árbol de coste mínimo considerando una coalición de usuarios determinada también pertenece al núcleo.

Proposición 1.3.13. *Sea (N_0, C) un problema mcst y sea (N, c) el juego de coste asociado. Sea $T = (N_0, E_T)$ un árbol generador de coste mínimo asociado y $L = (l_1, ..., l_n)$ la solución de árbol de coste mínimo correspondiente. Entonces para cada $Q \subseteq N$ el vector $wd^Q(L)$ pertenece al núcleo de (N, c).*

Demostración. Ver [16], teorema 3, página 328. □

Observación 1.3.14. *El teorema anterior no es cierto, en general, si en lugar de una solución de árbol de coste mínimo L, consideramos un vector del núcleo arbitario como puede observarse en [16], ejemplo 2, página 329.*

1.4. Formas irreducibles

En la presente sección se estudiará una clase distinguida dentro de los problemas *mcst*, las *formas irreducibles*. Las formas irreducibles surgen de modificar el coste de las conexiones en un problema *mcst* dado. En esta sección veremos cómo se construye la forma irreducible de un determinado problema, estudiaremos sus propiedades y se introducirá el concepto de *núcleo irreducible* de un juego *mcst*. La sección terminará con la demostración de un importante resultado acerca de la estructura del núcleo irreducible.

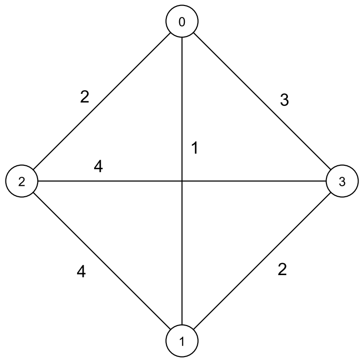
Definición 1.4.1. *Sea (N_0, C) un problema mcst, sea T un árbol generador de coste mínimo asociado y denotemos por E_T al conjunto de aristas de dicho árbol. Definiremos el problema (N_0, C^T) tomando:*

$$c_{ij}^T = \begin{cases} c_{ij} & si\ (i,j) \in E_T \\ max\{c_{hl} : (h,l) \text{ es una arista en el único camino entre } i \text{ y } j \text{ en } T\} & en\ otro\ caso\ . \end{cases}$$

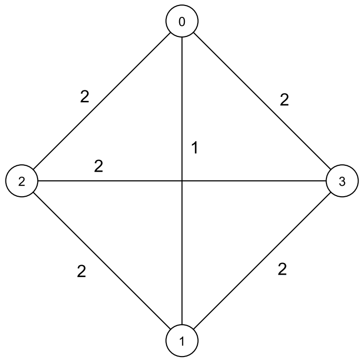
Además, denotaremos como hasta ahora (N, c) al juego derivado del problema (N_0, C) y (N, c^T) al juego obtenido a partir del problema (N_0, C^T).

Observación 1.4.2. *Sea (N_0, C) un problema mcst y T un árbol generador de coste mínimo asociado. Denotemos como E al conjunto de todas las conexiones posibles entre los componentes de N_0. De la anterior definición se tiene que $c_{ij}^T \leq c_{ij}$ para todo $(i,j) \in E$. En efecto, si $(i,j) \in E_T$ entonces se tiene la igualdad. En caso contrario, si $(i,j) \notin E_T$ y no se verificase la desigualdad anterior, se tendría una arista (h,l) en el único camino en T que conecta i y j tal* *que $c_{hl} > c_{ij}$, pero entonces sustituyendo dicha arista por (i,j) obtendríamos un árbol generador de menor coste, lo cual está en contradicción con que T sea un árbol generador de coste mínimo. De la misma forma se obtiene que T es también un árbol generador de coste mínimo asociado al problema (N_0, C^T).*

Ejemplo 1.4.3. *Consideremos el problema (N_0, C) dado por:*



Un árbol generador de mínimo coste T, asociado al problema vendrá dado por las aristas $\{(0,1),(0,2),(1,3)\}$. Aplicando entonces el procedimiento descrito en la definición anterior se obtiene el problema (N_0, C^T) dado por:



Proposición 1.4.4. *Sea un (N_0, C) un problema mcst y sea T un árbol generador de coste mínimo asociado. Consideremos el problema (N_0, C^T) obtenido mediante el procedimiento de la definición 1.4.1. Supongamos que existe otro árbol generador de coste mínimo T' para (N_0, C). Entonces T' es también un árbol generador de coste mínimo para (N_0, C^T) y $c_{ij}^T = c_{ij}^{T'}$ para toda arista $(i,j) \in E$.*

Demostración. Sea T' un árbol generador de coste mínimo asociado al problema (N_0, C). Entonces

$$\sum_{(i,j)\in E_{T'}} c_{ij}^T \leq \sum_{(i,j)\in E_{T'}} c_{ij} = \sum_{(i,j)\in E_T} c_{ij} = \sum_{(i,j)\in E_T} c_{ij}^T$$

donde se ha aplicado la observación 1.4.2 para la primera desigualdad. Ahora bien, como T es también un árbol generador de coste mínimo para (N_0, C^T) se tiene que la primera desigualdad es en realidad una igualdad y así, que T' es un árbol generador de coste mínimo para (N_0, C^T). Por tanto, $c_{ij}^T = c_{ij}$ para toda arista $(i,j) \in E_{T'}$. Ahora, aplicando el procedimiento descrito en la definición 1.4.1 con el árbol T' se obtiene que

$$c_{ij}^{T'} = \left(c^T\right)_{ij}^{T'} \leq c_{ij}^T \text{ para toda arista } (i,j) \in E.$$

Análogamente, cambiando los papeles de T y T' se obtiene

$$c_{ij}^T \leq c_{ij}^{T'} \text{ para toda arista } (i,j) \in E.$$

Por tanto $c_{ij}^T = c_{ij}^{T'}$ para toda $(i,j) \in E$. □

De la proposición anterior se sigue que el procedimiento aplicado en la definición 1.4.1 no depende del árbol generador de coste mínimo, ya que en cualquier caso se obtienen los mismos costes para las distintas conexiones. Por este motivo, en lo sucesivo utilizaremos la notación (N_0, C^*) y (N, c^*), en lugar de la empleada hasta ahora, (N_0, C^T) y (N, c^T), que hace referencia al árbol generador de coste mínimo T tomado.

Definición 1.4.5. *Sea un problema mcst (N_0, C) y sea (N, c) el juego derivado de dicho problema. Al problema obtenido de aplicar el procedimiento de la definición 1.4.1, se le denominará **forma irreducible** de (N_0, C) y se denotará (N_0, C^*).*

*De igual forma, al juego derivado de (N_0, C^*) se llamará **forma irreducible** de (N, c) y se denotará (N, c^*).*

*Diremos que (N_0, C^*) (respectivamente (N, c^*)) es una **forma irreducible**, si es la forma irreducible de algún problema mcst (N_0, C) (respectivamente de algún juego (N, c)).*

*Por último, si (N_0, C^*) es una forma irreducible, diremos que la matriz C^* es una **matriz irreducible** .*

La siguiente proposición servirá para caracterizar las formas irreducibles:

Proposición 1.4.6. *Un problema (N_0, C^*) es una forma irreducible si y solo si al reducir el coste de un arista cualquiera, el coste del árbol (o árboles) generador de coste mínimo asociado se reduce estrictamente.*

Demostración. $\Rightarrow$ Supongamos que reducimos el coste de una conexión (i,j) y sea $T = (N_0, E_T)$ un árbol generador de coste mínimo asociado. Si $(i,j) \in E_T$ entonces el resultado es claro. Supongamos que $(i,j) \notin E_T$ y que el coste del árbol generador de coste mínimo no se reduce. Añadiendo la arista (i,j) a T se obtiene un único ciclo C. Si el coste del árbol generador de mínimo coste no se reduce estrictamente, por la proposición 1.1.14, debe cumplirse que

$$c_{ij} > c_{lk} \text{ para cualquier } (l,k) \in C$$

lo que es incompatible con que (N_0, C^*) sea irreducible por la definición 1.4.1.

$\Leftarrow$ Consideremos un problema (N_0, C^*) tal que al reducir el coste de una arista cualquiera, el coste del árbol generador de coste mínimo asociado, $T = (N_0, E_T)$, se reduce estrictamente. Si (N_0, C^*) no fuese una forma irreducible, existirían $i, j \in N_0$ tal que $c_{ij} > c_{lk}$ para toda arista (l, k) en el único camino de i a j en T. Pero entonces tomando ε suficientemente pequeño, y tomando $c_{ij} := c_{ij} - \varepsilon$, el coste del árbol generador de coste mínimo asociado se mantendría, en contradicción con la hipótesis. Por tanto, (N_0, C^*) es irreducible. $\square$

Teniendo en cuenta la observación 1.4.2, podemos enunciar las siguientes proposiciones:

Proposición 1.4.7. *Sea un problema mcst (N_0, C) y sea (N_0, C^*) su forma irreducible. Entonces se verifica que $C \leq C^*$, en el sentido que $c_{ij} \leq c^*_{ij}$ para todo $i, j \in N_0$.*

Proposición 1.4.8. *Sea un problema mcst (N_0, C). Si $T = (N_0, E_T)$ es un árbol generador de coste mínimo asociado al problema (N_0, C), entonces T es también es un árbol generador de coste mínimo asociado a su forma irreducible (N_0, C^*) y además $c_{ij} = c^*_{ij}$ para cada $(i, j) \in E_T$.*

A continuación enunciaremos algunos resultados relacionados con la formas irreducibles que serán utilizados más adelante para estudiar las propiedades de algunas reglas de reparto. Con el objetivo de simplificar la notación, dada una permutación en el conjunto de agentes $\pi \in \Pi_N$, denotaremos al agente $i \in N$, cumpliendo que $\pi(i) = s$, como π_s.

Lema 1.4.9. *Un problema (N_0, C^*) es una forma irreducible si y solo si existe un árbol $T = (N_0, E_T)$ y $\pi \in \Pi_N$ satisfaciendo las siguientes condiciones:*

(A1) *$E_T = \{(\pi_{s-1}, \pi_s)\}_{s=1}^n$, donde $\pi_0 = 0$.*
(A2) *Dados $\pi_p, \pi_q \in N_0$ con $p < q$, se tiene que $c^*_{\pi_p \pi_q} = max\{c^*_{\pi_{s-1}\pi_s} : p < s \leq q\}$.*

Además, en las condiciones anteriores, T es un árbol generador de coste mínimo asociado a (N_0, C^*).

Demostración. Ver [2], proposición 3.1, página 335. $\square$

Observación 1.4.10. *Como consecuencia del resultado anterior se obtiene que un problema es una forma irreducible si y solo si existe un árbol generador de coste mínimo asociado que sea una cadena y además, el coste de conectar dos usuarios arbitrarios es igual al máximo de los costes de las aristas que los conectan en dicha cadena.*

Lema 1.4.11. *Sea (N_0, C^*) una forma irreducible y sea $\pi \in \Pi_N$ tal que $T = (N_0, E_T)$ con $E_T = \{(\pi_{s-1}, \pi_s)\}_{s=1}^n$ y $\pi_0 = 0$, es un árbol generador de coste mínimo asociado en las condiciones del lema anterior. Sea $S \subseteq N$. Podemos suponer que $S = \{\pi_{s(1)}, ..., \pi_{s(|S|)}\}$ con $s(q-1) < s(q)$ para todo $q \in \{1, ..., |S|\}$ y $s(0) = 0$. Entonces:*
(a) *$T' = (S_0, E_{T'})$ con $E_{T'} = \{(\pi_{s(q-1)}, \pi_{s(q)})\}_{q=1}^{|S|}$ es un árbol generador de coste mínimo del*

problema (S_0, C^*) *y* $c^*(S) = \sum_{q=1}^{|S|} c^*_{\pi_{s(q-1)}\pi_{s(q)}}$.

(b) $c^*(S) - c^*(S\backslash\{\pi_{s(p)}\}) = min\{c^*_{\pi_{s(p-1)}\pi_{s(p)}}, c^*_{\pi_{s(p)}\pi_{s(p+1)}}\}$ *si* $p < |S|$ *y* $c^*(S) - c^*(S\backslash\{\pi_{s(|S|)}\}) = c^*_{\pi_{s(|S|-1)}\pi_{s(|S|)}}$.

Demostración. Ver [2], proposición 3.3, página 336. □

Lema 1.4.12. *Sean dos problemas mcst* (N_0, C) *y* (N_0, C') *tales que* $C \leq C'$. *Entonces se verifica también que* $C^* \leq C'^*$.

Demostración. Ver [2], lema 4.2, página 338. □

Introduciremos ahora el concepto de *núcleo irreducible*:

Definición 1.4.13. *Sea* (N, c) *un juego mcst. Se define el* ***núcleo irreducible*** *de* (N, c) *y se denotará* $IC(N, c)$*, como el núcleo del juego* (N, c^*)*. Es decir:*

$$IC(N, c) = C(N, c^*)$$

Observación 1.4.14. *Obsérvese que, como* $c^*(N) = c(N)$ *y* $c^*(S) \leq c(S)$*, para toda coalición* $S \subseteq N$ *se verifica que* $IC(c) \subseteq C(c)$*, es decir, que todo vector perteneciente al núcleo irreducible de un juego mcst pertenecerá también al núcleo del juego.*

Lo que resta de sección estará dedicado a estudiar los resultados más importantes relacionados con el núcleo irreducible. Veremos que (N, c^*) es un juego cóncavo y, por otro lado, se verá una caracterización para el núcleo irreducible de un juego *mcst*. Para llegar a estos resultados necesitaremos estudiar un lema previo:

Lema 1.4.15. *Sea* (N_0, C) *un problema mcst y sea* (N_0, C^*) *su forma irreducible. Consideremos una coalición* $S \subseteq N$. *Sea* $T_S = (S_0, E_{T_S})$ *un árbol generador de coste mínimo asociado al problema* (S_0, C^*) *y sean* $i \in N\backslash S$, $j \in S_0$ *tales que*

$$c^*_{ij} = min\{c^*_{il} : l \in S_0\}.$$

Entonces el árbol $T_{S\cup\{i\}} = (S_0 \cup \{i\}, E_{T_{S\cup\{i\}}})$*, con* $E_{T_{S\cup\{i\}}} = E_{T_S} \cup \{(i, j)\}$*, es un árbol generador de coste mínimo del problema* $(S_0 \cup \{i\}, C^*)$.

Demostración. Ver [10], lema 6.3.4, página 139. □

El lema anterior establece que dado un problema *mcst* (N_0, C^*) y una coalición $S \subseteq N$, si tenemos un árbol generador de coste mínimo asociado al problema (S_0, C^*) y queremos obtener un árbol generador de coste mínimo asociado a un problema $(S_0 \cup \{i\}, C^*)$, donde $i \in N\backslash S$, basta añadir al árbol anterior la arista de menor coste que conecte i con algún miembro de S_0.

Teorema 1.4.16 (Granot y Huberman). *Sea un juego mcst (N, c) y sea (N, c^*) su forma irreducible. Entonces (N, c^*) es un juego cóncavo.*

Demostración. Sea un jugador $i \in N$ y consideremos dos coaliciones S^1, S^2 tales que $S^2 \subseteq S^1 \subseteq N \backslash \{i\}$. Entonces aplicando el lema anterior se obtiene que

$$c^*(S^2 \cup \{i\}) - c^*(S^2) = min\{c^*_{ij} : j \in S^2_0\} \geq min\{c^*_{ij} : j \in S^1_0\} = c^*(S^1 \cup \{i\}) - c^*(S^1)$$

Por tanto, c^* es cóncavo. □

Veremos ahora el resultado central de la sección, la caracterización del núcleo irreducible de un juego *mcst.*

Teorema 1.4.17 (Bird). *Sea (N, c) un juego mcst y sea (N, c^*) su forma irreducible. Entonces $IC(N, c)$ coincide con la envolvente convexa del conjunto de soluciones de árbol de coste mínimo del juego (N, c^*).*

Demostración. Sabemos por el teorema 1.3.8 que toda solución de árbol de coste mínimo del juego (N, c^*) pertenece al núcleo de dicho juego. Como el núcleo es un conjunto convexo, se verifica que la envolvente convexa del conjunto de soluciones de árbol de coste mínimo de (N, c^*) es un subconjunto de $IC(N, c) = C(N, c^*)$. Para probar el teorema tendremos que ver la inclusión contraria. Sabemos por el teorema anterior que (N, c^*) es un un juego cóncavo, por tanto, aplicando el teorema 1.2.25, se tiene que el núcleo del juego es igual al conjunto de Weber del mismo, es decir, a la envolvente convexa del conjunto de vectores de costes marginales de c^*. Tomemos uno de dichos vectores $x = m^{\pi}_{c^*}$, donde π es una permutación del conjunto de jugadores. Nuestro objetivo será construir el árbol generador de coste mínimo asociado al juego (N, c^*), para el cual x es una solución de árbol de coste mínimo del juego. Definamos las coaliciones $S^{\pi}(m)$ para todo $m \in N_0$ por:

$$\begin{aligned} S^{\pi}(0) &:= \emptyset \\ S^{\pi}(m) &:= \{\pi^{-1}(j) : 1 \leq j \leq m\} \end{aligned}$$

Entonces, por la definición anterior, el conjunto de predecesores de $\pi^{-1}(i)$ respecto a la permutación π vendrá dado por:

$$P_{\pi}(\pi^{-1}(i)) = \{j \in N : \pi(j) < i\} = S^{\pi}(i-1)$$

Llevaremos a cabo el siguiente procedimiento iterativo para construir el árbol generador de coste mínimo buscado. Sea T^x_1 el árbol generador de coste mínimo asociado al juego $(S^{\pi}(1), c^*)$. Ahora, dado un árbol generador de mínimo coste T^x_{i-1} asociado al juego $(S^{\pi}(i-1), c^*)$, extenderemos dicho árbol a un árbol generador de coste mínimo T^x_i del juego $(S^{\pi}(i), c^*)$, conectando el nodo $\pi^{-1}(i)$ con T^x_{i-1} mediante la arista más barata tal y como se describe en el lema 1.4.15. Así, T^x_n es un árbol generador de mínimo coste asociado al juego (N, c^*). Sea ahora $p(\pi^{-1}(i))$ el predecesor inmediato de $\pi^{-1}(i)$ en el camino que conecta el nodo 0 con $\pi^{-1}(i)$ en T^x_n. Entonces

$$c^*(S^\pi(m)) = \sum_{(j,l)\in E_{T^x_m}} c^*_{jl} \text{ para todo } m \in N$$

Así,

$$\begin{aligned} x_{\pi^{-1}(i)} &= c^*(S^\pi(i-1) \cup \{\pi^{-1}(i)\}) - c^*(S^\pi(i-1)) \\ &= c^*_{p(\pi^{-1}(i))i} \end{aligned}$$

luego en efecto, x es una solución de árbol de coste mínimo asociada al árbol T^x_n.

