ANÁLISIS COOPERATIVO DE CADENAS DE DISTRIBUCIÓN

Memoria de Tesis Doctoral realizada por
FEDERICO PEREA ROJAS-MARCOS
Directores de la Tesis:
Dr. Francisco Ramón Fernández García
Dr. Justo Puerto Albandoz

Universidad de Sevilla
Facultad de Matemáticas y Estadística
Departamento de Estadística e Investigación Operativa

FRANCISCO RAMÓN FERNÁNDEZ GARCÍA, CATEDRÁTICO DEL DEPARTAMENTO DE ESTADÍSTICA e INVESTIGACIÓN OPERATIVA, de la FACULTAD DE MATEMÁTICAS, de la UNIVERSIDAD de SEVILLA

JUSTO PUERTO ALBANDOZ, CATEDRÁTICO DEL DEPARTAMENTO DE ESTADÍSTICA e INVESTIGACIÓN OPERATIVA, de la FACULTAD DE MATEMÁTICAS, de la UNIVERSIDAD de SEVILLA

CERTIFICAMOS

Que la presente memoria titulada:

ANÁLISIS COOPERATIVO DE CADENAS DE DISTRIBUCIÓN

ha sido realizada bajo nuestra dirección por el Licenciado en Matemáticas

D. Federico Perea Rojas-Marcos,

y constituye su tesis para optar al grado de Doctor en Matemáticas.

Y para que conste, en cumplimiento de la legalidad vigente, y a los efectos que haya lugar, firmamos el presente en Sevilla, Febrero de 2007.

Fdo: Francisco Ramón Fernández García.

Fdo: Justo Puerto Albandoz.

Agradecimientos

Esta tesis no habría sido posible sin la ayuda de muchas personas, a todas y cada una de ellas va mi agradecimiento.

Mis estudios de doctorado comenzaron en 2003, gracias al proyecto europeo *MaMaEuSch*, el cual me dio la oportunidad de entrar en el fascinante mundo de la investigación. También contribuyeron al desarrollo de mi tesis la estancia que disfrute en Hengelo (Holanda) trabajando para *Thales Nederland* y actualmente el proyecto europeo *Arrival*.

Por supuesto que mi agradecimiento también va dirigido a todos esos compañeros de despacho, con los que compartí largas horas de trabajo. Y no solo por ser compañeros, sino también por ser mis amigos. Nunca olvidaré todos aquellos cafés, reuniones, charlas,... que compartimos durante el tiempo que trabajamos juntos en el departamento de Estadística e Investigación Operativa de la Universidad de Sevilla, en el departamento de Innovación en Thales Nederland y en el departamento de Matemática Aplicada II de la Universidad de Sevilla. Hicisteis mi trabajo agradable y entretenido.

Durante casi todos estos cuatro años hubo una persona especial junto a mí. Gracias a ella por apoyarme siempre.

Agradecimientos especiales para mis directores de tesis. No sólo porque me dieron los conocimientos y el ánimo necesarios para terminar este trabajo, sino porque se convirtieron en una segunda familia.

Mis amigos, sin ellos nada tiene sentido.

Mi familia.

A mi padre, mi madre,

mi hermano y mi hermana.

Índice general

Resumen en castellano

Comenzamos esta memoria de tesis doctoral con un resumen en castellano de sus principales resultados. Dichos resultados están estudiados en profundidad en la segunda parte de la tesis (a partir de la página 31), en la que se incluyen demostraciones y ejemplos para un mejor entendimiento de la materia analizada, y que está íntegramente redactada en inglés.

Introducción

En esta tesis estudiamos problemas de optimización desde el punto de vista de la teoría de juegos cooperativos. En problemas de optimización el objetivo es determinar un plan óptimo para maximizar (o minimizar) ciertos beneficios (o costes). Si el plan óptimo depende sólo de la decisión de un único agente, nos sería suficiente aplicar técnicas de optimización clásica. Pero no es extraño encontrar problemas de optimización en los que un grupo de agentes (personas, empresas, partidos políticos,...) con objetivos distintos, y a menudo encontraros, tienen el control de los recursos necesarios para producir el beneficio. Dichos recursos pueden ser dinero, medios de transporte, carreteras, ... Por lo tanto, el plan óptimo puede depender de las decisiones de varios agentes. Normalmente, cuando los agentes unen sus recursos, el beneficio global crece, luego parece lógico pensar que dichos agentes trabajarán juntos para obtener un mejor resultado. Pero ésta no es la única cuestión del problema, pues una vez que se ha conseguido el beneficio global surge una nueva pregunta: ¿cómo se reparte el beneficio obtenido? Todo el problema en conjunto se denomina juego cooperativo. A lo largo de este trabajo daremos respuesta a las cuestiones anteriormente propuestas para diversos juegos cooperativos, todos ellos teniendo como problema raíz un problema de programación lineal. Para este fin hemos dividido la tesis en 6 capítulos. Los dos primeros están dedicados a dar al lector una breve introducción sobre programación lineal, teoría de grafos y teoría de juegos cooperativos, con el fin de que no sea necesario consultar otros trabajos para un buen entendimiento de los capítulos posteriores. Los cuatro últimos capítulos de la tesis abordan cuatro diferentes clases de juegos.

En el capítulo 1 presentamos los problemas de programación lineal. Dichos problemas están pensados para identificar un conjunto de variables maximizando o minimizando una función objetivo lineal sujeta a un conjunto de restricciones que son también lineales. Algunos de los problemas de optimización con los que trabajaremos en esta tesis se pueden formular mediante la teoría de grafos. Un grafo es un conjunto de puntos que pueden estar unidos entre sí mediante arcos. En este capítulo también damos una breve introducción a la teoría de grafos. También presentamos algunos de los modelos más conocidos de la programación lineal, ya que serán de utilidad en el estudio de las cuatro clases de problemas que presentaremos en capítulos posteriores. En dichos problemas consideraremos que un conjunto de agentes tiene el control de los recursos necesarios para producir beneficios, un juego. Las clases que presentamos en esta tesis tienen la particularidad de que los agentes implicados pueden realizar acuerdos entre ellos, los llamados juegos cooperativos. Los juegos en los que no se permite que los jugadores lleguen a acuerdos se llaman juegos no cooperativos, pero no entran en el marco de esta tesis.

En el capítulo 2 introducimos la clase de juegos cooperativos con utilidad transferible. También mostramos diferentes formas de repartir (compartir) los beneficios (costes) generados. Además también presentamos algunas de las clases de juegos cooperativos más conocidas, que serán utilizadas en posteriores capítulos. Para terminar el capítulo se introduce brevemente una generalización de juegos cooperativos en la que cada jugador tiene varios objetivos diferentes y encontrados.

El capítulo 3 analiza más en detalle una de las clases de juegos presentadas en el capítulo 2, los juegos de producción lineal. Owen (1975) probó que dicha clase de juegos es totalmente balanceada, dando además un reparto del core que se puede calcular mediante un problema de programación lineal. Más recientemente, en un artículo firmado por van Gellekom y otros (2000), se proporciona una caracterización axiomática de las soluciones de Owen en los juegos de producción lineal. Aún así no siempre es lógico dar como reparto de un juego de producción lineal una solución de Owen, ya que a veces dichos repartos dan pago cero a jugadores que son absolutamente necesarios para obtener beneficio. Por lo tanto el resultado de dicho reparto puede ser económicamente contradictorio. En esta tesis evitamos dichos problemas con la introducción de un nuevo conjunto de soluciones para juegos de producción lineal, llamados repartos de Owen Extendido. En dichos repartos suponemos que los jugadores juegan un nuevo juego, en el que se han desecho previamente de los recursos sobrantes. Más tarde aplicamos las técnicas propuestas por Owen a el nuevo juego.

En el capítulo 4 abordamos el principal problema presentado en esta tesis, los juegos de cadenas de abastecimiento. Dichos juegos surgen cuando se considera la posible cooperación entre los nodos involucrados en un problema particular de cadenas de abastecimiento definido en la primera sección del capítulo. Los problemas de cadenas de abastecimiento se sitúan entre los modelos más complejos estudiados actualmente en investigación operativa. Incluyen diferentes aspectos que van desde localización o distribución hasta horarios o gestión de inventarios. La parte de optimización de los problemas de cadenas de abastecimiento se ha estudiado en profundidad desde muchos puntos de vista. Sin embargo, la cooperación en los modelos de cadenas de abastecimiento ha sido mayormente estudiada desde un punto de vista de teoría inventarios, obviando otras posibilidades. El análisis de la cooperación en problemas de inventario no es nuevo. Uno puede encontrar en la literatura varios modelos de inventarios (véase Eppen (1979), Gerchak y Gupta (1991), Hartman y Dror (1996, 2003, 2005), Hartman y otros (2000), Anupindi y otros (1991), Müller y otros (2002), Meca y otros (2003, 2004), Minner (2003), Tijs y otros (2005) o Slikker y otros (2006). En este capítulo estudiamos un aspecto diferente del modelo y nos centramos más en temas de distribución de productos que en control de inventarios. En la literatura se han estudiado varios modelos de cooperación en grafos. Nuestro modelo incorpora como novedad el hecho de que los proveedores son jugadores y que los costes de producción y envío, así como los beneficios de venta, son variables. Probamos en este capítulo que la cooperación es ventajosa para los agentes que participan en el juego, ya que al colaborar incrementan el beneficio total. Además demostramos que la cooperación es estable, es decir, hay repartos justos del beneficio general del sistema entre los jugadores. Nuestro modelo no debe interpretarse como un intento de dar un análisis completo de una cadena de abastecimiento, sino más bien como otra pieza más que, junto con todas las aportaciones anteriores, ayude a comprender la naturaleza compleja de la cooperación en modelos de cadenas de abastecimiento.

El problema de cadenas de abastecimiento que proponemos surge cuando, sobre un grafo, un grupo de nodos ofrece un cierto producto, otro grupo de nodos lo demanda y un tercer grupo de nodos no necesita dicho material ni lo ofrece pero es estratégicamente relevante para el plan de distribución. La entrega de una unidad de material en un nodo de demanda genera un cierto beneficio, y el envío de material a través de los arcos tiene un coste asociado. Dicho problema se define en la primera sección del capítulo. Mostramos que en este marco la cooperación es beneficiosa para los agentes. Probamos también que dicha situación cooperativa es totalmente balanceada. El capítulo también muestra la relación entre nuestros nuevos juegos cooperativos y otros conocidos juegos, léase juegos de producción lineal, juegos de flujo, juegos de asignación, juegos de transporte y juegos de camino más corto. Más adelante introducimos algunos conceptos de solución aplicados especialmente a nuestra nueva clase de juegos. La primera de ellas es la solución de Owen, que nos da un reparto del core en tiempo polinomial. Más adelante aplicamos el conjunto de Owen Extendido propuesto para juegos de producción lineal en el capítulo 3 a nuestra clase de juegos de cadenas de abastecimiento. El valor de Shapley, una conocida regla de reparto para juegos cooperativos, también se aplica a nuestra nueva clase de juegos. El cuarto concepto de solución presentado es la solución arco-proporcional. Dicha solución encuentra un plan de distribución óptimo y divide el beneficio total entre los jugadores teniendo en cuenta el uso que se hace de sus arcos en dicho plan de distribución. La tabla 1 presenta un resumen de los resultados obtenidos en la fase de experimentación. Para

Solución	Complejidad	Frecuencia en el core	Tiempo (segundos)
Owen	$O(n^2)$	100.000%	0.102464
Shapley	$O(n^2 2^n)$	73.875%	16.376855
Arco-Proporcional	$O(n^2)$	43.375%	0.078708

Table 1: Diferentes repartos para juegos de cadenas de abastecimiento

terminar esta sección proponemos un procedimiento para encontrar repartos que consiste en resolver diferentes juegos de cadenas de abastecimiento uno a uno, llamados repartos secuenciales. La última sección del capítulo presenta una extensión de nuestros juegos de cadenas de abastecimiento al caso multiobjetivo.

El capítulo 5 está dedicado a introducir y estudiar otra clase nueva de juegos cooperativos, llamada juegos de diámetro. Probamos que los juegos de diámetro son totalmente balanceados y estudiamos dos conceptos de solución, el valor de Shapley y el nucleolo, para esta clase de juegos. Es conocido que, en general, no se puede calcular de forma eficiente ni el valor de Shapley ni el nucleolo, pero presentamos unos algoritmos que demuestran que para la clase de juegos de diámetro ambas soluciones se pueden calcular en tiempo polinomial. Además, probamos que el cálculo del nucleolo de un juego de diámetro se reduce a la resolución de un problema de programación lineal continuo.

El último capítulo de esta tesis, el capítulo 6, estudia otra nueva clase de juegos cooperativos, llamados juegos de asignación multidimensional. Dichos juegos son una extensión de los juegos de asignación clásicos. Probamos que los juegos de asignación multidimensional tienen core no vació. Debido a que los problemas de programación lineal entera que dan lugar a esta clase de juegos, llamados problemas de asignación multidimensional, son NP duros, necesitamos usar algoritmos de aproximación para dar vectores de pagos en un tiempo razonable. Nótese que, para un juego en el que el valor de cada coalición se calcula a partir de un problema de programación lineal entera, el cálculo de la función característica puede ser imposible de realizar en una cantidad de tiempo razonable. Una clase de algoritmos de aproximación, llamada K-SGTS, ha sido utilizada para resolver problemas de asignación multidimensional y viene descrita en trabajos de investigación anteriores. Por lo tanto introducimos un conjunto de repartos para esta clase de juegos basados en los algoritmos K-SGTS, los cuales pueden ser calculados en tiempo polinomial.

La tesis termina con un índice de los principales conceptos utilizados en este trabajo y una lista de las referencias consultadas para su desarrollo.

En el resto de este resumen presentamos los resultados más importantes de los capítulos 3,4,5 y 6.

Capítulo 3: nuevos repartos para juegos de producción lineal

En este capítulo se analizan los juegos de producción lineal, que surgen de procesos de producción lineal en los que hay un conjunto finito de recursos $R = \{1, 2, \ldots, r\}$, a partir de los cuales se pueden producir ciertos productos $P = \{1, 2, \ldots, p\}$. La matriz

$A \in \mathbb{R}^{n\times p}$ es la matriz tecnológica, donde A_{ij} denota la cantidad de recurso i necesaria para producir una unidad del producto j, $\forall\ i = 1,\dots,r,\ j = 1,\dots,p$. El juego aparece cuando un conjunto de jugadores $N = \{1,\dots,n\}$ con objetivos encontrados controla los recursos. Suponemos que el jugador k tiene B_{ik} unidades del recurso i, $k = 1,\dots,n$, $i = 1,\dots,r$. Sea B la matriz correspondiente, denominada matriz de recursos-jugadores. Sea $(e_S) \in \mathbb{R}^{|S|}$ definido por $(e_S)_k = 1$ si $k \in S$ y cero en caso contrario, para todo $S \subset N$. Denotamos por $\mathcal{L}$ la clase de todos los juegos de producción lineal.

Para una coalición cualquiera $S \subset N$, definimos su función característica como el valor óptimo del siguiente problema:

$$\begin{array}{rl} \max & cx \\ \text{s.a.:} & Ax \leq Be_S \\ & x \geq 0 \end{array} \qquad (P(S)) \tag{1}$$

El dual de $P(S)$ es el problema

$$\begin{array}{rl} \max & yBe_S \\ \text{s.a.:} & yA \geq c \\ & y \geq 0 \end{array} \qquad (D(S)) \tag{2}$$

Sea $O_{\min}(A, B, c)$ el conjunto de soluciones óptimas del problema (2) para $S = N$.

Definición 1 *Sea (A, B, c) un juego de producción lineal. El conjunto de Owen de (A, B, c) es*

$$Owen(A, B, c) := \{yB : y \in O_{\min}(A, B, c)\}. \tag{3}$$

Es bien conocido que el conjunto de Owen de un juego de producción lineal está formado por repartos del core de dicho juego. A pesar de ser repartos del core, a menudo los repartos de Owen presentan ciertos problemas de justicia, tales como asignar un pago cero a jugadores que son absolutamente necesarios para producir cualquier tipo de beneficio. Debido a ese tipo de problemas, en este capítulo se presenta otro concepto de solución, llamado conjunto de Owen Extendido, ideado especialmente para los juegos de producción lineal.

Sea $O_{\max}(A, B, c)$ el conjunto de soluciones óptimas del problema (1) para $S = N$. Sea $x^* \in O_{\max}(A, B, c)$. Supongamos que todos los jugadores reducen su cantidad de recurso i hasta que la cantidad total de dicho recurso sea $(Ax^*)_i$, es decir, los recursos sobrantes se eliminan. Sea B^{x^*} la correspondiente matriz de recursos-jugadores. Definimos entonces (A, B^{x^*}, c) como el juego reducido asociado a x^*.

A menudo encontramos juegos de producción lineal en los que hay un único plan de producción óptimo x^*. Sea $\widehat{\mathcal{L}}$ la clase de dichos juegos de producción lineal. Sobre la clase $\widehat{\mathcal{L}}$ se define el conjunto de Owen Extendido así:

Definición 2 *Sea $(A, B, c) \in \widehat{\mathcal{L}}$. El conjunto de Owen Extendido de (A, B, c) es*

$$EOwen(A, B, c) = \{y^x B^x : y^x \in O_{\min}(A, B^x, c), \{x\} = O_{\max}(A, B, c)\}. \tag{4}$$

En este capítulo se demuestran los siguientes resultados para el conjunto de Owen Extendido.

Teorema 1 Sea $(A, B, c) \in \widehat{\mathcal{L}}$ y $\{x^*\} = O_{\max}(A, B, c)$. Se tiene que

1. $EOwen(A, B, c) = Owen(A, B^x, c)$.
2. $Owen(A, B, c) \subset EOwen(A, B, c)$.
3. $EOwen(A, B^x, c) = Owen(A, B^x, c)$.
4. Si $\gamma \in EOwen(A, B, c)$, entonces γ es eficiente.

