UNIVERSITAT DE BARCELONA

GRAU en MATEMÀTIQUES
GRAU en ADMINISTRACIÓ I DIRECCIÓ D'EMPRESES

Facultat de Matemàtiques i Informàtica
Facultat d'Economia i Empresa

Treball Final de Grau

Arbres de cost mínim i jocs cooperatius

Laura Camacho Martín

Directors: **Dr. Xavier Jarque**
Dept. de Matemàtiques i Informàtica
Dr. Javier Martínez de Albéniz
Dept. de Matemàtica Econòmica, Financera i Actuarial

Barcelona, Juny 2020

Resum

En aquest projecte estudiem els problemes d'arbres generadors de cost mínim i com associar-los a un joc cooperatiu d'utilitat transferible. Algunes situacions econòmiques reals com la construcció d'una xarxa elèctrica per l'abastiment d'energia a tot un poble des de una central en comú poden ser modelades mitjançant aquests problemes. Aquest cas va ser estudiat per Dutta i Kar (2004).

Per una banda, definim dos algoritmes capaços d'obtenir un arbre generador de cost mínim a partir d'un graf connex, concretament el de Prim (1957) i el de Kruskal (1956).

D'altra banda, definim una sèrie de regles d'assignació de costos, les quals s'interpreten sota el prisma dels jocs cooperatius. Es detalla una sèrie de propietats per veure quines d'aquestes propietats les determinen.

També estudiem la forma irreductible introduïda per Bird (1976) associada a un problema d'arbre generador de cost mínim. Es veu la forma s'associar-ne una a cada problema i s'analitza l'equivalència entre regles quan s'està en aquesta forma irreductible.

Abstract

In this project we study minimun cost spanning tree problems and how to associate them with a cooperative game of transferable utility. Some real economic situations such as the construction of an electricity network for the supply of energy to an entire village from a common power station can be modelled by these problems. This case was studied by Dutta and Kar (2004).

On the one hand, we define two algorithms capable of obtaining a minimum cost spanning tree of a connected graph, namely Prim's (1957) and Kruskal's (1956).

On the other hand, we explain a series of cost allocation rules, which are interpreted under the prism of cooperative games. A series of properties are detailed to see which of these properties determine them.

We also study the irreducible form introduced by Bird (1976) associated with a minimum cost spanning tree problem. We see how to associate one to each problem and analyze the equivalence between rules in this irreducible form.

2010 Mathematics Subject Classification. 05C05, 05C57, 91A12, 91A43, 91A80

Agraïments

Aquest treball no hauria estat possible sense l'ajuda i suport dels meus tutors, al Dr. Javier Martínez de Albéniz i al Dr. Xavier Jarque. És per això que els vull agrair la seva dedicació i el seu temps durant tots aquests mesos.

També vull agrair a la meva família per estar sempre que els necessito i donar-me suport durant tota la meva etapa acadèmica que està a punt d'arribar a la seva fi.

Per últim, i no menys important, als meus amics per la seva companyia i per estar sempre en els moments en què els ànims dequeien a causa d'aquestes circumstàncies no pas res fàcils que estem passant. Gràcies!

"The essence of math is not to make simple things complicated, but to make complicated things simple."

S. Gudder

Índex

Capítol 1

Introducció

Molts problemes que involucren la creació d'una xarxa han sigut estudiats en la literatura econòmica i en la d'investigació operativa per la seva gran aplicabilitat en diverses situacions quotidianes. La primera es centra més a estudiar la manera de distribuir els costos dins les xarxes i en el disseny de mecanismes que intenten explicar com formar-les, mentre que la investigació operativa s'enfoca en el disseny d'algoritmes eficients i la complexitat computacional.

El problema que es treballarà al llarg del treball sorgeix quan un grup de clients, situats en diferents punts geogràfics volen connectar-se a un determinat servei el qual únicament pot ser proveït per una font en comú per tots ells i aquests estan disposats a cooperar. Per aconseguir-ho requeriran connexions entre ells que comportaran una sèrie de costos. Els clients no han de connectar-se necessàriament a la font de manera directa sinó que també ho poden fer a través d'altres clients que ja es trobin connectats. L'objectiu principal és subminitrar el servei a tots els clients minimitzant el cost de connectar-los tots a la font. Això s'aconsegueix mitjançant una xarxa que connecti tots els clients i que no contingui cicles. Aquesta xarxa rep el nom d'*arbre generador de cost mínim.*

Hi ha moltes situacions econòmiques reals que poden ser modelades d'aquesta manera. Alguns autors han estudiat situacions concretes. Bergantiños i Lorenzo (2004) van estudiar el cas real d'un poble on els ciutadans havien de pagar el cost de la construcció de canonades des de les seves cases fins a l'estació proveïdora de l'aigua. Un altre va ser l'obtenció d'energia d'una central elèctrica comú, fet que comportava compartir el cost d'aquesta xarxa de distribució entre tots els beneficiaris. Aquest cas va ser estudiat per Dutta i Kar (2004) [10]. Altres exemples freqüents es troben en les xarxes de comunicació com internet, telefonia, etc.

D'aquesta manera es pot considerar un graf complet on un dels nodes representa la font i la resta són els usuaris interessats a connectar-se a ella. Cada aresta té un cost associat determinat. El problema consta de dues fases. La primera és com determinar aquest arbre capaç de connectar a tots els clients a la font amb el menor cost. Per això la literatura de problemes d'arbres generadors de cost mínim comença definint alguns algoritmes capaços de construir aquest arbre. Prim (1957) [16] i Kruskal (1956) [14] van dissenyar dos algoritmes eficients capaços de construir aquests arbres a partir d'un graf arbitrari connex. Malgrat que en els models els algorismes poden donar lloc a diferents arbres (de cost mínim) degut a la, teòricament possible, igualtat entre els costos de diferents arestes, en els casos reals aquest problema és irrellevant.

Una altra qüestió important, i la segona fase d'aquest problema, és la manera d'assignar entre tots els clients el cost associat a l'arbre generador de cost mínim. En aquest aspecte, es parla de quatre regles de repartició. Bird (1976) i Dutta i Kar (2004) van introduir dues regles basades en l'algoritme de Prim, d'altra banda Feltkamp et al. (1994) introdueixen la regla ERO (*Equal Remaining Obligation*) basada en l'algoritme de Kruskal i per últim Kar (2002) va estudiar el valor de Shapley del joc de costos associat a l'arbre generador de cost mínim.

L'evident analogia entre la divisió de costos i la divisió de beneficis que s'obtenen de la cooperació entre diferents jugadors, van convertir la teoria de jocs en una eina molt útil a l'hora de tractar problemes d'arbres generadors de cost mínim. Bird (1976) associa a cada problema d'arbre generador de cost mínim un joc cooperatiu d'utilitat transferible (TU). Segons Bird, el valor d'una coalició és el cost de connectar tots els agents de la coalició a la font, assumint que la resta de clients no hi són presents. Bergantiños i Vidal-Puga (2007) [4], en canvi, associen un altre joc cooperatiu TU a cada problema d'arbre generador de cost mínim on el cost d'una coalició s'obté assumint que els agents que no participen en la coalició ja estan connectats i els agents de la coalició poden connectar-se a la font a través d'ells. El primer joc, de Bird, rep el nom de joc pessimista, mentre que el segon joc rep el nom de joc optimista.

Un problema d'arbre generador de cost mínim és irreductible si reduint el cost de qualsevol arc, el cost de connectar els clients a la font també es redueix. Bird (1979) va definir la forma irreductible associada a cada problema. També va presentar un procediment per associar-ne una a cada problema generador de cost mínim. Més tard, Bergantiños i Vidal-Puga (2007) [4] van demostrar que les regles de Bird i Kar coincideixen en la forma irreductible, definint així una nova regla φ.

Bergantiños i Vidal-Puga (2005) [3] demostren que aquesta regla φ coincideix amb la regla ERO, a més, presenten algunes definicions alternatives per φ. Aquesta regla també ha estat estudiada en altres textos i amb altres noms. Branzei et al. (2004), per exemple, l'anomenen P-value.

Bergantiños i Vidal-Puga (2007) [4], enumeren una sèrie de propietats desitjables què una regla "justa"hauria de satisfer. La llista de propietats bàsiques inclou, des del nostre punt de vist, aquelles propietats que proporcionen la millor manera de procedir en aquest tipus de problemes. Es prova quines d'aquestes propietats són satisfetes per cadascuna de les regles estudiades, arribant a veure que la regla ERO satisfà moltes més propietats que la resta de regles.

Organització del treball

El treball està organitzat de la manera següent. El capítol 2 de Preliminars s'introdueixen els conceptes necessaris tant de teoria de jocs com de teoria de grafs.

El capítol 3 es divideix en tres seccions. A la primera s'introdueix el problema generador de cost mínim, a la segona els algoritmes de Prim i Kruskal i s'acaba amb l'explicació i l'aplicació de cadascuna de les regles d'assignació de costos.

En el quart i últim capítol s'associa un arbre generador de cost mínim a un joc cooperatiu i s'estudia tant el nucli d'aquest joc com l'aplicació de les regles de repartició a ell. Per acabar el capítol i el treball es parla de la forma irreductible d'aquests problemes, arribant a equivalències entre les regles i propietats d'aquestes.

Capítol 2

Preliminars

En aquest capítol fem una breu introducció tant a la teoria de grafs com a la teoria de jocs. De la primera definim el concepte de graf així com diverses característiques que es poden trobar en ells, per acabar definint el concepte d'arbre generador. D'altra banda de la part de teoria de jocs ens centrem a parlar dels jocs cooperatius d'utilitat transferible (Jocs cooperatius TU) i les seves peculiaritats, per acabar introduint alguns conceptes de solució. Els conceptes que s'introdueixen són solucions puntuals com el valor de Shapley i solucions conjuntistes com el *core* (nucli).

2.1 Introducció a la teoria de grafs

Qualsevol objecte matemàtic que involucra punts i connexions entre ells es pot considerar un graf, i depenent de si aquestes connexions són dirigides o no, tenim grafs orientats o no orientats.

La teoria de grafs és la branca de la matemàtica discreta què s'encarrega d'estudiar aquest tipus de problemes. Les nocions bàsiques de teoria de grafs es poden trobar a Gross et al.(2013)[12], Wilson (1996) [22] o a d'altres com Chartrand et al.(2012)[9].

Definició 2.1. Un *graf* (no orientat) és un parell (V, A) on V és un conjunt finit i no buit, als elements del qual anomenarem *vèrtexs* o *nodes*, i A és un conjunt de parelles no ordenades de vèrtexs diferents, anomenades *arestes* o *arcs*.

Aquesta definició de graf correspon amb el que alguns autors anomenen *graf simple,* ja que en ella hi ha algunes restriccions. En considerar que un element de V no pot aparèixer repetit en un mateix parell, estem eliminant l'existència de llaços o *loops*, i.e, arestes que comencen i acaben al mateix vèrtex. I en considerar A com un conjunt d'elements en lloc d'una família d'elements, entenem que tampoc existeixen diverses arestes entre un mateix parell de vèrtexs. A partir d'ara quan parlem de graf, ens estarem referint a un graf simple.

Donat un graf $G = (V, A)$ es diu que una aresta $a = (v_1, v_2) \in A$ és *incident* als vèrtexs v_1 i v_2 que la determinen i també que aquests dos vèrtexs que pertanyen a la mateixa aresta són *adjacents.* Una manera de representar els grafs és a través de les matrius d'*adjacència* i *incidència*, les quals es poden definir de la següent forma.

Definició 2.2. Sigui $G = (V, A)$ un graf, amb $V = (v_1, \ldots, v_n)$ i $A = (a_1, \ldots, a_m)$. Es defineix la matriu d'*adjacència* de G com una matriu $n \times n$

$$H(G) = (h_{ij})_{i,j=1,\ldots,n} \ ,$$

on $h_{ij} = 1$ si $v_i, v_j \in V$ estan connectats a través d'una aresta de A, és a dir, són adjacents i zero en cas contrari. Aquesta serà una matriu simètrica donat que es tracta d'un graf no orientat i amb zeros a la diagonal, ja que no existeixen llaços.

D'altra banda, la matriu d'*incidència* de G és una matriu $n \times m$ tal que

$$I(G) = (b_{ij})_{i=1,\ldots,n,\ j=1,\ldots,m} \ ,$$

on $b_{ij} = 1$, si el vèrtex i és incident a j, i zero en cas contrari.

Per tant, ambdues matrius d'un graf G dependran de l'ordre de numeració dels vèrtexs i arestes.

Exemple 2.3. Representació gràfica de grafs.

Considerem el graf $G = (V, A)$, on $V = \{1, 2, 3, 4\}$ és el conjunt de vèrtex i $A = \big\{(1,2), (1,3), (1,4), (2,3), (3,4)\big\}$ el conjunt d'arestes. Podem representar-los de la següent forma:

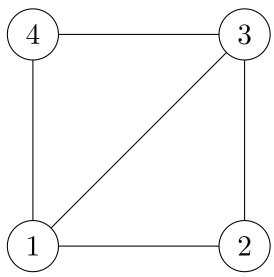


Figura 2.1: Representació gràfica del graf corrresponent a l'Exemple 2.3.

En aquest exemple les arestes (1,2), (1,3) i (1,4) són incidents al node 1, així com aquest node és adjacent als altres tres vèrtexs. En canvi, el node 4 únicament és adjacent als vèrtexs 1 i 3. I les matrius d'adjacència i incidència associades són:

$$H(G) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad I(G) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Definició 2.4. Donat un graf $G = (V, A)$ i un graf $G' = (V', A')$ es diu que G' és un *subgraf* de G si $V' \subseteq V$ i $A' \subseteq A$.

Per tant, en un subgraf el conjunt de vèrtexs i arestes de G' pertanyen a G.

Definició 2.5. Es diu que el graf $G = (V, A)$ és *complet* si per cada $v_i, v_j \in V$ amb $v_i \neq v_j$ es té que $(v_i, v_j) \in A$, és a dir, cada vèrtex de G està connectat a qualsevol altre per una aresta de G.

S'anomena *camí* a un graf $P = (V, A)$ on $V = \{v_1, ..., v_n\}$ i $A = \{(v_1, v_2), (v_2, v_3), (v_3, v_4) \ldots, (v_{n-1}, v_n)\}$, i *cicle* a un graf $C = (V, A)$ on el conjunt de nodes és $V = \{v_1, ..., v_n\}$ amb $n \geq 3$, i el conjunt d'arestes $A = \big\{(v_1, v_2), ..., (v_{n-1}, v_n), (v_n, v_1)\big\}$.

Definició 2.6. Donat $G = (V, A)$ un graf, aquest és *connex* si per tot $v_1, v_2 \in V$ existeix com a mínim un camí de v_1 a v_2 en G.

A la Figura 2.2 es pot veure un exemple bastant clar i visual de connexió. El graf de l'esquerra és connex però en canvi, el de la dreta no ho és.

Figura 2.2: Graf connex (esquerra) i graf no connex (dreta).

Ara introduïm la definició d'arbre, que és un tipus específic de graf, i en particular el concepte d'arbre generador, atès que aquests conceptes seran crucials dins de la nostra exposició.

Definició 2.7. Un graf $G = (V, A)$ es diu que és un *arbre* si és connex i no conté cicles. Un *arbre generador* d'un graf $G = (V, A)$ és un subgraf de G connex i sense cicles.

Podem trobar un arbre generador d'un graf connex de la següent manera. Donat qualsevol graf G connex, s'escull un cicle i s'elimina una de les seves arestes. Observem que el graf resultant segueix sent connex. Es repeteix aquest procés fins que ja no quedin més cicles, aleshores el graf obtingut és un arbre generador de G. A la Figura 2.3 apareix un graf, i un dels seus possibles arbres generadors.