□

Capítulo 2

Reglas de reparto

El objetivo principal dentro de los problemas de árbol generador de coste mínimo es el de, una vez conocido el coste total que les supone a los usuarios conectarse al servicio conjuntamente, distribuir los costes entre dichos usuarios. Para ello, diversas reglas de reparto de costes han sido propuestas en el estudio del tema. En este capítulo se presentarán algunas propiedades de reglas de reparto y se introducirán algunas propuestas de reglas realizadas en la literatura dedicada a los problemas de árbol generador de mínimo coste. Se analizarán también las propiedades que cumplen las reglas introducidas y se establecerá una comparación entre ellas en virtud de cuales de estas propiedades verifican.

2.1. Definición y propiedades

Definición 2.1.1. *Llamaremos* ***regla de reparto de costes****, o simplemente* ***regla****, a una aplicación ψ del conjunto de problemas de árbol generador de coste mínimo en $\mathbb{R}^N$, tal que dado un problema mcst (N_0, C), la componente i-ésima del vector $\psi(N_0, C) = (\psi_1(N_0, C), ..., \psi_n(N_0, C))$, representa el coste que la regla asigna al usuario i y además verifica que*

$$\sum_{i \in N} \psi_i(N_0, C) = c_{mcst}(N_0, C),$$

es decir, el vector proporcionado por la regla es eficiente.

A continuación definiremos algunas propiedades que puede cumplir una regla de reparto. El cumplimiento de estas propiedades por parte de una determinada regla de reparto dota a dicha regla de una serie de virtudes en el sentido económico que serán explicadas tras la definición de cada propiedad.

Definición 2.1.2. *Diremos que una regla de reparto de costes ψ satisface la propiedad* ***selección del núcleo (SN),*** *si para todo problema (N_0, C) y toda coalición $S \subseteq N$ se verifica que el vector $\psi(N_0, C)$ pertenece al núcleo del juego mcst asociado (N, c), es decir, que se verifique la* *desigualdad*

$$\sum_{i \in S} \psi_i(N_0, C) \leq c_{mcst}(S_0, C) \text{ para cada } S \subseteq N.$$

Esta propiedad implica que no existe una coalición que se encontrase en una posición más ventajosa al construir su propia red, en lugar de pagar lo que la regla de reparto asigna a esta coalición. Recordemos que para que un vector pertenezca al núcleo de un juego, además de satisfacer la condición anterior, debe ser eficiente, sin embargo, esto se verifica por definición para toda regla de reparto.

Definición 2.1.3. *Diremos que una regla de reparto de costes ψ satisface la propiedad **monotonicidad del coste(MC),** si para todo par de problemas (N_0, C), (N_0, C') para los que se verifica $c_{ij} < c'_{ij}$ para ciertos $i \in N$, $j \in N_0$ y para el resto de subíndices se verifica la igualdad $c_{kl} = c'_{kl}$, se tiene que*

$$\psi_i(N_0, C) \leq \psi_i(N_0, C')$$

La propiedad anterior implica que si una determinada conexión que atañe al usuario i aumenta su coste, pero el resto de conexiones mantienen su coste, entonces el usuario i no puede encontrarse en una mejor posición respecto a la cantidad a pagar que la regla le asigna.

Definición 2.1.4. *Diremos que una regla de reparto ψ satisface la propiedad **solidaridad (SOL),** si para cada par de problemas (N_0, C), (N_0, C') tales que $C \leq C'$ componente a componente, se verifica que*

$$\psi(N_0, C) \leq \psi(N_0, C')$$

Una regla que satisfaga esta propiedad verifica que si aumenta el coste de algunas conexiones, entonces ningún usuario puede encontrarse en una mejor posición que anteriormente respecto a la cantidad que la regla le asigna. Posteriormente veremos que la propiedad *solidaridad* es más fuerte que la de *monotonicidad del coste.*

Definición 2.1.5. *Se dice que una regla satisface la propiedad **monotonicidad de la población (MP),** si para todo problema (N_0, C), toda coalición $S \subseteq N$ y todo usuario $i \in S$, se tiene que*

$$\psi_i(N_0, C) \leq \psi_i(S_0, C)$$

En algunos textos, la anterior propiedad es conocida también como **monotonicidad fuerte del coste**. La verificación de esta propiedad implica que la adhesión de nuevos usuarios a una coalición no perjudica a los usuarios que inicialmente conformaban dicha coalición.

Definición 2.1.6. *Diremos que una regla de reparto de costes ψ es **positiva (POS),** si para todo problema mcst (N_0, C) y todo $i \in N$ se verifica que*

$$\psi_i(N_0, C) \geq 0$$

La interpretación de esta propiedad es sencilla: ningún usuario obtiene beneficios.

Definición 2.1.7. *Una regla de reparto ψ verifica la* ***separabilidad (SEP),*** *si para todo problema mcst (N_0, C) y cada coalición $S \subseteq N$, satisfaciendo $c_{mcst}(N_0, C) = c_{mcst}(S_0, C) + c_{mcst}((N \backslash S)_0, C)$, se tiene que*

$$\psi_i(N_0, C) = \begin{cases} \psi_i(S_0, C) & si\ i \in S, \\ \psi_i((N \backslash S)_0, C) & si\ i \in N \backslash S. \end{cases}$$

Dos subconjuntos de usuarios S y $N \backslash S$ pueden estar conectados a la fuente que proporciona el servicio cada uno por separado o conjuntamente. Si no hay ahorro cuando están conectados conjuntamente, esta propiedad implica que los usuarios pagarán lo mismo en ambos casos. En algunos textos, esta propiedad recibe el nombre de **descomposición**.

Antes de introducir la siguiente propiedad, veamos la siguiente la siguiente definición:

Definición 2.1.8. *Sea (N_0, C) un problema mcst. Diremos que dos usuarios $i, j \in N$ son* ***simétricos*** *si para todo $k \in N_0 \backslash \{i, j\}$, se tiene que $c_{ik} = c_{jk}$.*

Definición 2.1.9. *Una regla de reparto ψ es* ***simétrica (SIM),*** *si para todo par de usuarios simétricos $i, j \in N$ se tiene que*

$$\psi_i(N_0, C) = \psi_j(N_0, C)$$

Así, si existe un par de usuarios tales que el coste de sus conexiones con el resto de usuarios son idénticas, una regla de reparto simétrica asignará la misma cantidad a ambos.

Definición 2.1.10. *Una regla de reparto ψ satisface la propiedad* ***independencia de otros costes (IOC),*** *si dados dos problemas mcst (N_0, C) y (N_0, C'), se tiene que para cada $i \in N$ tal que $c_{ij} = c'_{ij}$ para todo $j \in N_0 \backslash \{i\}$, se verifica la igualdad*

$$\psi_i(N_0, C) = \psi_i(N_0, C')$$

La propiedad anterior implica que la cantidad asignada por la regla al usuario i depende exclusivamente del coste de las aristas que conectan i con el resto de usuarios y la fuente. Esta propiedad sería muy positiva para una regla de reparto, sin embargo, Bergantiños y Vidal-Puga demuestran en [2] (lema 4.1, página 338) que no existe ninguna regla que satisfaga *IOC*. Por ello, introduciremos dos propiedades relacionadas con la independencia de costes de aristas no incidentes en un determinado usuario menos restrictivas que esta última.

Definición 2.1.11. *Sean dos problemas mcst (N_0, C) y (N_0, C') y sea un usuario $i \in N$ satisfaciendo las siguientes tres condiciones:*

1. $c_{ik} = c'_{ik}$ *para todo* $k \in N_0$.

2. *Dados* $j, k \in N_0$*, se tiene que* $c_i^{min} \geq c_{jk}$ *si solo si* $c_i^{min} \geq c'_{jk}$*, donde* $c_i^{min} := min_{k \in N_0 \setminus \{i\}} \{c_{ik}\}$.

3. *Si* $j, k \in N_0$ *son tales que* $c_{jk} > c_i^{min}$ *entonces se verifica que* $c'_{jk} = c_{jk}$.

Diremos que una regla de reparto de costes ψ *satisface la propiedad* ***independencia de bajos costes (IBC),*** *si en las condiciones anteriores se verifica que*

$$\psi_i(N_0, C) = \psi_i(N_0, C').$$

La propiedad anterior implica que la cantidad asignada por la regla al usuario i no depende del coste de las aristas más baratas que la arista de menor coste incidente en i.

Definición 2.1.12. *Sean dos problemas mcst* (N_0, C) *y* (N_0, C') *y sea un usuario* $i \in N$ *satisfaciendo las siguientes tres condiciones:*

1. $c_{ik} = c'_{ik}$ *para todo* $k \in N_0$.

2. *Dados* $j, k \in N_0$*, se tiene que* $c_i^{max} \leq c_{jk}$ *si solo si* $c_i^{max} \leq c'_{jk}$*, donde* $c_i^{max} := max_{k \in N_0 \setminus \{i\}} \{c_{ik}\}$.

3. *Si* $j, k \in N_0$ *son tales que* $c_{jk} < c_i^{max}$ *entonces se verifica que* $c'_{jk} = c_{jk}$.

Diremos que una regla de reparto de costes ψ *satisface la propiedad* ***independencia de grandes costes (IGC),*** *si en las condiciones anteriores se verifica que*

$$\psi_i(N_0, C) = \psi_i(N_0, C').$$

La anterior propiedad implica que la cantidad asignada por la regla al usuario i no depende del coste de las aristas más caras que la arista de mayor coste incidente en i.

Definición 2.1.13. *Consideremos dos problemas mcst* (N_0, C) *y* (N_0, C')*. Consideremos dos cantidades* $c_0, c'_0 \geq 0$ *tales que* $c_0 < c'_0$*. Supongamos que* $c_{0i} = c_0$ *y* $c'_{0i} = c'_0$ *para todo* $i \in N$*, y* $c_{ij} = c'_{ij} \leq c_0$ *para todo* $i, j \in N$*. Diremos que una regla de reparto* ψ *satisface la propiedad de* ***igualdad de reparto de costes extra (IRCE),*** *si verifica que*

$$\psi_i(N_0, C') = \psi_i(N_0, C) + \frac{c'_0 - c_0}{|N|} \text{ para todo } i \in N.$$

Bajo las condiciones anteriores el árbol óptimo del problema (N_0, C) se obtendrá al establecer una conexión entre cualquier usuario ($c_{i0} = c_0$ para todo $i \in N$) y conectando dicho usuario directamente con el resto de usuarios ($c_{ij} \leq c_0$). Supongamos que todos los usuarios aceptan el reparto $\psi(N_0, C)$. Supongamos que la conexión de los usuarios con la fuente aumenta hasta una cantidad c'_0. La propiedad anterior establece que la cantidad $c'_0 - c_0$ es repartida a partes iguales entre los usuarios.

Antes de introducir la última propiedad daremos la siguiente definición:

Definición 2.1.14. *Diremos que dos problemas mcst* (N_0, C) *y* (N_0, C') *son* ***árbol-equivalentes*** *si existe un árbol generador* $T = (N_0, E_T)$ *tal que*

1. T es un árbol generador de coste mínimo para (N_0, C) y (N_0, C').

2. $c_{ij} = c'_{ij}$ para toda arista $(i, j) \in E_T$.

Definición 2.1.15. *Diremos que una regla de reparto ψ satisface la* ***independencia de árboles irrelevantes (IAI),*** *si para cada par de problemas (N_0, C), (N_0, C') árbol-equivalentes se verifica que*

$$\psi_i(N_0, C) = \psi_i(N_0, C')$$

La propiedad anterior implica que si dos problemas (N_0, C), (N_0, C') tienen el mismo árbol generador de coste mínimo asociado y el coste de las aristas de dicho árbol es idéntico, la regla proporcionará el mismo reparto en ambos problemas, independientemente del coste del resto de conexiones que no forman parte del árbol.

Las propiedades *simetría, positividad, selección del núcleo, monotonicidad del coste, monotonicidad de la población* y *separabilidad* son propiedades usuales que aparecen en varios textos de la literatura relacionada con los problemas de árbol generador de mínimo coste. Por su parte, las propiedades *solidaridad, independencia de otros costes, igualdad de reparto de costes extra* e *independencia de árboles irrelevantes* son propiedades introducidas en [2], mientras que *independencia de grandes costes* e *independencia de bajos costes* son propiedades estudiadas en [3]. En adelante nos referiremos a estas propiedades utilizando la nomenclatura abreviada.

Entre las distintas propiedades de reglas de coste existen diversas relaciones. Lo que resta de sección estará dedicado a introducir dichas relaciones. Los resultados que se mostrarán a continuación se deben a Bergantiños y Vidal-Puga y pueden encontrarse en [2] y [3].

Definición 2.1.16. *Sea un problema mcst (N_0, C). Diremos que una regla de reparto ψ* ***depende únicamente de la forma irreducible*** *si $\psi(N_0, C) = \psi(N_0, C^*)$, donde (N_0, C^*) es la forma irreducible de (N_0, C).*

Proposición 2.1.17. *Una regla de reparto depende únicamente de la forma irreducible si y solo si satisface IAI.*

Demostración. Ver [2], proposición 3.5, página 337. □

Proposición 2.1.18. *Toda regla de reparto que satisfaga SOL satisface también MC e IAI.*

Demostración. Sea ψ una regla de reparto que satisface SOL.
***SOL* implica *MC*.** Sean (N_0, C), (N_0, C') dos problemas *mcst* tales que $c_{ij} < c'_{ij}$ para ciertos $i \in N$, $j \in N_0$ y $c_{kl} = c'_{kl}$ en otro caso. Se verifica pues que $C \leq C'$. Ahora bien, ψ satisface SOL, luego $\psi(N_0, C) \leq \psi(N_0, C')$, por lo que en particular $\psi_i(N_0, C) \leq \psi_i(N_0, C')$, de donde se obtiene que ψ satisface MC.

***SOL* implica *IAI*.** Sea (N_0, C) un problema y sea (N_0, C^*) su forma irreducible. Entonces $\psi(N_0, C) \leq \psi(N_0, C^*)$, pues por la proposición 1.4.7, se tiene que $C \leq C^*$. Ahora, por la proposición 1.4.8, se tiene que $c_{mcst}(N_0, C) = c_{mcst}(N_0, C^*)$. Pero entonces $\psi(N_0, C) = \psi(N_0, C^*)$, es decir, ψ depende únicamente de la forma irreducible. Aplicando la proposición anterior se obtiene el resultado. □

Proposición 2.1.19. *Toda regla de reparto que satisfaga MP satisface también SN y SEP.*

Demostración. Sea ψ una regla de reparto que satisface MP

***MP* implica *SN*.** Sea un problema *mcst* (N_0, C) y una coalición arbitraria $S \subseteq N$. Se verifica entonces que $\psi_i(N_0, C) \leq \psi_i(S_0, C)$ para cada $i \in S$. Por tanto

$$\sum_{i \in S} \psi_i(N_0, C) \leq \sum_{i \in S} \psi_i(S_0, C) = c_{mcst}(S_0)$$

luego ψ verifica SN.

***MP* implica *SEP*.** Sea (N_0, C) un problema *mcst* y $S \subseteq N$ tales que $c_{mcst}(N_0, C) = c_{mcst}(S_0, C) + c_{mcst}((N \backslash S)_0, C)$. Se tiene entonces que $c_{mcst}(N_0, C) \geq c_{mcst}(S_0, C)$ y $c_{mcst}(N_0, C) \geq c_{mcst}((N \backslash S)_0, C)$. Ahora bien, como ψ verifica MP, se tiene que $\psi_i(N_0, C) \leq \psi_i(S_0, C)$ para cada $i \in S$ y $\psi_i(N_0, C) \leq \psi_i((N \backslash S)_0, C)$ para cada $i \in N \backslash S$. Por último, debe cumplirse que $\sum_{i \in S} \psi_i(S_0, C) = c_{mcst}(S_0)$, $\sum_{i \in N \backslash S} \psi_i((N \backslash S)_0, C) = c_{mcst}(N \backslash S_0)$ y $\sum_{i \in N} \psi_i(N_0, C) = c_{mcst}(N_0)$, luego

$$\psi_i(N_0, C) = \begin{cases} \psi_i(S_0, C) & \text{si } i \in S, \\ \psi_i((N \backslash S)_0, C) & \text{si } i \in N \backslash S. \end{cases}$$

□

Para mostrar la siguiente relación entre propiedades, será necesario en primer lugar estudiar un lema previo.