También en este capítulo se da una caracterización axiomática del conjunto de Owen Extendido en la clase $\widehat{\mathcal{L}}$, a partir de los siguientes cinco axiomas:

1. Un concepto de solución φ sobre $\widehat{\mathcal{L}}$ satisface *eficiencia unipersonal* si $\varphi(A, e_R, c) = \{v_{\min}(A, e_R, c)\}$ para todo $(A, e_R, c) \in \widehat{\mathcal{L}}$.
2. Un concepto de solución φ sobre $\widehat{\mathcal{L}}$ satisface reescalamiento si $\varphi(HA, HB, c) = \varphi(A, B, c)$ $\forall$ $H \in \mathbb{R}_+^{r \times r}$ matriz diagonal con entradas positivas, para todo $(A, B, c) \in \widehat{\mathcal{L}}$.

3. Un concepto de solución φ sobre $\widehat{\mathcal{L}}$ satisface la propiedad de *mezcla* si para toda $H \in {\mathbb{R}_+}^{n\times m}$ con $He_M = e_N$, y para todo$(A, b, c) \in \widehat{\mathcal{L}}$, se cumple que $\varphi(A, B, c)H = \varphi(A, BH, c)$, donde $\varphi(A, B, c)H = \{\alpha H : \alpha \in \varphi(A, B, c)\}$.

4. Un concepto de solución satisface *consistencia* si para cualquier $(A, I_N, c) \in \widehat{\mathcal{L}}$ con $r = n \geq 2$ y para todo $\alpha \in \varphi(A, I_N, c)$ se cumple que $(A_{-k\bullet}, I_{N\setminus k}, \widetilde{c}) \in \widehat{\mathcal{L}}$ y $\alpha_{-k} \in \varphi(A_{-k\bullet}, I_{N\setminus i}, \widetilde{c})$ para todo $k \in N$, donde $\widetilde{c}_j = c_j - \alpha_k A_{kj}$ para todo $j \in P$ y $A_{-k\bullet}$ denota la matriz resultante al borrar la fila k de la matriz A.

5. Un concepto de solución φ sobre $\widehat{\mathcal{L}}$ satisface *supresión* si para todo $(A, I_N, c) \in \widehat{\mathcal{L}}$ y para todo $J \subset P$ tales que $v_{\min}(A_{\bullet-J}, I_N, c_{-J}) = v_{\min}(A, I_N, c)$, entonces $\varphi(A, I_N, c) \subset \varphi(A_{\bullet-J}, I_N, c_{-J})$, donde $A_{\bullet-J}$ denota la matriz resultante después de borrar la columna j, $\forall\, j \in J$, de la matriz A.

Para el caso en el que haya más de un único plan de producción, se define el conjunto de Owen Extendido de tres formas diferentes:

Definición 3 *Sea* $(A, B, c) \in \mathcal{L}$ *y* $\{x^1, \ldots, x^w\}$ *las soluciones extremas del problema* $P(N)$.

1. *El conjunto de Owen Extendido 1 de* (A, B, c) *es el conjunto*

$$EOwen1(A, B, c) = \bigcap_{t=1}^{w} Owen(A, B^{x^t}, c).$$

2. *El conjunto de Owen Extendido 2 es el conjunto*

$$EOwen2(A, B, c) = \{\widehat{y}B^{\widehat{x}} : \widehat{y} \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c),\ \widehat{x} \in O_{\max}(A, B, c)\}.$$

3. *El conjunto de Owen Extendido 3 de* (A, B, c) *es el conjunto*

$$EOwen3(A, B, c) = \bigcup_{t=1}^{w} Owen(A, B^{x^t}, c).$$

A lo largo de este capítulo se demuestra que las tres extensiones del conjunto de Owen están bien definidas y son repartos eficientes. Además, para cualquier juego de producción lineal (A, B, c), se demuestran las siguientes propiedades

- $Owen(A, B, c) \subseteq EOwen1(A, B, c) \subseteq EOwen2(A, B, c) \subseteq EOwen3(A, B, c)$.
- $Owen(A, B, c) \not\supseteq EOwen1(A, B, c) \not\supseteq EOwen2(A, B, c) \not\supseteq EOwen3(A, B, c)$.
- $$\begin{aligned} &EOwen(A, B, c) = EOwen1(A, B, c) = EOwen2(A, B, c) \\ &= EOwen3(A, B, c) = Owen(A, B^{x^*}, c). \end{aligned}$$

El capítulo termina con una aplicación del conjunto de Owen Extendido a los juegos de flujo, definidos en el capítulo 2.

Capítulo 4: juegos de cadenas de abastecimiento

En este capítulo estudiamos modelos de cooperación sobre un problema de cadenas de abastecimiento particular. Actualmente, los problemas de cadenas de abastecimiento están entre los modelos de optimización más complejos analizados por la investigación operativa. Dichos problemas abarcan desde problemas de localización hasta problemas de horarios o gestión de inventarios. Los problemas de cadenas de abastecimiento han sido estudiados desde muchas perspectivas posibles y, sin embargo, la cooperación en estos problemas se ha estudiado principalmente desde un punto de vista de teoría de inventarios. El problema de cadenas de abastecimiento que proponemos surge cuando, sobre un grafo, un grupo de nodos ofrece un cierto producto (nodos oferta), otro grupo de nodos lo demanda (nodos demanda) y un tercer grupo de nodos no lo ofrece ni lo necesita pero es estratégicamente relevante para el plan de distribución óptimo (nodos transbordo). Sean P, Q y R dichos conjuntos. Supondremos que cada nodo oferta i ofrece b_i unidades y cada nodo demanda i pide b_i unidades. La entrega de una unidad de material en un nodo demanda i genera un cierto beneficio k_i, y el transporte de una unidad de material sobre los arcos tiene un coste que depende de cada arco, siendo c_{ij} el coste unitario de transportar una unidad sobre el arco (i, j). Suponemos también que la capacidad de cada arco es limitada, siendo h_{ij} la capacidad del arco (i,j). Sea (N,A) el grafo sobre el que definimos nuestro problema.

El juego que proponemos en este capítulo aparece cuando cada nodo tiene intereses distintos y encontrados, por lo que identificaremos el conjunto de nodos con el conjunto de jugadores.

Definición 4 *Un juego de cadenas de abastecimiento es un par (N,v) donde N es el conjunto de jugadores y v es la función característica, dada para cada coalición $S \subset N$ por el valor óptimo del siguiente problema:*

$$
(5) \quad \begin{array}{ll}
\max & \displaystyle\sum_{(i,j)\in A_S} (k_j - k_i - c_{ij})x_{ij} := f_S(x) \\
s.t. & \displaystyle\sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{j\in S:(j,i)\in A_S} x_{ji} \leq b_i \quad \forall\, i \in P_S \\
& \displaystyle\sum_{j\in S:(j,i)\in A_S} x_{ji} - \sum_{j\in S:(i,j)\in A_S} x_{ij} \leq 0 \quad \forall\, i \in P_S \\
& \displaystyle\sum_{j\in S:(j,i)\in A_S} x_{ji} - \sum_{j\in S:(i,j)\in A_S} x_{ij} \leq -b_i \quad \forall\, i \in Q_S \quad (Pr(S)) \\
& \displaystyle\sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{j\in S:(j,i)\in A_S} x_{ji} \leq 0 \quad \forall\, i \in Q_S \\
& \displaystyle\sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{j\in S:(j,i)\in A_S} x_{ji} = 0 \quad \forall\, i \in R_S \\
& 0 \leq x_{ij} \leq h_{ij} \quad \forall\, (i,j) \in A_S
\end{array}
$$

donde P_S, Q_S, R_S y A_S son los correspondientes conjuntos P, Q, R y A en la coalición S. La clase de todos los juegos (N,v) con v definida como antes se denota $SChG$.

Los juegos de cadenas de abastecimiento presentados en esta tesis cumplen las siguientes propiedades:

Teorema 2 Sea (N,v) un juego de cadenas de abastecimiento. Se cumple que (N,v)

1. está bien definido y es no negativo.

2. es 0-normalizado.

3. es superaditivo.

4. es monótono.

5. es totalmente balanceado.

A partir de la última propiedad enunciada en el teorema anterior, se deduce que los juegos de cadenas de abastecimiento tienen core no vacío.

También estudiamos la relación existente entre nuestros $SChG$ y otros juegos estudiados en la literatura de juegos cooperativos. Consideremos las siguientes clases de juegos: Juegos de flujo, FG; Juegos de programación lineal, $LPrG$; Juegos de asignación, AG; Juegos de transporte, TG; Juegos de camino más corto, SPG.

En la tesis se demuestran las siguientes inclusiones (estrictas):

$$SChG \subsetneq FG,\ SChG \subsetneq LPrG,\ AG \subsetneq SChG,\ TG \subsetneq SChG,\ SChG \subsetneq SPG.$$

El siguiente paso en el estudio de la clase $SChG$ es la búsqueda de repartos del beneficio obtenido entre los nodos del grafo correspondiente. Se estudiaron las siguientes soluciones:

El conjunto de Owen

Lo más relevante de esta sección es el hecho de que se puede dar un reparto del core de nuestros juegos de cadenas de abastecimiento en tiempo polinomial. Se calcula un reparto de Owen a partir del problema dual de (5), que tiene la forma

$$
(6)\quad
\begin{array}{lll}
\min & \sum_{i\in P_S} b_i u_i - \sum_{j\in Q_S} b_j u_j + \sum_{(i,j)\in A_S^{\mathbb{R}}} v_{ij} h_{ij} & \\
\text{s.a.:} & (u_i - t_i) - (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in P_S \times P_S \\
 & (u_i - t_i) + (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in P_S \times Q_S \\
 & (u_i - t_i) - u_j + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in P_S \times R_S \\
 & -(u_i - t_i) - (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in Q_S \times P_S \\
 & -(u_i - t_i) + (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in Q_S \times Q_S \\
 & -(u_i - t_i) - u_j + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in Q_S \times R_S \\
 & u_i - (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in R_S \times P_S \\
 & u_i + (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in R_S \times Q_S \\
 & u_i - u_j + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in R_S \times R_S \\
 & u_i, t_i \geq 0\ \forall\ i \in P_S \cup Q_S & \\
 & v_{ij} \geq 0\ \forall\ (i,j) \in A_S^{\mathbb{R}} &
\end{array}
$$

donde $A_S^{\mathbb{R}} = \{(i,j) \in A_S : h_{ij} < +\infty\}$. Notar que las variables v_{ij} no tienen sentido cuando $h_{ij} = +\infty$, y que las restricciones anteriores sólo son válidas para aquellos pares $\{i,j\}$ tales que $(i,j) \in A$.

Considerando el problema (6) con $S = N$, y $((u^*)_i, (t^*)_i, (v^*)_{ij})$ una solución de dicho problema, un reparto de Owen del correspondiente juego de cadenas de abastecimiento es el reparto $\alpha = (\alpha_1, \ldots, \alpha_n) \in C(N, v)$ con

$$
(7)\qquad \alpha_i = |b_i| u_i^* + \frac{1}{2}\{ \sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* + \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji} v_{ji}^* \} \quad \forall\ i \in N.
$$

El conjunto de todos los repartos generados a partir de soluciones óptimas del problema (6) es el conocido conjunto de Owen, que sabemos que está formado por repartos del core. Además, dichos repartos se pueden calcular en tiempo polinomial, como se afirma en el siguiente teorema que está demostrado en la tesis.

Teorema 3 Sea (N, v) un SChG. La complejidad computacional para calcular un reparto en el conjunto de Owen es polinomial, exactamente dada por

$$
(8)\qquad O(n^2).
$$

Debido a los problemas de justicia que a veces presentan los repartos de Owen, también aplicamos las soluciones de Owen Extendidas a los juegos $SChG$.

Conjunto de Owen Extendido

Definición 5 *Sea (N, v) un juego de cadenas de abastecimiento, y sea x^* un plan de distribución óptimo del correspondiente problema de cadenas de abastecimiento. El juego reducido de (N, v) asociado a x^*, (N, v^{x^*}), es el juego de cadenas de abastecimiento con los mismos datos que (N, v) pero cambiando*

$$(9)\qquad \begin{array}{l} b_i^{x^*} = \sum_{j:(i,j)\in A} x_{ij}^* - \sum_{j:(j,i)\in A} x_{ji}^* \ \forall\ i\ \in P \\ b_i^{x^*} = \sum_{j:(j,i)\in A} x_{ji}^* - \sum_{j:(i,j)\in A} x_{ij}^* \ \forall\ i \in Q \\ h_{ij}^* = x_{ij}^* \ \forall\ (i,j)\ \in A \end{array}$$

Por lo tanto, se define el conjunto de Owen Extendido para juegos de cadenas de abastecimiento de la siguiente forma:

Definición 6 *Dado (N, v) un SChG, el conjunto de Owen Extendido de dicho juego es*

$$(10)\qquad EOwen(N, v) = \{\alpha^{u,t,v,x^*} = (u, t, v) \in O_{\min}(N, v^{x^*}), x^* = O_{\max}(N, v)\}$$

donde

$$(11)\qquad \alpha_i^{u,t,v,x^*} = u_i b_i^{x^*} + \frac{1}{2}\{\sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij}^{x^*} v_{ij} + \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji}^{x^*} v_{ji}\},$$

y $O_{\min}(N, v^{x^*})$, $O_{\max}(N, v)$ son las soluciones óptimas de los problemas dual y primal de los correspondientes juegos.

Al igual que en los juegos de producción lineal, se puede demostrar que:

$$(12)\qquad EOwen(N, v) = Owen(N, v^{x^*}).$$

El valor de Shapley

Anteriormente se estudiaron dos conjuntos de soluciones. Ahora analizamos el valor de Shapley, que no nos da un conjunto sino un único punto, y que además siempre se puede calcular. Consideremos los siguientes axiomas:

1. Eficiencia, (EFF), dice que una regla de reparto reparte todo el beneficio generado por la cooperación de los jugadores.

2. Propiedad de jugadores irrelevantes, (IPP), afirma que si una regla de reparto lo cumple entonces los jugadores irrelevantes (aquellos nodos aislados) no deberían recibir pago alguno.

3. Justicia para jugadores no conectados, (FUP), exige que, para cualquier par i, j de nodos no conectados, i debe ganar o perder cuando j abandona el juego lo mismo que j gana o pierde cuando i abandona el juego.

4. Justicia para jugadores adyacentes, (FAP), el cual significa que, cuando se suprime un arco del grafo, los jugadores situados en los extremos de dicho arco se benefician o se ven perjudicados de la misma forma.

Y ahora se puede demostrar la caracterización axiomática del valor de Shapley para la clase de juegos de cadenas de abastecimiento.

Teorema 4 En nuestros juegos de cadenas de abastecimiento, hay una única regla de reparto Ψ satisfaciendo EFF, IPP, FUP and FAP. Dicha regla de reparto viene dada por

$$\Psi_i(B) = \phi_i(v_B) \tag{13}$$

para todo $B \subset A$ y todo $i \in N$, donde ϕ denota el valor de Shapley.

Desafortunadamente, el valor de Shapley de nuestros juegos de cadenas de abastecimiento no puede ser calculado, en general, en tiempo polinomial.

La solución arco-proporcional

Debido a los problemas de justicia de los repartos de Owen, y a que no se puede calcular el valor de Shapley de manera eficiente, se propone en la tesis otro concepto de solución especialmente creado para los juegos *SChG*. Dicho concepto de solución viene definido a partir de los planes de distribución óptimos del correspondiente problema de cadenas de abastecimiento.

Definición 7 *Sea (N, v) un juego de cadenas de abastecimiento, y sea x^* un plan de distribución óptimo del correspondiente problema de cadenas de abastecimiento. El reparto $\gamma(x^*)$ del juego (N, v) es:*

$$\gamma_i(x^*) = \frac{1}{2} \sum_{j:(i,j)\in A} \left(\frac{L(x^*)}{T(x^*)} - c_{ij} \right) x^*_{ij} + \frac{1}{2} \sum_{j:(j,i)\in A} \left(\frac{L(x^*)}{T(x^*)} - c_{ji} \right) x^*_{ji}, \ \forall\, i \in N \tag{14}$$

donde

$$L(x^*) = \sum_{(i,j)\in A} (k_j - k_i) x^*_{ij}, \quad T(x^*) = \sum_{(i,j)\in A} x^*_{ij}. \tag{15}$$

El conjunto Arco-Proporcional es:

$$\Omega = \{\gamma(x^*) \ : \ x^* \ \textit{es solución óptima de} \ Pr(N)\}. \tag{16}$$

El conjunto Arco-Proporcional satisface las siguientes propiedades:

1. Es una extensión del conjunto Arco-Igualitario propuesto para juegos de transporte.

2. Está formado por repartos eficientes.

3. Bajo ciertas condiciones está formado por repartos del core.

4. Está formado por repartos simétricos.

5. Satisface la propiedad IPP.

6. Satisface la propiedad estándar para dos jugadores.

En la tesis están demostradas todas esas propiedades, incluyendo ejemplos ilustrativos.

Además, se puede demostrar que los repartos del conjunto Arco-Proporcional se pueden calcular en tiempo polinomial.

Teorema 5 Sea (N, v) un SChG. La complejidad computacional para calcular una solución arco-proporcional es polinomial con respecto al número de jugadores, dada por

$$O(n^2). \tag{17}$$

Soluciones secuenciales

En esta sección se estudian formas de calcular repartos mediante la resolución de diferentes problemas secuencialmente. La forma de elegir dichos problemas es haciendo que todos los jugadores ofrezcan o soliciten la mínima demanda u oferta que tienen originalmente. En forma de pseudocódigo, dicho proceso se puede resumir de la siguiente forma:

Sea $G^0 = (N, A, C, b, k, H)$ un problema de cadenas de abastecimiento y (N, v) su juego asociado. Fijar $l = 1$, $y \in \mathbb{R}^n$, $y_i = 0 \ \forall \ i$ e ir al paso 1.

1. Sea $z^l = \max\{|b_i| : b_i \neq 0\}$. Considerar el SChP $G^l = (N, A, C, b^l, k, H)$ donde:
   - Si $b_i > 0$ entonces $b_i^l = z^l$.
   - Si $b_i < 0$ entonces $b_j^l = -z^l$.
   - Si $b_i = 0$ entonces $b_i^l = 0$.