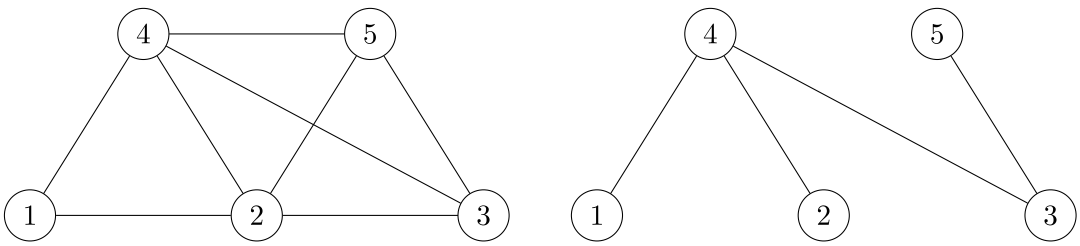


Figura 2.3: Graf connex i un dels seus arbres generadors.

2.2 Introducció als jocs cooperatius TU

La Teoria de Jocs és una branca de les matemàtiques que es dedica a l'estudi de situacions conflictives que sorgeixen quan un conjunt d'agents o jugadors, amb interessos generalment contraposats, han de prendre decisions què els hi afecten mútuament. Cadascuna d'aquestes situacions és el que es coneix com a joc.

Els inicis de la Teoria de Jocs daten de la primera meitat del segle passat, però la idea general dels jocs, amb la qual formular matemàticament processos de presa de decisions que es produeixen en activitats conjuntes de tipus competitiu, va ser introduïda per John von Neumann i Oskar Morgenstern el 1944 [21].

Es poden trobar diverses situacions conflictives en les quals participen jugadors, i per tant, hi ha molts tipus de jocs. Una primera distinció es fa entre els jocs cooperatius i els no cooperatius. Els darrers estudien el comportament dels agents en qualsevol situació on cada jugador tria la seva estratègia i el resultat depèn del que hagin fet tots els jugadors. Per tant el concepte que importa és el d'equilibri (de Nash), aquella combinació d'estratègies on ningú té incentius per moure's de l'estratègia triada. Cada jugador es preocupa només de la seva utilitat a cada resultat possible. Hi ha molts textos que estudien els jocs no cooperatius, i aquests són el fonament de la microeconomia. Vegeu Gibbons (1999) [11] o Binmore (2001) [6].

Els jocs cooperatius, a diferència dels no cooperatius, es permet comunicació entre els jugadors amb la finalitat de negociar o establir acords que els permetessin formar coalicions. En aquesta secció ens centrarem en una classe de jocs cooperatius que són útils per la seva aplicabilitat a diversos contextos que es treballaran més endavant. Parlem dels *jocs cooperatius d'utilitat transferible (TU):* vegeu Magaña (1996) [15]. Aquests jocs tracten aquelles situacions en les quals un grup d'agents pot negociar i coordinar les seves estratègies mitjançant la formació de coalicions i en el qual existeix un bé, totalment divisible, que els jugadors poden transferir lliurement els uns als altres. Una bona introducció es pot trobar a Rafels et al. (1999)[17].

Definició 2.8. Un *joc cooperatiu TU* és un parell (N, v) format per un conjunt finit $N = \{1, 2, \ldots, n\}$ i una funció $v : 2^N \longrightarrow \mathbb{R}$ la qual assigna a cada subconjunt $S \subseteq N$ un nombre real $v(S)$ amb la condició que $v(\emptyset) = 0$.[1] Cada element de N és un *jugador* i cada subconjunt de N és una *coalició.* La funció v s'anomena *funció caracteristica* del joc i $v(S) \in \mathbb{R}$ representa la *utilitat* (el guany) que poden obtenir els membres de S si cooperen.

A partir d'ara per abreujar, ens referirem als jocs cooperatius d'utilitat transferible com únicament jocs cooperatius. Vegem un exemple d'aplicació d'aquests tipus de jocs.

Exemple 2.9. *Repartició d'una herència.* Una persona deixa una herència d'un milió d'euros a repartir entre els seus tres fills amb la condició que almenys dos d'ells arribin a un acord sobre com efectuar el repartiment, ja que en cas contrari els diners anirien íntegrament destinats a una institució sense ànim de lucre.

Aquesta situació es pot tractar com un joc cooperatiu (N, v) on $N = \{1, 2, 3\}$, i v és la funció característica associada que descriurem tot seguit. Quan es forma una coalició individual cap d'ells surt guanyant, en canvi si decideixen cooperar i formar coalicions de dos jugadors o de tres, els que hi participin en elles guanyaran. Però aquell que es quedi

[1] On 2^N és el conjunt dels subconjunts de N, el que també s'anomena les parts de N: $\mathcal{P}(N)$.

fora no obtindrà cap guany. Per tant, la funció característica d'aquest joc és

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1,2\}) = v(\{1,3\}) = v(\{2,3\}) = v(N) = 1.$$

Algunes propietats semblen molt rellevants per als jocs cooperatius. D'aquestes, la primera és la monotonia:

Definició 2.10. Diem que un joc cooperatiu (N, v) és *monòton* si $v(S) \leq v(T)$ quan $S \subseteq T \subseteq N$.

D'això es desprèn que en aquests jocs, a mesura que augmenta el nombre de jugadors en una coalició, augmenta també el valor de la coalició. Com a conseqüència evident, un joc monòton pren valors positius, ja que $v(\emptyset) = 0$.

Una altra propietat és la superadditivitat, la qual implica que la utilitat obtinguda per la unió de dues coalicions disjuntes és major o igual que la de cadascuna per separat.

Definició 2.11. Diem que un joc cooperatiu (N, v) és *superadditiu* si per tot $S, T \subset N$ amb $S \cap T = \emptyset$,

$$v(S \cup T) \geq v(S) + v(T).$$

De la definició anterior podem extreure que en un joc superadditiu els jugadors tenen incentius per la cooperació en el sentit que la unió de dos grups qualssevol disjunts, mai disminueix els beneficis. En cas contrari, si la desigualtat es dóna en sentit oposat es diu que el joc és *subadditiu*.

Per últim, definirem la propietat de convexitat, una propietat més forta que l'anterior. En un joc convex la contribució marginal d'un jugador és més gran com més membres té la coalició a la qual s'incorpora. Dit d'una altra manera, la contribució a una coalició és creixent si a aquesta s'incorporen nous membres. Cal destacar que la superadditivitat és una condició necessària per a la convexitat.

Definició 2.12. Diem que un joc cooperatiu (N, v) és *convex* si per tot $i \in N$ i tot $S, T \subseteq N$ tal que $S \subseteq T \subset N \setminus \{i\}$ es compleix que

$$v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T).$$

Equivalentment es pot demostrar que (N, v) és *convex* si i només si per tot $S, T \subseteq N$ es té

$$v(S \cup T) \geq v(S) + v(T) - v(S \cap T).$$

Si i és un jugador que no pertany a la coalició S, l'increment $v(S \cup \{i\}) - v(S)$ mesura la contribució de i a la coalició S. La convexitat doncs com s'ha dit abans, implica que la contribució d'un jugador a una coalició no decreix contra més jugadors hi hagi a la coalició, això és el que es coneix com a bola de neu o *snowball effect.* Quan les desigualtats es donen en sentit oposat es diu que el joc és *còncau.*

Donat un joc cooperatiu es vol arribar a una assignació o un conjunt d'assignacions de $v(N)$ que puguin ser acceptades per tots els jugadors. Si bé hi ha un gran nombre de conceptes de solució, dels que més es parla són quatre, el *nucli*, el *kernel*, que són conjuntistes, i el *valor de Shapley* i el *nucleolus*, que són solucions puntuals. A continuació en farem cinc cèntims del nucli i del valor de Shapley que seran necessaris per més endavant, però primerament caldrà introduir algunes definicions i conceptes per finalment arribar a les definicions.

Definició 2.13. Un *vector de pagaments* x, és un element de $\mathbb{R}^n$ on la *i-éssima* coordenada representa el pagament corresponent en el joc cooperatiu al *i-éssim* jugador.

Donat $x \in \mathbb{R}^N$, per a qualsevol coalició $S \subseteq N$, denotem $x(S)$ com

$$x(S) = \sum_{i \in S} x_i.$$

I per definició $x(\emptyset) = 0$.

Un vector de pagaments $x \in \mathbb{R}^n$ es diu que compleix el *principi d'eficiència* si verifica la condició que $x(N) = v(N)$ i que satisfà el *principi d'individualitat racional* si $x_i \geq v(\{i\})$ per cada $i \in N$. Aquestes propietats semblen bastant lògiques que hi siguin presents en un vector de pagaments, ja que en ser eficients s'espera que el cost total no superi la suma de les quantitats assignades a cada jugador així com també d'altra banda, la individualitat racional assigna a cada jugador una quantitat major o igual a la que aconseguiria si ho fes per ell sol, ja que si fos al contrari, no ho faria.

Definició 2.14. S'anomena *imputació* a un vector de pagaments eficient i que, a més a més, satisfà el principi d'individualitat racional. El conjunt d'imputacions el denotarem com

$$I(v) = \{x \in \mathbb{R}^n \mid x(N) = v(N) \quad \text{i} \quad x_i \geq v(\{i\}), \quad \text{per tot } i \in N\}.$$

El *Core* o nucli d'un joc cooperatiu TU

Si estenem el principi de racionalitat individual mitjançant el principi de racionalitat coalicional, arribem llavors al concepte de *core* o nucli d'un joc cooperatiu. Formalment,

Definició 2.15. S'anomena *core* o *nucli* d'un joc cooperatiu TU (N, v) al conjunt

$$Core(N, v) = \{x \in \mathbb{R}^N \mid x(N) = v(N) \quad \text{i} \quad x(S) \geq v(S) \quad \text{per tot } S \subseteq N\}.$$

El nucli és el concepte de solució més elemental i fonamental de la majoria de problemes econòmics, i està format per vectors què cap coalició possible de jugadors tindria motius per rebutjar.

El proper exemple ens permet representar el nucli d'un joc cooperatiu.

Exemple 2.16. Tres jugadors poden cooperar amb l'objectiu d'aconseguir un benefici conjunt de 600 €. Se sap que si el jugador 1 coopera amb el 2, guanyen 300 €, i si ho fa amb el jugador 3, 400 €. Si decideixen cooperar els jugadors 2 i 3 el seu benefici és de 500 €. Finalment quan dos jugadors cooperen el que es queda fora no obté cap guany. Aquesta situació es pot modelar com un joc cooperatiu, amb la següent funció característica:

$$\begin{aligned}
&v(\{1\}) = v(\{2\}) = v(\{3\}) = 0, \\
&v(\{1,2\}) = 300, \ v(\{1,3\}) = 400, \ v(\{2,3\}) = 500, \\
&v(\{1,2,3\}) = 600.
\end{aligned}$$

Anem ara a buscar el nucli d'aquest joc, és a dir, busquem un vector $x = (x_1, x_2, x_3)$ de $\mathbb{R}^3$ tal que compleixi les condicions de nucli definides anteriorment. S'ha de complir doncs,

$$\begin{aligned}
&x_1 + x_2 + x_3 = 600,\\
&x_1 + x_2 \geq 300,\ x_1 + x_3 \geq 400,\ x_2 + x_3 \geq 500,\\
&x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0.
\end{aligned}$$

Fàcilment es veu que aquest sistema d'inequacions té com a única solució el vector de pagaments $x = (100, 200, 300)$. En aquest cas, el nucli del joc té un element. Si ho representem gràficament en el pla $x_1 + x_2 + x_3 = 600$,

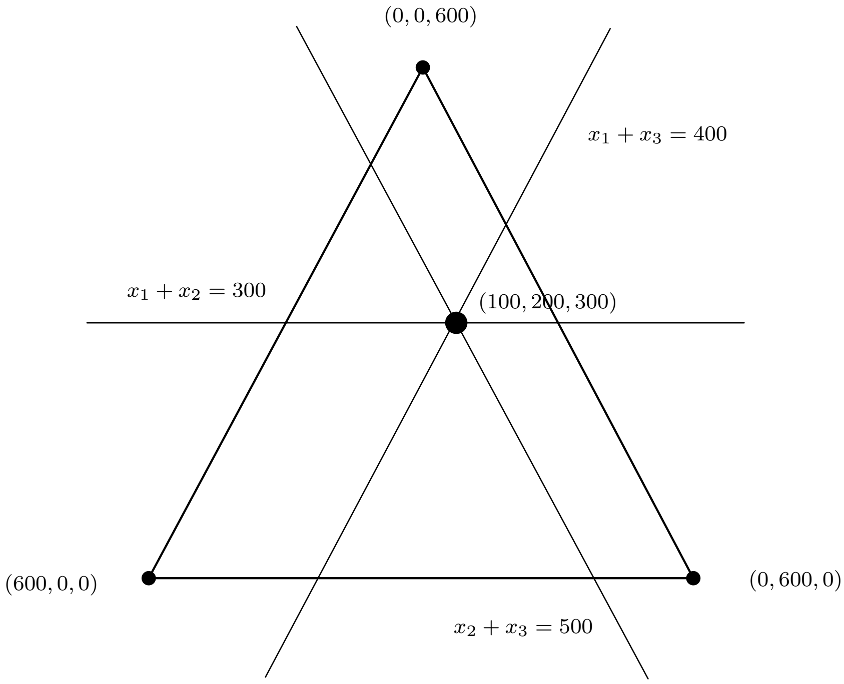


Figura 2.4: Representació del core del joc cooperatiu de l'Exemple 2.16 dins del pla del conjunt d'imputacions.

Si la situació canviés i el guany que obtindrien totes les coalicions de dos jugadors passés a ser igual en tots els casos, concretament de 360 €, aleshores la funció característica del nou joc serà:

$$\begin{aligned}
&v'(\{1\}) = v'(\{2\}) = v'(\{3\}) = 0,\\
&v'(\{1, 2\}) = v'(\{1, 3\}) = v'(\{2, 3\}) = 360,\\
&v'(\{1, 2, 3\}) = 600.
\end{aligned}$$

El *core* del nou joc seria diferent, mentre que el conjunt d'imputacions segueix sent el mateix. Tenim:

$$Core(N, v') = \Big\{x \in \mathbb{R}^N \mid x_1 + x_2 + x_3 = 600, 0 \leq x_1 \leq 240, 0 \leq x_2 \leq 240, 0 \leq x_3 \leq 240\Big\}.$$

En aquest cas, la representació del nou joc cooperatiu (N, v') en el pla d'imputacions és:

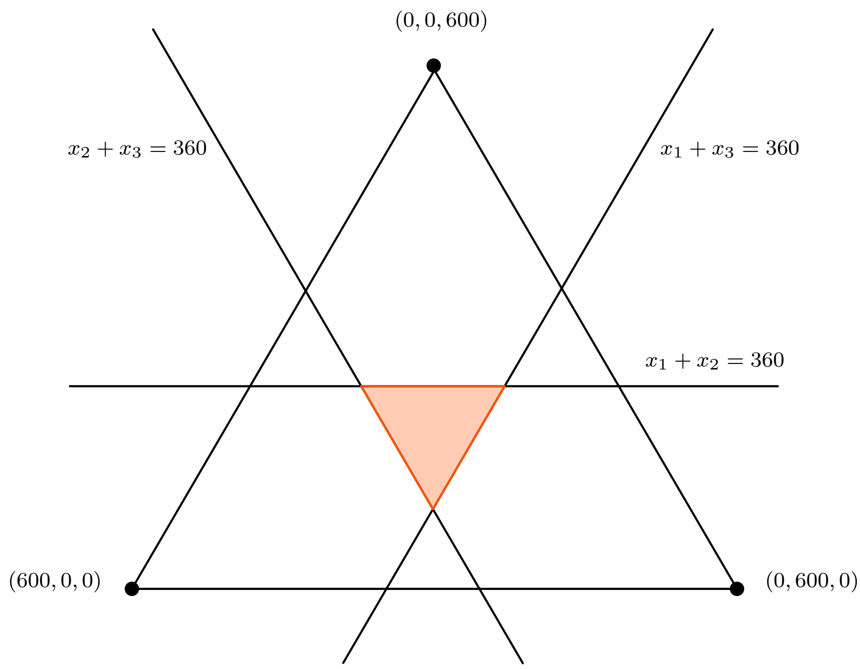


Figura 2.5: Representació del core del nou joc cooperatiu.