Lema 2.1.20. *Sea ψ una regla de reparto que satisface IAI. Sean dos problemas mcst (N_0, C), (N_0, C') e $i \in N$ en las condiciones de la definición 2.1.12. Entonces si existen $j, k \in N_0$ y $a > 0$ tal que para todo $l, m \in N_0$*

$$c_{lm} = \begin{cases} c'_{lm} - a & \textit{si } (l, m) = (j, k) \\ c'_{lm} & \textit{en otro caso.} \end{cases}$$

y $c_i^{max} \leq c_{jk}$. Entonces $\psi_i(N_0, C) = \psi_i(N_0, C')$.

Demostración. En primer lugar, como $c_i^{max} \leq c_{jk} < c'_{jk}$, se deduce que $i \neq j$ e $i \neq k$. Sea T un árbol generador de coste mínimo asociado a (N_0, C). Si $(j, k) \notin T$, entonces T es también un árbol generador de coste mínimo de (N_0, C'). Como ψ satisface IAI, se tiene que

$\psi_i(N_0, C) = \psi_i(N_0, C')$. Por contra, si $(j, k) \in T$, podemos suponer sin pérdida de generalidad que j es el primer nodo en el único camino que conecta k con el nodo 0. Existen dos casos posibles:

1. La conexión (j, k) no está en el único camino en T de i a 0. Definamos $T^* = T\backslash\{(j, k)\} \cup \{(i, k)\}$. Es claro que T^* es un árbol satisfaciendo que
$$c(T^*) - c(T) = c_{ik} - c_{jk}.$$
Como $c_i^{max} \leq c_{jk}$, se deduce que $c(T^*) \leq c(T)$. Por tanto, T^* es un árbol generador de (N_0, C) y (N_0, C'). Como ψ satisface IAI, $\psi_i(N_0, C) = \psi_i(N_0, C')$.

2. La arista (j, k) está en el único camino en T de i a 0. Definamos $T^* = (T\backslash\{(j, k)\} \cup \{(0, i)\}$. Usando un razonamiento análogo al caso anterior se obtiene que $\psi_i(N_0, C) = \psi_i(N_0, C')$.

□

Proposición 2.1.21. *Toda regla de reparto que satisfaga IAI satisface también IGC.*

Demostración. Sea ψ una regla de reparto que satisfaga IAI y sean dos problemas *mcst* (N_0, C), (N_0, C') e $i \in N$ en las condiciones de la definición 2.1.12. Sea $A_i = \{(i_l^1, i_l^2)\}_{l=1}^p$ el conjunto de aristas satisfaciendo $c_{i_l^1 i_l^2} \neq c'_{i_l^1 i_l^2}$. Tomemos $C^0 := C$. Para todo $l = 1, ..., p$ definamos el problema *mcst* (N_0, C^l), donde $c^l_{i_l^1 i_l^2} = c'_{i_l^1 i_l^2}$ para las aristas en A_i y $c^l_{lm} = c^{l-1}_{lm}$ en otro caso. Para cada $l = 1, ..., p$ tomemos $(j, k) = (i_l^1, i_l^2)$. Aplicando el lema anterior, se tiene que $\psi_i(N_0, C^{l-1}) = \psi_i(N_0, C^l)$ para todo $l = 1, ..., p$. Como $C^0 = C$ y $C^p = C'$, ψ satisface IGC.

□

2.2. Regla de Bird

La regla de Bird fue introducida por C.G. Bird, quien fue también el primero en realizar un enfoque teórico sobre los juegos de árbol generador de coste mínimo en [7] (1976). La idea de la regla de Bird es sencilla: consideremos un problema *mcst* (N_0, C), consideremos el árbol generador de coste mínimo de la red correspondiente y supongamos en primer lugar que es único. Entonces, según la regla de Bird, el coste que deberá pagar un usuario i será igual al coste de la arista incidente en el nodo i en el único camino que conecta i con el nodo que proporciona el servicio. A lo largo del texto denotaremos como $\mathcal{C}^1$ al conjunto de matrices tales que el problema (N_0, C) tiene un único árbol generador de coste mínimo siendo N un conjunto arbitrario de usuarios. Denotaremos como $\mathcal{C}^2$ al conjunto de matrices de $\mathcal{C}^1$ para las no existen dos aristas del mismo coste en el árbol generador de mínimo coste asociado a (N_0, C).

Teorema 2.2.1. *Sea* (N_0, C) *un problema mcst. Entonces todo vector de costes obtenido de considerar la regla de Bird pertenece al núcleo del juego asociado* (N, c)*.*

Demostración. Basta observar que todo vector de costes propuesto por la regla de Bird es una solución de árbol de coste mínimo y aplicar el teorema 1.3.8.

□

Como se vio en la observación 1.1.15, no existe necesariamente unicidad en el árbol generador de coste mínimo de una red, por tanto, uno de los inconvenientes de esta definición es que sólo se tiene garantizada la unicidad del vector de costes obtenido en el caso de problemas *mcst* en los que el árbol generador de coste mínimo sea único. Una alternativa para problemas en los que exista más de un árbol generador de coste mínimo, es tomar una combinación convexa de los vectores obtenidos para cada árbol:

Definición 2.2.2. *Sea* (N_0, C) *un problema mcst. Consideremos una permutación del conjunto de usuarios* $\pi \in \Pi_N$ *y definamos* $B^\pi(N_0, C)$ *como la asignación obtenida al considerar, en primer lugar, el árbol generador obtenido mediante la aplicación del algoritmo de Prim, tomando en caso de empates el primer usuario según el orden de* π *y, en segundo lugar, la regla de Bird para problemas con árbol generador de coste mínimo único. La* ***regla de Bird,*** $B(N_0, C)$***,*** *con independencia del número de árboles generadores de coste mínimo del problema vendrá dada por:*

$$B(N_0, C) = \frac{1}{|N|!} \sum_{\pi \in \Pi_N} B^\pi(N_0, C)$$

Observación 2.2.3. *Obsérvese que en el caso en que el árbol generador de coste mínimo del problema sea único,* $B^\pi(N_0, C)$ *será siempre la misma asignación, por lo que ambas definiciones coinciden.*

Observación 2.2.4. *Recordemos que en la definición de regla de reparto (definición 2.1.1), se exige que el vector obtenido sea eficiente, es decir, que la suma de las cantidades asignadas a cada usuario sea exactamente igual al coste de un árbol generador de mínimo coste asociado al problema. Es claro por la definición de la regla de Bird que cada* $B^\pi(N_0, C)$ *es eficiente, por tanto,* $B(N_0, C)$ *también lo será.*

Veamos un ejemplo de aplicación de la regla de Bird en un problema *mcst* concreto:

Ejemplo 2.2.5. *Consideremos el problema mcst* (N_0, C)*, con* $N_0 = \{0, 1, 2, 3\}$ *y la matriz de costes dada por*

$$C = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 1 & 0 & 4 & 2 \\ 3 & 4 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

Existe un único árbol generador de coste mínimo, que viene dado por las aristas $\{(0,1),(1,3),(2,3)\}$ *y su coste asociado es* 4*. Por tanto,* $B(N_0, C) = (1, 1, 2)$*.*

Veamos ahora un problema en el que el árbol generador de coste mínimo no es único:

Ejemplo 2.2.6. *Consideremos el problema mcst* (N_0, C)*, con* $N_0 = \{0, 1, 2, 3\}$ *y la matriz de costes dada por*

$$C = \begin{pmatrix} 0 & 6 & 3 & 3 \\ 6 & 0 & 3 & 3 \\ 3 & 3 & 0 & 1 \\ 3 & 3 & 1 & 0 \end{pmatrix}$$

En este problema, como vimos en el ejemplo 1.1.16, el árbol generador de coste mínimo no es único, por tanto, para obtener $B(N_0, C)$*, deberemos calcular* $B^\pi(N_0, C)$ *para cada* $\pi \in \Pi_N$*, aplicando el algoritmo de Prim:*
Considerando la permutación $\pi = (1, 2, 3)$*, la primera arista en añadirse será* $(0, 2)$*, pues aunque su coste sea igual que la arista* $(0, 3)$*, según la permutación, el usuario* 2 *debe conectarse antes al servicio en caso de empate. Tras ello, se añade la arista* $(2, 3)$*, y por último* $(1, 2)$*. Así, se obtiene que* $B^\pi(N_0, C) = (3, 3, 1)$*. Procediendo análogamente para cada permutación se obtiene:*

π		$B^\pi(N_0, C)$
$(1,2,3)$	$\longrightarrow$	$(3,3,1)$
$(1,3,2)$	$\longrightarrow$	$(3,1,3)$
$(2,1,3)$	$\longrightarrow$	$(3,3,1)$
$(2,3,1)$	$\longrightarrow$	$(3,3,1)$
$(3,1,2)$	$\longrightarrow$	$(3,1,3)$
$(3,2,1)$	$\longrightarrow$	$(3,1,3)$

Por tanto,

$$B(N_0, C) = \frac{1}{6} \sum_{\pi \in \Pi_N} B^\pi(N_0, C) = (3, 2, 2).$$

Una vez visto este ejemplo de aplicación, el siguiente teorema servirá para mostrar las propiedades que satisface la regla de Bird. Dicho teorema puede encontrarse en [3].

Teorema 2.2.7. *La regla de Bird satisface las propiedades SN, POS, SIM, IRCE e IBC. Sin embargo, no satisface MC, MP, SOL, SEP, IAI ni IGC.*

Demostración. *B* satisface *SN*: Aplicando el resultado 2.2.1 cada vector $B^\pi(N_0, C)$ pertenece al núcleo. El resultado es claro entonces si tenemos en cuenta que $B(N_0, C)$ es una combinación convexa de elementos del núcleo y este es un conjunto convexo.
***B* satisface *POS*:** Claro por la definición de B.
***B* satisface *SIM*:** Sean $i, j \in N$ dos usuarios simétricos de (N_0, C). Dada una permutación en el conjunto de usuarios $\pi \in \Pi_N$, definamos la permutación $\pi^{ij} \in \Pi_N$ dada por $\pi^{ij}(i) = \pi(j)$,

$\pi^{ij}(j) = \pi(i)$ y $\pi^{ij}(k) = \pi(k)$ para todo $k \in N \backslash \{i, j\}$. Es claro que $B_i^{\pi}(N_0, C) = B_j^{\pi^{ij}}(N_0, C)$. Entonces

$$\begin{aligned} B_i(N_0, C) &= \frac{1}{|N|!} \sum_{\pi \in \Pi_N} B_i^{\pi}(N_0, C) &= \frac{1}{|N|!} \sum_{\pi \in \Pi_N} B_j^{\pi^{ij}}(N_0, C) \\ &= \frac{1}{|N|!} \sum_{\pi \in \Pi_N} B_j^{\pi}(N_0, C) &= B_j(N_0, C). \end{aligned}$$

Luego en efecto, B satisface *SIM*.

***B* satisface *IRCE*:** Sean (N_0, C) y (N_0, C) en las condiciones de la definición 2.1.13. Entonces para cada $\pi \in \Pi_N$ se tendrá que

$$B_i^{\pi}(N_0, C') = \begin{cases} B_i^{\pi}(N_0, C) + (c_0' - c_0) & \text{si } \pi(i) = 1, \\ B_i^{\pi}(N_0, C) & \text{en otro caso.} \end{cases}$$

Entonces

$$B_i^{\pi}(N_0, C') = \sum_{\pi \in \Pi_N} \frac{(|N| - 1) B_i^{\pi}(N_0, C)}{|N|} + \frac{B_i^{\pi}(N_0, C) + (c_0' - c_0)}{|N|} = B_i^{\pi}(N_0, C) + \frac{c_0' - c_0}{|N|}$$

Luego en efecto, B satisface *IRCE*.

***B* satisface *IBC*:** Sean (N_0, C), (N_0, C') e $i \in N$ como en la definición de *IBC* (definición 2.1.11). Para ver el resultado es suficiente probar que $B_i^{\pi}(N_0, C) = B_i^{\pi}(N_0, C')$ para cada permutación $\pi \in \Pi_N$.

Supongamos en primer lugar que existe una única arista (j, k) tal que $c_{jk}' \neq c_{jk}$. Se tiene entonces que $i \neq j$ e $i \neq k$. Podemos suponer sin pérdida de generalidad que $c_{jk}' < c_{jk} \leq c_i^{min}$, donde recordemos $c_i^{min} := min_{k \in N_0 \backslash \{i\}} \{c_{ik}\}$. Al calcular $B^{\pi}(N_0, C)$ (respectivamente $B^{\pi}(N_0, C')$) a partir del algoritmo de Prim, los usuarios son conectados secuencialmente a la red en un orden específico. Denotemos este orden como π^* (respectivamente $\pi^{*'}$). Consideremos tres casos:

1. $\pi^*(i) < min\{\pi^*(j), \pi^*(k)\}$. En este caso $P_{\pi^*}(i) = P_{\pi^{*'}}(i)$, por tanto:

$$B_i^{\pi}(N_0, C) = B_i^{\pi}(N_0, C') = min_{l \in P_{\pi^*}(i)} \{c_{il}\}$$

2. $min\{\pi^*(j), \pi^*(k)\} < \pi^*(i) < max\{\pi^*(j), \pi^*(k)\}$. En este caso podemos suponer sin pérdida de generalidad que $\pi^*(j) < \pi^*(i) < \pi^*(k)$. Se tiene que $B_i^{\pi}(N_0, C) = c_{i^0 i}$, donde (i^0, i) o, equivalentemente (i, i^0), es la primera arista en el único camino de i a 0 en el árbol generador de coste mínimo. Así, se tiene que

$$c_i^{min} \leq c_{i^0 i} \leq c_{jk}.$$

Como $c_{jk} \leq c_i^{min}$, se obtiene que $c_i^{min} = c_{jk} = c_{i^0 i}$ y $B_i^\pi(N_0, C) = c_i^{min}$.
Además, como $c'_{jk} < c_{jk}$, $P_{\pi^*}(i) \subseteq P_{\pi^{*\prime}}(i)$, por lo que $B_i^\pi(N_0, C') \leq B_i^\pi(N_0, C) = c_i^{min}$.
Ahora bien, $B_i^\pi(N_0, C') = min_{l \in P_{\pi^{*\prime}}(i)}\{c_{il}\} \geq c_i^{min}$. Por tanto:

$$B_i^\pi(N_0, C) = B_i^\pi(N_0, C') = c_i^{min}$$

3. $max\{\pi^*(j), \pi^*(k)\} < \pi^*(i)$. En este caso $P_{\pi^*}(i) = P_{\pi^{*\prime}}(i)$, por tanto:

$$B_i^\pi(N_0, C) = B_i^\pi(N_0, C') = min_{l \in P_{\pi^*}(i)}\{c_{il}\}$$

Supongamos ahora que existen varias aristas (j, k) tal que $c'_{jk} \neq c_{jk}$. Sea $A = \{(j_1^h, j_2^h)\}_{h=1}^p$ el conjunto de estas aristas. Consideremos ahora la familia de problemas *mcst* $\{(N_0, C^h)\}_{h=0}^p$ tal que $C^0 = C$ y

$$c_{kl}^h = \begin{cases} c'_{j_1^h j_2^h} & \text{si } (k,l) = (j_1^h, j_2^h), \\ c_{kl}^{h-1} & \text{en otro caso} \end{cases}$$

para cada $h = 1, ..., p$. Ahora bien, para todo $h = 1, ..., p$, (N_0, C^{h-1}), (N_0, C^h) e $i \in N$ cumplen las condiciones de la defininción de *IBC*. Es más, la arista (j_1^h, j_2^h) es la única arista de diferente coste entre ambos problemas. Así, por lo demostrado anteriormente, $B_i^\pi(N_0, C^{h-1}) = B_i^\pi(N_0, C^h)$ para todo $h = 1, ..., p$. Como $C^p = C'$, $B_i^\pi(N_0, C) = B_i^\pi(N_0, C')$.

***B* no satisface *MC*:** Consideremos los problemas (N_0, C), (N_0, C), con $N = \{1, 2\}$ y las matrices de costes siguientes:

$$C = \begin{pmatrix} 0 & 4 & 5{,}5 \\ 4 & 0 & 3 \\ 5{,}5 & 3 & 0 \end{pmatrix} \qquad C' = \begin{pmatrix} 0 & 4 & 3{,}5 \\ 4 & 0 & 3 \\ 3{,}5 & 3 & 0 \end{pmatrix}$$

En este caso se tiene que $B(N_0, C) = (4, 3)$, mientras que $B(N_0, C') = (3, 3{,}5)$. Pero entonces $B_2(N_0, C) < B_2(N_0, C')$. Sin embargo, $c_{02} > c'_{02}$, mientras que el resto de conexiones tienen el mismo coste en ambos problemas. Por tanto, B no satisface *MC*.

***B* no satisface *SOL*:** Como B no satisface *MC*, aplicando la proposición 2.1.18 se obtiene el resultado.

***B* no satisface *SEP*:** Sea el problema *mcst* (N_0, C), con $N = \{1, 2, 3\}$, $S = \{1, 2\}$ y

$$C = \begin{pmatrix} 0 & 3 & 10 & 1 \\ 3 & 0 & 1 & 10 \\ 10 & 1 & 0 & 3 \\ 1 & 10 & 3 & 0 \end{pmatrix}$$

En este caso se tiene que

$$c_{mcst}(S_0, C) + c_{mcst}((N \backslash S)_0, C) = 4 + 1 = 5 = c_{mcst}(N_0, C).$$

Ahora bien, $B(N_0, C) = (2, 2, 1)$ y $B(S_0, C) = (3, 1)$, es decir, $B_1(N_0, C) \neq B_1(S_0, C)$ y $B_2(N_0, C) \neq B_2(S_0, C)$, por lo que B no satisface *SEP*.