   Considerar ahora (N, v^l) el juego asociado a G^l.
   - Si $v^l(N) = 0$, STOP.
   - Si $v^l(N) > 0$, calcular un plan de distribución óptimo x^l para G^l, y un reparto del juego (N, v^l). Denote dicho reparto por y^l. Ahora actualizar b y H:
     - Para todo $i \in N$ hacer,
       si $b_i > 0$, entonces $b_i = b_i - (\sum_{j:(i,j)\in A} x_{ij}^l - \sum_{j:(j,i)\in A} x_{ji}^l)$.
       si $b_i < 0$, entonces $b_i = b_i + (\sum_{j:(j,i)\in A} x_{ji}^l - \sum_{j:(i,j)\in A} x_{ij}^l)$.
       * Para todo $j \in N$, si $h_{ij} \neq +\infty$ hacer $h_{ij} = h_{ij} - x_{ij}^l$.

Fijar $y = y + y^l$ e ir al paso 1.

Es fácil encontrar ejemplos en los que el reparto final, y, no reparte todo el beneficio que la gran coalición podía haber obtenido en el juego original (N, v). Para evitar tal problema normalizamos y proponemos como reparto final $y = \frac{v(N)}{\sum_{i \in N} y_i} y$.

Nótese que, dependiendo del reparto calculado en cada paso, y^l, obtenemos diferentes repartos. Por lo tanto se pueden definir la solución de Owen secuencial, la solución de Owen Extendida secuencial, la solución arco-proporcional secuencial o el valor de Shapley secuencial.

A modo ilustrativo se realizaron experimentos para comprobar el tiempo de computación necesario para calcular los repartos propuestos en este capítulo. Además se calculó la frecuencia con la que dichos repartos pertenecían al core del juego correspondiente. La tabla 2 muestra un resumen de los resultados obtenidos.

Jugadores	Tiempo Shapley	Tiempo Owen	Tiempo AP	Core Shapley	Core AP
3	0.03353	0.00848	0.00854	99%	93%
4	0.12226	0.01683	0.01217	86%	71%
5	0.39523	0.03048	0.02207	66%	57%
6	1.27179	0.05362	0.04142	81%	35%
7	3.62249	0.08243	0.06557	72%	29%
8	10.95464	0.13888	0.10615	67%	29%
9	27.63470	0.19014	0.13953	74%	13%
10	86.98020	0.29885	0.23429	46%	20%

Table 2: Resultados experimentales.

Juegos de cadenas de abastecimiento multiobjetivo

Para terminar este capítulo se estudian los juegos de cadenas de abastecimiento multiobjetivo, en los que el transporte de material a través de los arcos tiene asociado un vector l-dimensional de costes, y tanto el beneficio como el coste generado en cada uno de los nodos no es un escalar sino otro vector l-dimensional. El resto de los datos son los mismos que en el caso uniobjetivo. Por lo tanto, la función característica de un juego de cadenas de abastecimiento multiobjetivo viene dada por el valor óptimo del siguiente problema de programación lineal multiobjetivo:

$$
(18) \qquad \begin{array}{ll}
\max & (f_S^1(x), \ldots, f_S^l(x)) \\
\text{s.a.:} & \displaystyle\sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{k\in S:(k,i)\in A_S} x_{ki} \leq b_i \quad \forall\, i \in P_S \\
& \displaystyle\sum_{k\in S:(k,i)\in A_S} x_{ki} - \sum_{j\in S:(i,j)\in A_S} x_{ij} \leq 0 \quad \forall\, i \in P_S \\
& \displaystyle\sum_{k\in S:(k,i)\in A_S} x_{ki} - \sum_{j\in S:(i,j)\in A_S} x_{ij} \leq -b_i \quad \forall\, i \in Q_S \\
& \displaystyle\sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{k\in S:(k,i)\in A_S} x_{ki} \leq 0 \quad \forall\, i \in Q_S \\
& \displaystyle\sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{k\in S:(k,i)\in A_S} x_{ki} = 0 \quad \forall\, i \in R_S \\
& 0 \leq x_{ij} \leq h_{ij} \quad \forall\, (i,j) \in A_S
\end{array} \qquad (MPr(S))
$$

donde $f_S^t(x) = \sum_{(i,j)\in A_S} (k_j^t - k_i^t - c_{ij}^t)x_{ij}\ \forall\, t = 1, \ldots, l.$

Estos juegos tienen core de dominancia no vacío, y en la tesis se proporcionan diversas formas de obtener repartos en dicho core.

Capítulo 5: Juegos de diámetro

En este capítulo se presenta una nueva clase de juegos cooperativos, llamada juegos de diámetro.

Dado un árbol $G = (N \cup v_0, A)$, donde v_0 es la raíz del árbol, se define la función característica v para toda posible coalición de N como

$$
(19) \qquad v(S) := d(S \cup \{v_0\})\ \forall\, S \subset N,
$$

donde $d(L)$ denota el diámetro del conjunto L sobre el árbol G, para todo $L \subset N \cup v_0$. El diámetro de un conjunto de nodos L sobre un árbol es la máxima distancia inducida sobre el árbol entre dos nodos de L. Notar que $v(\emptyset) = 0$. El juego cooperativo resultante

(N, v) se denomina juego de diámetro.

Dado un juego de diámetro (N, v), se puede demostrar que:

1. está bien definido y es no negativo.
2. es subaditivo.
3. es convexo.
4. tiene core no vacío.

Además proporcionamos un reparto del core de todo juego de diámetro, que puede ser calculado en tiempo lineal. Dicho reparto se basa en el hecho que el coste total de una coalición puede ser pagado por los nodos que dan el valor del diámetro de dicha coalición.

El valor de Shapley y el nucleolo

A pesar de que en general el valor de Shapley no se puede calcular de forma eficiente, debido a la estructura especial de los juegos de diámetro sí es posible calcularlo en tiempo polinomial para esta clase de juegos. En la tesis proporcionamos dos algoritmos que calculan el valor de Shapley con complejidad $O(n^4)$ y $O(n^3)$ respectivamente. Esto se consigue gracias a que los coeficientes que nos dan el valor de Shapley, $v(S \cup \{i\}) - v(S)$, se repiten con mucha frecuencia. Por lo tanto los algoritmos propuestos se basan en calcular todos los posibles valores que dichos coeficientes pueden tomar para después contar todas las coaliciones que tomas dichos valores, obteniendo así el valor de Shapley.

También se estudia el caso particular en el que el juego de diámetro viene generado por un árbol lineal. En esos casos el cálculo del valor de Shapley se simplifica.

Otro concepto de solución bien estudiado en la literatura de juegos cooperativos es el nucleolo. Dicha solución se obtiene resolviendo una serie de problemas de programación lineal que, en general, tienen $O(2^n)$ restricciones. Es decir, en general el cálculo del nucleolo no se puede hacer con complejidad polinomial con respecto al número de jugadores. Pero de nuevo, debido a la estructura especial de los juegos de diámetro, proporcionamos un algoritmo que calcula el nucleolo de nuestra nueva clase de juegos en tiempo polinomial.

Además, también se prueba un resultado que asegura que para calcular el nucleolo uno sólo tiene que resolver un problema de programación lineal continuo con $O(n^4)$ variables.

Aunque en la tesis todos los cálculos están realizados suponiendo que el juego de diámetro está generado por un árbol, tanto los algoritmos propuestos como los razonamientos utilizados se pueden extender al caso en el que el juego está generado por un grafo general.

Capítulo 6: Juegos de asignación multidimensional

En este último capítulo de la tesis se presenta otra clase nueva de juegos cooperativos, que surge a partir del problema combinatorio de asignación multidimensional. El problema de asignación multidimensional es una extensión del problema clásico de asignación. Un ejemplo típico del problema de asignación es cómo asignar trabajadores a tareas de tal forma que el rendimiento global se maximice. El problema de asignación multidimensional tiene el mismo objetivo, pero teniendo en cuenta otras dimensiones tales como localización física, punto en el tiempo,...

Un problema de asignación w-dimensional se compone de w conjuntos disjuntos dos a dos $N^1, \ldots, N^w$. La asignación de un elemento de cada conjunto a un elemento de cada uno de los otros conjuntos se traduce en un beneficio $a_{i^1,\ldots,i^w}$. Por lo tanto un problema de asignación w-dimensional es

$$(N^1, \ldots, N^w; a), \tag{20}$$

y su formulación como problema de programación lineal entera viene dada en la ecuación (6.9) (nótese que se han tomado los conjuntos N_S^k en lugar de N^k , $\forall\ k = 1, \ldots, w$. En la literatura se ha probado que los problemas de asignación multidimensional son NP-duros.

Los juegos de asignación multidimensional aparecen de forma natural a partir de un problema de asignación multidimensional $(N^1, \ldots, N^w; a)$. Por lo tanto, el conjunto de jugadores vendrá dado por $N = N^1 \cup \cdots \cup N^w$ y la función característica será el valor óptimo del siguiente problema de programación lineal:

$$
(21)\qquad
\begin{array}{lll}
\max & \displaystyle\sum_{i^1\in N_S^1}\cdots\sum_{i^w\in N_S^w} a_{i^1\dots i^w}x_{i^1\dots i^w} & \\
\text{s.a.:} & \displaystyle\sum_{i^2\in N_S^2}\cdots\sum_{i^w\in N_S^w} x_{i^1\dots i^w}\leq 1 & \forall\, i^1\in N_S^1 \\
& \displaystyle\sum_{i^1\in N_S^1}\cdots\sum_{i^{k-1}\in N_S^{k-1}}\sum_{i^{k+1}\in N_S^{k+1}}\cdots\sum_{i^w\in N_S^w} x_{i^1\dots i^w}\leq 1 & \forall\, i^k\in N_S^k,\ 2<k<w-1 \\
& \displaystyle\sum_{i^1\in N_S^1}\cdots\sum_{i^{w-1}\in N_S^{w-1}} x_{i^1\dots i^w}\leq 1 & \forall\, i^w\in N_S^w \\
& x_{i^1\dots i^w}\in\{0,1\} &
\end{array}\qquad (P_S)
$$

donde $N_S^k = N^k \cap S\ \forall\ k = 1,\dots,w$.

Los juegos de asignación multidimensional cumplen las siguientes propiedades:

1. están bien definidos y son no negativos.
2. son 0-normalizados por grupos, es decir, $v(S) = 0\ \forall\ S : |S| < w$.
3. son monótonos.
4. son superaditivos.

En general, los juegos de asignación multidimensional no son balanceados. Aún así, en la tesis proporcionamos una subclase de dichos juegos que sí es totalmente balanceada: la clase de juegos separables. Dichos juegos son aquellos en los que para todo $(i^1,\dots,i^w)\in N^1\times\cdots\times N^w$ existen $a^k_{i^k i^{k+1}}\ \forall\ k=1,\dots,w-1$ tales que $a_{i^1,\dots,i^w}=\sum_{k=1}^{w-1}a^k_{i^k i^{k+1}}$.

Algoritmos de aproximación en juegos de asignación multidmensional

Debido a que los problemas que generan los juegos de asignación multidimensional son NP-duros, se hace necesario el uso de algoritmos de aproximación para dar repartos en un tiempo razonable. La estrategia de aproximación que hemos seguido en esta tesis es aplicar algoritmos tipo greedy. Por lo tanto, una regla de reparto, en adelante llamada reparto greedy, se calcularía así:

Reparto Greedy.

1. $x = 0, C = N$

2. Repetir.
   Encontrar la asociación con beneficio más alto en C, s.
   Sea $a(s)$ el beneficio de la asociación s.
   Para cada $i \in s$ fijar $x_i = a(s)/w$.
   $C \leftarrow C \setminus s$.
   Hasta que no se puedan encontrar más asociaciones.

3. Salida x.

Nótese que no ha sido necesario calcular la función característica del juego para dar este reparto.

El reparto greedy cumple las siguientes propiedades.

Teorema 6 Sea $x = (x_1, \ldots, x_n)$ el reparto greedy de un juego de asignación multidimensional (N, v) que surge de un problema de asignación w-dimensional. Se cumplen las siguientes afirmaciones:

1. x es una preimputación de (N, v).

2. x satisface el principio de racionalidad individual.

3. $\frac{v(N)}{w} \leq x(N) \leq v(N)$.

4. x se puede calcular en tiempo polinomial.

Para terminar la tesis se estudia una generalización del reparto greedy, que consiste en elegir en cada etapa no la asociación que dé el máximo beneficio sino el grupo de K asociaciones disjuntas que maximicen la suma de beneficios. El reparto resultante se llamará reparto K-greedy. Notar que el reparto K-greedy cuando $K = 1$ coincide con el reparto greedy.

Teorema 7 Sea $x = (x_1, \ldots, x_n)$ el reparto K-greedy de un juego de asignación multidimensional (N, v) que surge de un problema de asignación w-dimensional. Se cumplen las siguientes afirmaciones:

1. x es una preimputación de (N, v).

2. x satisface el principio de racionalidad individual.

3. x se puede calcular en tiempo polinomial.

Una vez repasados todos los resultados estudiados en la tesis, a partir de ahora entramos a analizarlos en profundidad. Debido a que parte del trabajo se realizó en el extranjero, el resto de la memoria está redactado en inglés.

COOPERATIVE ANALYSIS ON SUPPLY CHAINS

PhD Thesis by
FEDERICO PEREA

PhD supervisors:
Prof. Dr. Francisco Fernández
Prof. Dr. Justo Puerto

University of Seville
Faculty of Mathematics and Statistics
Department of Statistics and Operations Research

Contents

Introduction

This thesis addresses optimization problems from a cooperative point of view. In such problems an optimal plan must be determined in order to maximize (minimize) certain benefit (cost). If the optimal plan depends only on the decision of one individual agent, the problem is solved by any of the classical optimization tools. But it is usual to find optimization problems in which a group of distinct agents (persons, companies, political parties,...) are in control of the necessary resources to produce benefit. Such resources can be money, means of transportation, roads,... Therefore, the optimal plan might depend on the decisions of several agents. Normally, when the agents controlling the resources join their efforts, the global benefit is maximized. Then it seems logical to think that those agents will work together. But this is not the end of the problem. Once they have joined their resources to obtain a maximal profit, the question of how to divide such profit is posed. This whole situation is called a cooperative game. Throughout this work we will answer such questions for several situations, all of them arising from the so-called linear programming problems. The thesis is divided into 6 different chapters. Chapters 1 and 2 are meant to give the necessary preliminary concepts on optimization, graph theory and game theory in order to follow the rest of the work. The following four chapters analyze four different classes games.

In Chapter 1 linear programming problems are presented. They are problems designed to identify a set of nonnegative variables minimizing a linear objective function subject to a set of linear constraints. Some of the optimization problems we will be working with in this thesis can be formulated by using Graph Theory. A Graph is a set of points which can be connected to each other by arcs. A brief introduction to graphs can also be found in Chapter 1. After having defined linear programming and some of its most visited models, in the following chapters we will study some situations that need linear programming methods in order to be analyzed. In such situations we will consider a group of agents with conflicting objectives, facing optimization problems to maximize a profit or to minimize costs, with the property that they can cooperate in order to increase their benefits: what we will call cooperative games.

In Chapter 2 the class of cooperative games with transferable utility is introduced. Several ways of allocating (sharing) the generated benefit (cost) will be shown. Besides, some classical classes of cooperative games, which will be referred to in following chapters, are presented. To finish, a generalization of games in which each player has k different (conflicting) objectives to be optimized is briefly presented, named multicriteria cooperative games.

Chapter 3 analyzes more in detail one of the games presented in Chapter 2, linear production games. Owen [47] proved that such class of games is totally balanced by providing a core allocation for every linear production game. For obtaining such allocations, called Owen allocations, one only needs to solve a linear programming problem. More recently van Gellekom et al. [24] gave an axiomatic characterization of the Owen solutions for linear programming games. Nevertheless one cannot propose Owen allocations in every linear production game, due to the fact that sometimes Owen allocations give null payoff to players that are absolutely necessary for any benefit to be obtained. Therefore the outcome of the game may lead to very economically counter-intuitive results. We overcome those problems that Owen allocations have by proposing a new set of allocations, named Extended Owen set. In such allocations we assume that players play a new game in which they get rid of the resource surplus, to later apply the techniques proposed by Owen to the new game.

Chapter 4 addresses the main problem presented in this thesis, named *Supply Chain Games*. Such a game arises when considering the possible cooperation between the nodes involved in a particular Supply Chain Problem as defined in the first section of this chapter. Supply chain problems are among the most complex models analyzed nowadays by Operations Research [7]. They include different aspects ranging from location or distribution to scheduling or inventory management. The optimization face of supply chain problems have been widely studied from most of its different perspectives (the interested reader is referred to [8]). Nevertheless, cooperation in supply chain models has been addressed mainly from the inventory point of view. These models assume that firms can make binding agreements on the convenience of the entire system and studies consolidation in ordering, holding, etcetera. The analysis of cooperation in inventory situations is not new. Thus, one can find in the literature several centralization inventory models approached from this point of view. The interested reader is referred to Eppen [15], Gerchak and Gupta [25], Hartman and Dror [29, 30, 31], Hartman et al. [32], Anupindi et al. [2], Müller et al. [44], Meca et al. [38, 37], Minner [40], Tijs et al. [68] and Slikker et al. [64] among others, for comprehensive literature on this subject. In this chapter we address a different aspect of the model and we focus on distribution matters rather than in inventory control. Several models of cooperation on graphs have been studied in the literature, see [23], [36], [42], [58] or [73]. A more related attempt are the newsvendor games, see for instance [75], which analyze the situation of a group of retailers that sell the same item at the same price, with common purchasing costs from a supplier that is not an active agent of the game. Comparing with the above approaches, our model incorporates the fact that suppliers are players, and that the purchasing costs, production costs and selling benefits are variable. We prove that cooperation is advantageous for the firms in the chain since they improve the overall gain. Moreover, we find that the cooperation is stable, i.e. there are fair divisions of the overall benefit of the system among the agents such that no group of them would like to leave the system. Needless to say, none of the previously mentioned approaches by themselves are enough to perform a complete analysis of a supply chain as a whole. In this regard, our approach can be seen as a new building block that together with all the previous attempts will help in understanding the complex nature of cooperation in supply chain models.