Per tant, s'ha vist que el nucli pot ser un únic punt, una regió del conjunt d'imputacions com s'ha donat en la modificació de l'exemple o inclús pot arribar a ser buit. És el cas de l'Exemple 2.9 anterior el qual té *core* buit.

Els jocs amb core no buit reben el nom de *jocs equilibrats.* El concepte de joc equilibrat pot fer-se més estricte. Així, es diu que un joc (N, v) és *totalment equilibrat* si el subjoc induït (S, v_S) és equilibrat per a tot $S \subseteq N, S \neq \emptyset$, on (S, v_S) es defineix com $v_S(T) = v(T)$ per a tot $T \subseteq S$. D'aquesta forma, direm que un joc és totalment equilibrat si i només si qualsevol subjoc induït té el core no buit.

Formalment tant els jocs de guanys, que són els que fins ara s'han considerat, on els vectors de pagaments són ingressos pels agents, com els jocs de costos, on els agents han de pagar per aconseguir alguna cosa, són tots dos jocs cooperatius. En el cas de jocs de costos, aquests es denoten com (N, c). Es pot passar dels jocs de costos als de guanys (i viceversa), i l'única diferència estriba en el fet que les desigualtats de la racionalitat individual i la coalicional estan canviades. Així doncs, com els jocs d'arbre generadors de cost mínim són considerats com jocs de costos, durant tota la memòria els jocs cooperatius es tractaran com jocs de costos on el valor de la coalició total es considerarà $c(N)$. I el nucli llavors passarà a ser considerat com el conjunt,

$$Core(N, c) = \left\{ x \in \mathbb{R}^N \mid x(N) = c(N) \quad \text{i} \quad x(S) \leq c(S) \quad \text{per tot} \quad S \subseteq N \right\}.$$

Valor de Shapley

El valor de Shapley és un altre concepte de solució què assigna un únic vector de pagament a cada joc cooperatiu, i es basa a intentar calcular quin grau de participació té cada jugador a la coalició. Per això, s'utilitza el concepte de contribució marginal. Aquesta idea va ser introduïda per Lloyd S. Shapley (1953)[18]. Shapley interpreta el seu valor com que cada jugador rep la contribució marginal que ha aconseguit quan els jugadors es posen en una fila (una ordenació). I el que fa és el promig considerant que totes les files o ordenacions són equiprobables.

Definició 2.17. El *valor de Shapley* d'un joc (N, v) és un vector de pagaments $Sh(N, v) \in \mathbb{R}^N$ definit component a component com

$$Sh_i(N,v) = \sum_{S \subseteq N \setminus \{i\}} \frac{s! \cdot (n-s-1)!}{n!} \big(v(S \cup \{i\}) - v(S)\big) \text{ per tot } i \in N.$$

L'expressió $v(S \cup \{i\}) - v(S)$ és la contribució marginal del jugador i a la coalició S en el joc v, $s = |S|$ i $n = |N|$. A més, $Sh(N, v)$ és independent del tipus de joc, es pot aplicar tant a jocs de guanys com de costos, i queda caracteritzat per una sèrie d'axiomes o propietats que són l'eficiència, additivitat, simetria i la del jugador nul.

Si calculem el valor de Shapley de l'Exemple 2.16 anterior, s'obtenen:

$$\begin{aligned} Sh_1(N,v) =& \frac{0! \cdot (n-0-1)!}{n!} \Big(v\left(\{1\}\right) - v\left(\emptyset\right)\Big) + \\ & \frac{\big|\{2\}\big|! \cdot \big(n - \big|\{2\}\big| - 1\big)!}{n!} \Big(v\left(\{1,2\}\right) - v\left(\{2\}\right)\Big) + \\ & \frac{\big|\{3\}\big|! \cdot \big(n - \big|\{3\}\big| - 1\big)!}{n!} \Big(v\left(\{1,3\}\right) - v\left(\{3\}\right)\Big) + \\ & \frac{\big|\{2,3\}\big|! \cdot \big(n - \big|\{2,3\}\big| - 1\big)!}{n!} \Big(v\left(\{1,2,3\}\right) - v\left(\{2,3\}\right)\Big) \\ & = 0 + \frac{1! \cdot 1!}{3!}(300-0) + \frac{1! \cdot 1!}{3!}(400-0) + \frac{2! \cdot 0!}{3!}(600-500) = 150. \end{aligned}$$

$$\begin{aligned} Sh_2(N,v) =& \frac{0! \cdot (n-0-1)!}{n!} \Big(v\left(\{2\}\right) - v\left(\emptyset\right)\Big) + \\ & \frac{\big|\{1\}\big|! \cdot \big(n - \big|\{1\}\big| - 1\big)!}{n!} \Big(v\left(\{1,2\}\right) - v\left(\{1\}\right)\Big) + \\ & \frac{\big|\{3\}\big|! \cdot \big(n - \big|\{3\}\big| - 1\big)!}{n!} \Big(v\left(\{2,3\}\right) - v\left(\{3\}\right)\Big) + \\ & \frac{\big|\{1,3\}\big|! \cdot \big(n - \big|\{1,3\}\big| - 1\big)!}{n!} \Big(v\left(\{1,2,3\}\right) - v\left(\{1,3\}\right)\Big) \\ & = 0 + \frac{1! \cdot 1!}{3!}(300-0) + \frac{1! \cdot 1!}{3!}(500-0) + \frac{2! \cdot 0!}{3!}(600-400) = 200. \end{aligned}$$

$$
\begin{aligned}
Sh_3(N,v) =& \frac{0!\cdot(n-0-1)!}{n!}\Big(v\left(\{3\}\right)-v\left(\emptyset\right)\Big)+\\
& \frac{\big|\{1\}\big|!\cdot\big(n-\big|\{1\}\big|-1\big)!}{n!}\Big(v\left(\{1,3\}\right)-v\left(\{1\}\right)\Big)+\\
& \frac{\big|\{2\}\big|!\cdot\big(n-\big|\{2\}\big|-1\big)!}{n!}\Big(v\left(\{2,3\}\right)-v\left(\{2\}\right)\Big)+\\
& \frac{\big|\{1,2\}\big|!\cdot\big(n-\big|\{1,2\}\big|-1\big)!}{n!}\Big(v\left(\{1,2,3\}\right)-v\left(\{1,2\}\right)\Big)\\
& = 0+\frac{1!\cdot 1!}{3!}(400-0)+\frac{1!\cdot 1!}{3!}(500-0)+\frac{2!\cdot 0!}{3!}(600-300)=250.
\end{aligned}
$$

S'ha fet servir la funció característica definida a l'Exemple 2.16 i en aquest cas s'ha arribat al vector de pagament $x=(150,200,250)$ en el qual es pot veure que al jugador 1 se li ha concedit 50 unitats més de la quantitat assignada en cas del *core*, i per contra, al jugador 3 se li ha disminuït en 50 unitats aquesta quantitat. A més, aquest exemple prova que el valor de Shapley en general no ha de pertànyer al nucli. En el cas dels jocs convexos, però, el valor de Shapley sempre pertany al core.

Capítol 3

Arbre generador de cost mínim

Aquest capítol es divideix en tres seccions. A la primera s'introdueix la definició del problema d'arbre generador de cost mínim i algunes de les seves aplicacions. La segona secció analitza dos algoritmes diferents i útils per trobar de manera eficient l'arbre generador de cost mínim a partir d'un graf connex donat. I en l'última s'exposen diferents mètodes d'assignació de costos que es poden aplicar un cop s'ha obtingut l'arbre generador.

3.1 Definició del problema

El model de l'arbre generador correspon a una sèrie de problemes d'assignació de costos, els quals es poden trobar sobretot en les indústries de telecomunicacions i transport, ja que han d'oferir un servei a molts clients situats en diferents punts geogràfics, i el que es busca és optimitzar els recursos i trobar la mínima configuració de xarxa que connecti tots els clients però amb els mínims costos possibles.

Donat un graf no orientat, connex i amb costos, es vol trobar un subgraf que a la vegada sigui un arbre generador i el qual minimitzi el cost total. És a dir, un subgraf que connecti tots els seus vèrtexs i que la suma dels costos de les seves arestes sigui menor o igual a la suma de costos de les arestes que formin qualsevol altre arbre generador del graf. Aquest és un problema antic ja tractat des de meitat del segle passat el qual rep el nom de problema de l'*arbre generador de cost mínim,* o més breument es coneix com problema *mcst*.

Es troben moltes situacions econòmiques que poden ser tractades d'aquesta manera. Per exemple, diverses ciutats obtenen l'energia d'una central elèctrica en comú, i aleshores han de distribuir el cost d'aquesta xarxa de distribució. Una altra situació real estudiada per Bergantiños (2004)[1] va ser la construcció de canonades en un poble pel subministrament de l'aigua a les cases, en el qual els habitants havien de pagar el cost d'aquesta construcció. O d'altres exemples es poden trobar a les xarxes de comunicació com el telèfon o internet.

Una explicació detallada d'aquests problemes es pot trobar a Sharkey (1995)[19] que és un capítol de *Netwoks Models in Economics.*

Es considera un conjunt de nodes $N = \{1, 2, \ldots, n\}$ que representaran diferents clients del servei i un node $\{0\}$ que representarà a un proveïdor del servei al qui anomenarem *font.* Denotarem per $N_0 = N \cup \{0\}$ el conjunt format pels nodes clients i per la font.

Sigui un graf $G_{N_0} = (V_{N_0}, A)$ complet i no orientat, i suposem que G_{N_0} no conté llaços ni arestes múltiples que connectin dos nodes donats. Una *matriu de costos* $C = (c_{ij})_{i,j \in N_0}$ en N representa el cost de connectar de manera directa dos nodes. Es dóna per fet que $c_{ij} = c_{ji} \geq 0$ per cada $i, j \in N_0$ i $c_{ii} = 0$ per cada $i \in N_0$. El fet que $c_{ij} = c_{ji}$ es desprèn de què ens trobem en un graf no orientat.

Podem doncs, donar una definició formal del nostre problema, que està determinat dient quins són els clients a servir i el cost de connectar-los entre ells i també amb la font.

Definició 3.1. Un *problema d'arbre generador de cost mínim* o breument problema *mcst,* és un parell (N_0, C) on N és un conjunt finit d'agents, 0 és la font i C és la matriu de costos associada.

3.2 Algoritmes de Prim y Kruskal

Donat un problema *mcst* (N_0, C) i una xarxa $g \in G_{N_0}$ formada per un conjunt d'arestes, definim el cost associat a g com

$$c(N_0, C, g) = \sum_{(i,j) \in g} c_{ij}.$$

Quan no hi hagi problema de nomenclatura, escriurem $c(g)$ en lloc de $c(N_0, C, g)$. Aleshores un *arbre generador de cost mínim* de (N_0, C) és un arbre $T \in G_{N_0}$ tal que $c(T) = \min_{g \in G_{N_0}} c(g)$.

Un arbre generador de cost mínim, pot ser construït eficientment per mitjà d'alguns algoritmes, com poden ser el de Prim o de Kruskal què explicarem a continuació.

Algoritme de Prim

Prim (1957)[16] va introduir un algoritme simple per trobar-ne aquest arbre. La idea és la següent: partint de la font, es construeix una xarxa connectant consecutivament tots els clients de menor cost. Més formalment,

Pas 0: Es comença amb el subconjunt $S_0 = \{0\}$ i $a_0 = \emptyset$.

Pas 1: Seleccionem l'aresta $a = (0, i)$ tal que $c_{0i} = \min_{i \in N}\{c_{0i}\}$. Si es dóna el cas que hi ha més d'una aresta que compleix la condició, es selecciona qualsevol d'elles. Ara, $S_1 = \{0, i\}$ i $a_1 = (0, i)$.

. . . . . .

Pas p+1: Suposem $S_p \subset N_0$ i $a_p \in G_N$. Agafem l'aresta (j, i) amb $j \in S_p$ i $i \in N_0 \backslash S_p$ tal que

$$c_{ji} = \min_{k \in S_p, l \in N_0 \backslash S_p \in N} \{c_{kl}\}.$$

Es torna a seleccionar un qualsevol si hi hagués dos o més que satisfan la condició. Definim doncs, $S_{p+1} = S_p \cup \{i\}$ i $a_{p+1} = a_p \cup \{j, i\}$.

Aquest procés s'acaba en n etapes. De l'algoritme resulta un arbre $T = S_{p+1}$ que pot ser diferent depenent de com es resolguin els empats en cas que més d'una aresta compleixi les condicions.

Es pot veure gràficament a la Figura 3.1 com s'aplicaria pas a pas l'algoritme de Prim fins a obtenir un arbre generador de cost mínim del graf donat inicialment. En aquest cas, el graf és connex però no complet, ja que hi ha vèrtexs els quals no es connecten directament. Suposem, sense pèrdua de generalitat, que els vèrtexs que hi falten tenen un cost associat molt alt i mai serien seleccionats per l'algoritme.

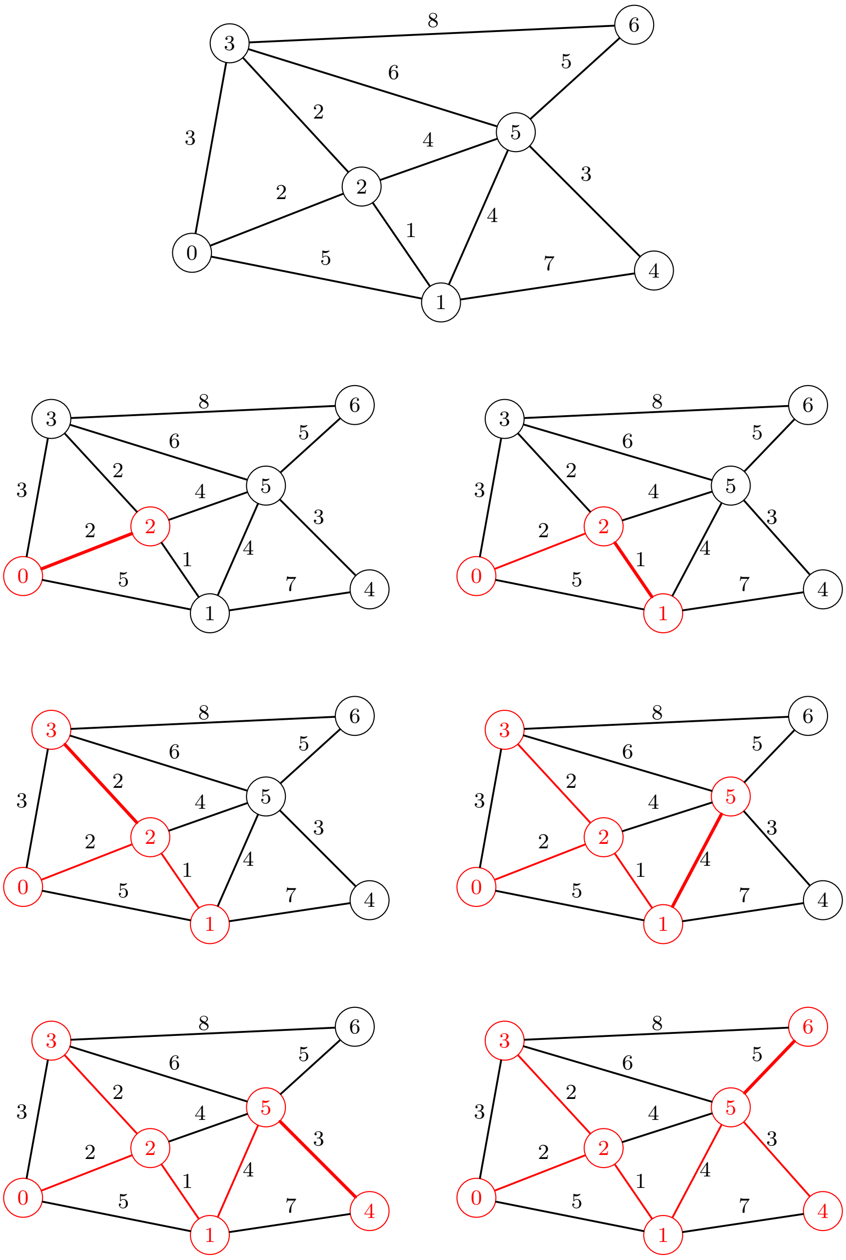


Figura 3.1: Aplicació per passos de l'algoritme de Prim.