***B* no satisface *MP*:** Como B no satisface *SEP*, aplicando la proposición 2.1.19 se obtiene el resultado.

***B* no satisface *IGC*:** Consideremos los problemas (N_0, C), (N_0, C), con $N = \{1, 2\}$ y las matrices de costes siguientes:

$$C = \begin{pmatrix} 0 & 10 & 10 \\ 10 & 0 & 2 \\ 10 & 2 & 0 \end{pmatrix} \qquad C' = \begin{pmatrix} 0 & 10 & 12 \\ 10 & 0 & 2 \\ 12 & 2 & 0 \end{pmatrix}$$

Entonces, (N_0, C), (N_0, C) y el usuario 1 están en las condiciones de la definición de IGC (definición 2.1.12). Sin embargo, $B_1(N_0, C) = 6$ y $B_1(N_0, C') = 10$.

***B* no satisface *IAI*:** Como B no satisface IGC, aplicando la proposición 2.1.21 se obtiene el resultado. $\square$

La regla de Bird ha sido caracterizada por varios autores a lo largo de la literatura dedicada al reparto de coste en problemas de árbol generador de coste mínimo. A continuación mostraremos la caracterización realizada por Dutta y Kar en [12].

Definición 2.2.8. *Sea $C \in \mathcal{C}^1$ y sea $i \in N$ una hoja del árbol generador de coste mínimo asociado al problema (N_0, C). Denotemos como C' a la restricción de C sobre el conjunto $N_0 \backslash \{i\}$. Una regla de reparto ψ satisface la propiedad **monotonicidad de hojas** si*

$$\psi_k(N_0, C) \leq \psi_k(N_0, C') \text{ para todo } k \in N \backslash \{i\}.$$

Observación 2.2.9. *Recordemos que una hoja en un grafo es un nodo de grado de incidencia 1.*

Consideremos inicialmente el problema sin tener en cuenta al usuario i. Como $i \in N$ es una hoja en el árbol generador de mínimo coste, ningún usuario se conecta al servicio a través de él. La propiedad anterior implica que ningún usuario pagará un coste mayor para incluir al nodo i en la red.

Para la definición de las próximas propiedades, consideremos lo siguiente: sea un problema *mcst* (N_0, C) con $C \in \mathcal{C}^1$. Sea $T = (N_0, E_T)$ el árbol generador de coste mínimo asociado y sea $i \in N$ tal que $(i, 0) \in E_T$. Sea x_i la cantidad asignada al usuario i a través de la aplicación de una determinada regla de reparto. Supongamos que i abandona la escena, pero el resto de usuarios pueden conectarse a través de él, aunque únicamente para conectarse al nodo fuente y no con otros usuarios. En este caso, el coste de las conexiones en el problema *mcst* (N_0, C') resultante viene dado por

- $c'_{j0} = min\{c_{j0}, c_{ji} + c_{i0} - x_i\}$ para todo $j \neq i$.

- $c'_{jk} = c_{jk}$ si $\{j,k\} \cap \{i,0\} = \emptyset$.

De la primera expresión se extrae que el coste de conectarse a la fuente de cada usuario j será la opción más barata de entre conectarse directamente o hacerlo a través de i, restando la cantidad x_i. El resto de conexiones mantienen su coste, pues únicamente se permite la utilización de las aristas incidente en i para conectarse a la fuente. Denotemos como $C^{sr}_{x_i}$ a la matriz resultante de aplicar el procedimiento anterior.

Una alternativa a lo anterior sería permitir que dos usuarios pudiesen conectarse entre ellos a través de i. Entonces el coste asociado a este problema vendría dado por:

$$c''_{jk} = min\{c_{jk}, c_{ji} + c_{ki} - x_i\} \text{ para cada } j,k \in N_0 \backslash \{i\}$$

Denotemos por $C^{tr}_{x_i}$ a la matriz que contiene los costes c''_{jk}. Definiremos ahora dos propiedades:

Definición 2.2.10. *Sea (N_0, C) un problema mcst con $C \in \mathcal{C}^1$, $T = (N_0, E_T)$ el árbol generador de coste mínimo asociado, $i \in N$ y $(0,i) \in E_T$. Diremos que una regla de reparto ψ satisface la propiedad **consistencia de fuente** si verifica que*

$$\psi_k((N\backslash\{i\})_0, C^{sr}_{\psi_i(N_0,C)}) = \psi_k(N_0, C) \textit{ para todo } k \in N\backslash\{i\}$$

siempre que $C^{sr}_{\psi_i(N_0,C)} \in \mathcal{C}^1$.

Definición 2.2.11. *Sea (N_0, C) un problema mcst con $C \in \mathcal{C}^2$, $T = (N_0, E_T)$ el árbol generador de coste mínimo asociado, $i \in N$ y $(0,i) \in E_T$. Diremos que una regla de reparto ψ satisface la propiedad **consistencia de árbol** si verifica que*

$$\psi_k((N\backslash\{i\})_0, C^{tr}_{\psi_i(N_0,C)}) = \psi_k(N_0, C) \textit{ para todo } k \in N\backslash\{i\}$$

siempre que $C^{tr}_{\psi_i(N_0,C)} \in \mathcal{C}^2$.

Ahora sí, podemos ver la caracterización de la regla realizada por Dutta y Kar:

Teorema 2.2.12. *Sobre el dominio de matrices $\mathcal{C}^1$, una regla de reparto ψ satisface las propiedades monotonicidad de hojas y consistencia de fuente si y solo si $\psi = B$.*

Demostración. Ver [12], teorema 3, página 241. □

Otras caracterizaciones de la regla de Bird han sido realizadas por Feltkamp [14] (teorema 2.27, página 18) y Özsoy [20] (teorema 1, página 6).

2.3. Regla de Kar

Otra de las reglas para el reparto de costes en problemas *mcst* introducidas en la literatura es la regla de Kar. Dado un problema *mcst*, la regla de Kar se define como el valor de Shapley del juego *mcst* asociado. La axiomatización de esta regla se debe a A. Kar [18]. En esta sección se estudiará la definición de la regla, se comprobará que propiedades de las introducidas en la primera sección del capítulo verifica y se mostrará un resultado de caracterización.

Recordemos, el valor de Shapley venía dado por la expresión:

$$Sh_i(N,c) = \frac{1}{|N|!} \sum_{\pi \in \Pi_N} \left[c\left(P_\pi(i) \cup \{i\}\right) - c\left(P_\pi(i)\right) \right]$$

donde:

- $P_\pi(i) = \{j \in N : \pi(j) < \pi(i)\}$ denota el conjunto de predecesores de i respecto de la permutación π.
- En el caso concreto de los juegos *mcst*, si $S \subseteq N$, $c(S) = c_{mcst}(S_0)$ representa el coste de un árbol generador de coste mínimo asociado al problema (S_0, C).

Definición 2.3.1. *Sea un problema mcst (N_0, C) y sea (N,c) el respectivo juego asociado. Se define la **regla de Kar**, la cual denotaremos K, como:*

$$K(N_0, C) = Sh(N, c)$$

Observación 2.3.2. *Observemos que, en efecto, el vector de costes propuesto por la regla de Kar es eficiente para todo problema mcst (N_0, C), pues por el teorema 1.2.19 el valor de Shapley de todo juego de coste es eficiente.*

Veamos ahora un ejemplo de aplicación de la regla de Kar:

Ejemplo 2.3.3. *Consideremos de nuevo el problema mcst (N_0, C), con $N_0 = \{0, 1, 2, 3\}$ y la matriz de costes dada por*

$$C = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 1 & 0 & 4 & 2 \\ 3 & 4 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

Considerando la permutación $\pi = (1, 2, 3)$ se tiene que el primer usuario en conectarse al servicio debe ser el número 1. Así, $c(\{1\}) - c(\emptyset) = 1$. Tras ello, debe conectarse el usuario 2, para el cual $c(\{1,2\}) - c(\{1\}) = 3$, y por último, $c(\{1,2,3\}) - c(\{1,2\}) = 0$. Se obtiene entonces *el vector de costes* $(1,3,0)$. *Procediendo análogamente para cada permutación se obtiene:*

π		m_c^π
$(1,2,3)$	$\longrightarrow$	$(1,3,0)$
$(1,3,2)$	$\longrightarrow$	$(1,1,2)$
$(2,1,3)$	$\longrightarrow$	$(1,3,0)$
$(2,3,1)$	$\longrightarrow$	$(0,3,1)$
$(3,1,2)$	$\longrightarrow$	$(0,1,3)$
$(3,2,1)$	$\longrightarrow$	$(0,1,3)$

Por tanto,

$$K(N_0,C) = \frac{1}{6}\sum_{\pi\in\Pi_N} m_c^\pi = (\frac{1}{2},2,\frac{3}{2}).$$

A continuación veremos las propiedades que verifica la regla de Kar en el siguiente teorema, que puede encontrarse en [3]. Para la demostración de dicho teorema necesitaremos enunciar un lema previo:

Lema 2.3.4. *Un árbol $T=(N_0,E_T)$ es un árbol generador de coste mínimo asociado a un problema (N_0,C) si y solo si para todo $S\subseteq N_0$ con $S\neq N_0$ y $S\neq\emptyset$ existe una arista $(i,j)\in E_T$ con $i\in S$, $j\in N_0\backslash S$ tal que $c_{ij}=min\{c_{hl}: h\in S, l\in N_0\backslash S\}$.*

Demostración. Ver [3], lema 1, página 253. □

Teorema 2.3.5. *La regla de Kar satisface las propiedades MC, SIM, IRCE e IBC. Sin embargo, no satisface SN, MP, SOL, POS, SEP, IAI ni IGC.*

Demostración. ***K* satisface *MC*:** Sean dos problemas *mcst* (N_0,C), (N_0,C') para los que se verifica $c_{ij}<c'_{ij}$ para ciertos $i\in N$, $j\in N_0$ y $c_{kl}=c'_{kl}$ para el resto de subíndices. Sean (N,c) y (N,c') los juegos asociados a dichos problemas. Entonces, para cada permutación $\pi\in\Pi_N$, es claro que se verifica que

$$m_c^\pi(i) = c(P_\pi(i)\cup\{i\}) - c(P_\pi(i)) \leq c'(P_\pi(i)\cup\{i\}) - c'(P_\pi(i)) = m_{c'}^\pi(i)$$

donde, recordemos, $P_\pi(i)=\{j\in N:\pi(j)<\pi(i)\}$. Por tanto,

$$K_i(N_0,C) = Sh_i(N,c) = \frac{1}{|N|!}\sum_{\pi\in\Pi_N} m_c^\pi(i) \leq \frac{1}{|N|!}\sum_{\pi\in\Pi_N} m_{c'}^\pi(i) = Sh_i(N,c') = K_i(N_0,C').$$

***K* satisface *SIM*:** Si los usuarios son simétricos en el problema (N_0,C), entonces son simétricos en el juego (N,c). Ahora bien, el valor de Shapley satisface la simetría (ver [23]), por lo que se obtiene el resultado.

***K* satisface *IRCE*:** Sean (N_0,C) y (N_0,C') dos problemas *mcst* en las condiciones de la defnción 2.1.13 y sean (N,c) y (N,c') los respectivos juegos derivados. Es fácil ver que $c'(S) = c_{mcst}(S_0,C') = c_{mcst}(S_0,C) + (c'_0 - c_0) = c(S) + (c'_0 - c_0)$ para toda coalición $S \subseteq N$. Por tanto:

$$K_i(N_0,C') = Sh_i(N,c') = Sh_i(N,c) + \frac{c'_0 - c_0}{|N|} = K_i(N_0,C) + \frac{c'_0 - c_0}{|N|}$$

luego, en efecto, K satisface *IRCE*.

***K* satisface *IBC*:** Sean (N_0,C), (N_0,C') e $i \in N$ en las condiciones de la definición de *IBC* (definición 2.1.11).

Supongamos, en primer lugar, que existe una única arista (j,k) tal que $c_{jk} \neq c'_{jk}$. Podemos suponer que $c_{jk} > c'_{jk}$. Como $K(N_0,C) = Sh(N,c)$ y $c(S) = c_{mcst}(S_0,C)$ para todo $S \subseteq N$, es suficiente probar que

$$c_{mcst}(S_0,C) - c_{mcst}((S\backslash\{i\})_0,C) = c_{mcst}(S_0,C') - c_{mcst}((S\backslash\{i\})_0,C')$$

con $i \in S$. Si los usuarios $\{j,k\} \nsubseteq S$, entonces el resultado es claro. Supongamos entonces que $\{j,k\} \subseteq S$. Sea $T_{S_0} = (S_0, E_{S_0})$ un árbol generador de coste mínimo asociado al problema (S_0,C). Consideremos el conjunto R de usuarios para los que i es un predecesor inmediato en el árbol T_{S_0} . Es decir, R es el conjunto de usuarios que se conectan al servicio inmediatamente a través de i. Podemos asumir, sin pérdida de generalidad que $R = \{1,...,r\}$. En caso de que i sea una hoja, R será vacío. Dado $l \in R$, definamos R^l como el conjunto de usuarios de S que están conectados al nodo fuente en el árbol T_{S_0} a través de l, aunque no nesariamente inmediatamente a través l. Es decir, R^l es el conjunto de usuarios p tales que l está en el único camino de p al nodo fuente en T_{S_0}. Consideramos que $l \in R^l$. Definamos ahora el conjunto

$$R^0 = (S_0\backslash\{i\})\backslash \bigcup_{l\in R} R^l$$

Para cada $l \in R$, consideremos el problema *mcst* $((R^l\backslash\{l\})_0, C^{+l})$ que surge al considerar que el usuario l está conectado a la fuente y el resto de usuarios pueden conectarse a través de él. Formalmente, para todo $p \in R^l\backslash\{l\}$, $c^{+l}_{0p} = min\{c_{lp}, c_{0p}\}$ y $c^{+l}_{qp} = c_{qp}$ cuando $q \neq 0$. Se verifica entonces la siguiente igualdad:

$$c_{mcst}(S_0,C) = c_{mcst}(R^0,C) + \sum_{l\in R} c_{mcst}\left((R^l\backslash\{l\})_0, C^{+l}\right) + \sum_{l\in R} c_{il} + c_{i^0 i}$$

donde (i^0,i) es la primera arista en el único camino de i al nodo fuente en el árbol generador de coste mínimo T_{S_0}. Sea $p^*_0 \in R^0$ y $p_1 \in \bigcup_{l\in R} R^l$ tal que el coste de la conexión entre ambos cumpla que

$$c_{p^*_0 p_1} = min\{c_{qs} : q \in R^0 \text{ y } s \in \bigcup_{l\in R} R^l\}$$

Podemos asumir sin pérdida de generalidad que $p_1 \in R^1$. Sea ahora $p^*_1 \in R^0 \cup R^1$ y $p_2 \in \bigcup_{R\backslash\{l\}} R^l$ tal que el coste de la conexión entre ambos cumpla que

$$c_{p^*_1 p_2} = min\{c_{qs} : q \in R^0 \cup R^1 \text{ y } s \in \bigcup_{l\in R\backslash\{l\}} R^l\}$$

De nuevo, podemos asumuir que $p_2 \in R^2$. Siguiendo este procedimiento, se obtiene un conjunto de aristas $\{(p^*_{l-1}, p_l)\}_{l\in R}$ tal que $p_l \in R^l$ para todo $l = 1, ..., r$. Ahora, aplicando el lema 2.3.4, se obtiene que el árbol $T^* = (S_0\backslash\{i\}, E_{T^*})$, con

$$E_{T^*} = \left(E_{S_0}\backslash\left(\{(i^0, i)\} \cup \bigcup_{l\in R}\{(i, l)\}\right)\right) \bigcup \{(p^*_{l-1}, p_l)\}_{l\in R}$$

es un árbol generador de mínimo coste asociado al problema $(S_0\backslash\{i\}, C)$. Así,

$$c_{mcst}(S_0, C) - c_{mcst}(S_0\backslash\{i\}, C) = \sum_{l\in R} c_{il} + c_{i^0 i} - \sum_{l\in R} c_{p^*_{l-1}p_l}$$

Probaremos ahora que este valor coincide con $c_{mcst}(S_0, C') - c_{mcst}(S_0\backslash\{i\}, C')$. Recordemos que se satisface que $c_{il} = c'_{il}$ para todo $l \in N_0$, $i \notin \{j, k\}$. Distingamos dos casos:

1. Existe $l \in \{0, 1, ..., r\}$ tal que $\{j, k\} \subseteq R^l$. Consideremos a su vez tres subcasos:

   a) $(j, k) \in E_{S_0}$. Como $c'_{jk} < c_{jk}$, T_{S_0} y T^* son árboles generadores de mínimo coste de los problemas (S_0, C') y $(S_0\backslash\{i\}, C')$ respectivamente y $(j, k) \in E_{T^*}$, se tiene que

   $$\begin{aligned} c_{mcst}(S_0, C') - c_{mcst}(S_0\backslash\{i\}, C') &= \sum_{l\in R} c'_{il} + c'_{i^0 i} - \sum_{l\in R} c'_{p^*_{l-1}p_l} \\ &= \sum_{l\in R} c_{il} + c_{i^0 i} - \sum_{l\in R} c_{p^*_{l-1}p_l} \\ &= c_{mcst}(S_0, C) - c_{mcst}(S_0\backslash\{i\}, C). \end{aligned}$$

   b) $(j, k) \notin E_{S_0}$ y $\{j, k\} \subseteq R^l$ con $l \neq 0$. Consideremos el grafo $\hat{T} = (S_0, E_{S_0} \cup \{(j, k)\})$. Como T_{S_0} es un árbol, aplicando el teorema 1.1.14, $\hat{T}$ contiene un ciclo que denotaremos g. Además, $(i, l) \notin g$, ya que $\{j, k\} \subseteq R^l$. Así, (i, l) pertenece al único camino que conecta j con el nodo 0 y también al único camino que conecta k con 0. Aplicando el lema 2.3.4, eliminando la arista de mayor coste en g se obtiene un árbol generador de coste mínimo $T' = (S_0, E_{T'})$ asociado al problema (S_0, C'). Observemos que $\{(i, l)\}_{l\in R} \in E_{T'}$, $(i^0, i) \in E_{T'}$ y $\{(p^*_{l-1}p_l)\}_{l\in R} \in E_{T'}$. Usando un argumento similar al llevado acabo en el apartado anterior se obtiene que

   $$c_{mcst}(S_0, C') - c_{mcst}(S_0\backslash\{i\}, C') = c_{mcst}(S_0, C) - c_{mcst}(S_0\backslash\{i\}, C).$$

   c) $(j, k) \notin E_{S_0}$ y $\{j, k\} \subseteq R^0$. Consideremos el grafo $\hat{T} = (S_0, E_{S_0} \cup \{(j, k)\})$. Como T_{S_0} es un árbol, aplicando el teorema 1.1.14, $\hat{T}$ contiene un ciclo g. Usando un razonamiento análogo que el anterior caso obtenemos que

   $$c_{mcst}(S_0, C') - c_{mcst}(S_0\backslash\{i\}, C') = c_{mcst}(S_0, C) - c_{mcst}(S_0\backslash\{i\}, C).$$

2. $j \in R^l$, $k \in R^q$ con $l \neq q$. Se tiene entonces que $l \neq 0$ o $q \neq 0$. Supongamos $l \neq 0$. Entonces tomando $\hat{T} = (S_0, \hat{E})$, con

   $$\hat{E} = (E_{S_0}\backslash\{(i, l)\}) \cup \{(j, k)\}$$

$\hat{T}$ es un árbol asociado al problema (S_0, C) cuyo coste, $c(\hat{T})$, será igual a $c(T_{S_0}) - c_{il} + c_{jk}$. Como T_{S_0} es un árbol generador de mínimo coste asociado al problema (S_0, C) y $c_{jk} \leq c_i^{min}$, podemos concluir que $c_{jk} = c_{il} = c_i^{min}$ y $\hat{T}$ es también un árbol generador de coste mínimo de (S_0, C). Podemos calcular $\hat{R}$ y $\{\hat{R}^l\}_{l \in R \cup \{0\}}$ en $\hat{T}$ de la misma forma que se ha calculado R y $\{R^l\}_{l \in R}$ para T_{S_0}. Ahora, existe $l \in R \cup \{0\}$ tal que $\{j, k\} \subseteq \hat{R}^l$ y puede procederse como en el caso 1.