The Supply Chain Problem we propose arises when, over a graph, a group of nodes offers certain commodity, other nodes require it and a third group of nodes does not need this material nor offer it but is strategically relevant to the distribution plan. The delivery of one unit of material to a demand node generates a fixed profit, and the shipping of the material through the arcs has an associated cost. Such problem is defined in the first section of this chapter. We show that in that framework cooperation is beneficial for the different parties. We prove that such a cooperative situation, which will be called a Supply Chain Game, is totally balanced by finding a fair allocation (in the core of an associated cooperative game). The chapter also shows the relation between these cooperative games and other well-known games: Linear Production, Flow, Assignment, Transportation and Shortest Path games. Later on we will introduce some solution concepts specifically applied to our new class of games. The first one is the Owen solution, which gives a core allocation in polynomial time. Afterwards we apply the Extended Owen set proposed for linear production games in Chapter 3 to our class of supply chain games. The Shapley value, a well-known value for cooperative games, is also applied to our supply chain games. An axiomatic characterization of the Shapley value is given for the class of Supply Chain Games. The fourth solution concept presented for our new class of games is the Arc-Proportional solution. This solution finds one optimal distribution plan and divides the total benefit among the players according with the use of their arcs in such distribution plan. Table 3 presents a summary of the results obtained through the tested games. To finish the section we proposed a procedure to find allocations consisting of finding

Solution	Complexity	Frequency in Core	Time (sec)
Owen	$O(n^2)$	100.000%	0.102464
Shapley	$O(n^2 2^n)$	73.875%	16.376855
Arc-Proportional	$O(n^2)$	43.375%	0.078708

Table 3: Different allocations for SChG

allocations for different supply chain games one by one, named sequential allocations. The last section of the chapter presents an extension of Supply Chain Games to the multicriteria case.

Chapter 5 is devoted to introduce and study a new class of cooperative games named *Diameter Games*. Such games arise when the value of a coalition is given by the maximum distance between two nodes belonging to the coalition. The game is played over a tree and its players are all the nodes of the tree but the root. In this chapter we prove that diameter games are balanced by giving a core allocation, which can be calculated efficiently. Such allocation is based on the fact that the two most distant nodes may pay the cost of the whole coalition. Two well-known solution concepts, the Shapley value and the nucleolus, are studied for the class of diameter games. It is known that, in general, one cannot efficiently compute nor the Shapley value nor the nucleolus. This is due to the fact that for calculating the Shapley value one, in general, needs to compute the characteristic function, leading to a number of operations growing exponentially with the number of players. For calculating the nucleolus one need to solve up to n, where n is the number of players, linear programming problems with n variables and $O(2^n)$

constraints. Nevertheless, making use of the special structure of this class of games, we provide algorithms that compute such solutions in polynomial time. Additionally, we also prove that the nucleolus of a diameter game can be obtained by solving a continuous linear programming problem.

The last chapter of the thesis, Chapter 6, studies a new class of cooperative games, named Multidimensional Assignment games. Such class of games extends the classical assignment games. Multidimensional Assignment games arise from a class of combinatorial optimization problems: Multidimensional Assignment Problems. The multidimensional assignment (MDA) problem is a higher dimensional version of the standard (two-dimensional) assignment problem in the literature. The general idea behind the MDA problem is that there are often additional scheduling dimensions besides just scheduling men to jobs that should be taken into consideration when making the optimal decision. These additional dimensions might be time, space or some other factors [54].

The multidimensional assignment problem has been shown to be a useful model to solve real world problems. Among them we mention *multi-target tracking*. According to [3], tracking is the processing of measurements obtained from a target in order to maintain an estimate of its current state, which typically consists of:

- Kinematic components - position, velocity, acceleration, turn rate, etc.
- Feature components - radiated signal strength, spectral characteristics, radar cross-section, target classification, etc.
- Constant or slowly varying parameters - aerodynamic parameters, etc.

One of the major difficulties in the application of multi-target tracking involves the problem of associating measurements received by a sensor, also called plots, with the appropriate target. Pattipati et al. ([50], [51]) and later Poore ([55], [56]) formulated the multi-target (multi-sensor) tracking problem as a MDA problem, and developed a multistage Lagrangian relaxation approach to solve the MDA problem as a series of classical (two dimensional) assignment problems, which are solvable in pseudo-polynomial time. More recently, new polynomial time approximation algorithms for solving the MDA problem have been developed, see [9, 52, 53].

In this chapter several properties of MDA games are proved. Unfortunately MDA games are not in general totally balanced. Nevertheless we provide a subclass of MDA games which is totally balanced. Since the integer linear programming problems that give rise to this class of games, MDA problems, are mathematically termed NP-hard, the use of approximation algorithms is needed to give payoff vectors in a reasonable amount of time. Note that, for a game in which the value of each coalition is calculated from an integer linear programming problem, computing the characteristic function might be impossible in a reasonable amount of time. From previous works on Multidimensional Assignment problems we know that a class of approximation algorithms, named K-SGTS, have successfully been used to solve them. Therefore we introduce a class of allocations for this class of games, based on K-SGTS algorithms, which can be computed in polynomial time.

The thesis is finished with an index of the main concepts of this work and a list of the references.

Chapter 1

Linear Programming and Graphs

Linear Programming has both theoretical and practical importance, and constitutes the basis of many computational algorithms to solve real situations. Many practical problems of economy, operations research, decision theory and engineering can be formulated as linear programming models.

A linear programming problem is designed to identify a set of nonnegative variables minimizing a linear objective function subject to a set of linear constraints.

A *standard form* of a *linear programming problem* is:

$$
\begin{aligned}
&\min && z = cx \\
&\text{subject to} && Ax = b \\
& && x \geq 0
\end{aligned}
\tag{1.1}
$$

where A is a given matrix of order $m \times n$, $m \leq n$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and x is an unknown vector of n components which shall be called decision variables.

Many algorithms have been proposed to solve Problem (1.1), among them we mention the *Simplex* method, introduced by Dantzig in 1947. Although in general the simplex algorithm has a good efficiency in practice, it has been shown to have exponential growth in effort on certain classes of problems as the problem size increases. Thus, it cannot be classified as a polynomial time algorithm for solving linear programming problems. However, there exist theoretically efficient algorithms for linear programming problems, for instance Khachian's or Karmarkar's. Both of these algorithms are nonlinear approaches to linear programming problems whose effort grows polynomially in the size of the problem, that is, they are polynomial time algorithms for solving linear programming problems, see [4]. Another class of algorithms for solving linear programming problems are the so-called interior-point methods. One of them, the Predictor-Corrector, is known to solve any linear programming problem with complexity $O(4L\sqrt{n})$, where 2^{-L} is the maximum error between the solution given by the Predictor-Corrector algorithm and the actual optimal value, see [72]. One usual threshold is 10^{-8}, having this way $L \approx 26$. This complexity is not polynomial itself, but can be bounded from above by a linear function on n, for $n > 2$.

There are some optimization models that can be formulated and solved in an alternative way: by using graph theory. In the development of this thesis we will work with such models. Thus, now we give some basic definitions. A *graph* is a pair (N, A) where N is a finite set of points (also called *nodes* or *vertices*) and A is a set of lines, called *edges* or *arcs*. Each arc joins a pair of distinct nodes. If $i, j \in N$, $i \neq j$, the edge joining i and j will be denoted by (i, j), and is said to be *incident* at vertices i and j. There is at most one edge between any pair of nodes, and every edge contains exactly two points of N.

If the edge (i, j) goes from i to j only in this direction, the edge is said to be *directed.* A graph with only directed arcs is said to be a *directed graph* or *digraph*. From now on we will mainly work with directed graphs, so every time we mention graph we will refer to directed graphs.

Given a graph $G = (N, A)$, and two nodes i and j of N, a *path* from i to j in G is a sequence of nodes of the form

$$\{i_0, \ldots, i_k\} \tag{1.2}$$

where $i_0 = i$, $i_k = j$ and $(i_{t-1}, i_t) \in A$ for all $t = 1, \ldots, k$. A path is said to be a *simple path* if every node along the path appears in the sequence only once. A *cycle* is a path joining a node to itself that contains at least two different nodes. A cycle is a *simple cycle* if it is a simple path from a node to itself.

Two nodes (i, j) are said to be *connected nodes* if there is a path from i to j. A graph is a *connected graph* if $\forall$ (i, j) pair of nodes, i and j are connected. Otherwise it is a *disconnected graph.* A connected graph that contains no cycles is called a *tree.* For a more detailed explanation on graphs see for instance [71].

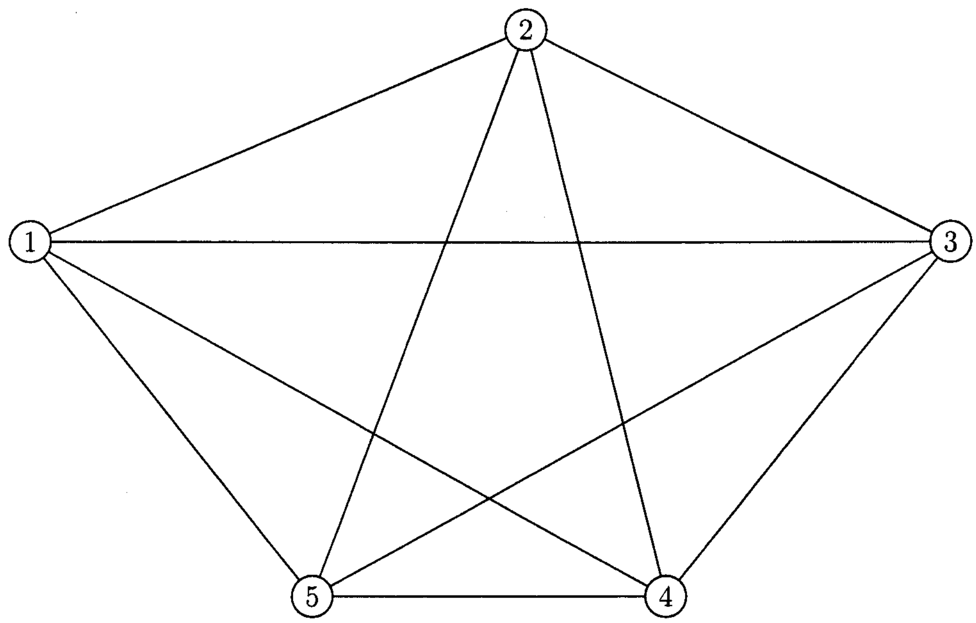

Figure 1.1: Network example

Figure 1.1 shows an example of a network or graph. This graph has the special feature that each node is directly connected by one arc to every other node. Those graphs satisfying that property are called *complete graphs.* A complete graph with n nodes is usually denoted by K_n. So, Figure 1.1 shows K_5.

In the rest of the chapter we present several problems that can be solved by graph theory as well as their formulation by using linear programming.

1.1 Minimum Cost Network Flows Problem

An important class of problems with many applications is the class of Minimum Cost Network Flow Problems. They can be used to model liquids flowing through pipes, parts through assembly lines, current through electrical networks, information through communication networks, ...

Given a digraph $G = (N, A)$ with arc capacities h_{ij} for all $(i, j) \in A$, demands or supplies b_i at each node $i \in N$, and unit flow cost c_{ij} for all $(i, j) \in A$, the minimum cost network flow problem consists of finding a feasible flow that satisfies all the demands at minimum cost.

This situation can be formulated as a linear programming problem as follows:

$$
\begin{array}{llll}
 & \min & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
(1.3) & \text{s.t.:} & \sum_{k\in B(i)} x_{ik} - \sum_{k\in E(i)} x_{ki} = b_i & \forall\, i \in N \\
 & & 0 \leq x_{ij} \leq h_{ij} & \forall (i,j) \in A
\end{array}
$$

where x_{ij} denotes the flow running through arc (i, j), $B(i) = \{k : (i, k) \in A\}$ (arcs beginning at i) and $E(i) = \{k : (k, i) \in A\}$ (arcs ending at i). For the problem to be feasible, the total sum of demands must be equal to the total sum of supplies, that is, $\sum_{i\in N} b_i = 0$.

Sometimes, we may be interested in shipping integer units. In such a case we should ask the decision variables x_{ij} to be integer, which turns Problem (1.3) into an *Integer Linear Programming* (*ILP* for short) Problem, Problem (1.4). An *ILP* problem is a linear programming problem in which the decision variables are forced to be integer.

$$
\begin{array}{llll}
 & \min & \sum_{(i,j)\in A} c_{ij}x_{ij} & \\
(1.4) & \text{s.t.:} & \sum_{k\in B(i)} x_{ik} - \sum_{k\in E(i)} x_{ki} = b_i & \forall\, i \in N \\
 & & 0 \leq x_{ij} \leq h_{ij} & \forall (i,j) \in A \\
 & & x_{ij} \in \mathbb{Z} & \forall (i,j) \in A
\end{array}
$$

Solving an *ILP* problem is, in general, *NP*-hard. Nevertheless, the matrix that gives us the constraints of Problem (1.3) satisfies the *unimodularity* property, see [74]. Linear programs whose constraint matrix is unimodular are proven to have an integer optimal solution. From this argument, the following result can be concluded.

Theorem 1.1.1 In a minimum cost flow problem, if the demands and supplies (b_i) and the capacities (h_{ij}) are integer, then each optimal feasible flow (x^*_{ij}) of Problem (1.4) is integer.

Proof. See [74]. □

This theorem allows us to say that solving Problem (1.3) is equivalent to solving the associated ILP problem (1.4) provided that all b_i and h_{ij} are integer. An ILP problem without the integer constraints is called a *relaxed problem*. For instance, Problem (1.3) is a relaxation of Problem (1.4).

Many different algorithms have been developed to solve the minimum cost flow problem. To mention one of them, the *Out-of kilter algorithm* solves a minimum cost flow problem with complexity $O(nU)$, where U denotes the largest magnitude of any supply/demand or finite arc capacity. For a more detailed summary on algorithms for solving the minimum cost flow problem see [21].

Some of the other problems in networks that will be presented in this chapter, named *Maximum Flow Problems*, *Transportation Problems*, *Assignment Problems* and *Shortest Path Problems*, can be considered as particular instances of the minimum cost network problem. Let us introduce them.

1.2 Maximum Flow Problems

Consider a network with q nodes and r arcs where only one good flows through. We associate an upper bound h_j with each arc $j = 1, \dots, r$ of the network (the maximum amount of flow that can pass through j), which shall be called *capacity* of j. The function h that assigns to every arc its capacity is called *capacity function*. For every node $i = 1, \dots, q$, let $B(i)$ denote the set of arcs that start in node i and $E(i)$ the set of arcs that end in node i. In the *maximum flow problem*, the goal is to find the maximum amount of flow from node 1, which shall be called *source* and denoted by s, to node q, which shall be called *sink* and denoted by t, without violating the capacity constraints of the arcs.

If we denote the amount of source-to-sink flow by f, then the maximum flow problem can be formulated via the following linear program:

$$
\begin{aligned}
\max \quad & f \\
\text{s.t.} \quad & \sum_{j \in B(1)} x_j - \sum_{k \in E(1)} x_k = f \\
& \sum_{j \in B(i)} x_j - \sum_{k \in E(i)} x_k = 0 \quad i = 2, \ldots, q-1 \\
& \sum_{j \in B(q)} x_j - \sum_{k \in E(q)} x_k = -f \\
& 0 \leq x_j \leq h_j \quad j = 1, \ldots, r
\end{aligned}
\tag{1.5}
$$

Example 1.2.1 *Consider the graph in Figure 1.2, in which the capacity of each arc is framed on it. The objective is to send as much material as possible from the source s, node 1, to the sink t, node 4, taking into account the capacity constraints of the arcs. Equation*

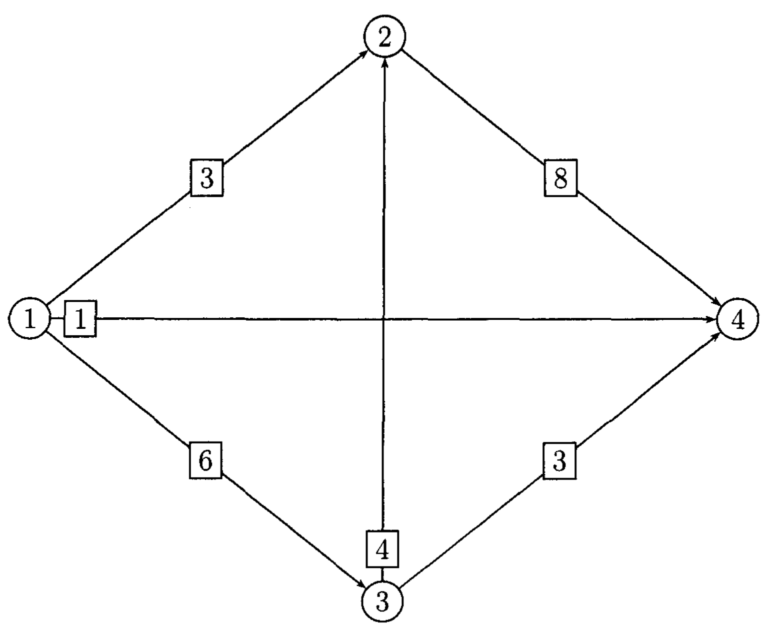

Figure 1.2: Flow Problem

(1.6) shows the linear programming problem arising from this example.

$$
\begin{aligned}
\max \quad & f \\
& x_{12} + x_{13} + x_{14} = f \\
& x_{24} - x_{12} - x_{32} = 0 \\
& x_{34} + x_{32} - x_{13} = 0 \\
& x_{14} + x_{24} + x_{34} = f \\
& 0 \leq x_{12} \leq 3 \\
& 0 \leq x_{13} \leq 6 \\
& 0 \leq x_{14} \leq 1 \\
& 0 \leq x_{24} \leq 8 \\
& 0 \leq x_{32} \leq 4 \\
& 0 \leq x_{34} \leq 3
\end{aligned}
\tag{1.6}
$$

One optimal solution to Problem (1.6) is:

$$x^*_{12} = 3,\ x^*_{13} = 6,\ x^*_{14} = 1,\ x^*_{24} = 8,\ x^*_{32} = 4,\ x^*_{34} = 2. \tag{1.7}$$

Although any Flow Problem can be written as a linear programming problem, there are several algorithms to solve flow problems and calculate their maximum flow that considerably improve the methods for solving general linear programming problems. Among them we mention the *labelling algorithm* by Ford and Fulkerson, see [21]. This algorithm finds the minimum *cutset*, which has been proven to give us the maximum value of feasible flow vectors. A cutset is a set of arcs such that, without them, it is not possible to connect the source s to the sink t. The following theorem shows the relationship between cutsets and maximum flow.