En primer lloc, es comença amb el subconjunt $S_0 = \{0\}$ i les possibles arestes a escollir són $(0,1), (0,2)$ i $(0,3)$ les quals tenen un cost de connexió de 5, 2 i 3, respectivament. Per tant, s'agafa $a = (0,2)$. Ara $S_1 = \{(0,2)\}$. En el següent pas, es selecciona l'aresta de menor cost entre totes les incidents als vèrtexs 0 i 2, sempre que no es formi cap cicle. Les alternatives són $(0,1), (0,3)$ trobades anteriorment, més $(2,3)$ i $(2,1)$. Si es mira els costos de cadascuna $a = (2,1)$ és la que el té més petit. Ara $S_2 = \{(0,2) \cup (2,1)\}$. En el tercer pas, les possibles arestes són $(0,1),(0,3),(1,4),(1,5),(2,3)$ i $(2,5)$. La de menor cost és $(2,3)$, i per tant, s'obté $S_3 = \{(0,2)\cup(2,1)\cup(2,3)\}$. El quart pas, si es repeteix el procés, en principi la que compleix la condició seria $a = (0,3)$ però, es formaria un cicle i això no és possible. Per tant, les dues arestes següents que compleixen la condició són $(1,5)$ i $(2,5)$. En aquest cas, per exemple s'agafa $a = (1,5)$. Si s'hagués seleccionat l'altre, s'arribaria a un arbre generador diferent. Ara $S_4 = \{(0,2)\cup(2,1)\cup(2,3)\cup(1,5)\}$. Si es segueix repetint el procés, l'algoritme s'acabaria amb $S_5 = \{(0,2)\cup(2,1)\cup(2,3)\cup(1,5)\}\cup(5,4)$ i $S_6 = \{(0,2)\cup(2,1)\cup(2,3)\cup(1,5)\cup(5,4)\cup(5,6)\}$. On S_6 és l'arbre generador de cost mínim resultant.

Algoritme de Kruskal

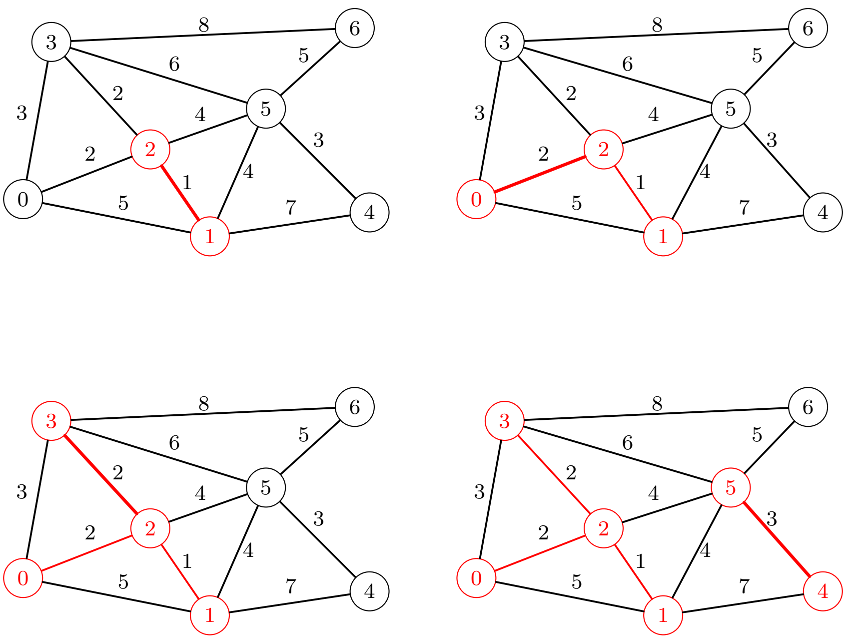
Hi ha un segon algoritme també capaç d'obtenir un arbre generador de cost mínim. Joseph Kruskal el va proposar l'any 1956[14]. La idea principal d'aquest algoritme es basa en connectar arestes i no nodes, de manera seqüencial. En aquest cas no serà necessari l'elecció d'un node de partida.

Procediment: Donat un graf $G = (V, A)$ no dirigit i una matriu de costos C. Es comença agafant els subconjunts $S_0 = \emptyset$ i $A_0 = \emptyset$.

Seguidament, es selecciona $a_{ij} \in A \setminus A_0$ tal que $c_{ij} = \min_{i,j \in N; i \neq j}\{c_{ij}\}$. En cas que hi hagi més d'una aresta que compleix la darrera condició, es selecciona qualsevol d'elles de manera indiferent. Ara, $S_1 = \{i, j\}$ i $A_1 = (i, j)$.

Es va repetint aquest procediment, tenint en compte que no es poden formar cicles, fins que tinguem $n-1$ arestes connectades, i és aleshores quan tots els nodes estaran units formant un arbre generador de cost mínim.

Si apliquem també aquest algoritme al graf inicial de la Figura 3.1 es faria de la següent forma:



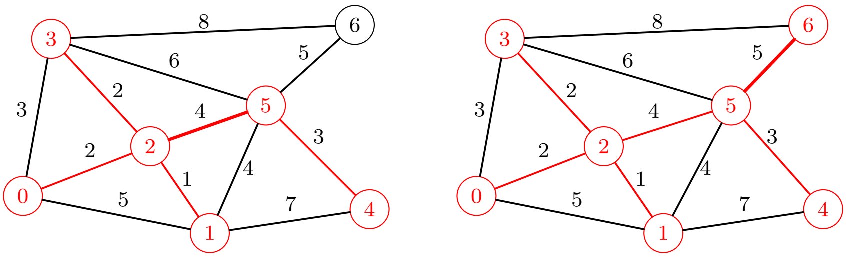


Figura 3.2: Aplicació per passos de l'algoritme de Kruskal.

En aquest cas, el que es selecciona són arestes en ordre creixent de costos. També s'ha de tenir en compte de no formar cap cicle. Al primer pas es selecciona l'aresta $a = (1, 2)$, ja que té un cost d'1. Seguidament, tenim dues opcions que tenen el mateix cost de 2, són les arestes $(0, 2)$ i $(2, 3)$. S'agafa una qualsevol i al següent pas l'altre. Es va repetint aquest procés fins a seleccionar 6, és a dir, $(n-1)$ arestes del graf. Al final obtenim l'arbre generador de cost mínim $S_6 = \{(1, 2) \cup (2, 0) \cup (2, 3) \cup (4, 5) \cup (2, 5) \cup (5, 6)\}$.

Es pot veure que l'arbre obtingut en aquest algoritme $T = S_6$ és diferent de l'obtingut anteriorment amb el de Prim. És l'arbre que s'hagués obtingut si es seleccionés l'altra aresta de cost 4 en el pas quart de l'anterior algoritme, però veiem que el cost obtingut en els dos arbres és el mateix.

3.3 Regles de repartició de costos

Un altre aspecte important a considerar en aquests problemes d'assignació de costos és la manera de distribuir el cost total d'aquesta xarxa entre els diferents clients.

Una *regla de distribució de costos* és una funció ψ que associa a cada problema d'arbre generador de cost mínim (N_0, C) un vector $\psi(N_0, C) \in \mathbb{R}^n$ tal que

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

on cada component del vector és el cost que paga el jugador i corresponent i $c_{mcst}(N_0, C)$ és el cost associat a qualsevol arbre generador de cost mínim per (N_0, C).

Quatre de les regles més conegudes d'assignació de costos són la regla de Bird (Bird, 1976 [7]), la regla de Kar (Kar, 2002 [13]), la regla de Dutta-Kar (Dutta i Kar, 2004 [10]) i la regla de Tijs et al. (Tijs et al., 1994 [20]). Les dues primeres regles estan basades en l'algoritme de Prim, la de Kar utilitza el valor de Shapley i per últim la de Tijs et al. es defineix a través de l'algoritme de Kruskal.

Es poden trobar a Bergantiños et al. (2008)[5] una bona explicació de les regles i també a Cesco et al. (2013)[8] amb exemples clars d'aplicació d'aquestes.

Regla de Bird

Es considera un problema d'arbre de cost mínim (N_0, C) i el seu arbre generador de cost mínim corresponent obtingut mitjançant l'algoritme de Prim i es suposa que aquest és únic, és a dir que no hi han aparegut casos en què més d'una aresta complia les condiciones en cap pas de l'algoritme.

La regla de Bird es basa en la idea que el cost que haurà de pagar el client i serà el cost de l'aresta incident a ell en l'únic camí que connecta i amb la font. Per tant, la *regla de Bird* (Bird, 1976 [7]) està definida com:

$$B_i(N_0, C) = c_{i^0 i} \text{ per cada } i \in N.$$

On i^0 representa el predecessor immediat de i, i anomenarem com $B(N_0, C)$ el vector que té per components el valor que ha de pagar cada client i segons aquesta regla.

Quan existeix més d'un arbre generador de cost mínim, es fa un promig entre tots els possibles arbres. Denotarem per Π_N el conjunt de totes les permutacions sobre N. Per cada $\pi \in \Pi_N$, diem $B^\pi(N_0, C)$ a la distribució obtinguda amb la regla de Bird d'arbre únic, tenint en compte les prioritats descrites per π, és a dir, en cas d'empat, aquest es resol fent servir l'ordre indicat. Es defineix la regla de Bird, independentment del nombre d'arbres generadors de cost mínim obtinguts, com:

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

Seguim amb l'exemple donat per la Figura 3.1 d'aquest capítol. Es calculen quines serien les reparticions de costos que es durien a terme segons aquesta regla.

Com ja s'havia comentat, l'arbre generador de cost mínim trobat mitjançant l'algoritme de Prim, no era l'únic possible. Per tant, per obtenir $B(N_0, C)$, s'haurà de calcular $B^\pi(N_0, C)$ per cada $\pi \in \Pi_N$. Considerant la permutació $\pi = (1, 2, 3, 4, 5, 6)$ i assignant el cost de cada aresta al nou client que es va afegint, s'obté $B^\pi(N_0, C) = (1, 2, 2, 3, 4, 5)$. Es procedeix anàlogament per cada permutació, en aquest cas $6! = 720$. Però és fàcil veure que amb totes s'arriba al mateix repartiment. Per tant,

$$B(N_0, C) = \frac{1}{720} \sum_{\pi \in \Pi_N} B^\pi(N_0, C) = (1, 2, 2, 3, 4, 5).$$

Regla de Dutta-Kar

Una segona regla per l'assignació de costos que comentarem és la regla de Dutta-Kar. Aquesta el que fa és repartir els costos de forma que cada client escull el mínim cost entre que li correspondria segons la regla de Bird i el del seu successor immediat. Se suposa que hi ha un únic arbre generador de cost mínim i que els clients estan connectats a la font segons l'algoritme de Prim en ordre $i_1, i_2, \ldots, i_n$. Aleshores la *regla de Dutta-Kar* (Dutta i Kar, 2004 [10]) ve donada per

$$DK_{i_k}(N_0, C) = x_{i_k}, \quad \text{per} \quad k = 1, 2, \ldots, n,$$

on x_{i_k} es calcula de la forma que s'explica a continuació. Primer el client i_1 es connecta a $\{0\}$ i $p_1 = c_{0i_1}$. Al segon pas, el client i_2 es connecta a i_2^0 que és el camí format per l'algoritme de Prim abans de la incorporació de l'agent i_2 i es defineix $x_{i_1} = \min\{p_1, c_{i_2^0 i_2}\}$ i $p_2 = \max\{p_1, c_{i_2^0 i_2}\}$. Al tercer, el client i_3 es connecta a i_3^0 i anàlogament es defineix $x_{i_2} = \min\{p_2, c_{i_3^0 i_3}\}$ i $p_3 = \max\{p_2, c_{i_3^0 i_3}\}$. Es repeteix el procediment fins a la incorporació del client i_n amb $x_{i_{n-1}} = \min\{p_{n-1}, c_{i_n^0 i_n}\}$ i $x_{i_n} = p_n$. Aquesta regla afavoreix als clients que estan connectats prèviament a la font.

Anomenarem com $DK(N_0, C)$ el vector que té per components el valor que ha de pagar cada client segons aquesta regla. En el cas que hi hagués més d'un arbre generador, aquesta regla s'estén de manera anàloga que la de Bird. És a dir,

$$DK(N_0, C) = \frac{1}{n!} \sum_{\pi \in \Pi_N} DK^{\pi}(N_0, C).$$

Es torna a la Figura 3.1 i es considera la mateixa permutació $\pi = (1, 2, 3, 4, 5, 6)$ com en el cas anterior i l'arbre generador de l'algoritme de Prim. S'aplica la regla de Dutta-Kar per buscar l'assignació de la següent manera:

Els agents estan connectats amb l'ordre $v_2, v_1, v_3, v_5, v_4, v_6$. En el primer pas, el client v_2 es connecta a la font i $p_1 = 2$. En el següent, $p_2 = \max\{2, 1\} = 2$, i $DK_2^{\pi} = \min\{2, 1\} = 1$. Es va repetint el procediment i en el penúltim pas s'obté $p_6 = \max\{4, 5\} = 5$, i $DK_4^{\pi} = \min\{4, 5\} = 4$. Per acabar $DK_6^{\pi} = 5$. Per tant, $DK^{\pi}(N_0, C) = (2, 1, 2, 4, 3, 5)$. En aquest cas, també s'arriba al mateix vector d'assignació de costos en totes les permutacions possibles.

$$DK(N_0, C) = \frac{1}{720} \sum_{\pi \in \Pi_N} DK^{\pi}(N_0, C) = (2, 1, 2, 4, 3, 5).$$

Es pot veure que segons els vectors d'assignació de costos obtinguts amb els dos mètodes explicats fins ara, els clients v_1 i v_4 preferiran que es realitzi una repartició de costos segons la regla de Bird perquè el cost que haurien de suportar seria menor, en canvi, els clients v_2 i v_3 preferiran l'assignació de Dutta-Kar, mantenint-se els clients v_3 i v_6 indiferents davant aquesta decisió.

Regla de Kar

Una tercera regla és la regla de Kar. Aquesta es defineix com el valor de Shapley del joc d'arbre generador de cost mínim associat. El valor de Shapley va ser definit al Capítol de preliminars d'aquesta memòria. Aleshores donat un problema *mcst* (N_0, C) i sigui (N, c) el respectiu joc associat,[1] on N és el conjunt de clients i c la funció característica, es defineix la *regla de Kar* (Kar, 2002 [13]), què anomenarem K, com:

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

Si es considera el conjunt de permutacions Π sobre el conjunt de jugadors N. Sigui $\pi \in \Pi$, $P^i(i, \pi) = \{j \in N \mid \pi(j) < \pi(i)\}$ és el conjunt de jugadors que precedeixen a i en π, on $P^i(i, \pi) = \emptyset$ si el jugador i és el primer en l'ordre π. Aleshores podem definir també el valor de Sapley com:

$$Sh_i(N, c) = \frac{1}{n!} \sum_{\pi \in \Pi_N} [c(P^i(i, \pi) \cup \{i\}) - c(P^i(i, \pi))].$$

[1] El joc de costos associat al problema es definirà al capítol següent, i essencialment el cost per una coalició és el cost necessari per servir a tots els clients de la coalició. Es veurà que si cal es poden fer servir arestes que passen per clients de fora de la coalició.

Regla ERO (Equal Remaining Obligation)

Aquesta darrera regla va ser introduïda per Tijs et al. (1994) [20]. És una regla que a diferència de la de Bird i Dutta-Kar es defineix a partir de l'algoritme de Kruskal. Es vol arribar a construir un vector d'assignació de costos el qual s'anomenarà ERO (Equal Remaining Obligations). La idea d'aquesta regla és que en cada etapa de l'algoritme on s'afegeix una aresta a l'arbre, el cost d'aquesta serà pagat per tots els clients que es beneficiaran d'ella i cadascun pagarà la diferència entre la seva obligació a pagar abans d'afegir-la i la d'un cop afegida.