Así, se ha probado el resultado para el caso en que existe una única arista (j, k) tal que $c_{jk} \neq c'_{jk}$. Para el caso en el que existe más de una arista de diferente coste basta proceder de la misma forma que en el teorema 2.2.7 para ver que B satisface IBC.

***K* no satisface *SN*:** Consideremos el problema *mcst* (N_0, C), donde $N = \{1, 2, 3\}$ y la matriz de costes es la siguiente:

$$C = \begin{pmatrix} 0 & 4 & 12 & 4 \\ 4 & 0 & 4 & 30 \\ 12 & 4 & 0 & 4 \\ 4 & 30 & 4 & 0 \end{pmatrix}$$

Consideremos la coalición $S = \{1, 2\}$. El vector de costes que resulta de aplicar la regla de Kar es $K(N_0, C) = \left(\frac{8}{3}, \frac{20}{3}, \frac{8}{3}\right)$. Ahora bien

$$\sum_{i=1}^{2} K(N_0, C) = \frac{28}{3} > 8 = c_{mcst}(S_0)$$

Por tanto, K no satisface *SN*.

***K* no satisface *IGC*:** Consideremos los problema *mcst* (N_0, C) y (N_0, C') donde $N = \{1, 2\}$ y las matrices de costes son las siguientes:

$$C = \begin{pmatrix} 0 & 10 & 10 \\ 10 & 0 & 2 \\ 10 & 2 & 0 \end{pmatrix} \qquad C' = \begin{pmatrix} 0 & 10 & 12 \\ 10 & 0 & 2 \\ 12 & 2 & 0 \end{pmatrix}$$

Así, tomando ambos problemas y seleccionando el usuario 1, estamos en las condiciones de la definición de *IGC* (definición 2.1.12), sin embargo, $K_1(N_0, C) = 6$ y $K_1(N_0, C') = 5$.

***K* no satisface *IAI*:** Como K no satisface *IGC*, aplicando la proposición 2.1.21 se obtiene el resultado.

***K* no satisface *SOL*:** Como K no satisface *IAI*, aplicando la proposición 2.1.18 se obtiene el resultado.

***K* no satisface *MP*:** K no satisface *SN*, por tanto, aplicando la proposición 2.1.19 se obtiene el resultado.

***K* no satisface *POS*:** Consideremos el problema (N_0, C), donde $N = \{1, 2\}$ y los costes vienen dados por la matriz

$$C = \begin{pmatrix} 0 & 1 & 10 \\ 1 & 0 & 2 \\ 10 & 2 & 0 \end{pmatrix}$$

Entonces, aplicando la regla de Kar se obtiene que $K_1(N_0, C) = -3$. Por tanto, K no es positiva.

***K* no satisface *SEP*:** Sea el problema (N_0, C), con $N = \{1, 2, 3\}$ y la matriz de costes dada por

$$C = \begin{pmatrix} 0 & 10 & 100 & 20 \\ 10 & 0 & 10 & 100 \\ 100 & 10 & 0 & 40 \\ 20 & 100 & 40 & 0 \end{pmatrix}$$

Tomemos la coalición $S = \{1, 2\}$. Se tiene que

$$c_{mcst}(S_0, C) + c_{mcst}((N \backslash S)_0, C) = 20 + 20 = 40 = c_{mcst}(N_0, C).$$

Sin embargo, $K_1(S_0, C) = -35 \neq -15 = K_1(N_0, C)$. □

Observación 2.3.6. *Del teorema anterior se obtiene que en la clase de juegos mcst, el valor de Shapley no pertenece necesariamente al núcleo.*

A continuación veremos un importante resultado que relaciona la regla de Bird y la regla de Kar para problemas *mcst* en forma irreducible:

Proposición 2.3.7. *Sea (N_0, C^*) un problema mcst. Si (N_0, C^*) es una forma irreducible entonces*

$$K(N_0, C^*) = B(N_0, C^*).$$

Demostración. Ver [2], proposición 3.4, página 337. □

Por tanto, sobre la clase de formas irreducibles, las reglas de Bird y Kar coinciden.

La última parte de la seción estará dedicada a estudiar una caracterización de la regla. Con el objeto de realizar dicha caracterización, Kar consideró propiedades de reglas de reparto diferentes a las propuestas en la primera sección del capítulo. A continuación introduciremos las propiedades consideradas por Kar, que nos permitirán mostrar el teorema de caracterización de K.

Definición 2.3.8. *Diremos que un grafo es un* ***grafo estrella*** *si existe un nodo que está conectado al resto de nodos y no existen otras conexiones en el grafo. Al nodo conectado con el resto se le denominará* ***centro*** *.*

Definición 2.3.9. *Una regla de reparto ψ satisface la propiedad de* ***ausencia de subsidio cruzado*** *si para todo problema mcst (N_0, C) en el que el árbol generador de coste mínimo sea un grafo estrellado en el que el centro es la fuente que proporciona el servicio se verifica*

$$\psi_i(N_0, C) = c_{i0} \text{ para todo } i \in N.$$

La anterior propiedad establece que si el árbol generador de coste mínimo de un problema es un grafo estrella con cuyo centro es la fuente, entonces la regla asigna a cada usuario i el coste de la conexión entre i y la fuente.

Antes de introducir la siguiente propiedad necesitaremos algunas definiciones previas:

Definición 2.3.10. *Diremos que una conexión entre dos usuarios i, j es* ***irrelevante*** *si $c_{ij} > max(c_{i0}, c_{j0})$. Una conexión que no es irrelevante recibirá el nombre de conexión* ***relevante****.*

Observación 2.3.11. *Es claro que un árbol generador de coste mínimo de una red cualquiera no contendrá conexiones irrelevantes.*

Definición 2.3.12. *Se dice que existe un* ***camino relevante*** *entre el usuario i y el j si existen $i_1, ..., i_K \in N$ para los que (i_k, i_{k+1}) es una conexión relevante para cada $k < K$ y se tiene que $i_1 = i$, $i_K = j$.*

Definición 2.3.13. *Supongamos que existe una partición del conjunto de usuarios $N = [N_1, ..., N_p]$ tal que*

1. *Si $i \in N_k$, $j \in N_k$ entonces existe un camino relevante de i a j en N_k.*

2. *Si $i \in N_k$, $j \in N_l$ con $k \neq l$ entonces no existe un camino relevante de i a j en N.*

Entonces, a cada elemento de la partición se le denominará ***grupo****.*

Estamos, ahora sí, en condiciones de introducir la siguiente propiedad:

Definición 2.3.14. *Una regla de reparto ψ satisface la propiedad* ***independencia de grupos*** *si dado un problema (N_0, C) en el que el conjunto de usuarios está particionado en p grupos, $N = [N_1, ..., N_p]$, tomando una matriz de coste C' y un par de usuarios $i, j \in N_l$, tales que $c'_{mn} = c_{mn}$ para todo $(m, n) \neq (i, j)$, se verifica que*

$$\psi_k(N_0, C) = \psi_k(N_0, C') \text{ para todo } k \in N_t \text{ y todo } t \neq l.$$

Que una regla de reparto satisfaga la independencia de grupos implica que si existe un cambio en el coste de una conexión dentro de un determinado grupo, esto no afecta a los agentes de otros grupos.

Definición 2.3.15. *Sean dos problemas mcst* (N_0, C) *y* (N_0, C') *tales que* $c_{mn} = c'_{mn}$ *para todo* $(m, n) \neq (i, j)$*, donde* $i, j \in N$*. Una regla de reparto* ψ *satisface la propiedad de* ***tratamiento igualitario*** *si satisface que*

$$\psi_i(N_0, C') - \psi_i(N_0, C) = \psi_j(N_0, C') - \psi_j(N_0, C).$$

Pensemos una situación inicial en el que los costes vienen dados por la matriz C. Supongamos que se mantienen los costes de todas las conexiones excepto la conexión (i, j). La propiedad anterior implica que la regla de reparto divide el coste igualitariamente entre los usuarios i y j.

Estamos ahora sí, en condiciones de enunciar el teorema de caracterización de la regla de Kar, o equivalentemente, del valor de Shapley de un juego *mcst*:

Teorema 2.3.16. *La regla de Kar es la única regla de reparto de costes en la clase de los problemas mcst que satisface las propiedades ausencia de subsidio cruzado, independencia de grupos y tratamiento igualitario.*

Demostración. Ver [18], teorema 1, página 271. □

Además de esta caracterización para el valor de Shapley de juegos *mcst*, en el teorema 1.2.19 vimos una caracterización para el valor de Shapley de juegos de coste arbitrarios.

2.4. Regla de Dutta-Kar

La regla de Dutta-Kar fue introducida por B. Dutta y A. Kar [12] en 2004. Las dos reglas anteriormente introducidas, la regla de Bird y la regla de Kar, son reglas que no cumplen simultáneamente las propiedades *selección del núcleo* y *monotonicidad del coste*, dos propiedades de gran importancia para una regla de reparto. En la presente sección veremos que la regla de Dutta-Kar cumple ambas propiedades. Veamos cómo se introduce la regla:

Recordemos que denotamos como $\mathcal{C}^1$ al conjunto de matrices tales que el problema (N_0, C) tiene un único árbol generador de coste mínimo, siendo N un conjunto arbitrario de usuarios. Denotábamos también $\mathcal{C}^2$ al conjunto de matrices de $\mathcal{C}^1$ para las no existen dos aristas del mismo coste en el árbol generador de mínimo coste asociado a (N_0, C). La regla de Dutta-Kar, que será denotada como *DK*, se define a través del algoritmo de Prim mediante el método que introduciremos a continuación. Consideraremos en primer lugar un problema *mcst* (N_0, C) con $C \in \mathcal{C}^2$ y denotaremos por A_c al complementario de una coalición $A \subseteq N$, es decir, $A_c = N_0 \backslash A$. El método es el siguiente:

Sea $A^0 = \{0\}$, $g^0 = \emptyset$, $t^0 = 0$.

Etapa 1: Seleccionamos el par ordenado (a^1, b^1) tal que

$$(a^1, b^1) = argmin\ c_{ij} \text{ con } (i, j) \in A^0 \times A^0_c.$$

Definimos $t^1 = max(t^0, c_{a^1b^1})$, $A^1 = A^0 \cup \{b^1\}$, $g^1 = g^0 \cup \{(a^1, b^1)\}$.

Etapa k: Seleccionamos el par ordenado (a^k, b^k) tal que

$$(a^k, b^k) = argmin\ c_{ij} \text{ con } (i,j) \in A^{k-1} \times A_c^{k-1}.$$

Tomamos $t^k = max(t^{k-1}, c_{a^kb^k})$, $A^k = A^{k-1} \cup \{b^k\}$, $g^k = g^{k-1} \cup \{(a^k, b^k)\}$ y definimos

$$DK_{b^{k-1}}(N_0, C) := min(t^{k-1}, c_{a^kb^k}).$$

El algoritmo termina cuando $k = |N| = n$, donde $DK_{b^n}(N_0, C) = t^n$.

En cualquier paso k, la coalición A^{k-1} es el conjunto de nodos que ya han sido conectados a la fuente. En dicho paso, se construye una conexión tomando la arista de menor coste entre un nodo de A^{k-1} y otro de A_c^{k-1}. El coste que la regla asigna al usuario b^{k-1} se decide en el paso k, tomando el mínimo de t_{k-1} (que es el coste máximo de las conexiones establecidas en pasos anteriores) y $c_{a^kb^k}$ que es el coste de la conexión establecida en el paso k. Finalmente, se tiene que el último usuario en ser conectado, b^n, deberá pagar el máximo coste de entre todos los usuarios. Observemos que g^n es un conjunto que contiene las aristas del árbol generador de coste mínimo asociado al problema.

Observación 2.4.1. *Para la aplicación del algoritmo y la definición de la regla de Dutta-Kar, se ha tomado un problema (N_0, C) con $C \in \mathcal{C}^2$. Supongamos ahora que $C \notin \mathcal{C}^2$. En este caso se tiene que el algoritmo no está bien definido, pues en cierto paso k existirán aristas distintas (a^k, b^k) y (a'^k, b'^k) conectando A^{k-1} y A_c^{k-1} con el mismo coste. El algoritmo puede ser fácilmente extendido a matrices no necesariamente en $\mathcal{C}^2$ (ni en $\mathcal{C}^1$) tomando una permutación $\pi \in \Pi_N$ y seleccionando de entre b^k y b'^k el usuario que sea anterior respecto la permutación π. Si $b^k = b'^k$, entonces aplicamos el mismo criterio para a^k y a'^k.*

Definición 2.4.2. *Sea (N_0, C) un problema mcst arbitrario. Denotemos como $DK^\pi(N_0, C)$ la asignación de costes obtenida de aplicar el algoritmo anterior usando la permutación $\pi \in \Pi_N$ como método de desempate ante aristas del mismo coste. La* ***regla de Dutta-Kar, DK,*** *vendrá dada por*

$$DK(N_0, C) = \frac{1}{|N|!} \sum_{\pi \in \Pi} DK^\pi(N_0, C).$$

Observación 2.4.3. *Por definición, cada vector $DK^\pi(N_0, C)$ es eficiente, por tanto, $DK(N_0, C)$ verificará que*

$$\sum_{i \in N} DK_i(N_0, C) = c_{mcst}(N_0, C)$$

por lo que, en efecto, es una regla de reparto.

Veamos ahora un ejemplo de aplicación de la regla de Dutta-Kar:

Ejemplo 2.4.4. *Consideremos el problema mcst* (N_0, C)*, con* $N_0 = \{0, 1, 2, 3\}$ *y la matriz de costes dada por*

$$C = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 1 & 0 & 4 & 2 \\ 3 & 4 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

Tomemos $t^0 = 0$, $A^0 = \{0\}$.
Etapa 1: *Tomamos* $(a_1, b_1) = (0, 1)$, $t^1 = max\{t^0, c_{01}\} = 1$ *y* $A^1 = \{0, 1\}$.
Etapa 2: *Tomamos* $(a_2, b_2) = (1, 3)$, $t^2 = max\{t^1, c_{13}\} = 2$, $A^2 = \{0, 1, 3\}$ *y* $DK_1^\pi = min\{t^1, c_{13}\} = 1$.
Etapa 3: *Tomamos* $(a_3, b_3) = (3, 2)$, $t^3 = max\{t^2, c_{32}\} = 2$, $A^3 = \{0, 1, 2, 3\}$, $DK_3^\pi = min\{t^2, c_{32}\} = 1$ *y* $DK_2^\pi = t^3 = 2$.
Se obtiene así que $DK^\pi(N_0, C) = (1, 2, 1)$.