Theorem 1.2.1 In the single commodity flow problem (1.5), the maximum source-to-sink flow is equal to the minimum capacity of the cutsets separating the source and the sink.

Proof. See [41] □

Example 1.2.2 *The minimum cutset separating the source and the sink in Example 1.2.1 is* $\{(1,2),(1,4),(1,3)\}$. *So the maximum source to sink flow in this example is* $3+1+6=10$, *the sum of the capacities of the arcs in the minimum cutset. This amount of flow can be achieved from the flow vector* x^* *obtained as a solution in Example 1.2.1.*

Although the labelling algorithm is easy to implement and flexible, it is not very efficient in practice. A better method for practical implementations is the *FIFO preflow-push algorithm*(first-in first-out), whose complexity is $O(n^3)$, see [26]. Goldberg and Tarjan also showed an implementation of the push-relabel method with dynamic trees, taking $O(nm\log(n^2/m))$ time, see [27] and [28]. Cheriyan and Maheshwari, see [10], and Tunçel, [69], got to solve the problem with complexity $O(n^2\sqrt{m})$. Further improvements were given by Ahuja, Orlin and Tarjan, see [1]. For a more detailed summary on algorithms for solving the minimum cost flow problem see [21].

1.3 Transportation Problems

Assume that we have a *directed network* defined by a set of nodes N and a set of directed edges A. In the transportation problem the network is *bipartite* and *complete*, that is, all its nodes can be arranged in two groups; *supply nodes* numbered $i=1,2,\dots,m$ and *demand* nodes numbered $j=1,2,\dots,n$. Every supply node i offers a_i units of a commodity and every demand node j needs b_j units of the same commodity. Every supply node has n outgoing edges to all demand nodes. The edge from the supply node i to the demand node j has a shipping cost per unit of transported commodity equal to c_{ij}. The problem is to determine the shipping amounts $x_{ij}\ \forall\ i=1,\dots,m,\ j=1,\dots,n$, which minimize the total transportation cost satisfying the requirements of the demand nodes. The formulation of such a problem as a linear program is:

$$
\begin{aligned}
\text{(1.8)} \qquad \min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} \le a_i \quad i = 1, \ldots, m \\
& \sum_{i=1}^{m} x_{ij} \ge b_j \quad j = 1, \ldots, n \\
& x_{ij} \ge 0 \quad i = 1, \ldots, m, \; j = 1, \ldots, n
\end{aligned}
$$

The transportation problem has a feasible solution if the total offer is greater than or equal to the total demand,

$$\text{(1.9)} \qquad \sum_{i=1}^{m} a_i \ge \sum_{j=1}^{n} b_j.$$

In such a case, the demand constraints hold as strict equalities at the optimal solution, that is,

$$\text{(1.10)} \qquad \sum_{i=1}^{m} x_{ij} = b_j \quad \forall \, j = 1, \ldots, n.$$

Moreover, if

$$\text{(1.11)} \qquad \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

then every feasible solution satisfies all the inequality constraints in (1.8) as equalities.

The transportation problem is another instance of how linear programming models network situations and can be solved by a specialized primal simplex method. There is also an alternative approach to solve (1.8) known as the primal-dual algorithm due to Ford and Fulkerson [21], which is of complexity $O(\alpha(n+m))$, where α is the maximum between U as previously defined for minimum cost network flow problems and $C = \max_{i,j}\{c_{ij}\}$.

1.4 Assignment Problem

A very special and important kind of transportation problems is that where $m = n$, and $a_i = b_j = 1 \; \forall \; i, j = 1, \ldots, m$. This very special case is called *assignment problem.* As an example, let us suppose that we are running a company where there are m workers and m jobs to be done. If worker i is assigned to job j, there will be a benefit of c_{ij} units. One wants to determine the assignment between people and jobs that gives us the maximum benefit. If we consider the decision variables x_{ij}, where $x_{ij} = 1$ means that worker i is going to do job j, and $x_{ij} = 0$ means that i does not do job $j \; \forall \; i, j = 1, \ldots, m$, the mathematical formulation of the assignment problem as a LP problem is:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} = 1 \quad i = 1, \ldots, m \\
& \sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, m \\
& x_{ij} = \{0, 1\} \quad i, j = 1, \ldots, m
\end{aligned} \tag{1.12}
$$

Since the assignment problem is a particular case of the transportation problem, the primal-dual algorithm is valid. Nevertheless, there are faster algorithms to solve this problem, for instance the *Hungarian algorithm* due to Kuhn, see [4]. This algorithm is a direct implementation of the primal-dual algorithm for the minimum cost flow problem. It solves the assignment problem with complexity $O(m^2)$, see [21].

When the objective is to match d-tuples, $d > 2$, of objects in such a way that the solution with the optimum total cost is found, the problem is called a Multidimensional Assignment (MDA) problem. The case $d = 2$ is the classical Assignment Problem as presented in this section. The MDA problem will be presented in more detail in Chapter 6.

1.5 Shortest Path Problems

Shortest path problems are among the most fundamental and also the most commonly encountered problems in the study of transportation and communication networks. There are many types of shortest path problems. For instance, we may be interested in finding the shortest path from one specified node in the network to another specified node; or we may need to determine the shortest path from a fixed node to all other nodes; or we might want to find the shortest path between all pair of nodes in the network; or we may need to find a shortest path from one given node to another given node that passes through certain specified intermediate nodes.

The case presented in this section assumes that the given network is directed and there is a weight associated to each edge (which could be interpreted as distances). Formally, a shortest path problem σ is a 5-tuple (N, A, c, s, t) such that:

- (N, A) is a directed graph without loops. The elements of N and A are called nodes and arcs respectively.
- c is a map assigning to every arc $a \in A$ a non-negative real number $c(a)$, which can be interpreted as the length of a.
- s and t are non-empty and disjoint subsets of N. The elements of s and t are called sources and sinks, respectively.

The goal of this model is to find the shortest path between the set s and the set t by using the arcs of A. To show the mathematical formulation of this Shortest Path Problem, let us start first by assuming that $|s| = |t| = 1$, that is, we have to find the shortest path between node s and node t.

We may think of the shortest path problem in a minimum cost network flow context if we set up a network in which we wish to send a single unit of flow from node 1 (the node source) to node n (the node sink) at minimal cost. Thus $b_1 = 1$, $b_n = 1$ and $b_i = 0 \; \forall \; i \in N, \; i \neq 1, n$. The mathematical formulation of this problem becomes:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\
\text{s.t.:} \quad & \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1, n \\ -1 & \text{if } i = n \end{cases} \\
& x_{ij} = 0 \text{ or } 1 \ \forall \ (i,j) \in A
\end{aligned} \tag{1.13}
$$

where c_{ij} denotes the length of arc $(i,j) \in A$.

This formulation can automatically be extended to the case in which we have more than one node in s or in t as follows:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\
\text{s.t.:} \quad & \sum_{i\in s}\Big(\sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki}\Big) = 1 \\
& \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} = 0 \ \forall \ i \notin s \cup t \\
& \sum_{i\in t}\Big(\sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki}\Big) = -1 \\
& x_{ij} = 0 \text{ or } 1 \ \forall \ (i,j) \in A
\end{aligned} \tag{1.14}
$$

To solve the shortest path problem one can use the algorithm due to Dijkstra, see [13], of complexity $O(n^2)$.

1.6 Minimum Cost Spanning Tree problem

The minimum cost spanning tree, or just minimum spanning tree, (MST) of a graph defines the cheapest subset of edges that keep the graph in one connected component. Telecommunication companies are particularly interested in minimum spanning trees, because the minimum spanning tree of a set of sites defines the wiring scheme that connects the sites using as little wire as possible.

Formally, given a weighted directed graph $G = (N, A)$ we want to find the minimum spanning tree. A *tree* of G is a connected subgraph of G such that it contains no cycles. If it also includes every node of the graph, it is called a *spanning tree*. Then, the minimum spanning tree problem consists of finding a spanning tree minimizing the sum of weights. In Figure 1.3 and Figure 1.4 a connected graph and one of its minimum spanning trees are shown.

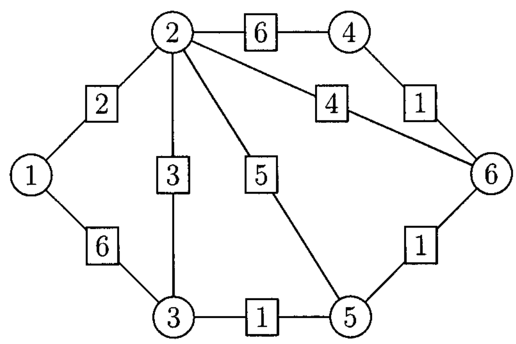

Figure 1.3: Graph.

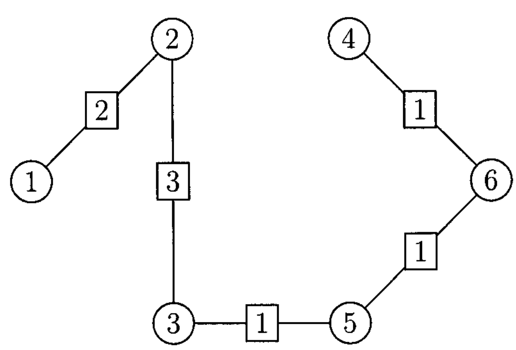

Figure 1.4: Minimum spanning tree of graph in Figure 1.3.

Two classical algorithms efficiently construct minimum spanning trees, namely Prim's, see [57] and Kruskal's, see [35].

The Prim, or Prim-Dijkstra, algorithm starts with an arbitrary node and builds a connected graph by every time choosing a cheapest arc. This way it makes a connection between the part of the tree that has already been constructed and the nodes that are not connected yet. Kruskal's algorithm selects arcs to belong to the tree to be constructed, by starting with a cheapest edge and adding each time a cheapest edge among the ones that have not been selected yet, which does not create a cycle. The computational complexity of both algorithms is $O(m + n \log n)$, where n denotes the number of nodes $|N|$ and m is the number of arcs $|A|$ of the network.

To formulate the MST problem as a linear programming problem we define for each $a \in A$ the variable x_a, where

$$x_a = \begin{cases} 1 & \text{if edge } a \text{ is included in the tree} \\ 0 & \text{otherwise} \end{cases} \tag{1.15}$$

Since a spanning tree should have $n-1$ arcs, the constraint $\sum_{a \in A} x_a = n-1$ is introduced. In every subset S of N, the number of arcs with both endpoints in S must be less than or equal to $|S| - 1$.

If we denote by c_a the length of arc a and for every $S \subset N$ we define $E(S) = \{\{i,j\} \in A | i,j \in S\}$, the formulation of the minimum spanning tree problem as a linear programming problem results in:

$$\begin{aligned} \min \quad & \sum_{a \in A} c_a x_a \\ \text{s.t.:} \quad & \sum_{a \in A} x_a = n - 1 \\ & \sum_{a \in E(S)} x_a \leq |S| - 1, \ \emptyset \subsetneq S \subsetneq N \\ & x_a \in \{0, 1\} \end{aligned} \tag{1.16}$$

1.7 Multiobjective Linear Programming

Multiobjective programming is a part of mathematical programming dealing with decision problems characterized by multiple and conflicting objective functions that are to be optimized over a feasible set of decisions. Such problems, referred to as multiobjective programs (MOPs), are commonly encountered in areas such as engineering, management and others. A typical formulation of a multiobjective linear programming program is given by the problem

$$
\begin{aligned}
\min \quad & f(x) = (c_1 x, \ldots, c_p x) \\
\text{s.t.} \quad & Ax = b \\
& x \geq 0
\end{aligned} \tag{1.17}
$$

where A is a given matrix of order $m \times n$, $m \leq n$, $c_i \in \mathbb{R}^n \; \forall\, i = 1, \ldots p$, $b \in \mathbb{R}^m$ and x is an unknown vector of n components which shall be called decision variables.

A feasible solution x to Problem (1.17) is evaluated by p objective functions ($c_i x$, $i = 1, ..., p$) producing the outcome $f(x)$. We define the set of all attainable outcomes or criterion vectors for all feasible solutions in the objective space $Y := f(X) \subset \mathbb{R}^p$, where $X = \{x \in \mathbb{R}^n : Ax = b\}$.

The symbol min is generally understood as finding optimal or preferred outcomes in Y and their pre-images in X, where the preference between the outcomes results from a binary relation $\mathcal{R}$ defined on Y. One such relation could be

$$x \leq_{\mathcal{R}} y \Leftrightarrow x_i \leq y_i \; \forall\, i. \tag{1.18}$$

For a more detailed description on multiobjective linear programming see, for instance, [20] or [65].

In following chapters we will consider situations in which optimizations problems have to be solved. Additionally, we will also consider that a group of agents having conflicting objectives are involved in the optimization process. Those situations are called games, and will be briefly introduced in Chapter 2.

Chapter 2

TU-Games

A general decision process is a situation in which a group of agents converge and act, independently or collectively, under certain rules in order to obtain a benefit. The part of the mathematics that studies these situations is called Game Theory. Those agents that act in the decision process are called *players* and the result they obtain after the process is called *payoff*. The whole situation is a *Game.*

Game Theory can be divided into two main fields of research: Noncooperative Game Theory and Cooperative Game Theory, the greatest difference between those two areas being the possibility for the players to make agreements between each other. Since in the situations we are going to study in this work it is possible for the players to cooperate, we start by giving a brief introduction to Cooperative Games.

Examples of cooperative situations are pacts and coalitions made in a council or a city hall after an election in order to form a majority or to have a better representation. Those examples are situations in which every player tries to get their best payoff personally, but this payoff can be improved if they cooperate between each other.

For a more complete description on Game Theory, including also noncooperative theory, see [22] or [48].

2.1 Cooperative Games: Definition and properties

Given the set of players $N = \{1, \dots, n\}$, a *coalition* of N is any $S \subset N$. The set of all possible coalitions of N shall be denoted by 2^N.

For a game with set of players $N = \{1, \dots, n\}$, we shall define its *characteristic function* as the map

$$v : 2^N \longrightarrow \mathbb{R}, \tag{2.1}$$

defined for every coalition $S \subset N$ as the maximum profit that the coalition S can make by acting on its own, without taking into account what the other players $N \setminus S$ can do. So, $v(N)$ is the best payoff that the coalition formed by all the players can obtain. This coalition, N, is called *the grand coalition.*

Therefore, a cooperative game can be represented by $\Gamma = (N, v)$ where N is its set of players $\{1, 2, ..., n\}$ and v is its characteristic function

$$\begin{array}{rccc} v: & 2^N & \longrightarrow & \mathbb{R} \\ & S & \longrightarrow & v(S) \end{array} \quad . \tag{2.2}$$

having $v(\emptyset) = 0$ and $v(S)$ the maximum profit that the coalition S can make without the help of any of the other players.

In cooperative situations we can distinguish between two main cases:

- Transferable Utility games (TU-games for short): those where the profit obtained by each player can be transferred between each other. In those games $v(S) \in \mathbb{R}\ \forall\ S \subset N$.
- Non Transferable Utility games (NTU-games for short): those in which the profit made by each coalition is valued in a different way by different players, and it is not possible any transfer of payoffs. In those games $v(S) \in \mathbb{R}^{|S|}\ \forall\ S \subset N$.

Since NTU-games will not be used in the rest of the work, nor will noncooperative games, from now on we will refer to TU-games by saying "cooperative games" or just "games".

We are interested in games satisfying some properties. Among them we enumerate:

Definition 2.1.1 (0-normality) *The game (N, v) is said to be 0-normalized if and only if*

$$v(\{i\}) = 0 \quad \forall\, i \in N. \tag{2.3}$$

In games with this property players have incentives to act cooperatively since, if they did not, they would receive no payoff at all.

Definition 2.1.2 (Superadditivity) *The game $\Gamma = (N, v)$ is superadditive if*

$$\forall\, S, T \subset N : S \cap T = \emptyset \Rightarrow v(S) \;+\; v(T) \;\leq\; v(S \cup T). \tag{2.4}$$

Having a superadditive characteristic function is a desirable property, since players will try to make bigger coalitions so that their benefit is higher. Sometimes we do not require that much and it is enough to have a characteristic function satisfying the following property:

Definition 2.1.3 (Weak superadditivity) *We shall say that the game $\Gamma = (N, v)$ is weakly superadditive if*

$$\forall\, S \subset N, \quad v(N) \geq v(S) + \sum_{i \in N \setminus S} v(\{i\}). \tag{2.5}$$

One can easily see that superadditivity implies weak superadditivity.

Definition 2.1.4 (Monotonicity) *The game $\Gamma = (N, v)$ is monotonic if*

$$\forall S \subset T, \quad v(S) \leq v(T). \tag{2.6}$$

This property states that, if a coalition accepts new members, then its global benefit is not decreased.

Some of the conditions above must be satisfied for the cooperation to be possible, that is, the structure of the characteristic function is of great importance for the cooperation between players to arise. For instance, if we have a game (N, v) in which there exists a player $i \in N$ such that $v(\{i\}) \geq v(S)\ \forall\ S \subset N$, it is clear that player i will not want to cooperate with other players, as the benefit it can get by its own is the maximum. In such game the grand coalition would not be constituted.

Definition 2.1.5 (Convexity) *We say that a game* $\Gamma = (N, v)$ *is convex if*

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T) \quad \forall\ S, T \subset N. \tag{2.7}$$

It can be seen that this definition is equivalent to

$$\forall\ i \in N,\ d_i(S) \leq\ d_i(T), \quad \forall\ S \subset T, \tag{2.8}$$

where

$$d_i(S) = \begin{cases} v(S \cup \{i\}) - v(S) & \text{if } i \notin S \\ v(S) - v(S \setminus \{i\}) & \text{if } i \in S \end{cases} \tag{2.9}$$

$d_i(S)$ is the *marginal contribution* of i to S.