Sigui (N_0, C) un problema d'arbre generador de cost mínim. L'algoritme d'aquesta regla s'aplica de la manera següent:

- **Pas 1**: S'agafa $t = 0$, $\tau = |N_0| - 1$ i $A_0 = \emptyset$.
- **Pas 2**: $t := t + 1$.
- **Pas 3**: Es selecciona l'aresta a_t aplicant l'algoritme de Kruskal, i,e., l'aresta de cost mínim que no pertanyi ja a l'arbre i sense que es formin cicles al graf $G_t = (N_0, A_{t-1} \cup a_t)$.
- **Pas 4**: Si anomenem $W^t = W_i^{t-1} \cup W_j^{t-1}$ a la component connexa formada en connectar W_i^{t-1} i W_j^{t-1} de G_{t-1} a través de l'aresta $a_t = (i, j)$. Definim el vector $f^t = (f_k^t)_{k \in N}$ com,

$$f_k^t = \begin{cases} \frac{1}{|W_i^{t-1}|} - \frac{1}{|W^t|} & si \qquad k \in W_i^{t-1} \quad i \quad \{0\} \notin W^t, \\[2ex] \frac{1}{|W_j^{t-1}|} - \frac{1}{|W^t|} & si \qquad k \in W_j^{t-1} \quad i \quad \{0\} \notin W^t, \\[2ex] \frac{1}{|W_i^{t-1}|} & si \qquad k \in W_i^{t-1} \quad i \quad \{0\} \in W_j^{t-1}, \\[2ex] \frac{1}{|W_j^{t-1}|} & si \qquad k \in W_j^{t-1} \quad i \quad \{0\} \in W_i^{t-1}, \\[2ex] 0 & \text{en altre cas.} \end{cases}$$

- **Pas 5**: Es defineix $A_t := A_{t-1} \cup \{a_t\}$.
- **Pas 6**: Si $t < \tau$, tornem al pas 2.
- **Pas 7**: A_τ és l'arbre generador de cost mínim associat al problema.

Aleshores la *regla ERO* es defineix com:

$$ERO(N_0, C) = \sum_{t=1}^{\tau} f^t c_{ij}.$$

On c_{ij} és el cost associat a l'aresta a_t en el pas t.

En general, com es va veure, l'arbre resultant de l'aplicació de l'algoritme de Kruskal no era únic, ja que podia haver-hi més d'una aresta amb el mateix cost que complissin les condicions. Tot i això, Tijs et al. (Tijs, Muto i Feltkamp, 1994 [20, Proposició 4.2]) demostren que aquest valor ERO és independent de l'arbre escollit.

Es defineix pel problema (N_0, C) l'obligació inicial o_i del jugador i com

$$o_i := \begin{cases} \frac{1}{|W_i|} & si \quad \{0\} \notin W_i, \\ \\ 0 & si \quad \{0\} \in W_i, \end{cases}$$

On W_i és el component de $G = (N_0, A)$ que conté al jugador i.

Per la seqüència de vectors $\mathcal{F} = (f_1 \cdots f_\tau)$, després del pas $t \leq \tau$ un jugador $i \in N$ ha pagat $\sum_{s \leq t} f_i^s$, mentre que l'obligació inicial era o_i. Aleshores l'obligació restant (*remaining obligation*) o_i^t satisfà

$$o_i^t = o_i - \sum_{s \leq t} f_i^s = o_i^{t-1} - f_i^t.$$

Per tant, cada jugador $i \in N$ paga exactament la diferència f_i^t entre l'obligació restant o_i^{t-1} i o_i^t.

Podem veure un exemple de com s'aplica aquesta regla a continuació, on es fa el càlcul de la regla ERO aplicada a la Figura 3.2 anterior.

Primerament, seguint l'algoritme de Kruskal, es selecciona l'aresta $(1, 2)$, ja que és la que té el cost més baix i $W^1 = \{(1, 2)\}$ serà la component connexa. Per calcular el vector f^1 veiem que $\{0\}$ no pertany a la component connexa i per $k = 1, \ldots, 6$, calculem el vector. En aquest cas únicament 1 i 2 pertanyen a W^1 aleshores $f_1^1 = f_2^1 = \frac{1}{2}$ i zero en cas contrari.

En el segon pas, l'aresta seleccionada és $(0, 2)$. Ara $W^2 = \{(1, 2) \cup (0, 2)\}$. En aquest cas $\{0\}$ si pertany a la component connexa i calculant el vector com s'ha definit en la regla ERO, en aquest cas també s'obté $f_1^2 = f_2^2 = \frac{1}{2}$ i zero en cas contrari.

Repetint el procediment en totes les altres arestes que s'incorporen mitjançant l'algoritme de Kruskal s'arriba a calcular tots els vectors f^t per $t = 1, \ldots, 6$.

Es calcula ara la regla ERO amb els vectors obtinguts anteriorment i els costos associats a cada aresta del graf de la següent manera:

$$ERO(N_0, C) = \sum_{t=1}^{6} f^t c_{ij} = c_{12}\left(\frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0\right) + c_{02}\left(0, \frac{1}{2}, \frac{1}{2}, 0, 0, 0\right)$$

$$+ c_{23}\left(0, \frac{1}{2}, \frac{1}{2}, 0, 0, 0\right) + c_{45}\left(0, 0, 0, \frac{1}{2}, \frac{1}{2}, 0\right) + c_{25}\left(0, \frac{1}{2}, 0, 0, \frac{1}{2}, 0\right)$$

$$+ c_{56}\left(0, 0, 0, 0, \frac{1}{2}, \frac{1}{2}\right) = \left(\frac{3}{2}, \frac{9}{2}, 1, \frac{3}{2}, 6, \frac{5}{2}\right).$$

Capítol 4

Jocs cooperatius d'arbres generadors de cost mínim

En aquest capítol es relacionen les regles anteriors amb els jocs cooperatius. Als preliminars ja es van estudiar els jocs cooperatius de costos i el que ara veurem és com es relaciona el nostre problema de l'arbre generador de cost mínim amb un joc cooperatiu de forma natural. Essencialment es tracta de definir per cada coalició el cost de l'arbre mínim que connecta tots els agents de la coalició amb la font. De seguida es veu que el joc així construït pot no ser ni tan sols monòton, i això porta a definir el joc monòton associat.

S'estudien les propietats d'aquest joc, i en particular es veu que sempre el joc és equilibrat (té el *core* no buit) i s'identifiquen les regles com a punts del *core* del joc. També estudiem la forma irreductible d'un problema, cosa que permet identificar en aquest cas l'equivalència de regles. Les propietats de les regles ara es poden estudiar atenent al joc associat, i permeten caracteritzar axiomàticament les regles.

4.1 Core d'un joc d'arbre generador de cost mínim

A un problema d'arbre generador de cost mínim se li pot associar un joc cooperatiu TU, més concretament un joc de cost. Per qualsevol coalició $S \subseteq N$ de clients, $c(S)$ denotarà el mínim cost de connectar tots els vèrtexs de la coalició a la font, el que defineix un subproblema del problema original. Per tant, tenim la següent definició.

Definició 4.1. Donat un graf complet $G_N = (V_N, A)$ amb la matriu de costos $C = \big(c_{ij}\big)$ definida anteriorment, un *joc d'arbre generador de cost mínim* o *joc mcst*, (N, c), està definit pel conjunt de jugadors $N = \{1, 2, \ldots, n\}$ i la funció de cost c que compleix

$$c(\emptyset) = 0 \quad \text{i} \quad c(S) = \sum_{a_{ij} \in A_S} c_{ij} \quad \text{per cada} \quad S \subseteq N,$$

on A_S és el conjunt d'arestes de l'arbre generador de cost mínim en el graf complet $G_S = (S_0, A_S)$ amb $S_0 = S \cup \{0\}$ i $A_S \subseteq A$.

Un joc d'arbre generador de cost mínim, conegut més breument com joc *mcst (minimum cost spanning tree)*, és aquell que connecta cada client de S amb la font utilitzant tots els vèrtexs de S i va creant una xarxa la qual té un cost més petit que qualsevol altra possible. Sempre es pot escollir una xarxa més barata sense cicles, és a dir, un arbre.

Cada joc *mcst* té associada una funció característica c que és subadditiva, però aquesta no ha de ser neccesàriament monòtona. Recordem que un joc de costos (N, c) és subadditiu si $c(S) + c(T) \geq c(S \cup T)$ per qualssevol $S, T \subseteq N$ disjuntes, $S \cap T = \emptyset$. En aquest cas és evident, ja que el cost dels arbres per separat és més gran o igual del que seria un arbre mínim que connecti tots els vèrtexs de la unió de les dues coalicions.

Es defineix doncs, un *joc mcst monòton* $(N, \overline{c})$ derivat d'un joc *mcst* (N, c) com un joc on la seva funció de cost $\overline{c}$ està definida de la següent manera:

$$\overline{c}(S) = \min_{S \subseteq T \subseteq N} c(T).$$

Aquesta funció de cost permet a una coalició S utilitzar vèrtexs de $N \setminus S$ en la construcció de l'arbre de cost mínim. Aquesta funció també és subadditiva i, a més a més, $\overline{c}(N) = c(N)$ i $\overline{c}(S) \leq c(S)$ per tot $S \subseteq N$. També es verifica que $\overline{c}(S) \leq \overline{c}(T)$ per tot $S \subseteq T \subseteq N$, és a dir, $\overline{c}$ és una funció monòtona creixent.

El proper exemple està basat en un exemple del llibre de Rafels et al.(1999) [17].

Exemple 4.2. Es considera un cas particular amb tres clients, $N = \{1, 2, 3\}$, la font o origen del servei $\{0\}$ des d'on es vol subministrar i uns costos associats c_{ij} representats mitjançant la següent matriu de costos,

$$C = \begin{pmatrix} 0 & 5 & 9 & 10 \\ 5 & 0 & 3 & 4 \\ 9 & 3 & 0 & 2 \\ 10 & 4 & 2 & 0 \end{pmatrix}$$

A la Figura 4.1 es pot veure el graf que li correspon a aquesta situació i a cada aresta es troben indicats els costos de connexió entre cada parell de vèrtexs adjacents a ella.

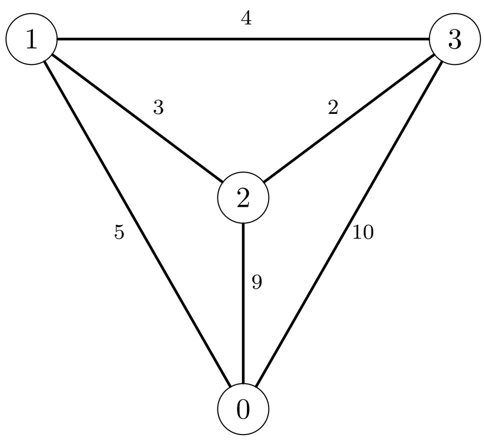


Figura 4.1: Graf corresponent al joc de l'Exemple 4.2.

Aquesta situació plantejada com un joc d'arbre generador de cost mínim, fàcilment es pot veure que la funció característica per coalicions individuals ve donada per

$$c(\{1\}) = 5, \quad c(\{2\}) = 9, \quad c(\{3\}) = 10.$$

Per trobar la resta de funció característica, ens fixem en les coalicions de dos clients, per exemple $\{1,2\}$ i s'obté tres possibilitats per connectar-los a la font. Es poden veure quines són aquestes a la Figura 4.2.

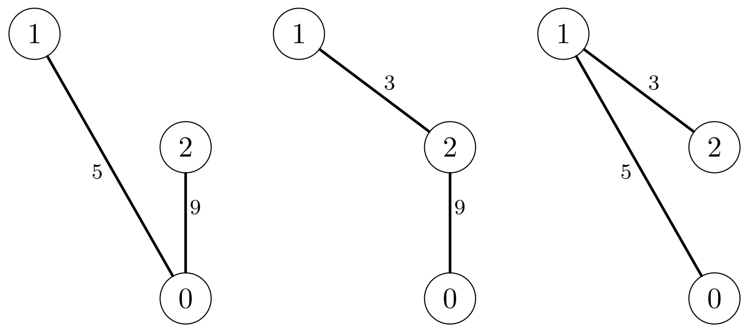


Figura 4.2: Possibles arbres generadors per la coalició {1,2}.

Clarament, l'última opció és la que té el cost mínim amb un valor de 8, per contra, els altres tenen un cost de $c(S_1) = 14$ i de $c(S_2) = 12$, respectivament. Es repeteix el mateix procediment per les altres dues coalicions possibles, $\{1,3\}$ i $\{2,3\}$. En aquests casos, en tenir únicament una coalició de dos clients ha sigut fàcil veure-ho d'aquesta manera. Quan el nombre de clients a la coalició és més elevat, el nombre de combinacions possibles es dispara, i llavors es pot arribar a trobar el camí de cost mínim aplicant els algoritmes de Prim i Kruskal explicats a l'anterior capítol. Aplicant aquests arribem a completar la funció característica del joc descrit,

$$c(\{1,2\}) = 5+3 = 8, \quad c(\{1,3\}) = 5+4 = 9, \quad c(\{2,3\}) = 9+2 = 11,$$

$$c(\{1,2,3\}) = 5+3+2 = 10.$$

Aquesta funció trobada podem veure que com tota funció característica associada a un joc d'arbre generador de cost mínim és subadditiva, ja que per tot subconjunt $S, T \subset N$, amb $S \cap T = \emptyset$, $c(S \cup T) \leq c(S) + c(T)$. En canvi, en aquest cas no és una funció monòtona, ja que, per exemple, $c(\{1,2\}) = 8 \leq 9 = c(\{2\})$ o $c(\{1,3\}) = 9 \leq 10 = c(\{3\})$.

La Taula 4.1 que trobem a continuació mostra la funció característica que acabem de trobar associada al joc cooperatiu descrit, així com també la funció del joc cooperatiu d'arbre generador de cost mínim monòton derivat del problema.

	$c(S)$	$\overline{c}(S)$		$c(S)$	$\overline{c}(S)$		$c(S)$	$\overline{c}(S)$
$\{1\}$	5	5	$\{1,2\}$	8	8	$\{1,2,3\}$	10	10
$\{2\}$	9	8	$\{1,3\}$	9	9	$\emptyset$	0	0
$\{3\}$	10	9	$\{2,3\}$	11	10			

Taula 4.1: Funció característica del joc de l'Exemple 4.2 i la del joc monòton derivat d'aquest.

L'arbre generador de cost mínim obtingut en el cas que tots els clients participin, $N = \{1, 2, 3\}$, mitjançant l'algoritme de Prim és:

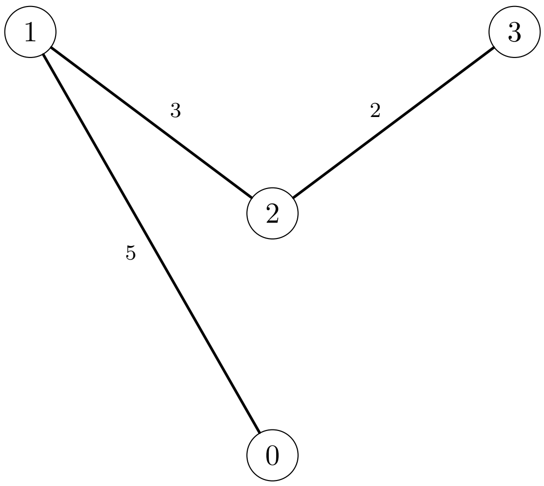


Figura 4.3: Arbre generador de cost mínim per a la coalició $N = \{1, 2, 3\}$.