A continuación se verá un ejemplo de aplicación en el que el problema tiene más de un árbol generador de coste mínimo asociado:

Ejemplo 2.4.5. *Consideremos el problema mcst* (N_0, C)*, con* $N_0 = \{0, 1, 2, 3\}$ *y la matriz de costes dada por*

$$C = \begin{pmatrix} 0 & 6 & 3 & 3 \\ 6 & 0 & 3 & 3 \\ 3 & 3 & 0 & 1 \\ 3 & 3 & 1 & 0 \end{pmatrix}$$

Como el árbol generador de coste mínimo no es único, para obtener $DK(N_0, C)$*, deberemos calcular* $DK^\pi(N_0, C)$ *para cada* $\pi \in \Pi_N$*. Consideremos la permutación* $\pi = (1, 2, 3)$ *y tomemos* $t^0 = 0$, $A^0 = \{0\}$.
Etapa 1: *Existen dos aristas del mismo coste que podrían ser añadidas:* $(0, 2)$ *y* $(0, 3)$*, sin embargo, según* π*, el usuario 2 debe ser conectado antes que el 3, por tanto, tomamos* $(a_1, b_1) = (0, 2)$, $t^1 = max\{t^0, c_{02}\} = 3$ *y* $A^1 = \{0, 2\}$.
Etapa 2: *Tomamos* $(a_2, b_2) = (2, 3)$, $t^2 = max\{t^1, c_{23}\} = 3$, $A^2 = \{0, 2, 3\}$ *y* $DK_2^\pi = min\{t^1, c_{23}\} = 1$.
Etapa 3: *Tomamos* $(a_3, b_3) = (2, 1)$, $t^3 = max\{t^2, c_{21}\} = 3$, $A^3 = \{0, 1, 2, 3\}$, $DK_3^\pi = min\{t^2, c_{21}\} = 3$ *y* $DK_1^\pi = t^3 = 3$.
Se obtiene así que $DK^\pi(N_0, C) = (3, 1, 3)$*. Procediendo de esta forma para cada permutación se*

obtiene:

π		$DK^{\pi}(N_0,C)$
$(1,2,3)$	$\longrightarrow$	$(3,1,3)$
$(1,3,2)$	$\longrightarrow$	$(3,3,1)$
$(2,1,3)$	$\longrightarrow$	$(3,1,3)$
$(2,3,1)$	$\longrightarrow$	$(3,1,3)$
$(3,1,2)$	$\longrightarrow$	$(3,3,1)$
$(3,2,1)$	$\longrightarrow$	$(3,3,1)$

Por tanto,

$$DK(N_0,C) = \frac{1}{6}\sum_{\pi\in\Pi_N} DK^{\pi}(N_0,C) = (3,2,2).$$

Observemos que el vector de costes obtenido es igual al que se obtuvo al aplicar la regla de Bird al mismo problema en el ejemplo 2.2.6, sin embargo, $DK^{\pi}(N_0,C) \neq B^{\pi}(N_0,C)$ para cada $\pi \in \Pi_N$.

En el siguiente resultado comprobaremos las propiedades que verifica la regla de Dutta-Kar. Dicho resultado puede encontrarse en [3].

Teorema 2.4.6. *La regla de Dutta-Kar satisface las propiedades SN, MC, POS y SIM. Sin embargo no satisface las propiedades SOL, MP, SEP, IRCE, IAI, IBC ni IGC.*

Demostración. *DK* satisface *SN*: Sea (N_0,C) un problema *mcst*. Supongamos en primer lugar que $C \in \mathcal{C}^2$. Tenemos que ver que para todo $S \subseteq N$, $\sum_{i\in S} DK_i(N_0,C) \leq c(S)$. Podemos suponer, sin pérdida de generalidad, que para cada $i \in N$, $b^i = i$. Denotemos $c^k := c_{a^k b^k}$. Veamos en primer lugar que si $S = \{1,...,K\}$, entonces $\sum_{i\in S} DK_i(N_0,C) \leq c(S)$. Consideremos el grafo $T_{S_0} = \left(S_0, \bigcup_{k=1}^{K}(a^k,k)\right)$. Se tiene que T_{S_0} es un árbol generador de coste mínimo de (S_0,C), por tanto $c(S) = \sum_{k=1}^{K} c^k$. Entonces:

$$\sum_{i\in S} DK_i(N_0,C) = \sum_{k=1}^{K+1} c^k - max\{c^k : 1 \leq k \leq K+1\} \leq \sum_{k=1}^{K} c^k = c(S)$$

Por tanto, para toda coalición de la forma $S = \{1,...,K\}$ se cumple la desigualdad deseada. Supongamos que existe alguna coalición S para la cual se verifica que $\sum_{i\in S} DK_i(N_0,C) > c(S)$ y supongamos que dicha coalición es maximal respecto a esta característica, es decir, si $S \subseteq T$ entonces $\sum_{i\in T} DK_i(N_0,C) \leq c(T)$. Consideremos $T' = (S_0, E_{T'})$ un árbol generador de coste mínimo asociado a (S_0,C). Distingamos dos casos:

1. $1 \notin S$. Sea $K = min\{j : j \in S\}$. Consideremos $T := \{1,...,K-1\}$. Se tiene entonces que

$$\begin{aligned}\sum_{i\in T\cup S} DK_i(N_0,C) &= \sum_{i\in S} DK_i(N_0,C) + \sum_{i\in T} DK_i(N_0,C)\\ &> c(S) + \sum_{k=1}^{K} c^k - max\{c^k : 1 \leq k \leq K\}\\ &\geq c(S) + \sum_{k=1}^{K} c^k - c_{0s}\end{aligned}$$

donde $(0, s)$ pertenece al árbol generador de coste mínimo asociado a (S_0, C). La última desigualdad se obtiene al verificarse que $c^k \leq c_{0s}$ para todo $k \in \{1, ...K\}$. Ahora, el grafo $G = \left(T \cup S \cup \{0\}, \bigcup_{k=1}^{K} \{(a^k, b^k)\} \cup (E_{T'} \backslash \{(0, s)\}\right)$ es conexo, por tanto debe verificarse que

$$c(S) + \sum_{k=1}^{K} c^k - c_{0s} \geq c(S \cup T)$$

por tanto,

$$\sum_{i \in T \cup S} DK_i(N_0, C)) > c(S \cup T)$$

en contradicción con que S era maximal respecto de esta propiedad.

2. $1 \in S$. Por lo demostrado anteriormente, S no puede ser de la forma $S = \{1, ..., K\}$. Por tanto, podemos encontrar una partición de S, $\{S_1, ..., S_k\}$, donde cada S_k consiste en una serie de usuarios consecutivos y tal que si $i \in S_k$ y $j \in S_{k+1}$, entonces $i + 1 < j$. Denotemos $m := max\{j : j \in S_1\}$ y $r := min\{j : j \in S_2\}$. Definamos $T := \{m + 1, ..., r - 1\}$. Se tiene entonces que

$$\begin{aligned}
\sum_{i \in T \cup S} DK_i(N_0, C) &= \sum_{i \in S} DK_i(N_0, C) + \sum_{i \in T} DK_i(N_0, C) \\
&> c(S) + \sum_{i \in S_1 \cup T} DK_i(N_0, C) - \sum_{i \in S_1} DK_i(N_0, C) \\
&= c(S) + \left(\sum_{i=1}^{r} c^i - max_{1 \leq i \leq r} c^i\right) - \left(\sum_{i=1}^{m+1} c^i - max_{1 \leq i \leq m+1} c^i\right) \\
&= c(S) + \left(\sum_{i=m+2}^{r} c^i + max_{1 \leq i \leq m+1} c^i - max_{1 \leq i \leq r} c^i\right) \\
&\geq c(S) + \sum_{i=m+1}^{r} c^i - max_{1 \leq i \leq r} c^i
\end{aligned}$$

Ahora como T' es un árbol, y por tanto conexo, existe $s_2 \in S \backslash S_1$ y $s_1 \in S_1 \backslash \{0\}$ tal que $(s_1, s_2) \in E_{T'}$. Es más, $c_{s_1 s_2} \geq max_{1 \leq i \leq r} c^i$. Ahora bien, el grafo

$$G = \left(T \cup S \cup \{0\}, \bigcup_{k=m+1}^{r} \{(a^k, b^k)\} \cup (E_{T'} \backslash \{(s_1, s_2)\}\right)$$

es conexo, por lo que debe cumplirse que

$$\sum_{i \in T \cup S} DK_i(N_0, C)) > c(S) + \sum_{i=m+1}^{r} c^i - max_{1 \leq i \leq r} c^i \geq c(S \cup T)$$

obteniéndose de nuevo una contradicción con la maximalidad de S respecto a esta propiedad.

Hemos visto pues, que se verifica el resultado para matrices $C \in \mathcal{C}^2$, el resultado para matrices arbitrarias es consecuencia de que $DK(N_0, C) = \frac{1}{|N|!}\sum_{\pi \in \Pi} DK^{\pi}(N_0, C)$ es una combinación convexa de vectores que pertenecen al núcleo por lo demostrado anteriormente y de que el núcleo de un juego es un conjunto convexo.

***DK* satisface *MC*:** Sean (N_0, C) y (N_0, C') dos problemas *mcst*. Supongamos en primer lugar que $C, C' \in \mathcal{C}^2$. Para demostrar el resultado, hay que ver que si $i \in N, j \in N_0$ son tales que $c'_{ij} < c_{ij}$ y $c_{jk} = c'_{jk}$ para el resto de pares, entonces se verifica que $DK_i(N_0, C') \leq DK_i(N_0, C)$. A lo largo de la demostración utilizaremos la notación A^k, t^k, a^k, b^k, g^n para referirnos a los conjuntos y valores fruto de la aplicación del algoritmo descrito en el inicio de la sección para definir la regla de Dutta-Kar con la matriz C y usaremos la notación $A'^k, t'^k, a'^k, b'^k, g'^n$ para los conjuntos y valores relacionados con la matriz C'. Así, $T := (N_0, g^n)$ y $T' := (N_0, g'^n)$ serán árboles generadores de coste mínimo asociados a (N_0, C) y (N_0, C') respectivamente.

Si $(i, j) \notin g'^n$. Entonces $g^n = g'^n$, es decir, el árbol generador de coste mínimo asociado a ambos problemas es el mismo y las aristas tienen todas el mismo coste. Por tanto se verificará que $DK_k(N_0, C) = DK_k(N_0, C')$ para todo $k \in N$.

Veamos ahora el caso en que $(i, j) \in g'^n$. Supongamos en primer lugar que i es el predecesor inmediato de j en el único camino de 0 a j en T'. Tomemos $i = b'^{k-1}$. Como el coste del resto de aristas es igual, $A^{k-1} = A'^{k-1}$ y $t^{k-1} = t'^{k-1}$. Además, $DK_i(N_0, C') = min\{t'^{k-1}, c'_{a'_k b'_k}\}$ y $DK_i(N_0, C) = min\{t^{k-1}, c_{a^k b^k}\}$. Como $c'_{a'^k b'^k} \leq c_{a^k b^k}$, se obtiene el resultado. Supongamos ahora que i es el sucesor inmediato de j en T'. Sea $i = b^l$ e $i = b'^m$. Como $c'_{ij} < c_{ij}$, debe cumplirse que $l \geq m$, $A'^m \subseteq A^l$ y $t^l \geq t'^m$. Se cumple en este caso que $DK_i(N_0, C') = min\{t'^m, c'_{a'^{m+1} b'^{m+1}}\}$ y $DK_i(N_0, C) = min\{t^l, c_{a^{l+1} b^{l+1}}\}$. Como $t^l \geq t'^m$, debemos probar que $c'_{a'^{m+1} b'^{m+1}} \leq c_{a^{l+1} b^{l+1}}$. Distigamos los siguientes casos:

1. $a^{l+1} \in A'^m$. Como $b^{l+1} \in N_0 \backslash A'^m$, entonces $c'_{a'^{m+1} b'^{m+1}} \leq c'_{a^{l+1} b^{l+1}} \leq c_{a^{l+1} b^{l+1}}$. Obteniendo así lo deseado.

2. $a^{l+1} \notin A'^m$. Entonces se tiene que $a^{l+1} \neq i$ y $a^{l+1} \in A^l$, por tanto, $c_{a^{l+1} b^{l+1}} \geq c_{a^l b^l}$. Consideremos dos subcasos:

- $a^l \in A^{l-1} \backslash A'^{m-1}$. Entonces, como $A^l = A^{l-1} \cup \{i\}$ y $A'^m = A'^{m-1} \cup \{i\}$, se tiene que $a^l \in A^l \backslash A'^m$. Ahora, como $i \in A'^m$ y $a^l \notin A'^m$ se tiene que

$$c'_{a'^{m+1} b'^{m+1}} \leq c'_{ia^l} \leq c_{ia^l} = c_{a^l b^l}$$

  por lo que

$$c'_{a'^{m+1} b'^{m+1}} \leq c_{a^l b^l} \leq c_{a^{l+1} b^{l+1}}.$$

- $a^l \in A'^{m-1} = A^{m-1}$. Entonces, se verifica que $c_{a^m b^m} \leq c_{a^l b^l}$, pues $m \leq l$. Además, que $A'^m \subseteq A^l$ y $a^{l+1} \in A^l \backslash A'^m$ implica que $|A'^m| < |A^l|$, y por tanto que $m < l$. Así,

$b^m \neq i = b^l$, por lo que $b^m \notin \left(A'^{m-1} \cup \{i\}\right) = A'^m$.

Por otro lado, $a^m \in A^{m-1} = A'^{m-1}$, por lo que $a^m \in A'^m$. Pero esto, junto con que $b^m \notin A'^m$, implica que

$$c'_{a'^{m+1}b'^{m+1}} \leq c'_{a^m b^m} \leq c_{a^m b^m}$$

por lo que

$$c'_{a'^{m+1}b'^{m+1}} \leq c_{a^m b^m} \leq c_{a^l b^l} \leq c_{a^{l+1}b^{l+1}}.$$

Se ha demostrado el resultado para el caso en el que las matrices pertenecen a $\mathcal{C}^2$. El resultado para el caso general es consecuencia de que cada par de vectores $DK^\pi(N_0, C)$ y $DK^\pi(N_0, C')$ cumplen la propiedad para cada $\pi \in \Pi_N$, por tanto, se verificará también para $DK(N_0, C)$ y $DK(N_0, C')$.

***DK* satisface *POS*:** Claro, teniendo en cuenta que los costes de las conexiones son positivos y la definición de *DK*.

***DK* satisface *SIM*:** La demostración es análoga a la realizada en el teorema 2.2.7 para ver que la regla de Bird satisface *SIM*.

***DK* no satisface *IGC*:** Consideremos los problema *mcst* (N_0, C) y (N_0, C') donde $N = \{1, 2\}$ y las matrices de costes son las siguientes:

$$C = \begin{pmatrix} 0 & 10 & 10 \\ 10 & 0 & 2 \\ 10 & 2 & 0 \end{pmatrix} \qquad C' = \begin{pmatrix} 0 & 10 & 12 \\ 10 & 0 & 2 \\ 12 & 2 & 0 \end{pmatrix}$$

Así, tomando ambos problemas y seleccionando el agente 1, estamos en las condiciones de la definición de *IGC* (definición 2.1.12), sin embargo, $DK_1(N_0, C) = 6$ y $DK_1(N_0, C') = 2$.

***DK* no satisface *IAI*:** Como *DK* no satisface *IGC*, aplicando la proposicón 2.1.21 se obtiene el resultado.

***DK* no satisface *SOL*:** Como *DK* no satisface *IAI*, aplicando la proposición 2.1.18 se obtiene el resultado.

***DK* no satisface *SEP*:** Sea el problema *mcst* (N_0, C), con $N = \{1, 2, 3\}$, $S = \{1, 2\}$ y

$$C = \begin{pmatrix} 0 & 3 & 10 & 1 \\ 3 & 0 & 1 & 10 \\ 10 & 1 & 0 & 3 \\ 1 & 10 & 3 & 0 \end{pmatrix}$$

En este caso se tiene que

$$c_{mcst}(S_0, C) + c_{mcst}((N \backslash S)_0, C) = 4 + 1 = 5 = c_{mcst}(N_0, C).$$

Ahora bien, $DK(N_0, C) = (2, 2, 1)$ y $DK(S_0, C) = (1, 3)$, es decir, $DK_1(N_0, C) \neq DK_1(S_0, C)$ y $DK_2(N_0, C) \neq DK_2(S_0, C)$, por lo que B no satisface *SEP*.

***DK* no satisface *MP*:** Como no satisface *SEP*, aplicando la proposición 2.1.19 se obtiene el resultado.

***DK* no satisface *IRCE*:** Consideremos los problema *mcst* (N_0, C) y (N_0, C') donde $N = \{1, 2\}$ y las matrices de costes son las siguientes:

$$C = \begin{pmatrix} 0 & 10 & 10 & 10 \\ 10 & 0 & 10 & 10 \\ 10 & 10 & 0 & 6 \\ 10 & 10 & 6 & 0 \end{pmatrix} \qquad C' = \begin{pmatrix} 0 & 16 & 16 & 16 \\ 16 & 0 & 10 & 10 \\ 16 & 10 & 0 & 6 \\ 16 & 10 & 6 & 0 \end{pmatrix}$$

Observemos que en C' aumentan las conexiones de los usuarios con la red en 6 unidades. Aplicando DK se obtienen los vectores de costes $DK(N_0, C) = (10, 8, 8)$ y $DK(N_0, C') = (14, 9, 9)$, es decir, $DK(N_0, C') \neq DK(N_0, C) + (2, 2, 2)$, por lo que DK no satisface *IRCE*.