The interpretation of this definition is that, for any player $i \in N$, the bigger a coalition is, the greater the contribution of i to that coalition becomes.

Definition 2.1.6 (Veto player) *Given a game* (N, v)*, we say that player* i *is a veto player if*

$$v(N - \{i\}) = 0. \tag{2.10}$$

Veto players are absolutely necessary for the cooperation to arise, because without them there is no benefit at all.

Another special kind or players are those that have no direct influence in the game since they contribute nothing to any coalition. Those players are called *dummy* players.

Definition 2.1.7 (Dummy player) *Let* (N, v) *be a game. We say that* $i \in N$ *is a dummy player if*

$$v(S \cup \{i\}) = v(S)\ \forall\ S \subset N. \tag{2.11}$$

Let us see now possible incentives that players have in order to constitute the grand coalition, that is, the coalition including all players.

In a cooperative game (N, v) such that

$$v(N) < \sum_{i \in N} v(\{i\}), \tag{2.12}$$

there will be no cooperation to form the grand coalition, since there are players that prefer to act individually because they get a better payoff this way. A game satisfying this property is said to be *irrational.* On the other hand, if we have that

$$v(N) \geq \sum_{i \in N} v(\{i\}), \tag{2.13}$$

then players may want to join all together and form coalitions. In this case we say that the game is *rational*. This kind of games can also be divided in two groups, those where

$$v(N) > \sum_{i \in N} v(\{i\}),$$

which shall be called *essentials*, and those where

$$v(N) = \sum_{i \in N} v(\{i\}), \tag{2.14}$$

named *inessential* games.

The main problem we face when dealing with cooperative games is how to distribute the total benefit generated among the players. We define an *allocation* for the game $\Gamma = (N, v)$ as a vector $x \in \mathbb{R}^n$, where its i^{th} coordinate represents the payoff that player i receives from the allocation x. For each coalition $S \subset N$ and for each $x \in \mathbb{R}^n$, let $x(S)$ denote the payoff that coalition S receives from the allocation x, that is,

$$x(S) = \sum_{i \in S} x_i. \tag{2.15}$$

For a game $\Gamma = (N, v)$ we define its set of *feasible allocations* as the set

$$I^{**}(N,v) = \{x/x \in \mathbb{R}^n,\ x(N) \leq v(N)\}. \tag{2.16}$$

The set above can be bounded by applying an efficiency principle, demanding that players should divide among them the maximum profit they can obtain. Mathematically this set is defined as

$$I^{*}(N,v) = \{x/x \in \mathbb{R}^n,\ x(N) = v(N)\}, \tag{2.17}$$

and its elements are called *preimputations.*

When we assign an allocation it is logical to think that each player should receive from such allocation, at least, what they would obtain by acting individually, that is, $x_i \geq v(\{i\})\ \forall\ i \in N$. This condition is called *individual rationality principle.* Every preimputation satisfying this principle shall be called *imputation.* The set of all imputations of a game (N, v), denoted by $I(N, v)$, is

$$I(N,v) = \{x/x \in \mathbb{R}^n,\ x(N) = v(N),\ x_i \geq v(\{i\})\ \forall i \in N\}. \tag{2.18}$$

Once we have presented what games are, in the rest of this chapter it will be shown how to find allocations satisfying certain properties. Later on some classes of games will be introduced. To finish the chapter we will consider cooperative games in which each player has several objectives.

2.2 Solutions in Cooperative Games

The goal of the cooperative game theory analysis is to find allocations belonging to the preimputation set that have properties we consider acceptable. Giving a procedure that assigns to every game a set of allocations is what we call a *solution concept.* Formally, we have the following definition:

Definition 2.2.1 (Solution Concept) *A solution concept over the class of Cooperative Games is a map* ψ *that assigns to each Cooperative Game* Γ *a subset* $\psi(\Gamma) \subset I^*$*, where* I^* *is the set of preimputations of the game* Γ.

An instance of a solution concept is the proportional rule, which assigns to each player i the value $\frac{v(N)}{n}$. This rule is not always acceptable, see the following example.

Example 2.2.1 *Consider the game* (N, v) *with* $N = \{1, 2\}$ *and* $v(\{1\}) = 1,\ v(\{2\}) = 4,\ v(\{1, 2\}) = 6$. *According to the proportional rule, both player 1 and player 2 should receive 3. But player 2 can make 4 units on its own, therefore it will not want to constitute the grand coalition, since its profit acting individually is higher than what the proportional rule would give him.*

In the rest of the section we see some different solution concepts that are well accepted because of the properties they satisfy.

2.2.1 Stable sets

The idea of stable sets was first introduced in 1944 by Von Neumann and Morgenstern [45]. Those sets are described in terms of a relation between imputations called *dominance*.

Definition 2.2.2 (Dominance) *Given a cooperative game* $\Gamma = (N, v)$ *and two imputations of this game,* x *and* y*, we say that* x dominates y*, and we represent it as* x *dom* y*, if there exists a nonempty coalition* S *such that*

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

From the first condition we conclude that all the members of the coalition S prefer the imputation x to y, whereas the second condition states that if the members of the coalition S cooperate between each other, they receive at least the same benefit as they would obtain from the allocation x. That is, condition 2 states that the members of S are capable of obtaining what x gives them.

Informally, we can define a *stable set* V as a subset of imputations satisfying that no imputation of V dominates another imputation of V, and that any imputation out of V is dominated by an imputation of V. Formally, stable sets are defined as follows:

Definition 2.2.3 *Given is $\Gamma = (N, v)$ a cooperative game. We say that the set $V \subset I(N, v)$ is a stable set if it satisfies the following conditions:*

1. *If $x, y \in V$, then one has that x no dom y (*Internal stability*).*
2. *If $y \in I(N, v)$, $y \notin V \implies \exists\ x \in V\ :\ x$ dom y (*External stability*).*

The concept of dominance leads us to build a set of undominated allocations. Such a set is the core.

2.2.2 The core

Given the imputation set of a game, we are interested in those allocations satisfying some properties. An acceptable property is the *collective rationality principle*, which assures that every coalition S of N receives a better payoff than the one they would obtain by acting without the help of the other players $N \setminus S$. Formally, we define the core via the coalitionally rational property.

Definition 2.2.4 (Coalitionally rational) *Given a cooperative game (N, v), we say that one of its imputations x is coalitionally rational if*

$$\forall S \subset N \quad \sum_{i \in S} x_i \geq v(S). \tag{2.19}$$

The core of the game Γ is defined as the set of imputations satisfying the collective rationality property.

Definition 2.2.5 (Core) *Let (N, v) be a cooperative game. The core of (N, v) is defined by*

$$C(N, v) = \{x \in \mathbb{R}^n\ :\ x(S) \geq v(S)\ \forall S\ \subset N,\ \sum_{i=1}^{n} x_i = v(N)\}. \tag{2.20}$$

We remark that $x \in C(N, v)$ if and only if no coalition can improve upon x. Thus, each member of the core consists of a highly stable payoff distribution. It is known that the set of all undominated imputations is the core, see [48].

Unfortunately not all games have imputations in the core. The concept of balancedness provides us with a theorem that characterizes those games with non-empty core.

Definition 2.2.6 (Balancedness) *Let (N, v) and $\Psi = \{S_1, S_2, \ldots, S_r\}$ be a cooperative game and a collection of coalitions of N, respectively. We say that Ψ is a balanced collection if there exist some coefficients $\gamma_1, \gamma_2, \ldots, \gamma_r$, with $\gamma_l \geq 0 \;\forall\; l = 1, 2, \ldots, r$, such that*

$$\sum_{l:i \in S_l} \gamma_l = 1 \quad \forall\, i \in N. \tag{2.21}$$

The coefficients γ_l are called balancing weights.

Bondareva, [6], and Shapley, [63], independently identified the class of games that have non-empty core as the class of balanced games.

Theorem 2.2.1 [Bondareva and Shapley] The core of the game (N, v) is non-empty if and only if for every balanced collection $\{S_1, \ldots, S_k\}$ with balancing weights $\lambda_1, \ldots, \lambda_k$ the inequality

$$\sum_{j=1}^{k} \lambda_j v(S_j) \leq v(N) \tag{2.22}$$

holds.

The relationship between the core and the stable sets can be summarized in the following proposition:

Proposition 2.2.1 Let (N, v) be a cooperative game and V a stable set of imputations of (N, v). Then $C(N, v) \subset V$.

Proof. See [48]. □

Thus, each member of the core is a highly stable allocation.

So far we have seen two desirable solution concepts but they have a problem, their existence is not assured. In the following section we see other solution concepts, the Shapley value and the nucleolus, that do always exist.

2.2.3 The Shapley value

One of the alternatives for the problem of the no general existence of core allocations has been solved by searching for other solution concepts. Among them there are the so-called *values*. One of those values is the *Shapley value*, which is a solution concept that always gives us a unique solution, unlike the stable sets and the core which are not just one unique point but a set. The Shapley value assigns to each player a convex combination of their marginal contributions ($d_i(S)$ as previously defined). There are more values with the same idea, but it was Shapley who first introduced one of them in 1953, defining this value as follows:

Definition 2.2.7 (Shapley value) *Given a game (N, v), we define the Shapley value of the game as the vector $\phi \in \mathbb{R}^n$ where*

$$\phi_i = \sum_{S \subset N : i \in S} \frac{(s-1)!(n-s)!}{n!} d_i(S)$$

where

$$|S| = s; \; |N| = n \text{ and } d_i(S) = v(S) - v(S \setminus \{i\}). \tag{2.23}$$

It has been proven that the Shapley value is always a preimputation, if the game is superadditive then it is also an imputation and if the game is convex then the Shapley value is an allocation in the core, see [22]. So, a game with convex characteristic function is guaranteed to have at least one core allocation, this allocation being the Shapley value. But we must note that the computational complexity of the calculus of the Shapley value is, in general, exponential, as for every $i \in N$ we have to calculate $O(2^n)$ weights. At the end of this thesis we present a class of games, Diameter Games, for which the calculation of the Shapley value is of polynomial complexity.

2.2.4 The nucleolus

Another well-studied solution concept in the literature is the *nucleolus*, as introduced by Schmeidler in [61]. For its definition, an excess vector is defined for each allocation of a cooperative game.

Definition 2.2.8 *Let (N, v) be a cooperative game and $x \in \mathbb{R}^n$ a payoff vector. We define the excess vector of x as the vector $\theta(x) \in \mathbb{R}^{2^n}$*

$$\theta(x) = (e(S,x)), \quad \text{with } e(S,x) = v(S) - x(S) \ \forall\ S \subset N. \tag{2.24}$$

Each component of the excess vector gives us a measure of the complaints that the corresponding coalition might have if the allocation x is taken.

After defining the excess vector, the nucleolus of a cooperative game can be introduced.

Definition 2.2.9 *Given a function $\psi : \mathbb{R}^{2^n} \longrightarrow \mathbb{R}$, consider the linear programming problem*

$$\begin{array}{rl} \min & \psi(\theta(x)) \\ s.t.: & x \in I(N, v) \end{array} \tag{2.25}$$

If Problem (2.25) selects a payoff vector minimizing the excess vector according to a lexicographic order, i.e., minimizing the maximum complaint, then the solution obtained is that introduced by Schmeidler in [61] called the nucleolus.

Now we see a process for computing such allocation.

Define $\tau(x)$ to be the 2^n-vector of excesses of game (N, v), where the excesses are arranged in nondecreasing order,

$$\tau_i(x) \geq \tau_j(x)\ \forall\ 1 \leq i \leq j \leq 2^n. \tag{2.26}$$

Define the minimum set T of Problem (2.25) under the lexicographic minimization:

$$T = \{x \in I : y \in I \Rightarrow \tau(x) \leq_L \tau(y)\} \tag{2.27}$$

where the order $<_L$ is defined as

$$x <_L y \Rightarrow \exists\, k < n : x_j = y_j \;\forall\, j = 1, ..., k \text{ and } x_{k+1} < y_{k+1}, \tag{2.28}$$

$x \leq_L y$ meaning that either $x <_L y$ or $x = y$.

Theorem 2.2.2 T as defined in Equation (2.27) is a single point, the nucleolus.

Proof. See [22]. □

To finish the introduction of the nucleolus, we show an iterative procedure for calculating it. If $e_1(x)$ is the first component of $\tau(x)$, then the nucleolus must be a solution to

$$\min_{x \in I} \max_{S \subset N} (v(S) - x(S)) \tag{2.29}$$

which can be formulated as a linear programming problem via

$$P_1 \quad \begin{array}{ll} \min & w_1 \\ \text{s.t.:} & x(S) + w_1 \geq v(S) \;\forall\, S \subset N \\ & x(N) = v(N) \end{array} \tag{2.30}$$

If P_1 has a unique optimal solution x^*, then x^* is the nucleolus. Otherwise, we compute the minimum of the second largest excess $e_2(x)$ among those imputations which are optimal solutions to P_1. Let $\mathcal{S}_1$ be such set of imputations and let w_1^* be the value of problem P_1. The new linear programming problem to be solved is

$$P_2 \quad \begin{array}{ll} \min & w_2 \\ \text{s.t.:} & x(S) + w_2 \geq v(S) \;\forall\, S \in 2^N \setminus \mathcal{S}_1 \\ & x(N) = v(N) \\ & x(S) = v(S) - w_1^* \;\forall\, S \in \mathcal{S}_1 \end{array} \tag{2.31}$$

Again, if the solution to P_2 is unique, such solution is the nucleolus. Otherwise we build problems $P_3, P_4, ...$ By Theorem 2.2.2, this process is finished after a finite number of steps.

By the construction of the nucleolus, it is easy to prove the following result.

Theorem 2.2.3 Let (N, v) be a game. If $C(N, v) \neq \emptyset$, then the nucleolus is a core allocation.

2.3 Linear programming games

In this section we deal with the special class of cooperative games whose characteristic function is given by the optimal value of a linear program. Consider the following linear programming problem:

$$\begin{aligned} \max \quad & cx \\ \text{s.t.:} \quad & xA \leq b \\ & xH = d \\ & x \geq 0 \end{aligned} \tag{2.32}$$

where $c \in \mathbb{R}^m$, $b \in \mathbb{R}^p$, $d \in \mathbb{R}^r$, $A \in \mathbb{R}^{m \times p}$, $H \in \mathbb{R}^{m \times r}$. The value of Problem (2.32) shall be denoted by $v_p(A, H, b, d, c)$, and is equal to cx^*, where x^* is any optimal solution to Problem (2.32).

One can check that the dual problem of (2.32) is

$$\begin{aligned} \min \quad & yb + zd \\ \text{s.t.:} \quad & Ay + Hz \geq c \\ & y \geq 0 \end{aligned} \tag{2.33}$$

The value of (2.33) shall be denoted by $v_d(A, H, b, d, c)$. The duality theorem, see [4], assures that (2.33) is feasible and bounded if and only if (2.32) is feasible and bounded, and in this case $v_p(A, H, b, d, c) = v_d(A, H, b, d, c)$.

Suppose now that there is a set of players $N = \{1, \ldots, n\}$, each of them having their own vectors $b(i)$ and $d(i)$, for all $i \in N$. Thus, we can construct a cooperative game from 2.32. This can be done in different ways. One of them could be assuming that for every coalition $S \subset N$, its characteristic function $v(S)$ is the value of the problem

$$\begin{array}{rl} \max & cx \\ \text{s.t.:} & xA \leq b(S) \\ & xH = d(S) \\ & x \geq 0 \end{array} \tag{2.34}$$

where $b_k(S) = \sum_{i \in S} b_k(i),\ d_k(S) = \sum_{i \in S} d_k(i)$.

Definition 2.3.1 (Linear programming game) *A cooperative game (N, v) is said to be a linear programming game if there exist $A \in \mathbb{R}^{m \times p},\ H \in \mathbb{R}^{m \times r}$ and vectors $b(S) \in \mathbb{R}^p,\ d(S) \in \mathbb{R}^r$ such that $v(S) = v_p(A, H, b(S), d(S), c)\ \forall\ S \subset N \setminus \emptyset$.*

To obtain a cooperative game, $v(S)$ must exist for all $S \subset N$, that is, $v_p(A, H, b(S), d(S), c)$ should be a real number for all $S \in 2^N \setminus \emptyset$. So Problem (2.34) should be feasible and bounded $\forall\ S \subset N$. Whenever the equality conditions are void, a sufficient but not necessary condition for (2.34) to be feasible and bounded is that $b(S) \geq 0$ and $A(i, j) \geq 0\ \forall\ i = 1, \ldots, m,\ j = 1, \ldots, p$, with at least one positive entry in each row.

The class of linear programming games coincides with the class of games with non empty core, as stated in the following theorem.

Theorem 2.3.1 A cooperative game is totally balanced if and only if it is a linear programming game.

Proof. See [12]. □

In the proof of this theorem it is described an efficient way of building a core allocation (in general a linear program with 2^n constraints must be solved). For calculating such core allocation it is not necessary to compute the value of $v(S)$ for all $2^n - 1$ non-empty coalitions, we just need to solve the dual program (2.33). In the following chapters of this work we shall make use of this construction, thus we are going to briefly describe how to obtain such core allocation.

Let $y_N^*,\ z_N^*$ be an optimal solution to (2.33). Then, we have that the allocation

$x \in \mathbb{R}^n$ defined as

$$x_i = y_N^* b(i) + z_N^* d(i) \quad \forall i = 1, \ldots, n \tag{2.35}$$

is in the core of the Game, see [12]. The set consisting of all solutions calculated this way is the well-known *Owen set*, see [47]. To complete this section we briefly introduce some classes of linear programming games that will be referred to in the rest of the thesis.