Un cop obtingut l'arbre generador de cost mínim es vol trobar una repartició eficient del cost total entre els tres clients, en aquest cas un cost $c(N) = 10$, que també compleixi la racionalitat coalicional i per consegüent que pertanyi al nucli del joc. Per tant, aquesta assignació que serà un vector de pagaments $x = (x_1, x_2, x_3) \in \mathbb{R}^3$, ha de complir les següents condicions:

$$\begin{aligned}
&x_1 + x_2 + x_3 = 10,\\
&x_1 \leq 5,\ x_2 \leq 9,\ x_3 \leq 10,\\
&x_1 + x_2 \leq 8,\\
&x_1 + x_3 \leq 9,\\
&x_2 + x_3 \leq 11.
\end{aligned}$$

Substituint i simplificant, s'arriba al conjunt

$$\begin{aligned}
Core(N,c) =& \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_1 + x_1 + x_3 = 10,\\
& \quad 0 \leq x_1 \leq 5, 1 \leq x_2 \leq 9, 2 \leq x_3 \leq 10\}.
\end{aligned}$$

Existeixen distribucions que compleixin aquestes condicions? En el capítol anterior s'han explicat diferents regles de repartició d'aquests costos.

Si es busca una assignació de costos d'aquest joc mitjançant la regla de Bird, on primerament s'ha obtingut l'arbre generador de cost mínim mitjançant l'algoritme de Prim, s'obté el vector de la regla de Bird $B(N_0, C) = (c_{01}, c_{12}, c_{23}) = (5, 3, 2)$, és a dir, el primer client hauria d'assumir un cost de 5, el segon de 3 i el tercer de 2. Si en canvi es fes amb la regla de Dutta-Kar s'obtindria $DK(N_0, C) = (3, 2, 5)$, on el primer hauria de pagar 3, el segon 2 i el tercer 5. Es comprova fàcilment que ambdues distribucions pertanyen al nucli del joc d'arbre generador de cost mínim descrit anteriorment.

També es pot buscar una repartició amb la regla de Kar, la qual es fa buscant el valor de Shapley del joc. Si es calcula aquest valor en aquest joc, s'obté,

$$
\begin{aligned}
Sh_1(N,c) &= \frac{1}{3}c(\{1\}) + \frac{1}{6}(c(\{1,2\}) - c(\{2\}) + \frac{1}{6}(c(\{1,3\}) - c(\{3\}) + \frac{1}{3}(c(\{1,2,3\}) - \\
&\quad - c(\{2,3\}) = \frac{1}{3}(5) + \frac{1}{6}(8-9) + \frac{1}{6}(9-10) + \frac{1}{3}(10-11) = 1. \\
Sh_2(N,c) &= \frac{1}{3}c(\{2\}) + \frac{1}{6}(c(\{1,2\}) - c(\{1\}) + \frac{1}{6}(c(\{2,3\}) - c(\{3\}) + \frac{1}{3}(c(\{1,2,3\}) - \\
&\quad - c(\{1,3\}) = \frac{1}{3}(9) + \frac{1}{6}(8-5) + \frac{1}{6}(11-10) + \frac{1}{3}(10-9) = 4. \\
Sh_3(N,c) &= \frac{1}{3}c(\{3\}) + \frac{1}{6}(c(\{1,3\}) - c(\{1\}) + \frac{1}{6}(c(\{2,3\}) - c(\{2\}) + \frac{1}{3}(c(\{1,2,3\}) - \\
&\quad - c(\{1,2\}) = \frac{1}{3}(10) + \frac{1}{6}(9-5) + \frac{1}{6}(11-9) + \frac{1}{3}(10-8) = 5.
\end{aligned}
$$

En aquest cas, s'ha arribat al vector d'assignació de costos

$$Sh(N,c) = K(N_0, C) = (1, 4, 5),$$

i es pot veure que també pertany al nucli d'aquest joc.

Per últim, després d'aplicar l'algoritme de Kruskal per trobar l'arbre generador de cost mínim, s'aplica la regla ERO per trobar-hi aquesta repartició. Si es fa, s'arriba a un vector que com els altres també pertany al nucli. Aquest es calcula de la següent manera:

$$ERO(N_0, C) = \sum_{t=1}^{3} f^t c_{ij} = c_{23}\left(1-1, 1-\frac{1}{2}, 1-\frac{1}{2}\right) + c_{12}\left(1-\frac{1}{3}, \frac{1}{2}-\frac{1}{3}, \frac{1}{2}-\frac{1}{3}\right)$$

$$+c_{01}\left(\frac{1}{3}-0, \frac{1}{3}-0, \frac{1}{3}-0\right) = 2\left(0, \frac{1}{2}, \frac{1}{2}\right) + 3\left(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right) + 5\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) =$$

$$= \left(\frac{11}{3}, \frac{19}{6}, \frac{19}{6}\right).$$

En resum, a la Taula 4.2 podem veure els repartiments obtinguts amb cadascuna de les regles aplicades anteriorment.

Regla/Client	Client 1	Client 2	Client 3
Bird	5	3	2
Dutta-Kar	3	2	5
Kar	1	4	5
ERO	11/3	19/6	19/6

Taula 4.2: Regles de repartició de l'Exemple 4.2

Demostrem a continuació que tots els jocs d'arbres generadors de cost mínim són equilibrats, és a dir, que tenen nucli no buit.

Proposició 4.3. *Tot joc d'arbre generador de cost mínim és equilibrat.*

Demostració. Sigui T_N l'arbre generador de cost mínim per V_N i c_i el cost de l'aresta $a_{i,p(i)}$, on $p(i)$ és el predecessor de i en l'arbre.

Demostrem que el vector de pagaments $x \in \mathbb{R}^N$ tal que $x_i = c_i$ per tot $i \in N$ satisfà les condicions de nucli.

Primerament, veiem que $x \in \mathbb{R}^N$ és eficient. Si sumem els costos de connectar cada client amb el seu predecessor en l'arbre generador de cost mínim T_N, s'obté el cost total, ja que com s'ha definit el joc, $x(N) = \sum_{i=1}^{n} c_i = c(N)$. Per tant, x és eficient.

Suposem ara que $x(S) > c(S)$ per algun $S \subset N$. Sigui T_S l'arbre generador de cost mínim per V_S amb els costos associats c_i^S. Aleshores $\sum_{i \in S} c_i > \sum_{i \in S} c_i^S$. Sigui A_S el conjunt d'arestes de T_N i $A_{S'}$ els arcs associats a $S' = N \setminus S$ en T_N.

És fàcil demostrar que el graf $G = (V, A_S \cup A_{S'})$ és un arbre, ja que per cada $i \in N$, existeix un camí format per les arestes de A_S que connecten i amb algun vèrtex $j \in S$. En efecte, T_N no pot ser l'arbre generador de cost mínim, contradient la hipòtesi inicial. Llavors és veritat que $x(S) \leq c(S)$ per cada $S \subset N$, segona condició necessària perquè el vector x pertanyi al nucli del joc. Queda demostrat que el vector x pertany al nucli del joc. $\square$

De fet, com tot subjoc *mcst* és també un joc *mcst* i en conseqüència és equilibrat, pot afirmar-se doncs que els jocs d'arbre generador de cost mínim són totalment equilibrats.

S'ha vist en el joc d'arbre generador de cost mínim de l'Exemple 4.2 que totes les assignacions de costos mitjançant cadascuna de les regles pertany al nucli del joc. En el cas de les regles de Bird, Dutta-Kar i ERO es compleix en general, però en el cas de la regla de Kar no té per què ser així.

De la definició de nucli d'un joc de cost, es dedueix que el nucli d'un joc *mcst* (N, c) és un poliedre convex en $\mathbb{R}^n$. Qualsevol assignació de costos de l'arbre de cost mínim amb la regla de Bird és un punt en el poliedre definit pel nucli de (N, c). De fet, és un extrem del poliedre.

Teorema 4.4. *Cada assignació de la regla de Bird de l'arbre generador de cost mínim és un extrem del nucli del joc* mcst *associat.*

Demostració. Sigui $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ un repartiment segons la regla de Bird de T, l'arbre generador de cost mínim corresponent a (N_0, C). Anomenem als nodes $\{1, \ldots, n\}$ de T de forma que si $i > j$, aleshores i no està a l'únic camí en T que connecta el vèrtex j amb 0. Per veure que el vector x és un extrem del poliedre que forma el nucli, hem de veure que hi ha n desigualtats linealment independents en

$$\sum_{i \in S} x_i \leq c(S) \quad \text{per tot } S \subseteq N,$$

que són satisfetes en igualtat pel vector x. En efecte, les desigualtats que corresponen als conjunts de vèrtexs consecutius de l'arbre, per construcció, són les igualtats buscades. Si aquest fos un arbre lineal serien $\{1, \ldots, i\}$ amb $i = 1, \ldots, n$. □

La prova que la regla de Dutta-Kar també és una assignació del nucli, es pot trobar a Dutta i Kar (2004) [10, Teorema 1] i la de ERO a Bergantiños (2007) [4, Teorema 4.1].

En canvi, la regla de Kar tot i que en l'Exemple 4.2 pertany al nucli, no és així en general. Veiem-ho en un exemple.

Exemple 4.5. Considerem un problema *mcst* (N_0, c) amb $N = \{1, 2, 3\}$ i la matriu de costos associada següent:

$$C = \begin{pmatrix} 0 & 2 & 6 & 2 \\ 2 & 0 & 2 & 15 \\ 6 & 2 & 0 & 2 \\ 2 & 15 & 2 & 0 \end{pmatrix}$$

La funció característica d'aquest joc, c, juntament amb la funció característica del joc monòton derivat d'aquest, $\bar{c}$, estan representades a la següent taula:

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

Calculant el nucli d'aquest joc, s'arriba a que té un core amb únic punt,

$$Core(N, c) = \{(2, 2, 2)\}.$$

Ara bé, el vector de costos resultant a l'aplicar la regla de Kar de manera anàloga a l'exemple anterior, s'obté l'assignació

$$K(N_0, C) = Sh(N, c) = \left(\frac{4}{3}, \frac{10}{3}, \frac{4}{3}\right).$$

Per tant, la repartició segons la regla de Kar en aquest exemple no pertany al core del joc com s'ha pogut veure.

4.2 La forma irreductible

Aquesta secció està dedicada a l'estudi de la forma irreductible d'un problema d'arbre generador de cost mínim. Va ser introduïda per Bird el 1976 [7]. La forma irreductible d'un problema consisteix a modificar els valors de cost de les arestes sense modificar el problema ni la seva solució. Es rebaixa el valor d'arestes no essencials fins que no es poden baixar més.

Aquesta forma té la propietat que si reduïm el cost de qualsevol arc, també es redueix el cost de connectar els clients a la font. A més, presentem un procediment per associar a cada problema *mcst* amb la seva forma irreductible. Finalment s'arriba a unes equivalències entre les regles de repartició en la forma irreductible.

Donat un problema *mcst* (N_0, C) i un arbre generador de cost mínim T, es defineix *problema de graf mínim* (N_0, C^T) associat a T de la següent manera. El cost de connectar dos vèrtexs, $i, j \in N$, es defineix com $c^T_{ij} = \max_{(k,l) \in g_{ij}} \{c_{kl}\}$, on g_{ij} és l'únic camí en T des de i fins a j. Si no hi ha confusió, escrivim C^* en lloc de C^T.

Tot i que aquesta definició depèn de l'elecció de l'arbre generador de cost mínim T, és independent del T escollit.

Definició 4.6. Es defineix la *forma irreductible* d'un problema *mcst* (N_0, C) com el problema de graf mínim (N_0, C^*) associat a un arbre generador de cost mínim T. Si (N_0, C^*) és una forma irreductible, aleshores direm que C^* és la *matriu irreductible.*

Observem de la definició que la forma irreductible associada a un arbre generador de cost mínim només depèn d'aquest arbre generador. Per tant, si tenim dos problemes *mcst* (N_0, C) i (N_0, C') que són arbre-equivalents, és a dir, que existeix un arbre tal que aquest és arbre generador de cost mínim d'ambdós problemes i a més, $c_{ij} = c'_{ij}$ per totes les arestes de l'arbre, aleshores $C^* = C'^*$.

Així doncs, (N_0, C^*) és una forma irreductible si i només si, reduint el cost dels arcs, el cost de connectar els clients a la font també es redueix. Aleshores tenim els següents resultats:

Lema 4.7. *Sigui (N_0, C) un problema d'arbre generador de cost mínim i (N_0, C^*) la forma irreductible associada. Llavors:*

a) Per tot $i, j \in N_0$ existeix un arbre mínim T en (N_0, C^*) tal que $(i, j) \in T$.

b) $C^* \leq C$.

Un problema d'arbre generador de cost mínim és irreductible si i només si podem trobar un arbre tal que la connexió directa entre dos vèrtexs qualssevol sigui el cost màxim de les arestes que els connecten en aquest arbre. Aquesta és la propera proposició.

Introduïm la següent notació, que és necessària dins del que segueix. Donada una permutació dels clients $\pi \in \Pi_N$, denotarem al client $i \in N$ que ocupa la posició s, és a dir $\pi(i) = s$ com π_s. Els clients anomenats d'aquesta manera resulten consecutius.

Proposició 4.8. *El problema* mcst (N_0, C^*) *és irreductible si i només si existeix un arbre T en (N_0, C^*) que satisfà les següents condicions:*

(A) $T = \{(\pi_{s-1}, \pi_s)\}_{s=1}^n$ per una permutació $\pi \in \Pi_N$, i on $\pi_0 = 0$.

(B) Donats $\pi_p, \pi_q \in N_0$ amb $p < q$, llavors $c^*_{\pi_p \pi_q} = \max_{s \mid p<s\leq q}\{c^*_{\pi_{s-1}\pi_s}\}$.

A més, T és un arbre generador de cost mínim associat a (N_0, C^*).

Demostració. Veure Bergantiños i Vidal-Puga (2007) [4, Proposició 3.1]. □

La proposició anterior indica que en un problema irreductible els clients es poden ordenar de forma que existeix un arbre generador de cost mínim on els clients estan alineats amb la font. Aquest arbre, a més, compleix que el cost de les arestes directes és el cost més gran de les arestes de l'arbre que els uneix.

El següent exemple mostra la construcció de la forma irreductible.

Exemple 4.9. Tornem a considerar el joc d'arbre generador de cost mínim de l'Exemple 4.2 que correspon a la Figura 4.1 anterior. En primer lloc ressaltem en vermell l'arbre generador de cost mínim que s'havia trobat anteriorment, que en aquest cas resultava un únic arbre.

D'altra banda, el graf de la dreta mostra la forma irreductible associada a aquest problema així com els costos reduïts en les arestes produïdes per la definició de forma irreductible definida anteriorment. En aquest cas, podem veure que no hi ha un únic arbre generador de cost mínim. Està assenyalat un dels possibles arbres.

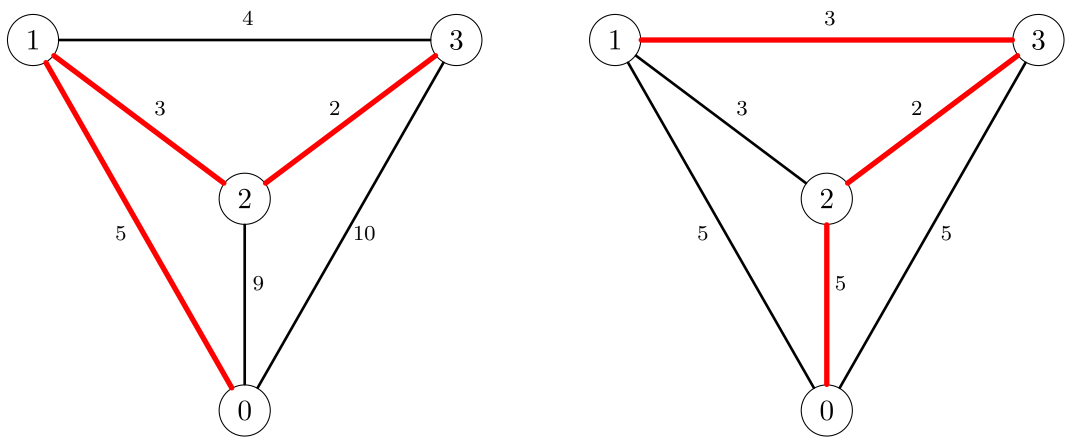


Figura 4.4: Problema *mcst* (N_0, C) i (N_0, C^*), respectivament.