***DK* no satisface *IBC*:** Consideremos los problema *mcst* (N_0, C) y (N_0, C') donde $N = \{1, 2, 3\}$ y las matrices de costes son las siguientes:

$$C = \begin{pmatrix} 0 & 100 & 110 & 120 \\ 100 & 0 & 6 & 6 \\ 110 & 6 & 0 & 10 \\ 120 & 6 & 10 & 0 \end{pmatrix} \qquad C' = \begin{pmatrix} 0 & 100 & 110 & 120 \\ 100 & 0 & 3 & 6 \\ 110 & 3 & 0 & 10 \\ 120 & 6 & 10 & 0 \end{pmatrix}$$

Así, ambos problemas junto con el usuario 3, se encuentran en las condiciones de la definición de *IBC* (definición 2.1.11), sin embargo, $DK_3(N_0, C) = 53$ y $DK_3(N_0, C') = 100$. □

Dutta y Kar caracterizaron la regla en términos de las propiedades *monotonicidad de hojas* (definición 2.2.8) y *consistencia de árbol* (definición 2.2.11):

Teorema 2.4.7. *Sobre el dominio de matrices $\mathcal{C}^2$, una regla de reparto ψ satisface las propiedades monotonicidad de hojas y consistencia de árbol si y solo si $\psi = DK$.*

Demostración. Ver [12], teorema 2, página 236. □

2.5. Regla de Feltkamp-Tijs-Muto

La siguiente sección estará dedicada a la regla de Feltkamp-Tijs-Muto. En ella se definirá dicha regla, se estudiarán las propiedades que verifica, se mostrará un ejemplo de aplicación y se enunciará un resultado para su caracterización. Feltkamp, Tijs y Muto [13], definen un regla de reparto que, a diferencia de la regla de Bird y la regla de Dutta-Kar que se definen a partir del algoritmo de Prim, se vale del algoritmo de Kruskal para construir un vector de costes que denominan el ***ERO* valor** (*equal remaining obligations*). Presentemos a continuación el algoritmo para obtener dicho valor:

Sea (N_0, C) un problema *mcst*.

Etapa 1: Tomar $t = 0$, $\tau = |N_0| - 1$ y $E_0 = \emptyset$.

Etapa 2: $t := t + 1$.

Etapa 3: Elegir la arista más barata e^t, no perteneciente a E^{t-1}, tal que el grafo $G^t = (N_0, E^{t-1} \cup e^t)$ no contiene ciclos.

Etapa 4: Si $C^t = C_i^{t-1} \cup C_j^{t-1}$ es la componente conexa recién formada por la conexión de las componentes conexas C_i^{t-1} y C_j^{t-1} del grafo G^{t-1} al añadir la arista $e^t = \{(i,j)\}$, definir el vector $f^t = \left(f_k^t\right)_{k \in N}$ dado por

$$f_k^t = \begin{cases} \dfrac{1}{|C_i^{t-1}|} - \dfrac{1}{|C^t|} & \text{si } k \in C_i^{t-1} \text{ y } 0 \notin C^t, \\[2ex] \dfrac{1}{|C_j^{t-1}|} - \dfrac{1}{|C^t|} & \text{si } k \in C_j^{t-1} \text{ y } 0 \notin C^t, \\[2ex] \dfrac{1}{|C_i^{t-1}|} & \text{si } k \in C_i^{t-1} \text{ y } 0 \in C_j^{t-1}, \\[2ex] \dfrac{1}{|C_j^{t-1}|} & \text{si } k \in C_j^{t-1} \text{ y } 0 \in C_i^{t-1}, \\ 0 & \text{en otro caso.} \end{cases}$$

Etapa 5: Definir $E^t := E^{t-1} \cup e^t$.

Etapa 6: Si $t < \tau$, ir a la etapa 2.

Etapa 7: $T = (N_0, E^\tau)$ es un árbol generador de coste mínimo asociado al problema. Denotemos por $\Gamma = (e^1, ..., e^\tau)$ al conjunto de aristas añadidas.

Etapa 8: Definir $ERO^\Gamma(N_0, C) := \sum_{t=1}^{\tau} c_{e^t} f^t$, donde c_{e^t} representa el coste de la arista e^t.

Detengámonos a estudiar el significado del vector f^t, conocido como **vector de obligaciones**. Es claro que f^t es un vector de números entre 0 y 1. A cada f_i^t se le denomina **obligación del usuario i** sobre la arista e^t. En una iteración t del algoritmo, se observa que la obligación del usuario k sobre la arista añadida e^t es 0 si k no pertenece a ninguna de las componentes conexas que son conectadas por la adhesión de dicha arista y por tanto, k no tendrá que aportar nada por su establecimiento. Si k pertenece a una de las componentes conexas, digamos C_i^{t-1}, y la otra componente conexa, C_j^{t-1}, ya estaba conectada al servicio, entonces la obligación de k sobre la arista añadida será inversamente proporcional al número de usuarios que forman parte de la componente C_i^{t-1}. Por último, si la arista no sirve para conectar a ninguna de las componentes al servicio, la obligación de un usuario $k \in C_i^{t-1}$ será igual a $\frac{1}{|C_i^{t-1}|} - \frac{1}{|C^t|}$. Por tanto, su obligación será mayor cuanto menor sea el número de integrantes de la componente C_i^{t-1} y cuanto mayor sea el número de integrantes de la componente conexa formada.

En general, la secuencia de aristas Γ no es única, pues puede haber aristas del mismo coste y, salvo que que se formen ciclos, podría añadirse cualquiera de ellas a E^t. Sin embargo, Feltkamp, Tijs y Muto demuestran que el ERO valor no depende de la secuencia tomada ([13], proposición 4.2, página 16). Así, podemos dar la siguiente definición:

Definición 2.5.1. *Se define el **ERO valor**, $ERO(N_0, C)$, asociado a un problema mcst (N_0, C) como*

$$ERO(N_0, C) := ERO^{\Gamma}(N_0, C)$$

siendo Γ una secuencia cualquiera de aristas tomada en el anterior algorimo.

Más tarde, Bergantiños y Vidal-Puga [2], apoyándose en que la regla de Bird y la de Kar coinciden para formas irreducibles, definen una regla de reparto de la siguiente forma:

Definición 2.5.2. *Sea (N_0, C) un problema mcst. Se define la regla de reparto φ como*

$$\varphi(N_0, C) = K(N_0, C^*) = B(N_0, C^*)$$

donde (N_0, C^*) es la forma irreducible irreducible asociada a (N_0, C).

Observación 2.5.3. *Observemos que por definición la definición de φ, las reglas de reparto φ, B y K coinciden sobre formas irreducibles.*

Pese que a primera vista, la regla definida por Feltkamp, Tijs y Muto no guarda relación con la de Bergantiños y Vidal-Puga, estos últimos demostraron en [1] (teorema 1, página 6), que ambas reglas coinciden y son, en consecuencia, la misma. En toda la literatura relacionada relacionada con los problemas de árbol generador de coste mínimo, la regla de Feltkamp-Tijs-Muto ha sido ampliamente estudiada. Además de la realizada por Bergantiños y Vidal-Puga en [2], dicha regla ha sido definida e introducida de diversas formas, todas ellas equivalentes: Brânzei, Moretti, Norde y Tijs [8] introducen la regla con el nombre de *p-valor*. En otro trabajo, Bergantiños y Vidal-Puga [3] introducen la regla como el valor de Shapley de la *visión optimista* del juego, que consiste en considerar el coste asociado a una coalición $S \subseteq N$, $c^+(S)$, de la siguiente forma

$$c^+(S) = c_{mcst}(N_0, C^{+(N \setminus S)})$$

donde $c_{ij}^{+(N\backslash S)} = c_{ij}$ para todo $i, j \in S$ y $c_{i0}^{+(N\backslash S)} = min_{j\in N_0\backslash S} c_{ij}$ para cada $i \in S$. Es decir, se considera que los agentes de $N\backslash S$ están conectados al servicio y cada usuario de S puede conectarse al mismo a través de los individuos de $N\backslash S$. Por otro lado, Bergantiños y Vidal-Puga, demuestran en [5] que φ puede introducirse como el equilibrio de un juego no cooperativo. En un juego no cooperativo, a diferencia de los juego cooperativos considerados en todo el trabajo, cada jugador toma decisiones de forma independiente al resto de jugadores para obtener el mayor beneficio propio posible. Más tarde, de nuevo Bergantiños y Vidal-Puga [6] introducen la regla a partir del algoritmo de Boruvka para el cálculo del árbol generador de coste mínimo. De aquí en adelante, utilizaremos φ para denotar la regla.

Veamos ahora un ejemplo de aplicación de la regla φ a un problema concreto:

Ejemplo 2.5.4. *Consideremos el problema mcst (N_0, C), con $N_0 = \{0, 1, 2, 3\}$ y la matriz de costes dada por*

$$C = \begin{pmatrix} 0 & 1 & 3 & 3 \\ 1 & 0 & 4 & 2 \\ 3 & 4 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

La matriz irreducible asociada vendrá dada por

$$C^* = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 2 & 2 \\ 2 & 2 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix}$$

Aplicando ahora la regla de Kar al problema (N_0, C^*) se obtiene:

π		$m_{C^*}^{\pi}$
$(1,2,3)$	$\longrightarrow$	$(1,2,1)$
$(1,3,2)$	$\longrightarrow$	$(1,1,2)$
$(2,1,3)$	$\longrightarrow$	$(1,2,1)$
$(2,3,1)$	$\longrightarrow$	$(1,2,1)$
$(3,1,2)$	$\longrightarrow$	$(1,1,2)$
$(3,2,1)$	$\longrightarrow$	$(1,1,2)$

Así,

$$\varphi(N_0, C) = K(N_0, C^*) = (1, \frac{3}{2}, \frac{3}{2})$$

Como se vio en la primera sección del capítulo, ninguna regla de reparto satisface la propiedad IOC, sin embargo, la regla φ sí satisface dicha propiedad si nos restringimos a formas irreducibles.

Proposición 2.5.5. *φ satisface IOC para toda forma irreducible (N_0, C^*).*

Demostración. Ver [2], lema 4.1, página 338. □

En siguiente teorema, comprobaremos que φ satisface todas las propiedades introducidas en la primera sección del capítulo. Dicho teorema es obra de Bergantiños y Vidal-Puga [3]. Por su parte, las caracterizaciones dadas posteriormente se deben a los mismos autores y pueden encontrarse en [2].

Teorema 2.5.6. *φ satisface las propiedades SN, MC, SOL, MP, POS, SEP, SIM, IRCE, IAI, IBC e IGC.*

Demostración. **φ satisface *SOL*:** Sea (N_0, C) un problema *mcst*, $S \subseteq N$ e $i \in S$. Sea (N_0, C^*) la forma irreducible del problema. Aplicando la proposición 1.4.15 se tiene que

$$c^*(S) - c^*(S \backslash \{i\}) = min\{c^*_{ij} : j \in S_0 \backslash \{i\}\}.$$

Sean (N_0, C) y (N_0, C') dos problemas *mcst* tales que $C \leq C'$. Aplicando el resultado 1.4.12, se tiene que $C^* \leq C'^*$. Entonces

$$c^*(S) - c^*(S \backslash \{i\}) \leq c'^*(S) - c'^*(S \backslash \{i\}).$$

Ahora bien, Young [26] demostró que en estas condiciones se tiene que

$$Sh(N, c^*) \leq Sh(N, c'^*)$$

propiedad que se conoce como *monotonicidad fuerte* del valor de Shapley. Por tanto,

$$\varphi(N_0, C) = K(N_0, C^*) = Sh(N, c^*) \leq Sh(N, c'^*) = K(N_0, C'^*) = \varphi(N_0, C'),$$

por lo que, en efecto, φ satisface SOL.
φ satisface *MC*: Como φ satisface SOL, aplicando la proposición 2.1.18 se obtiene el resultado.
φ satisface *MP*: Sea (N_0, C) un problema *mcst*, $S \subseteq N$ e $i \in S$. Tenemos que ver que $\varphi_i(N_0, C) \leq \varphi_i(S_0, C)$. Para ello, es suficiente probarlo para $S = N \backslash \{k\}$ con $k \in N$. Podemos suponer, sin pérdida de generalidad, que $k = n$. Para realizar la demostración necesitaremos utilizar un resultado previo:
Lema. *Si C es tal que $c_{0n} = \alpha$, $c_{in} = \beta$ para todo $i \in N \backslash \{n\}$ y además, $\beta > \alpha > max\{c_{ij} : i, j \in N_0 \backslash \{n\}\}$, se tiene que*

$$\varphi_i(N_0, C) = \begin{cases} \alpha & \text{si } i = n, \\ \varphi_i((N \backslash \{n\})_0, C) & \text{en otro caso.} \end{cases}$$

La demostración de este resultado puede verse en [2] (prueba A.7., página 348).
Tomemos ahora $\alpha = 1 + max_{i,j \in N_0} c_{ij}$ y $\beta = \alpha + 1$. Consideremos la matriz C^0 con $c^0_{0n} = \alpha$ y

$c^0_{ij} = c_{ij}$ en otro caso. Definamos ahora para cada $k = 1, ..., n-1$, C^k de forma que $c^k_{kn} = \beta$ y $c^k_{ij} = c^{k-1}_{ij}$. Tomemos $i \in N \backslash \{n\}$. Como φ satisface SOL se tiene que

$$\varphi_i(N_0, C) \leq \varphi_i(N_0, C^0) \leq \varphi_i(N_0, C^1) \leq \cdots \leq \varphi_i(N_0, C^{n-1}).$$

Aplicando el lema,

$$\varphi_i(N_0, C^{n-1}) = \varphi_i((N \backslash \{n\})_0, C^{n-1}) = \varphi_i((N \backslash \{n\})_0, C).$$

Así, $\varphi_i(N_0, C) \leq \varphi_i((N \backslash \{n\})_0, C)$.

φ satisface *SEP*: Como φ satisface MP, aplicando la proposición 2.1.19 se obtiene el resultado.

φ satisface *SN*: Como φ satisface MP, aplicando la proposición 2.1.19 se obtiene el resultado.

φ satisface *SIM*: Consecuencia de que $\varphi(N_0, C) = K(N_0, C^*) = Sh(N, c^*)$ y de que el valor de Shapley satisface la simetría (ver [23]).

φ satisface *IRCE*: Sean (N_0, C) y (N_0, C') dos problemas en las condiciones de la definición de la propiedad (definición 2.1.13). Es directo ver que entonces las formas irreducibles asociadas, (N_0, C^*) y (N_0, C'^*), cumplen también las condiciones de la definición. Además, $c^*_0 = c_0$ y $c'^*_0 = c'_0$. Como (N_0, C^*) es irreducible, aplicando la proposición 1.4.9, existirá un árbol generador de mínimo coste asociado a (N_0, C^*) que satisfaga las condiciones (A1) y (A2) de 1.4.9. El mismo árbol será también un árbol generador de mínimo coste para (N_0, C'^*) y cumplirá las condiciones (A1) y (A2) para dicho problema. Así, para cualquier $\pi \in \Pi_N$, se tiene que $B^\pi_i(N_0, C^*) = B^\pi_i(N_0, C'^*)$ si $i \neq \pi_1$ (donde, recordemos, utilizábamos la notación π_s para referirnos al usuario i tal que $\pi(i) = s$), $B^\pi_{\pi_1}(N_0, C^*) = c_0$ y $B^\pi_{\pi_1}(N_0, C'^*) = c'_0$. Por tanto, dado $i \in N$, se tendrá que

$$\begin{aligned}
\varphi_i(N_0, C') &= \frac{1}{|N|!} \sum_{\pi \in \Pi_N} B^\pi_i(N_0, C'^*) \\
&= \frac{1}{|N|!} \sum_{\pi \in \Pi_N} B^\pi_i(N_0, C^*) + \frac{(|N|-1)!(c'_0 - c_0)}{|N|!} \\
&= \varphi_i(N_0, C) + \frac{c'_0 - c_0}{|N|}.
\end{aligned}$$

Obteniéndose así que φ satisface $IRCE$.

φ satisface *POS*: Consecuencia de que $\varphi(N_0, C) = B(N_0, C^*)$ y que B satisface POS.

φ satisface *IAI*: Como φ depende únicamente de la forma irreducible, el resultado se obtiene aplicando la proposición 2.1.17.

φ satisface *IGC*: Como φ satisface IAI, aplicando la proposición 2.1.21 se obtiene el resultado.

φ satisface *IBC*: Sean (N_0, C), (N_0, C') e $i \in N$ cumpliendo las condiciones de la definición de IBC (definición 2.1.11). Supongamos en primer lugar que existe una única arista (j, k) tal que $c_{jk} \neq c'_{jk}$. Podemos suponer que $c'_{jk} < c_{jk}$. Como $c'_{jk} < c_{jk} \leq c^{min}_i$, se tiene que $i \neq j$ e $i \neq k$. Distingamos los siguientes tres casos:

1. Existe un árbol generador de coste mínimo $T = (N_0, E_T)$ en (N_0, C) tal que $(j, k) \in E_T$. Podemos suponer, sin pérdida de generalidad que j es predecesor de k en T. Supongamos también que en T, los agentes se conectan al servicio en el orden $\pi = (1, ..., n)$ siguiendo el algoritmo de Prim. Como T es un árbol generador de coste mínimo asociado a (N_0, C) y $(j, k) \in E_T$, T es también un árbol generador de coste mínimo del problema (N_0, C'). Sea $\pi' = (\pi'_1, ..., \pi'_n)$ tal que los agentes se conectan al servicio en T, en el orden π', siguiendo el algoritmo de Prim. Podemos encontrar π' tal que para cada $l = 1, ..., j$, se tenga que $l = \pi'_l$ y $c^*_{(l-1)l} = c'^*_{(l-1)l} = c_{l^0 l}$, donde (l^0, l) es la primera arista en el único camino en T de l a 0.
Sea p tal que $k = \pi'_p$. Entonces, $j < p \leq k$. Es más, podemos tomar π' tal que $l = \pi'_l$ para todo $l = j + 1, ..., p - 1$ y $c^*_{(l-1)l} = c'^*_{(l-1)l} = c_{l^0 l} < c_i^{min}$ para todo $l = p, ..., k$.
Supongamos ahora que podemos encontrar $m \in N$ tal que $m > k$, $c^*_{(m-1)m} = c_{m^0 m} \geq c_i^{min}$ y $c^*_{(l-1)l} = c'^*_{(l-1)l} = c_{l^0 l} < c_i^{min}$ para todo $l = k + 1, ..., m - 1$. Entonces, para todo $l = m, ..., n$, se tiene que $l = \pi'_l$ y $c^*_{(l-1)l} = c'^*_{(l-1)l} = c_{l^0 l}$. Como $c_{l^0 l} < c_i^{min}$ para todo $l = p, ..., m - 1$, se tiene que $\pi'_i = i$. Además, $i < j$ o $i \geq m$.
Si no podemos encontrar tal m, entonces $\pi'_i = i < j$. En este caso tomamos $m = n + 1$ y probaremos que $c^*_{il} = c'^*_{il}$ para todo $l \in N_0 \backslash \{i\}$. Supongamos que $i < j$ (el caso $i \geq j$ es análogo). Aplicando la sentencia (A2) de la proposición 1.4.9, se tiene que $c^*_{il} = c'^*_{il}$ para $l \leq j$. Si $j < l \leq m - 1$, entonces $c^*_{il} = c^*_{ij} = c'^*_{ij} = c'^*_{il}$. Por último, si $l \geq m$, entonces $c^*_{il} = max\{c^*_{ij}, c^*_{(m-1)l}\} = max\{c'^*_{ij}, c'^*_{(m-1)l}\} = c'^*_{il}$.
Como φ satisface IOC sobre la clase de matrices de coste irreducibles se tiene que $\varphi_i(N_0, C^*) = \varphi_i(N_0, C'^*)$. Ahora, por la definición, φ depende únicamente de su forma irreducible, por tanto, se tiene que $\varphi(N_0, C) = \varphi(N_0, C^*)$ y $\varphi(N_0, C') = \varphi(N_0, C'^*)$, por lo que $\varphi_i(N_0, C) = \varphi_i(N_0, C')$.