2.3.1 Linear Production Games

Linear production games arise from linear production processes in which there is a finite set of resources $R = \{1, 2, \ldots, r\}$ and from those resources a set $P = \{1, 2, \ldots, p\}$ of consumption goods can be produced. The production technologies are given by a production matrix $A \in \mathbb{R}^{n \times p}$, where A_{ij} denotes the amount of resource i necessary to produce one unit of product j, $\forall\, i = 1, \ldots, r,\ j = 1, \ldots, p$. It is also assumed that the demand of every product is large enough to sell all produced products, the unitary market price of product j being c_j. The game arises when a bunch of players $N = \{1, \ldots, n\}$ with conflicting objectives is in control of the resources. Assume that player k owns B_{ik} units of resource i, $k = 1, \ldots, n$, $i = 1, \ldots, r$. Therefore, let $B = (B_{ik})_{r \times n}$ be the resource-player matrix. Let $b \in \mathbb{R}^r$ be the resource vector, that is $b = Be_N$, where $e_S \in \mathbb{R}^n$ such that $(e_S)_k = 1$ if $k \in S$ and zero otherwise for all $S \subset N$. In other words, b_i is the total amount of resource i owned by the grand coalition, $b_i = \sum_{k=1}^n B_{ik}\ \forall\ i \in R$. Thus, the maximum profit that can be made by the grand coalition is given by the value of the following linear programming problem:

$$\begin{array}{rll} \max & cx & \\ \text{s.t.:} & Ax \leq b & \qquad (P(N)) \\ & x \geq 0 & \end{array} \tag{2.36}$$

The dual program of $P(N)$, denoted by $D(N)$, is

$$
\begin{array}{lll}
\max & yb & \\
\text{s.t.:} & yA \geq c & (D(N)) \\
 & y \geq 0 &
\end{array} \tag{2.37}
$$

It is easy to check that, although players can try to produce individually, it is always more profitable to join their resources as the benefit they obtain this way is at least as high as the sum of the individual profits separately. For a coalition $S \subset N$, we define its characteristic function via the optimal value of the problem

$$
\begin{array}{lll}
\max & cx & \\
\text{s.t.:} & Ax \leq Be_S & (P(S)) \\
 & x \geq 0 &
\end{array} \tag{2.38}
$$

The dual of $P(S)$ is the linear program

$$
\begin{array}{lll}
\max & yBe_S & \\
\text{s.t.:} & yA \geq c & (D(S)) \\
 & y \geq 0 &
\end{array} \tag{2.39}
$$

It is easy to check that $P(S)$ is feasible and bounded for all possible coalitions if

- $Be_S > 0$.
- $c \geq 0$.
- If $c_j > 0$ there is at least one resource $i \in R$ with $A_{ij} > 0$.

Definition 2.3.2 *$\mathcal{L}$ is the class of linear production games (LP games for short). That is to say, $\mathcal{L}$ is the class of cooperative games (N, v) such that there exists a 3-tupla (A, B, c) satisfying that $v(S)$ is the optimal value of Problem 2.38, for every coalition $S \subset N$.*

It is easy to check that the class of linear production games is included in the class of linear programming games, as stated in the following theorem.

Theorem 2.3.2 Linear Production Games are Linear Programming Games.

2.3.2 Flow Games

Another class of games that can be expressed as linear programming games is that of *Flow Games* (*FG* for short), which arise from Flow Problems and we describe as follows.

Consider a directed network G with node and arc sets $M = \{1, \ldots, q\}$ and $L = \{1, \ldots, r\}$, respectively. Let $N = \{1, \ldots, n\}$ be the set of players. Suppose that each arc $i \in L$ belongs to a unique player $p_i \in N$. However, we allow a given player $j \in N$ to own several arcs. Let o be the function

$$\begin{array}{rccc} o: & L & \longrightarrow & N \\ & i & \longrightarrow & o(i) \end{array} \tag{2.40}$$

that assigns to every arc a player, the owner of that arc (o is called *ownership function*).

Let node 1 and node q be the *source* and the *sink* of this network respectively. For each coalition of players $S \subset N$, let G^S be the network consisting of all the nodes of G but only the arcs that belong to the members of S. We denote by $f(S)$ the value of the maximum flow from the source to the sink in G^S.

Definition 2.3.3 (Flow game) (N, v) *is said to be a Flow Game if there exists a network G such that*

$$v(S) = f(S) \ \forall \ S \subset N. \tag{2.41}$$

Now we will see how FG can be transformed into Linear Programming Games, proving this way that they have non-empty core. To do so we have to give the necessary matrices and vectors to define the characteristic function of any Flow Game as an optimal solution of a linear program.

Theorem 2.3.3 Flow Games are Linear Programming Games.

Proof. The proof is taken from [12], but we include it since it gives us the way to express Flow Games as Linear Programming Games.

Let (N, v) be a Flow Game with associated network G, consisting of r arcs and q nodes. Let $l_1, l_2, \cdots, l_r$ be an ordering of the arcs of G such that $\{l_1, \cdots, l_g\}$ are those arcs beginning at the source and $\{l_{g+1}, \cdots, l_h\}$ are the arcs that finish at the source.

For each $i \in N$, define vector $b(i) \in \mathbb{R}^r$ as follows:

$$b_k(i) := \begin{cases} c(l_k) & \text{if } o(l_k) = i \\ 0 & \text{otherwise} \end{cases} \tag{2.42}$$

Let $p_1, \cdots, p_q$ be the nodes of G, satisfying that p_1 is the source and p_q is the sink. We define the matrix $H = (h_{km}) \in \mathcal{M}_{r \times q}$ via:

$$h_{km} := \begin{cases} 1 & \text{if } m \neq 1, q \text{ and } l_k \text{ starts at } p_m \\ -1 & \text{if } m \neq 1, q \text{ and } l_k \text{ ends at } p_m \\ 0 & \text{otherwise} \end{cases} \tag{2.43}$$

For each $i \in N$ we define $d(i)$ as the q-dimensional vector with all its entries null. Let $p \in \mathbb{R}^q$ be the vector with the first g coordinates equal to 1, the following $h-g$ coordinates equal to -1 and the rest of the coordinates equal to 0. If we consider the following linear program,

$$\begin{array}{lll} \max & px & \\ \text{s.t.:} & xI \leq b(S) & (1) \\ & xH = d(S) & (2) \\ & x \geq 0 & \end{array} \qquad P_S \tag{2.44}$$

where $b(S) = \sum_{i \in S} b(i)$, $d(S) = 0$, then we have that $v(S) = px^*_S$, where x^*_S is an optimal solution to (2.44), $\forall\ S \in\ 2^N \setminus \{\emptyset\}$. This concludes the proof. In (2.44) the inequalities (1) correspond to the capacity constraints and the inequalities (2) refer to the flow conservation constraints.

The dual program of (2.44), which will be useful in following sections, is

$$\begin{array}{lll} \min & yb(S) & \\ & Iy + Hz \geq p & D_S \\ & y \geq 0 & \end{array} \quad (2.45)$$

□

Example 2.3.1 *Consider the Flow Problem described in Example 1.2.1. Suppose that there are three players that own the six arcs. In Figure 2.1 we represent this situation where the three numbers on the arcs mean: the number we assign to it, the player that owns it and its capacity, in that order.*

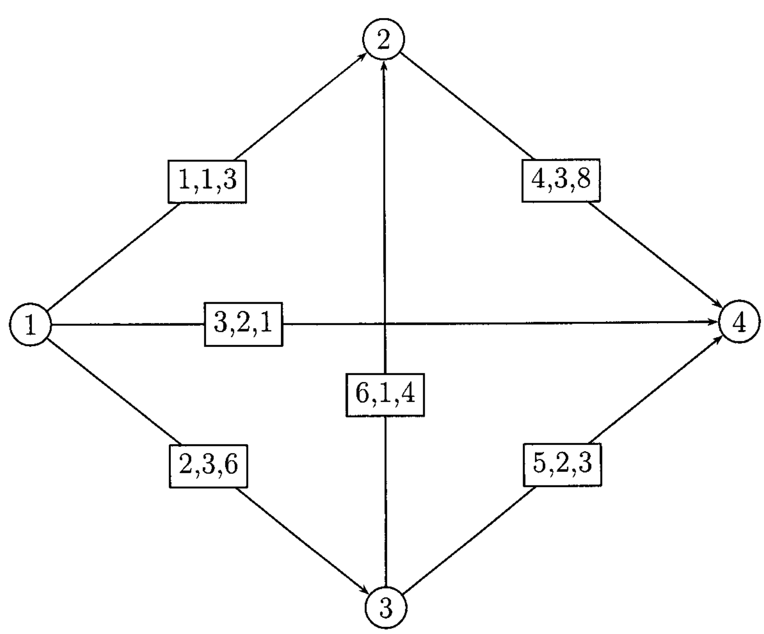

Figure 2.1: Flow game.

The characteristic function of the resultant game, that is, the maximum source-to-sink flow that each coalition can transport without the help of other players, is expressed *in the following table:*

(2.46)

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
$v(S)$	0	1	0	1	7	4	10

By performing the above construction to transform a flow game into a linear programming game, we obtain the following data:

$$b(1) = (3, 0, 0, 0, 0, 4),\ b(2) = (0, 0, 1, 0, 3, 0),\ b(3) = (0, 6, 0, 8, 0, 0),$$

$$H = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix} \tag{2.47}$$

and

$$p = (1, 1, 1, 0, 0, 0). \tag{2.48}$$

Then, by solving the respective linear programs we obtain the characteristic function of the game, v, described above.

2.3.3 Assignment Games

Let $N = M \cup M'$, where M and M' are the supply node set (worker set) and the demand node set (job set) respectively, see Section 1.4. Let $M = \{1, 2, \ldots, m\}$ and $M' = \{m + 1, m + 2, \ldots, 2m\}$. Then the coalition consisting of players $i \in M$ and $m + j \in M'$ can obtain a profit

$$v(\{i, m + j\}) = c_{ij}. \tag{2.49}$$

For any $S \subset M$ or $S \subset M'$, clearly $v(S) = 0$. For other S containing both jobs and workers, $v(S)$ is equal to the maximum total profit generated by the assignment of jobs among the members of S, subject to the constraints that no job can be *assigned* to more than one worker, and no worker can do more than one job. Thus, if S has no more jobs than workers, we assign worker $m + j(i)$ to job i, and the total profit is

$$\sum_{i \in S \cap M} c_{i,j(i)}. \tag{2.50}$$

Then, $v(S)$ is given by the value of the linear program

$$\begin{aligned} \max \quad & \sum_{i \in S \cap M} \sum_{j \in S \cap M'} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j \in S \cap M'} x_{ij} = 1 \quad i \in S \cap M \\ & \sum_{i \in S \cap M} x_{ij} = 1 \quad j \in S \cap M' \\ & x_{ij} = \{0, 1\} \quad i, j \in S \end{aligned} \tag{2.51}$$

or equivalently,

$$v(S) = \max \sum_{i \in S \cap M} c_{i,j(i)} \tag{2.52}$$

where the maximum is taken over all such assignments. A different treatment in case S has more jobs than workers gives us

$$v(S) = \max \sum_{m+j \in S \cap M'} c_{i(j),j}. \tag{2.53}$$

For the set N we have

$$v(N) = \max \sum_{i=1}^{m} c_{i,j(i)}, \tag{2.54}$$

where the maximum is taken over all permutations $(j(1), j(2), \ldots, j(m))$ of the set M. This can be written as

$$v(N) = \max \sum_{i=1}^{m} \sum_{j=1}^{m} x_{ij} c_{ij}, \tag{2.55}$$

with the usual interpretation for the coefficients x_{ij}, if j^{th} worker is to do i^{th} job, then $x_{ij} = 1$, and $x_{ij} = 0$ otherwise. Consider the linear program (1.12) in section 1.4. Due to the unimodularity of its constraint matrix, it can be shown that the solution to this program coincides with the solution to the relaxed problem

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} = 1 \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, m \\ & x_{ij} \geq 0 \quad i, j = 1, \ldots, m \end{aligned} \tag{2.56}$$

It is well known that, among all the properties that they have, the core in these games is nonempty and coincides with the Owen set, as we can see in the following construction.

Consider, next, the dual program of (2.56), which is

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} y_i + \sum_{j=1}^{m} z_j \\ \text{s.t.:} \quad & y_i + z_j \geq c_{ij} \quad i, j = 1, \ldots, m. \end{aligned} \tag{2.57}$$

Note that there is no nonnegativity restriction on the dual variables y_i, z_j. However it is easy to see that the minimum vector is not unique, for instance if

$$(u^*; z^*) = (y_1^*, \ldots, y_m^*; z_1^*, \ldots, z_m^*) \tag{2.58}$$

is an optimal vector in (2.57), then so is

$$(y'; z') = (y_1^* - t, \ldots, y_m^* - t; z_1^* + t, \ldots, z_m^* + t), \tag{2.59}$$

for any real number t. In particular, if we choose

$$t = \min_i y_i^* \tag{2.60}$$

then

$$\min_i y_i' = 0. \tag{2.61}$$

Thus, $y_i' \geq 0$ and there exists k such that $y_k' = 0$. For every j we have that

$$z_j' = y_k' + z_j' \geq c_{ij} \geq 0. \tag{2.62}$$

Then, $(y'; z')$ constructed this way is a minimizing vector for (2.57) with all components non-negative. Now, by duality

$$\sum_{i=1}^{m} y_i' \sum_{j=1}^{m} z_j' = v(N) \tag{2.63}$$

and so $(y'; z')$ is an imputation for the game (N, v). We shall show that in this allocation the collective rationality property holds.

For any $S \subset N$ we have

$$v(S) = c_{i_1 j_1} + \cdots + c_{i_q j_q}, \tag{2.64}$$

where $i_1, \ldots, \imath_q, j_1, \ldots, j_q$ are distinct members of S. Then

$$\sum_{i \in S} y_i' \sum_{m+j \in S} z_j' \geq y_{i_1}' + \cdots + y_{i_q}' + z_{i_1}' \cdots + z_{i_1}' \geq c_{i_1 j_1} + \cdots + c_{i_q j_q} = v(S), \tag{2.65}$$

which concludes that $(y'; z') \in C(N, v)$.

In the following section we analyze a class of games that generalizes the Assignment Games.

2.3.4 Transportation games

A Transportation Game is defined by a 5-tuple (P, A, B, p, q) where:

- P is the set of supply nodes.
- Q is the set of demand nodes.
- B is the matrix of profits (b_{ij} is the profit generated when transporting one unit from $i \in P$ to $j \in Q$).
- p is the vector of offers (p_i is the amount of material that player $i \in P$ offers).
- q is the vector of demands (d_j is the amount of material that player $j \in Q$ demands).

The characteristic function of a Transportation Game (N, v), where $N = P \cup Q$ and B, p, q are defined as before, is given by the optimal value of the following linear program:

$$
\begin{aligned}
\max \quad & \sum_{i \in P_S} \sum_{j \in Q_S} b_{ij} x_{ij} \\
\text{s.t.:} \quad & \sum_{j \in Q_S} x_{ij} \leq p_i \quad \forall\, i \in P_S \\
& \sum_{i \in P_S} x_{ij} \leq q_j \quad \forall\, j \in Q_S \\
& x_{ij} \geq 0 \quad \forall\, i \in P_S,\ \forall\, j \in Q_S
\end{aligned} \tag{2.66}
$$

where $P_S = P \cap S$, $Q_S = Q \cap S$. That is, each coalition plays the game only with the arcs and nodes completely owned by them. By taking $p_i = q_j = 1\ \forall\ i \in P,\ j \in Q$ we obtain an Assignment Game. So, the reader may note that the class of Assignment Games is included in the class of Transportation Games. For a more detailed description of Transportation Games (TG for short) see [59] or [60]. In those works some properties of TG are proven, among them we underline the following:

Proposition 2.3.1 TG are 0-normalized.

That means that no player wants to act individually, as they would receive no payoff at all if they did.

Proposition 2.3.2 TG are totally balanced.

From this result we conclude that it is always possible to find core allocations in Transportation Games.

Proposition 2.3.3 Let (N, v) be a Transportation Game. For every $\lambda \in [0, +\infty)$, the game $(N, \lambda v)$ is a Transportation Game.

This result states that the multiplication times a non-negative scalar is a closed operation in Transportation Games.

2.3.5 Shortest Path Games

From a Shortest Path Problem, see Section 1.5, in [23] the class of shortest path games is introduced as follows. Consider a shortest path problem σ whose nodes are owned by a finite set of players N according to a map $o : X \to N$, such that $o(x) = i$ means that player i is the owner of node x. For any path P connecting a source and a sink, $o(P)$ denotes the owners of the nodes in P (only paths connecting a source and a sink are considered). Suppose that the transportation of a certain good from a source to a sink of σ produces an income g and a cost given by the length of the path that was used. Suppose also that a coalition $S \subset N$ can transport the good only through paths owned by its members (a path is owned by a coalition S if $o(P) \subset S$). A Shortest Path cooperative situation σ is any 4-tuple (σ, N, o, g). With σ it is associated the game (N, v_σ) whose characteristic function v_σ is given by:

$$v_\sigma = \begin{cases} g - L_S & \text{if } S \text{ owns a path in } \sigma \text{ and } L_S < g \\ 0 & \text{otherwise} \end{cases} \tag{2.67}$$

for every $S \subset N$, where L_S is the length of the shortest path owned by S. A *Shortest Path Game* is any such game (N, v_σ) associated with a shortest path cooperative situation σ. SPG denotes the class of shortest path games.

In [23] the relationship between SPG and monotonic games, see Definition 2.1.4, is studied. If we denote by MO the class of monotonic games with finite set of players, the following result holds:

Theorem 2.3.4 The class of Shortest Path Games and the class of Monotonic Games coincide.

In the same paper it is shown that, in general, SPG are not totally balanced and some conditions for SPG to have non-empty core are given.

2.3.6 Minimum Cost Spanning Tree Games

Let $G = (N \cup \{0\}, A)$ be a graph. We define a minimum cost spanning tree (MCST) game over G as the cooperative game (N, v) such that $v(S) = \sum_{a \in T_S} c_a$, where T_S is a minimum cost spanning tree in the complete graph made out of the nodes $S \cup \{0\}$, see Section 1.6.

Bird, [5] proposed the following cost allocation rule. For each $i \in N$, let x_i be the amount that i has to pay. Then x_i is equal to the cost of the edge incident upon i on the unique path from i to 0 in T, where T is a minimum cost spanning tree over G. Since there could be more than one minimum spanning tree, this way of dividing costs leads, in general, to more than one allocation. The following result shows the quality of the bird cost allocation scheme.

Theorem 2.3.5 Let (N, v) be a MCST game. Let x be a Bird allocation for (N, v). Then $x \in C(N, v)$.

Proof. See [5] □

This theorem shows that MCST games are totally balanced.