Ara, donat un problema (N_0, C) introduirem un procediment per associar una matriu C^* a cada matriu C i un arbre generador de cost mínim que satisfaci les propietats *(A)* i *(B)*.

Donat T' un arbre generador de cost mínim en (N_0, C), la permutació $\pi \in \Pi_N$ indica els clients en l'ordre que els ha triat l'algoritme. Denotem $\pi_0 = 0$.

Els nodes es connecten a la font via T' mitjançant l'algoritme de Prim. En el pas p, l'aresta seleccionada és (π_p^0, π_p), on escrivim π_p^0 per indicar el predecessor de π_p en l'arbre. Això significa que si $\pi_p = \pi_q^0$, llavors $p < q$ per tot $\pi_p, \pi_q \in N_0$. A més, per cada $\pi_s \in N$,

$$c_{\pi_s^0 \pi_s} = \min_{(p,q) \mid p<s\leq q} c_{\pi_p \pi_q}.$$

Aleshores, veiem que si $\pi_p = \pi_q^0$,

$$c_{\pi_s^0 \pi_s} \leq c_{\pi_q^0 \pi_q} \qquad \text{per tot } s \text{ tal que } p < s \leq q.$$

Per tant podem definir C' de la manera següent. Per tot $\pi_p, \pi_q \in N_0$ amb $p < q$,

$$c'_{\pi_p \pi_q} = \max_{s|p<s\leq q} c_{\pi_s^0 \pi_s}. \tag{4.1}$$

Llavors per tot $\pi_s \in N$,

$$c'_{\pi_{s-1} \pi_s} = c_{\pi_s^0 \pi_s} \quad \text{i} \quad c'_{\pi_s^0 \pi_s} = c_{\pi_s^0 \pi_s}. \tag{4.2}$$

Proposició 4.10. *Donat un problema msct (N_0, C), la matriu C' obtinguda anteriorment és la matriu irreductible associada a C, és a dir, $C' = C^*$. I més, T l'arbre generador de (N_0, C), també és un arbre generador a (N_0, C^*).*

Demostració. Primerament demostrarem que T satisfà les condicions *(A)* i *(B)*.

(A) Trivial per la construcció del arbre.

(B) Donat $\pi_p, \pi_q \in N_0$ amb $p < q$,

$$c'_{\pi_p \pi_q} = \max_{s|p<s\leq q} \{c_{\pi_s^0 \pi_s}\} = \max_{s|p<s\leq q} \{c'_{\pi_{s-1} \pi_s}\}.$$

En la primera igualtat s'ha fet servir (4.1) i en la segona (4.2). Per tant, s'ha vist que T compleix les dues propietats. Aleshores per la Proposició 4.8, C' és una matriu irreductible i T és l'arbre generador de cost mínim en C'.

Ara per (4.2),

$$c(N_0, C', T') = \sum_{s=1}^{n} c'_{\pi_s^0 \pi_s} = \sum_{s=1}^{n} c'_{\pi_{s-1} \pi_s} = c(N_0, C', T).$$

Per tant, T' és un arbre generador de cost mínim en C' que satisfà $c'_{i^0 i} = c_{i^0 i}$ per tot $i \in N$. Llavors, com C i C' son arbre- equivalents, s'obté $C^* = (C')^* = C'$. □

Proposició 4.11. *Sigui (N_0, C^*) una forma irreductible i sigui $\pi \in \Pi_N$ tal que $T = \{(\pi_{s-1}, \pi_s)\}_{s=1}^n$ amb $\pi_0 = 0$ és l'arbre generador de cost mínim en (N_0, C^*) satisfent* (A) *i* (B). *Per $S \subset N$, podem suposar que $S = \{(\pi_{s(1)}, \ldots, \pi_{s(|S|)}\}$ amb $s(q-1) < s(q)$ per tot $q \in \{1, \ldots, |S|\}$ i $s(0) = 0$. Aleshores:*

(a) $T' = \{(\pi_{s(q-1)}, \pi_{s(q)})\}_{q=1}^{|S|}$ *és un arbre generador de cost mínim de (S_0, C^*) i* $c^*(S) = \sum_{q=1}^{|S|} c^*_{\pi_{s(q-1)} \pi_{s(q)}}$,

(b) $c^*(S) - c^*\left(S \setminus \{\pi_{s(p)}\}\right) = \min\left\{c^*_{\pi_{s(p-1)} \pi_{s(p)}}, c^*_{\pi_{s(p)} \pi_{s(p+1)}}\right\}$ *si* $p < |S|$ *i* $c^*(S) - c^*\left(S \setminus \{\pi_{s(|S|)}\}\right) = c^*_{\pi_{s(|S|-1)} \pi_{s(|S|)}}$,

(c) (N, c^*) *és un joc còncau.*

Demostració. Veure a Bergantiños i Vidal-Puga (2007) [4, Proposició 3.3] □

Exemple 4.12 (Continuació de l'Exemple 4.9)**.** Seguint amb l'Exemple 4.9, calculem de nou la regla de Bird, la regla de Kar i la regla de Dutta-Kar amb la matriu irreductible de la forma irreductible associada al problema. Primerament veiem que

$$C^* = \begin{pmatrix} 0 & 5 & 5 & 5 \\ 5 & 0 & 3 & 3 \\ 5 & 3 & 0 & 2 \\ 5 & 3 & 2 & 0 \end{pmatrix}$$

Amb funció característica:

$$\begin{aligned} &c^*(\{1\}) = c^*(\{2\}) = c^*(\{3\}) = 5 \\ &c^*(\{1,2\}) = c^*(\{1,3\}) = 8,\ c^*(\{2,3\}) = 7, \\ &c^*(\{1,2,3\}) = 10. \end{aligned}$$

En aquest cas, no hi ha un únic arbre generador de cost mínim, per tant, per calcular $B(N_0, C^*)$ hem de calcular $B^\pi(N_0, C^*)$ per cada $\pi \in \Pi_N$ mitjançant l'algoritme de Prim. D'aquesta manera arribem als següents resultats:

π		$B^\pi(N_0, C^*)$
$(1,2,3)$	$\Longrightarrow$	$(5,3,2)$
$(1,3,2)$	$\Longrightarrow$	$(5,2,3)$
$(2,1,3)$	$\Longrightarrow$	$(3,5,2)$
$(2,3,1)$	$\Longrightarrow$	$(3,5,2)$
$(3,1,2)$	$\Longrightarrow$	$(3,2,5)$
$(3,2,1)$	$\Longrightarrow$	$(3,2,5)$

Obtenim doncs,

$$B(N_0, C^*) = \frac{1}{6} \cdot \sum_{\pi \in \Pi_N} B^\pi(N_0, C^*) = \left(\frac{13}{3}, \frac{17}{6}, \frac{17}{6} \right).$$

Procedint de manera similar, calculem la regla de Dutta-Kar pel problema d'aquest Exemple 4.9. Per calcular $DK(N_0, C^*)$ també hem de calcular aquest vector per cada permutació $\pi \in \Pi_N$ mitjançant l'algoritme de Prim. En aquest cas s'obté:

π		$DK^\pi(N_0, C^*)$
$(1,2,3)$	$\Longrightarrow$	$(3,2,5)$
$(1,3,2)$	$\Longrightarrow$	$(3,5,2)$
$(2,1,3)$	$\Longrightarrow$	$(5,2,3)$
$(2,3,1)$	$\Longrightarrow$	$(5,2,3)$
$(3,1,2)$	$\Longrightarrow$	$(5,3,2)$
$(3,2,1)$	$\Longrightarrow$	$(5,3,2)$

Aplicant la fórmula obtenim,

$$DK(N_0, C^*) = \frac{1}{6} \cdot \sum_{\pi \in \Pi_N} DK^\pi(N_0, C^*) = \left(\frac{11}{3}, \frac{19}{6}, \frac{19}{6} \right).$$

Per últim determinem quina seria l'assignació de costos segons la regla de Kar.

$$\begin{aligned}
Sh_1(N,c^*) = & \frac{1}{3}c^*(\{1\}) + \frac{1}{6}(c^*(\{1,2\}) - c^*(\{2\}) + \frac{1}{6}(c^*(\{1,3\}) - c^*(\{3\}) + \\
& \frac{1}{3}\left(c^*(\{1,2,3\}) - c^*(\{2,3\})\right) = \frac{1}{3}\cdot 5 + \frac{1}{6}\cdot 3 + \frac{1}{6}\cdot 3 + \frac{1}{3}\cdot 3 = \frac{11}{3}. \\
Sh_2(N,c^*) = & \frac{1}{3}c^*(\{2\}) + \frac{1}{6}(c^*(\{1,2\}) - c^*(\{1\}) + \frac{1}{6}(c^*(\{2,3\}) - c^*(\{3\}) + \\
& \frac{1}{3}\left(c^*(\{1,2,3\}) - c^*(\{1,3\})\right) = \frac{1}{3}\cdot 5 + \frac{1}{6}\cdot 3 + \frac{1}{6}\cdot 2 + \frac{1}{3}\cdot 2 = \frac{19}{6}. \\
Sh_3(N,c^*) = & \frac{1}{3}c^*(\{3\}) + \frac{1}{6}(c^*(\{1,3\}) - c^*(\{1\}) + \frac{1}{6}(c^*(\{2,3\}) - c^*(\{2\}) + \\
& \frac{1}{3}\left(c^*(\{1,2,3\}) - c^*(\{1,2\})\right) = \frac{1}{3}\cdot 5 + \frac{1}{6}\cdot 3 + \frac{1}{6}\cdot 2 + \frac{1}{3}\cdot 2 = \frac{19}{6}.
\end{aligned}$$

Obtenim doncs,

$$K(N_0, C^*) = Sh(N, c^*) = \left(\frac{11}{3}, \frac{19}{6}, \frac{19}{6}\right).$$

Aleshores veiem que en aquest exemple $B(N_0, C^*) = K(N_0, C^*)$, és a dir, que ambdues regles de repartició coincideixen en la forma irreductible. Això es manté en general.

La propera proposició estableix la relació d'igualtat que existeix entre les regles de Bird i Kar a la forma irreductible, com s'ha vist a l'Exemple 4.12.

Proposició 4.13. *Sigui* (N_0, C^*) *una forma irreductible, aleshores* $K(N_0, C^*) = B(N_O, C^*)$.

Demostració. Es pot trobar a Bergantiños et al.(2007) [4, Proposició 3.4]. La demostració es fa utilitzant un argument d'inducció sobre el nombre de clients. Si $n = 1$ és clar que les dues regles coincideixen. Se suposa cert per $n-1$ clients i aleshores es prova per n clients. Es consideren dos casos. Si l'aresta $(0, \pi_1)$ no és la més cara de T, es divideix el problema *mcst* en dos problemes amb menys de n clients cadascun. Sota les hipòtesis de separabilitat i inducció es veu la igualtat. El segon cas, si l'aresta $(0, \pi_1)$ és la més cara de T, la prova es fa a través dels ordres. □

Es pot definir doncs, una nova regla φ per qualsevol problema generador de cost mínim. La manera de definir aquesta regla consisteix a aplicar la regla de Kar (o de Bird) a la forma irreductible. Ja sabem que la regla de Kar i de Bird coincideixen en la forma irreductible (Proposició 4.13). Ara podem definir aquesta nova regla.

Definició 4.14. Donat un problema generador de cost mínim (N_0, C) definim la *regla* φ com

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

amb C^* la matriu irreductible associada a C.

Bergantiños i Vidal-Puga (2005)[3] demostren que aquesta regla φ coincideix amb la regla ERO (*Equal Remaining Obligation*). En l'exemple estudiat al llarg del capítol veiem que clarament això es compleix.

$$\varphi(N_0, C) = K(N_0, C^*) = B(N_0, C^*) = ERO(N_0, C) = \left(\frac{11}{3}, \frac{19}{6}, \frac{19}{6}\right).$$

4.3 Propietats de les regles

Ara introduïm diverses propietats de les regles. Aquestes es poden trobar a Bergantiños i Vidal-Puga (2007) [4].

Sigui un problema *mcst* (N_0, C) i $S \subset N$, escriurem amb un cert abús de notació el problema (S_0, C) per indicar el problema que només involucra els elements de $S \cup \{0\}$ i de la restricció de la matriu C a aquests agents. Així mateix, donat el conjunt N d'agents, denotarem $\mathcal{C}^N$ com el conjunt de totes les matrius de costos sobre N_0. Per últim denotarem per $c_{msct}(N_0, C)$ com el cost associat a qualsevol arbre generador de cost mínim per (N_0, C).

Donada una regla ψ, i un problema *mcst* (N_0, C) podem considerar les següents propietats:

- **Selecció del nucli (*Core selection*) (CS):**
  Per tot problema *mcst* (N_0, C) i tot $S \subset N$,
  $$\sum_{i \in S} \psi_i(N_0, C) \leq c_{msct}(S_0, C),$$
  és a dir, que el vector $\psi(N_0, C)$ pertany al nucli del joc associat (N, c). Aquesta propietat implica que cap grup d'agents estaria millor construint la seva pròpia xarxa en lloc de pagar el cost que la regla li assigna a cadascun d'ells.

- **Monotonia del cost (*Cost monotonicity*) (CM):**
  Per tot parell de problemes *mcst* (N_0, C) i (N_0, C') i $i \in N$ tal que $c_{ij} < c'_{ij}$ per algun $j \in N_0$ i per la resta de subíndexs es verifica $c_{kl} = c'_{kl}$, aleshores
  $$\psi_i(N_0, C) \leq \psi_i(N_0, C').$$
  Aquesta propietat implica que si el cost del jugador i d'una determinada connexió augmenta, però la resta de costos de connexió resten iguals, llavors l'agent i no pot estar millor.

- **Solidaritat (*Solidarity*)(SOL):**
  Per tots els problemes (N_0, C) i (N_0, C') tal que $C \leq C'$, tenim que
  $$\psi(N_0, C) \leq \psi(N_0, C').$$
  Aquesta propietat implica que si augmenta el cost d'algunes connexions i la resta (si n'hi ha) es queden iguals, cap agent trobaria una situació millor. Veiem que SOL és més forta que CM.

- **Monotonia de la població (*Population monotonicity*) (PM):**
  Per tot problema *mcst* (N_0, C), $S \subset N$, i $i \in S$, aleshores
  $$\psi_i(N_0, C) \leq \psi_i(S_0, C).$$
  Aquesta propietat implica que si un nou client (o més) s'uneix a la coalició, cap jugador que formava inicialment la coalició surt perjudicat.

- **Simetria (*Symmetry*) (SYM):**

  Diem que $i, j \in N$ són simètrics si per tot $k \in N_0 \setminus \{i,j\}, c_{ik} = c_{jk}$. Per tant, per tot problema *mcst* (N_0, C) i tot parell de clients simetrics $i, j \in N$,

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

  Així, si existeix un parell de jugadors simètrics, la regla assignarà el mateix cost a ambdós.

- **Continuïtat (*Continuity*) (CON):**

  Per tot N, $\psi(N_0, \cdot)$ és una funció contínua de $\mathcal{C}^N$ en $\mathbb{R}^N$.

  Aquesta propietat implica que els petits canvis en els costos de connexió no poden donar lloc a grans canvis en la quantitat que han de pagar.

- **Positivitat (*Positivity*) (POS):**

  Per tot problema *mcst* (N_0, C) i tot $i \in N$, tenim

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

  És a dir, cap client rep quantitats de la resta de clients.

- **Separabilitat (*Separability*) (SEP):**

  Per tot problema *mcst* (N_0, C) i $S \subset N$ tal que $c_{mcst}(N_0, C) = c_{mcst}(S_0, C) + c_{mcst}((N \setminus S)_0, C)$ tenim

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

  Dos subconjunts de clients, S i $N \setminus S$ poden estar connectats a la font cadascun per separat o conjuntament. Si no hi ha estalvi quan estan connectats conjuntament, aquesta propietat implica que els clients pagaran el mateix en ambdós casos. Aquesta propietat apareix en alguns textos com *descomposició.*

- **Independència d'altres costos (*Independence of other costs*) (IOC):**

  Per tot problema *mcst* (N_0, C) i (N_0, C'), i per tot $i \in N$ tal que $c_{ij} = c'_{ij}$ per tot $j \in N_0 \setminus \{i\}$, tenim

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

  Aquesta propietat implica que la quantitat assignada al client i depèn únicament del cost de les arestes que connecten i amb la resta de clients i la font.