2. (j, k) no pertenece a ningún árbol generador de (N_0, C) ni de (N_0, C').
Sea T un árbol generador de mínimo coste asociado a (N_0, C). Entonces, T es también un árbol generador de coste mínimo asociado a (N_0, C'). Como φ satisface IAI, $\varphi_i(N_0, C) = \varphi_i(N_0, C')$.

3. (j, k) no pertenece a ningún árbol generador de mínimo coste asociado al problema (N_0, C), pero existe algún árbol generador de mínimo coste $T' = (N_0, E_{T'})$ tal que $(j, k) \in E_{T'}$. Claramente, se cumple que $c_{mcst}(N_0, C) \geq c_{mcst}(N_0, C')$. Definamos el problema $mcst$ (N_0, C'') donde $c''_{jk} := c'_{jk} + c_{mcst}(N_0, C) - c_{mcst}(N_0, C')$ y $c''_{ij} := c_{ij}$ en otro caso. Se tiene entonces que $C \geq C'' \geq C'$. Es directo ver que si T es un árbol generador de coste mínimo asociado a (N_0, C), entonces lo es también del problema (N_0, C''). Como φ satisface IAI, se verifica que $\varphi_i(N_0, C) = \varphi_i(N_0, C'')$. Si $c_{mcst}(N_0, C) - c_{mcst}(N_0, C') = 0$, entonces el resultado es claro, pues $(N_0, C') = (N_0, C'')$ y se tiene que $\varphi_i(N_0, C') = \varphi_i(N_0, C'')$. En caso contrario, si vemos que existe un árbol generador de mínimo coste T'' asociado a (N_0, C'') que contenga la arista (j, k), podremos aplicar el caso número 1 con los problemas (N_0, C') y (N_0, C''). Consideremos $T = (N_0, E_T)$ un árbol generador de mínimo coste asociado a (N_0, C) y sea $G = (N_0, E_T \cup \{(j, k)\})$. Entonces

G contiene exactamente un ciclo, digamos Λ. Ahora, como $(j,k) \in E_{T'}$, aplicando el resultado 1.1.14, existe alguna arista $(l,m) \in \Lambda$ con $(l,m) \neq (j,k)$ tal que $c'_{lm} \geq c'_{pq}$ para toda arista $(p,q) \in \Lambda$ y $c'_{lm} > c'_{jk}$. Por tanto, $c_{mcst}(N_0, C) - c_{mcst}(N_0, C') = c'_{lm} - c'_{jk}$. Pero entonces, $c''_{jk} = c'_{jk} + c'_{lm} - c'_{jk} = c'_{lm}$, por lo que $c''_{jk} = c''_{lm}$ y se tendrá que $T'' = (N_0, (E_{T'} \backslash \{(l,m)\}) \cup \{(j,k)\})$ es un árbol generador de coste mínimo asociado a (N_0, C'').

Se ha probado el resultado para el caso en que existe una única arista (j,k) tal que $c_{jk} \neq c'_{jk}$. Para el caso en el que existe más de una arista de diferente coste basta proceder de la misma forma que en el teorema 2.2.7 para ver que B satisface IBC. □

Los anteriores resultados muestran que φ satisface IOC para formas irreducibles y el resto de propiedades introducidas para problemas *mcst* cualesquiera. Sin embargo, en el siguiente teorema se verá que para caracterizar la regla, basta considerar tres de estas propiedades:

Teorema 2.5.7. *Una regla de reparto ψ satisface IAI, SEP e $IRCE$ si y solo si $\psi = \varphi$.*

Demostración. Por el teorema anterior, sabemos que φ satisface IAI, SEP e $IRCE$, veamos ahora la unicidad. Supongamos que existe una regla de reparto ψ que cumpla estas propiedades. Procederemos por inducción sobre el número de usuarios. Si $|N| = 1$, entonces el resultado es claro. Supongamos que el resultado es cierto para $n-1$ usuarios.
Consideremos ahora el número de usuarios igual a n. Como ψ satisface IAI, en virtud de la proposición 2.1.17 podemos restringirnos a formas irreducibles. Sea (N_0, C^*) un problema *mcst* irreducible y sea $T = (N_0, E_T)$, con $E_T = \{(\pi_{s-1}, \pi_s)\}_{s=1}^n$, un árbol generador de coste mínimo asociado satisfaciendo (A1) y (A2) de la proposición 1.4.9. Sea $\pi_r \in N$ tal que $c^*_{\pi_{r-1}\pi_r} = max\{c^*_{\pi_{s-1}\pi_s} : 0 < s \leq n\}$. Distingamos dos casos:

1. $r > 1$. Tomemos $S = \{\pi_s\}_{s=1}^{r-1}$. Aplicando la proposición 1.4.11(a), se tiene que $T_S = \big(S_0, \{(\pi_{s-1}, \pi_s)\}_{s=1}^{r-1}\big)$ es un árbol generador de coste mínimo del problema (S_0, C^*) y $T_{N\backslash S} = \big((N\backslash S)_0, \{(0, \pi_r)\} \cup \{(\pi_{s-1}, \pi_s)\}_{s=r+1}^n\big)$ es un árbol generador de coste mínimo del problema $((N\backslash S)_0, C^*)$. Además, $c^*_{0\pi_r} = max\{c^*_{\pi_{s-1}\pi_s} : 0 < s \leq r\} = c^*_{\pi_{r-1}\pi_r}$. Por tanto,

$$c_{mcst}(S_0, C^*) + c_{mcst}((N\backslash S)_0, C^*) = \sum_{s=1}^{n} c^*_{\pi_{s-1}\pi_s} = c_{mcst}(N_0, C^*).$$

Ahora bien, como ψ satisface SEP, $\psi_i(N_0, C^*) = \psi_i(S_0, C^*)$ si $i \in S$ y $\psi_i(N_0, C^*) = \psi_i((N\backslash S)_0, C^*)$ si $i \in N\backslash S$. Además, sabemos que $S \neq \emptyset$ y $N\backslash S \neq \emptyset$, pues $r > 1$. Aplicando la hipótesis de inducción, $\psi(S_0, C^*) = \varphi(S_0, C^*)$ y $\psi((N\backslash S)_0, C^*) = \varphi((N\backslash S)_0, C^*)$. Como φ satisface SEP, se tiene que $\psi(N_0, C^*) = \varphi(N_0, C^*)$.

2. $r = 1$. Sea $p > 1$ tal que $c^*_{\pi_{p-1}\pi_p} = max\{c^*_{\pi_{s-1}\pi_s} : 1 < s \leq n\}$. Claramente, $c^*_{\pi_{p-1}\pi_p} \leq c^*_{0\pi_1}$. Definamos la matriz C' de la siguiente forma: $c'_{ij} = c^*_{ij}$ si $i, j \in N$ y $c'_{0i} = c^*_{0i} - \alpha$ para todo

$i \in N$, donde $\alpha := c^*_{0\pi_1} - c^*_{\pi_{p-1}\pi_p}$. Como ψ y φ satisfacen $IRCE$, se tiene para todo $i \in N$,

$$\psi_i(N_0, C^*) = \psi_i(N_0, C') + \frac{\alpha}{n}$$

$$\varphi_i(N_0, C^*) = \varphi_i(N_0, C') + \frac{\alpha}{n}$$

Ahora bien, como C' es una matriz irreducible que cumple que

$$c'_{\pi_{r-1}\pi_r} = max\{c'_{\pi_{s-1}\pi_s} : 0 < s \leq n\},$$

aplicando el caso 1, se obtiene que $\psi(N_0, C') = \varphi(N_0, C')$ y, en consecuencia, que $\psi(N_0, C^*) = \varphi(N_0, C^*)$. $\square$

Consecuencia de esta caracterización, que SOL implica IAI, PM implica SEP y que φ satisface PM y SOL, se obtiene el siguiente resultado:

Corolario 2.5.8. *Una regla de reparto ψ satisface SOL, PM e $IRCE$ si y solo si $\psi = \varphi$.*

Otras caracterizaciones de la regla de Feltkamp-Tijs-Muto pueden encontrarse en [13] (proposición 5.9, página 23), [8] (teorema 1, página 57), [2] (proposición 4.1, página 339) y [4] (teoremas 1 y 2, páginas 41 y 42).

2.6. Consideraciones finales

A lo largo del capítulo se han tratado cuatro de las reglas de reparto en problemas de árbol generador de coste mínimo más estudiadas en la literatura. En el siguiente cuadro realizaremos una síntesis de las propiedades que verifica cada una de ellas.

Propiedades	B	K	DK	φ
SN	Sí	No	Sí	Sí
MC	No	Sí	Sí	Sí
SOL	No	No	No	Sí
MP	No	No	No	Sí
POS	Sí	No	Sí	Sí
SEP	No	No	No	Sí
SIM	Sí	Sí	Sí	Sí
IOC	No	No	No	No
IBC	Sí	Sí	No	Sí
IGC	No	No	No	Sí
$IRCE$	Sí	Sí	No	Sí
IAI	No	No	No	Sí

Como muestra la tabla anterior, φ, a diferencia de las reglas B, K, y DK, satisface todas las propiedades introducidas en la primera sección del capítulo, excepto IOC, que no es satisfecha por ninguna regla de reparto. Observemos que el hecho de que φ satisfaga todas ellas, hace que el resto de reglas no puedan caracterizarse en términos de estas propiedades. Cabe destacar también que las propiedades SOL, MP, SEP, IGC e IAI son únicamente satisfechas por φ. Además de las que aparecen en el cuadro, en el trabajo aparecen definidas otras propiedades de reglas de reparto, como son *monotonicidad de hojas* y *consistencia de fuente* para la caracterización de B, *ausencia de subsidio cruzado, independencia de grupos* y *tratamiento igualitario* para la caracterización de K y *consistencia de árbol* y de nuevo *mononicidad de hojas* para mostrar la caracterización de DK.

Además, en los textos relacionados con los problemas de árbol generador de coste mínimo se han considerado otras propiedades que serían interesantes para una regla de reparto como la *aditividad.* Una regla de reparto ψ satisface esta propiedad si dados dos problemas *mcst* (N_0, C) y (N_0, C') se tiene que

$$\psi(N_0, C + C') = \psi(N_0, C) + \psi(N_0, C').$$

Sin embargo, como se muestra en [4], ninguna regla de reparto satisface esta propiedad. Algunas propiedades menos restrictivas que la aditividad pueden encontrarse también en [4] y [8]. En [4], Branzei, Moretti, Norde y Tijs, definen la propiedad *cone-wise positive linearity.* Una regla de reparto ψ cumple dicha propiedad si verifica que

$$\psi(N_0, C + C') = \psi(N_0, C) + \psi(N_0, C')$$

para cada par de problemas *mcst* (N_0, C) y (N_0, C') para los que exista una aplicación

$$\sigma : \{(i, j)\}_{i,j \in N_0, i<j} \longrightarrow \{1, 2, ..., \frac{|N|(|N| + 1)}{2}\}$$

que verifique que si $i, j, k, l \in N_0$ son tales que $i < j$, $k < l$ y $\sigma(i, j) \leq \sigma(k, l)$ entonces se cumpla que $c_{ij} \leq c_{kl}$ y $c'_{ij} \leq c'_{kl}$. En dicho artículo se demuestra que el p-valor, o equivalentemente, φ satisface dicha propiedad.

Posteriormente, Bergantiños y Vidal-Puga [4] introducen la propiedad *aditividad restringida.* Una regla de reparto ψ satisface esta propiedad si verifica que

$$\psi(N_0, C + C') = \psi(N_0, C) + \psi(N_0, C')$$

para todo par de problemas *mcst* (N_0, C) y (N_0, C') para los que exista $\pi \in \Pi_N$ y un árbol generador de coste mínimo $T = \big(N_0, \{(i^0, i)\}_{i \in N_0}\big)$ asociado a los tres problemas (N_0, C), (N_0, C') y $(N_0, C + C')$ tal que

$$c_{\pi_1^0 \pi_1} \leq c_{\pi_2^0 \pi_2} \cdots \leq c_{\pi_n^0 \pi_n} \text{ y } c'_{\pi_1^0 \pi_1} \leq c'_{\pi_2^0 \pi_2} \cdots \leq c'_{\pi_n^0 \pi_n}$$

donde π_i hace referencia al usuario $j \in N$ tal que $\pi(j) = i$ y (π_i^0, π_i) es la primera arista en el único caminto en T de π_i a 0. Bergantiños y Vidal-Puga demuestran que φ satisface esta propiedad y que toda regla de reparto que satisfaga la aditividad restringida satisface también *cone-wise positive linearity*

Bibliografía

[1] G. Bergantiños, J. Vidal-Puga, *Several approaches to the same rule in minimum cost spanning tree problems*, Mimeo, Universidad de Vigo, 2007.

[2] G. Bergantiños, J. Vidal-Puga, *A fair rule in minimum cost spanning tree problems*, Journal of Economic Theory 137 (2007), 326-352.

[3] G. Bergantiños, J. Vidal-Puga, *On some properties of cost allocation rules in minimum cost spanning tree problems*, AUCO Czech Economic Review 2 (2008), 251–267.

[4] G. Bergantiños, J. Vidal-Puga, *Additivity in minimum cost spanning tree problems*, Journal of Mathematical Economics 45 (2009), 38-42.

[5] G. Bergantiños, J. Vidal-Puga, *Realizing fair outcomes in minimum cost spanning tree problems through non-cooperative mechanisms*, European Journal of Operational Research 201 (2010), 811-820.

[6] G. Bergantiños, J. Vidal-Puga, *The folk solution and Boruvka's algorithm in minimum cost spanning tree problems*, Discrete Applied Mathematics 159 (2011), 1279-1283.

[7] C.G. Bird, *On cost allocation for a spanning tree: a game theoretic approach*, Networks 6 (1976), 335-350.

[8] R. Brânzei, S. Moretti, H. Norde, S. Tijs. *The P-value for cost sharing in minimum cost spanning tree situations*, Theory and Decision Library 56 (2004), 47-61.

[9] L. Cánovas, P. Fernández, B. Pelegrín, *Algoritmos en grafos y redes*, PPU, Barcelona, 1992.

[10] I. Curiel, *Cooperative game theory and applications: cooperative game arising from combinatorial optimization problems*, Springer, 1997.

[11] T.S.H. Driessen, S.H. Tijs, *Game theory and cost allocation problems*, Management Science 32 (1986), 1015-1028.

[12] B. Dutta, A. Kar, *Cost monotonicity, consistency and minimum cost spanning tree games*, Games and Economic Behavior 48 (2004), 223-248.

[13] V. Feltkamp, S. Tijs, S. Muto, *On the irreducible core and the equal remaining obligation rule of minimum cost extensions problems*, Mimeo, Tilburg University, 1994.

[14] V. Feltkamp, *Cooperation in controlled network structures*, PhD thesis. NWO, Netherlands, 1995.

[15] D. Granot, G.Huberman, *Minimum cost spanning tree Games*, Mathematical Programming 21 (1981), 1-18.

[16] D. Granot, G.Huberman, *On the core and nucleolus of minimum cost spanning tree games*, Mathematical Programming 29 (1984), 323-347.

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

[18] A. Kar, *Axiomatization of the Shapley value on minimum cost spanning tree games*, Games and Economic Behavior 38 (2002), 265-277.

[19] A. Marín, *Apuntes asignatura: Grafos y Optimización Discreta*, Universidad de Murcia, curso 2014/2015.

[20] H. Özsoy, *A characterizacion of Bird's rule*, Working paper, Rice University, 2006.

[21] M.A. Pulido, Tesis doctoral: *Estructuras de coaliciones y aplicaciones a la teoría de juegos*, Universidad Miguel Hernández, Elche, 2001.

[22] A. E. Roth, L. S Shapley, *The Shapley value : essays in honor of Lloyd S. Shapley / edited by Alvin E. Roth*, Cambridge University Press, Cambridge, 1988.

[23] L. S. Shapley, *A value for n-person games*, Ann. of Math. Studies 28 (1953), 307-317.

[24] L.S. Shapley, *Cores of convex games*, International Journal of Game Theory 1 (1971), 11-26.

[25] R.J. Weber, *Probabilistic values for games*, Cowles Foundation Discussion Paper 417R (1978), Yale University, New Haven.

[26] H.P. Young, *Monotonic solutions of cooperative games*, International Journal of Game Theory 14 (1985), 65-72.

Índice terminológico