- **Repartició equitativa dels costos extra (*Equal share of extra costs*) (ESEC):**

  Siguin (N_0, C) i (N_0, C') dos problemes *mcst* i $c_0, c'_0 \geq 0$. Suposem $c_{0i} = c_0$ i $c'_{0i} = c'_0$ per tot $i \in N$, $c_0 < c'_0$, i $c_{ij} = c'_{ij} \leq c_0$ per tot $i, j \in N$, tenim

  $$\psi_i(N_0, C') = \psi_i(N_0, C) + \frac{c'_0 - c_0}{n} \qquad \text{per tot } i \in N.$$

Aquesta propietat implica que si un grup de clients N s'enfronten a un problema (N_0, C) en el qual tots ells tenen el mateix cost de connexió a la font i en el qual aquest cost és major que el cost entre clients. Aleshores una xarxa òptima implica que un client qualsevol es connecta directament a la font i la resta es connecten a la font a través d'aquest client. A més, estan tots d'acord que el repartiment correcte és $\psi_i(N_0, C)$. Suposem que la connexió amb la font té un increment fins c_0', aleshores aquest cost extra la regla diu que serà repartit entre tots els agents en parts iguals.

- **Independència d'arbres irrellevants (*Independence of irrelevants trees*) (IIT):**

  Si dos problemes *mcst* (N_0, C) i (N_0, C') són arbre-equivalents,

  $$\psi_i(N_0, C) = \psi_i(N_0, C') \qquad \text{per tot } i \in N.$$

  Dos problemes *mcst* (N_0, C) i (N_0, C') són arbre-equivalents si existeix un arbre T tal que T és un arbre generador de cost mínim dels dos problemes i a més, $c_{ij} = c'_{ij}$ per totes les arestes $(i, j) \in T$. Per tant, aquesta propietat implica que si dos problemes tenen el mateix arbre generador de cost mínim i el cost de les arestes de l'arbre són iguals, la regla proporcionarà la mateixa assignació de costos, independent del cost de les arestes que no formen part d'aquest arbre.

A partir d'ara ens referirem a les propietats per les seves abreviatures. En el pròxim teorema es veurà quines de totes aquestes propietats són satisfetes per cadascuna de les diferents regles estudiades durant tota la memòria. Però abans veurem unes implicacions que es compleixen entre les propietats i que seran útils en algunes demostracions posteriors. Aquestes es poden trobar, juntament amb el teorema a Bergantiños (2008) [5].

Proposició 4.15. *Els enunciats següents es compleixen per tot problema mcst.*

1. *Si una regla de repartició verifica SOL (Solidarity) implica que també satifà CM (Cost monotonicity)i IIT (Independence of irrelevant trees).*
2. *Si una regla de repartició verifica PM (Population monotonicity) implica que també satisfà CS (Core selection) i SEP (Separability).*
3. *Cap regla compleix IOC (Independence of other costs).*

Demostració. Les dues primeres es poden trobar a Bergantiños (2007) [4]. I a continuació demostrarem la tercera, és a dir, que cap regla compleix la propietat d'independència d'altres costos (IOC).

Sigui ψ una regla de repartició i suposem que satisfà IOC. Donat $N = \{1, 2\}$, $x > 0$ i $y > 0$ considerem els problemes representats a la Figura 4.5 següent.

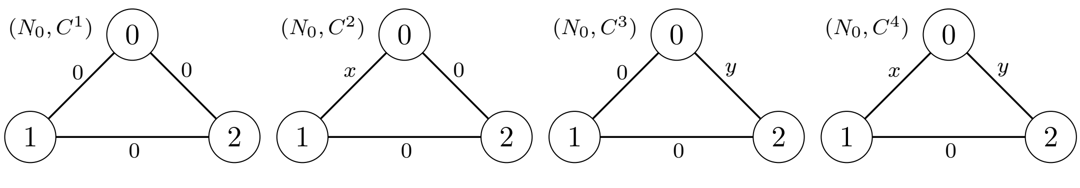


Figura 4.5

Sabem que $\psi(N_0, C^1) = (a, -a)$ per alguna $a \in \mathbb{R}$. Com ψ satisfà IOC, tenim que $\psi(N_0, C^2) = \psi(N_0, C^1)$, $\psi(N_0, C^3) = \psi(N_0, C^1)$, $\psi_1(N_0, C^4) = \psi_1(N_0, C^2) = a$ i $\psi_2(N_0, C^4) = \psi_2(N_0, C^3) = -a$. Aleshores, $\psi(N_0, C^4) = (a, -a)$, que és una contradicció, ja que $c_{mcst}(N_0, C^4) = \min\{x, y\} > 0$.

Per tant, hem vist que cap regla de repartició verifica la propietat IOC. □

Un cop sabem que es compleixen aquestes implicacions, ja podem passar a enunciar i demostrar quines propietats verifiquen cadascuna de les regles de reparticions.

Teorema 4.16. *Els enunciats següents es compleixen per tot problema mcst.*

(i) La regla de Bird satisfà les propietats de CS, POS, SYM i ESEC. Però no satisfà les de CM, PM, CON, SEP, IOC, SOL i IIT.

(ii) La regla de Dutta-Kar satisfà les propietats de CS, CM, POS i SYM. Però no satisfà PM, CON, SEP, IOC, SOL, ESEC i IIT.

(iii) La regla de Kar satisfà les propietats de CM, CON, SYM i ESEC. Però no satisfà les de CS, PM, POS, SEP, IOC, SOL i IIT.

(iv) La regla φ satisfà totes les propietats excepte IOC.

Demostració. (i):

- B satisfà CS: Vist en el Teorema 4.4.
- B satisfà POS: És trivial per la definició de la regla de Bird.
- B satisfà SYM: Siguin i, j dos clients simètrics en (N_0, C) i per tota $\pi \in \Pi_N$ definim $\pi^{ij} \in \Pi_N$ tal que $\pi^{ij}(i) = \pi(j)$, $\pi^{ij}(j) = \pi(i)$, i $\pi^{ij}(k) = \pi(k)$ per tot $k \in N \setminus \{i, j\}$, on només hem canviat de lloc els clients i i j.

  És fàcil veure que $B_i^\pi(N_0, C) = B_j^{\pi^{ij}}(N_0, C)$. Aleshores,

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

  Queda demostrat doncs, que la regla de Bird compleix la propietat de simetria.

- B satisfà ESEC: Siguin (N_0, C) i (N_0, C') dos problemes *mcst* definits com a la definició de ESEC. Aleshores per tot $\pi \in \Pi_N$,

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

  Llavors,

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

  verificant així que la regla de Bird satisfà ESEC.

– B no satisfà CON ni SEP: Es poden trobar contraexemples en Bergantiños et al. (2008) [5].

– B no satisfà CM: Veure Dutta-Kar (2004) [10, Exemple 2].

– B no satisfà SOL: Per la Proposició 4.14 (1), com la regla de Bird no satisfà CM pel contrarecíproc, B no satisfà SOL.

– B no satisfà PM: De manera anàloga a l'anterior i per la Proposició 4.14 (2), com la regla de Bird no verifica SEP, aleshores tampoc verifica PM.

– B no satifà IIT: Veure Bergantiños et al. (2008) [5].

– B no satisfà IOC: Vist a la Proposició 4.14 (3).

(ii):

– DK satisfà CS i CM: Es pot trobar a Dutta-Kar (2004) [10, Teorema 1].

– DK satisfà POS: És trivial per la definició de la regla i que els costos de connexió són positius.

– DK satisfà SYM: Usant els mateixos arguments que en el cas de la regla de Bird, es pot demostrar que DK també satisfà SYM.

– DK no satisfà IIT: Veure Bergantiños et al. (2008) [5].

– DK no satisfà SOL: Per la Proposició 4.14 (1), com la regla de Dutta-Kar no satisfà IIT pel contrarecíproc, DK no satisfà SOL.

– DK no satisfà ESEC, SEP ni CON: Es poden trobar contraexemples a Bergantiños et al. (2008) [5].

– DK no satisfà PM: Per la Proposició 4.14 (2), com la regla de Dutta-Kar no satisfà SEP pel contrarecíproc, DK no satisfà PM.

– DK no satisfà IOC: Vist a la Proposició 4.14 (3).

(iii):

– K satisfà CM: Es pot trobar a Dutta-Kar (2004) [10].

– K satisfà CON: Sabem que $K(N_0, C) = Sh(N, c)$ i com c és una funció continua en C, aleshores es verifica que la regla compleix aquesta propietat.

– K satisfà SYM: És fàcil veure que si dos agents són simètrics en (N_0, C), també ho són en (N, c). Com $K(N_0, C) = Sh(N, c)$ i el valor de Shapley és simètric (Veure [18]), aleshores queda demostrat el resultat.

– K satisfà ESEC: Siguin dos problemes *mcst* (N_0, C) i (N_0, C') tals com están definits a la definició de la propietat ESEC. És fàcil veure que $c'(S) = c'_{mcst}(S_0, C') = c_{mcst}(S_0, C) + (c'_0 - c_0) = c(S) + (c'_0 - c_0)$ per tot $S \subset N$. Aleshores,

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

Queda demostrat doncs, que la regla de Kar verifica ESEC.

– K no satisfà CS: Vist en l'Exemple 4.5.

– K no satisfà POS ni SEP: Es troben contraexemples en Bergantiños et al. (2008) [5].

– K no satisfà PM: Per la Proposició 4.14 (2), com la regla de Kar s'ha vist que no compleix CN, aleshores pel contrarecíproc tampoc verifica PM.

– K no satisfà IOC: Vist a la Proposició 4.14 (3).

– K no satisfà IIT: Veure Bergantiños et al. (2008) [5].

– K no satisfà SOL: Per la Proposició 4.14 (1), com la regla de Kar s'ha vist que no verifica IIT, aleshores pel contrarecíproc tampoc verifica PM.

(iv):

– φ satisfà SOL: Veure Bergantiños (2007) [4, Teorema 4.1].

– φ satisfà CM: Per la Proposició 4.14 (1), com SOL implica CM, aleshores φ satifà CM.

– φ satisfà IIT: Per la Proposició 4.14 (1), com SOL implica IIT, aleshores φ satisfà IIT.

– φ satisfà PM: Veure Bergantiños (2007) [4, Teorema 4.1].

– φ satisfà SEP: Per la Proposició 4.14 (2), com PM implica SEP, aleshores φ satisfà SEP.

– φ satisfà CS: Per la Proposició 4.14 (2), com PM implica CS, aleshores φ satisfà CS.

– φ satisfà CON: Es pot veure que $\varphi(N_0, C) = (f \circ g \circ h)(C)$ per tota $C \in \mathcal{C}^N$, on $h(C) = C^*$ i $g(C) = c$ per tota $C \in \mathcal{C}^N$, i $f(v) = Sh(N, v)$ per tot v. Aleshores, com f, g i h són funcions contínues, φ també ho és.

– φ satisfà POS: Sigui $S \subset N$, i per la Proposició 4.11 (b), $c^*(S) - c^*\Big(S \setminus \{\pi_{s(p)}\}\Big) \geq 0$ per tot $i \in N$. Aleshores, $\varphi_i(N_0, C) = K_i(N_0, C^*) = Sh_i(N, c^*) \geq 0$.

– φ satisfà SYM: Veure Bergantiños i Vidal-Puga (2004) [2].

– φ satisfà ESEC: Veure Bergantiños i Vidal-Puga (2007) [4].

– φ no satisfà IOC: Vist a la Proposició 4.14 (3).

$\square$

A la següent Taula 4.3 es resumeixen les propietats que satisfan les regles mencionades anteriorment.

	Bird	Dutta-Kar	Kar	φ
CS	✓	✓	✗	✓
CM	✗	✓	✓	✓
PM	✗	✗	✗	✓
CON	✗	✗	✓	✓
POS	✓	✓	✗	✓
SEP	✗	✗	✗	✓
SYM	✓	✓	✓	✓
IOC	✗	✗	✗	✗
SOL	✗	✗	✗	✓
ESEC	✓	✗	✓	✓
IIT	✗	✗	✗	✓

Taula 4.3: Propietats que compleix o no cada regla.

Aquesta taula mostra clarament que φ, és a dir la regla ERO, satisfà moltes més propietats i en conseqüència es pot considerar una regla més "justa"que les altres. Més concretament, compleix totes les propietats menys l'*Independència d'altres costos*, que com s'ha demostrat abans no la compleix cap regla.

Bergantiños i Vidal Puga (2008) [5] van introduir dues versions més dèbils d'aquesta propietat. L'*Independència de baixos costos* (ISL) la qual diu que la quantitat pagada per cada client no depèn del cost de les arestes que tenen un cost més baix que ell. I l'*Independència de grans costos* (ILC) que diu que la quantitat pagada per cada client no depèn del cost dels arcs que tenen un cost major que ell. En aquests casos la regla ERO sí que compleix aquestes dues propietats.

Bibliografia

[1] Bergantiños, G.; Lorenzo-Freire S. (2004). A non-cooperative approach to the cost spanning tree problem. Mathematical Methods of Operations Research. 59, 393–403.

[2] Bergantiños, G.; Vidal-Puga, J.J. (2004). Additivity in cost spanning tree problems. Journal of Mathematical Economics. 45(1–2), 38–42.

[3] Bergantiños, G.; Vidal-Puga, J.J. (2005). Several approaches to the same rule in minimum cost spanning trees problems. Mimeo. Universidad de Vigo.

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

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

[6] Binmore, K. (2011). La teoría de juegos. Una breve introducción. Alianza Editorial.

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

[8] Cesco, J.C.; Risma, L.B.P.; Quintas, L. (2013). Reglas proporcionales en problemas de árboles de mínimo costo. Perspectivas. Revista de Análisis de Economía, Comercio y Negocios Internacionales. 7 (2), pp. 75 – 100.

[9] Chartrand, G.; Zhang, P. (2012). A first course in graph theory. Dover Publications.

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

[11] Gibbons, R. (1999). Un primer curso de teoría de juegos. Antoni Bosch Editor.

[12] Gross, J. L.; Yellen, J.; Zhang, P. (2013). Handbook of Graph Theory, Second Edition. CRC Press, pp. 2–21.

[13] Kar, A (2002). Axiomatization of the Shapley value on minimum cost spanning tree games. Games and Economic Behavior. 38(2), 265–277.

[14] Kruskal, J.B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society. 7(1), 48—50.

[15] Magaña, A. (1996). Formación de coaliciones en los juegos cooperativos y juegos con múltiples alternativas. Universitat Politècnica de Catalunya. Capítol 1. pp. 15–26.

[16] Prim, R.C. (1957). Shortest connection networks and some generalizations. Bell System Technical Journal. 36(6), 1389–1401.

[17] Rafels, C.; Izquierdo, J.M.; Marín-Solano, J.; Martínez de Albéniz, F.J.; Núñez, M.; Ybern, N. (1999). Jocs cooperatius i aplicacions econòmiques. Edicions de la Universitat de Barcelona.

[18] Shapley, L. S. (1953). A value for n-person games, in: Kuhn, H. and Tucker, A., Eds., Contributions to the Theory of Games II. Princeton University Press, Princeton, pp. 307–317.

[19] Sharkey, W. W. (1995). Networks Models in Economics. Handbook in Operation Research and Managment Science. Chapter 9, (8), pp. 713–765.

[20] Tijs, S.; Feltkamp, V.; Muto, S. (1994). On the irreducible core and the equal remaining obligations rule of minimum cost spanning extension problems. Mimeo. Tilburg University.

[21] Von Neumann, J.; Morgenstern, O. (1944). Theory of games and economic behavior. Princeton University Press.

[22] Wilson, R. J. (1996). Introduction to Graph Theory, Fourth Edition. Prentice Hall.