Once defined linear programming games and some of its more important subclasses, we finish the chapter by introducing an extension of classical cooperative games.

2.4 Multicriteria Cooperative Games

We might find games in which the payoffs that coalitions receive are valued in a k-dimensional space, that is, players have different objectives to maximize in the same cooperative TU-game. This kind of games arise when players consider how to allocate among them the undominated outcomes obtained as solutions to the corresponding multicriteria optimization problem. For a description of multicriteria optimization see [14] and for a complete guide on multicriteria games see [18].

The idea of multicriteria cooperative games is formalized in the next definition.

Definition 2.4.1 (Multicriteria game) *A combinatorial multicriteria cooperative game is a pair (N, V), where $N = \{1, 2, \ldots, n\}$ is the set of players and V is a function which assigns to each coalition $S \subseteq N$ a finite subset $V(S)$ of $\mathbb{R}^k$, the* characteristic set *of coalition S, such that $V(\emptyset) = \{0\}$.*

Vectors in $V(S)$ represent the payoffs, in terms of k criteria, that the members of coalition S can guarantee by themselves. If the characteristic functions in these games are set-to-set maps instead of the usual set-to-point maps, the arising class of cooperative games is a subclass of the so called Set Valued TU games that have recently been explored in [17]. Let us see an example of a Set Valued TU game.

Example 2.4.1 *Consider a game with three players whose payoffs are measured with respect to two criteria. The characteristic function of this multicriteria combinatorial game is given in the following table:*

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	N
$V(S)$			$\left\{\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix}\right\}$	$\left\{\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}2.5\\1.3\end{pmatrix},\begin{pmatrix}2\\2\end{pmatrix}\right\}$		$\begin{pmatrix}2\\2\end{pmatrix}$	$\left\{\begin{pmatrix}3\\5\end{pmatrix},\begin{pmatrix}4\\4\end{pmatrix}\right\}$

The goal of a scalar TU-game is to allocate in a fair way the value obtained by the grand coalition. Quite frequently, in the set-valued case, there is more than one element (efficient outcomes) that may be considered to be divided among the players; and the question is how an achievable vector $z^N \in V(N)$ will be fairly allocated.

An allocation in a multicriteria combinatorial cooperative game consists of a payoff matrix $X \in \mathbb{R}^{k \times n}$. The i^{th} column, X^i, represents the payoffs of the i^{th} player for each criteria; therefore $X^i = (x_{i1}, x_{i2}, \ldots, x_{ik})^t$ are the payoffs for player i. The j^{th} row, X_j, is an allocation of the total amount obtained in each criteria; $X_j = (x_{1j}, x_{2j}, \ldots, x^{nj})$ are the payoffs for each player corresponding to criteria j. The sum $X^S = \sum_{i\in S} X^i$ is the overall payoff obtained by coalition S.

Definition 2.4.2 (Multicriteria allocation) *An allocation of the combinatorial multicriteria cooperative game (N, V) is a matrix $X \in \mathbb{R}^{k\times n}$ such that $X^N = \sum_{i\in N} X^i \in V(N)$. The set of allocations of the game is denoted by $I^*(N, V)$.*

As in the classical TU-games, we look for allocations satisfying some properties such as collective rationality, that is, the core. Allocations in the core provide, to every player or coalition, payoffs that are not worse than any of those that they can guarantee on their own. To simplify, in the following $X^S \not\leq V(S)$ means that there is no $z^S \in V(S)$ such that $X^S_j \leq z^S_j$, $j = 1, 2, \ldots, k$, and $X^S \neq z^S$.

Definition 2.4.3 (Dominance core) *The dominance core of a combinatorial multicriteria game (N, V) is the set of allocations, $X \in I^*(N, V)$, such that $X^S \not\leq V(S)\ \forall\ S \subset N$. We will denote this set as $C(N, V; \not\leq)$.*

As long as $C(N, V; \not\leq) \neq \emptyset$, it is possible to find dominance core allocations by a scalarization method. For each vector $\lambda \in \Lambda = \{\lambda \in \mathbb{R}^k\ \lambda_j > 0,\ j = 1, \ldots, k,\ \sum_{j=1}^{k} \lambda_j = 1\}$, and any vector $z^N \in V(N)$, we define the scalar game $(N, v_\lambda^{z^N})$ as the game whose characteristic function is given by:

$$(2.68) \qquad v_\lambda^{z^N}(\emptyset) = 0, \quad v_\lambda^{z^N}(S) = \max_{z^S\in V(S)} \lambda^t z^S,\ \forall\ S \subset N, S \neq \emptyset, \quad v_\lambda^{z^N}(N) = \lambda^t z^N.$$

We will denote by $C(N, v_\lambda^{z^N})$ the core of this scalar game, that is,

$$(2.69) \qquad C(N, v_\lambda^{z^N}) = \{x \in \mathbb{R}^n, \sum_{i\in N} x^i = \lambda^t z^N,\ x(S) \geq v_\lambda^{z^N}(S),\ \forall\ S \subset N\}.$$

The next result states that for some $z^N \in V(N)$ it is possible to find dominance core allocations of multicriteria games from core allocations of the corresponding scalarized game.

Theorem 2.4.1 Let $\lambda \in \Lambda$ and $z^N \in V(N)$ verifying that $\lambda^t z^N \neq 0$. If $x = (x^1, \ldots, x^n) \in C(N, v_\lambda^{z^N})$, then $X \in C(N, V; \not\leq)$, where $X^i = \frac{x^i}{\lambda^t z^N} z^N, \ \ \forall\, i \in N$.

The proof of this result can be seen in [17], but we include it for its applicability in searching for core allocations in multicriteria games.

Proof. Let $\lambda \in \Lambda$ and let z^N be a vector in $V(N)$ such that $\lambda^t z^N \neq 0$ and $x \in C(N, v_\lambda^{z^N})$.

Now consider the matrix $X \in \mathbb{R}^{k\times n}$ whose columns are:

$$X^i = \frac{x^i}{\lambda^t z^N} z^N \quad \forall\, i \in N.$$

Let us prove first that $X \in I^*(N, V)$.

$$X^N = \sum_{i=1}^{n} \frac{x^i}{v_\lambda^{z^N}(N)} z^N = z^N \Rightarrow X \in I^*(N, V). \tag{2.70}$$

Let us prove now that X is an allocation in the core. If $X \notin C(N, V; \not\leq)$, then there would exist a coalition $S \subset N$ and a vector $\widehat{z}^S \in V(S)$ such that $X_j^S \leq \widehat{z}_j^S$, $j = 1, 2, \ldots, k$, $X^S \neq \hat{z}^S$. Therefore $\lambda^t X^S < \lambda^t \widehat{z}^S$ and the following chain of inequalities

(2.71)

$$\max_{z^S \in V_S} \lambda^t z^S \geq \lambda^t \widehat{z}^S > \lambda^t X^S = \sum_{i\in S} \lambda^t X^i = \frac{\sum_{i\in S} x^i}{\lambda^t z^N} \lambda^t z^N = x^S \geq v_\lambda^{z^N}(S) = \max_{z^S \in V_S} \lambda^t z^S,$$

would hold, which is a contradiction. □

The following result for the existence of the dominance core is automatically derived from the previous theorem.

Corollary 2.4.1 $\exists\, \lambda \in \Lambda,\ z^N \in V(N) : C(N, v_\lambda^{z^N}) \neq \emptyset \Rightarrow C(N, V; \not\leq) \neq \emptyset$.

Example 2.4.2 *From Example 2.4.1, consider the vector $z^N = (3,5)^t \in V(N)$ and $\lambda = (0.2, 0.8)$. The scalar game $(N, v_\lambda^{z^N})$ is:*

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	N
$v_\lambda^{z^N}(S)$		0.8		2		2	4.6

As $x = (1, 2, 1.6)$ *is an allocation in* $C(N, v_\lambda^{z^N})$*, the matrix*

$$X = \begin{pmatrix} \frac{15}{23} & \frac{30}{23} & \frac{24}{23} \\ \frac{25}{23} & \frac{50}{23} & \frac{40}{23} \end{pmatrix}, \tag{2.72}$$

constructed as in the previous theorem, is a dominance core allocation for the multicriteria combinatorial game.

In some situations the order assumed by the players could be stronger than the dominance. In such situations the coalitions would only accept payoffs that are at least as high as what they can obtain by themselves in every coordinate.

So, given $x, y \in \mathbb{R}^k$, we say that

$$x \succeq y \Leftrightarrow x^i \geq y^i \ \forall\, i = 1, \ldots, k. \tag{2.73}$$

Under such conditions, we define the *preference core* as follows:

Definition 2.4.4 (Preference core) *The preference core of a combinatorial multicriteria game* (N, V)*, denoted by* $C(N, V; \succeq)$*, is the set of allocations* $X \in I^*(N, V)$ *such that* $X^S \succeq z^S \ \forall\, z^S \in V(S)$.

Analogously to the dominance core, an existence theorem for the preference core can be stated from the associated scalar games.

Theorem 2.4.2 Let $\lambda_1, \ldots, \lambda_k$ be the extreme points of Λ. One has that

$$C(N, V; \succeq) \neq \emptyset \Leftrightarrow \exists\, z^N \in V(N) \ : \ C(N, v_{\lambda_i}^{z^N}) \neq \emptyset \ \forall\, i = 1, \ldots, k. \tag{2.74}$$

Proof. See [17] □

Once cooperative games have been introduced, in the following chapter we will study possible ways of allocating the benefits generated by the grand coalition in the class of Linear Production Games.

Chapter 3

New allocations in Linear Production Games

In Section 2.3.1 the class of Linear Production Games was introduced. Now a natural question arises: how to divide the profit made by the grand coalition among the players. The following sections present two ways of doing so. The first one is the so-called Owen set, see [47]. The second one is a new allocation, specifically created for linear production games in this thesis, named Extended Owen set.

3.1 The Owen set

In this section we describe a well-known solution concept for LP games: the Owen set. Although it will be proven that the Owen set consists of core allocation, several questions on the fairness of such allocations are proposed. To begin with, let us introduce some notation that will be useful in the following.

Definition 3.1.1 *Let $(A, B, c) \in \mathcal{L}$. The feasible regions of problems $P(N)$ and $D(N)$, see equations 2.36 and 2.37 in Section 2.3.1, are denoted by*

$$\begin{aligned} F_{\max}(A, B, c) &:= \{x \in \mathbb{R}^p_+ : Ax \leq b\} \\ F_{\min}(A, B, c) &:= \{y \in \mathbb{R}^n_+ : yA \geq c\} \end{aligned} \tag{3.1}$$

respectively. The optimal values of problems $P(N)$ and $D(N)$ are denoted by

$$(3.2)\qquad \begin{aligned} v_{\max}(A,B,c) &:= \max\{cx : x \in F_{\max}(A,B,c)\} \\ v_{\min}(A,B,c) &:= \min\{yb : y \in F_{\min}(A,B,c)\} \end{aligned}$$

respectively, and the set of optimal solutions to $P(N)$ and $D(N)$ by

$$(3.3)\qquad \begin{aligned} O_{\max}(A,B,c) &:= \{x \in F_{\max}(A,B,c) : cx = v_{\max}(A,B,c)\} \\ O_{\min}(A,B,c) &:= \{y \in F_{\min}(A,B,c) : yb = v_{\min}(A,B,c)\} \end{aligned}$$

In the rest of the chapter we will search for ways of dividing the profit generated by the common action of all players, $v(N)$, among the agents: the so-called *solution rules.*

Definition 3.1.2 *A solution rule φ on $\mathcal{L}$ is a map assigning to every game $(A,B,c) \in \mathcal{L}$ a subset of $\mathbb{R}^n$. Each member of such subset is an* allocation.

A well-known solution rule for cooperative games is the core, see Section 2.2.2. Every allocation in the core of a game distributes the general benefit among the players in such a way that no group of players can obtain a better payoff by acting separately from the rest of players. Mathematically, for every game (N,v) its core is defined as the set

$$(3.4)\qquad Core(N,v) = \{x \in \mathbb{R}^n : x(N) = v(N), x(S) \geq v(S)\ \forall\ S \subset N\},$$

where $x(S) = \sum_{i \in S} x_i$ for all S. One well accepted solution rule specific for linear production games is that introduced by Owen [47], and named *Owen set.*

Definition 3.1.3 *Let $(A,B,c) \in \mathcal{L}$. The Owen set of (A,B,c) is*

$$(3.5)\qquad Owen(A,B,c) := \{yB : y \in O_{\min}(A,B,c)\}.$$

In [47] it is proven that for every $(A,B,c) \in \mathcal{L}$, $Owen(A,B,c) \subset Core(A,B,c)$. That is to say, given an allocation in the Owen set of a LP game, no coalition of players can obtain a better payoff by acting on their own than the payoff they receive from this allocation. The following example shows how to calculate the Owen set of a given LP game.

Example 3.1.1 *Consider the game $(A,B,c) \in \mathcal{L}$ where*

$$(3.6) \qquad A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \; c = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

The corresponding dual problem $D(N)$ is

$$(3.7) \qquad \begin{array}{rll} \min & 2y_1 + 4y_2 + y_3 & \\ s.t.: & y_1 + y_2 & \geq 1 \\ & \quad y_2 + y_3 & \geq 2 \\ & y_1, y_2, y_3 \geq 0 & \end{array}$$

It can be checked that $O_{\min}(A, B, c) = \{(1, 0, 2)\}$. So, $Owen(A, B, c) = \{(1, 0, 2)B\} = \{(3, 0, 1)\}$.

This allocation is in the core of the game but, is it a "fair" allocation? Note that player 2 receives nothing for this allocation but, without his resources, players 1 and 3 cannot make any benefit at all. So, the Owen allocation gives null payoff to players that are absolutely necessary to obtain the maximum benefit $v(N)$.

What happened in Example 3.1.1 is a general drawback of linear production games. It comes from the fact that, by the Complementary Slackness theorem, see [4], if there is some surplus of resource i in an optimal solution $x^* \in O_{\max}(A, B, c)$ (meaning $(Ax^*)_i < b_i$), then $y_i^* = 0 \; \forall \; y \in O_{\min}(A, B, c)$. This means that only players owning resources that generate no surplus have the chance of receiving a positive payoff from Owen allocations. This fact could make players get rid off their surplus so that the corresponding dual variables are not forced to be null and they have the possibility of receiving a positive reward from allocations in the Owen set.

The following section presents a new solution concept on LP games that avoids the drawbacks mentioned in the previous example.

3.2 The Extended Owen set

In this section a new allocation for linear production games is presented, named the Extended Owen set (EOwen for short). It is based on the idea that players owning resources that produce surplus in the optimal production plan can get rid of them. Let $(A, B, c) \in \mathcal{L}$ and $x^* \in O_{\max}(A, B, c)$ one solution to the corresponding problem $P(N)$. The coordinates of x^* define the amount of consumption goods to be produced. Now consider the linear production game in which each player k reduces the amount of its resource i so that the total amount of this resource owned by all agents is $(Ax^*)_i$.

For any optimal solution x, let B^x_{ik} be the updated amount of resource i owned by agent k, satisfying that $0 \leq B^x_{ik} \leq B_{ik}$ and $\sum_{k=1}^n B^x_{ik} = (Ax)_i$. Consider the vector $b^x \in \mathbb{R}^n$ where $b^x_i = (Ax)_i = \sum_{k=1}^n B^x_{ik}$.

So, for every $x \in O_{\max}(A, B, c)$ a new linear production game $(A, B^x, c) \in \mathcal{L}$ is defined. Its corresponding problems $P^x(S)$ and $D^x(S)$ are:

$$
\begin{array}{lll}
\max & cx & \\
\text{s.t.:} & Ax \leq B^x e_S & P^x(S) \\
 & x \geq 0 &
\end{array}
\tag{3.8}
$$

$$
\begin{array}{lll}
\min & yB^x e_S & \\
\text{s.t.:} & yA \geq c & D^x(S) \\
 & y \geq 0 &
\end{array}
\tag{3.9}
$$

where $(e_S)_k = 1 \ \forall \ k \in S$ and zero otherwise.

Definition 3.2.1 *Given a LP game (A, B, c) and $x \in O_{\max}(A, B, c)$, the set of all possible reduced resource-player matrices associated to x is*

$$
\mathcal{B}(A, B, x) = \{B^x : 0 \leq B^x_{ik} \leq B_{ik}, \sum_{k=1}^{n} B^x_{ik} = (Ax)_i, \forall \ i \in R, k \in N\}. \tag{3.10}
$$

The LP game (A, B^x, c) is called the reduced game of (A, B, c) associated to x.

Building reduced resource-player matrices

Note that $b^x = B^x e_N$ and that for a given $x \in O_{\max}(A, B, c)$, there might be infinitely many ways of obtaining matrix B^x as defined before. The problem of building an appropriate matrix in $\mathcal{B}(A, B, x)$ deserves a whole research by itself. One logical way to do so is by minimizing the total cost. Suppose that player k has to pay r_{ik} monetary units for each unit of resource i, for all $k \in N,\ i \in R$. Thus, B^x is given by one optimal solution to

$$
\text{(3.11)} \qquad \begin{aligned} \min \ & \sum_{i=1}^{r}\sum_{k=1}^{n} r_{ik} B^x_{ik} \\ \text{s.t.:} \ & \sum_{k=1}^{n} B^x_{ik} = b^x_i \quad \forall\, i \in R \\ & 0 \leq B^x_{ik} \leq B_{ik} \ \forall\, i \in R, k \in N. \end{aligned}
$$

Due to the fact that the previous linear program is separable in r different knapsack problems in their continuous version, it is easy to check that one optimal solution to this problem can be calculated as follows.

For any resource $i \in R$, the player that can buy resource i at the cheapest price keeps all his resource. Later, the player buying i at the second cheapest price remains with all his resource. Continue this process until we have reached b^x_i units of resource i. The other players have to get rid of their amount of resource i.

Nevertheless, it is usual that all players buy any given resource at the same price. In such case we build B^x following the proportional rule, that is, all players get rid of their resources in equal proportions. Thus, one has that

$$
\text{(3.12)} \qquad B^x_{ik} = \frac{b^x_i}{b_i} B_{ik}.
$$

Let us see that B^x built this way is a reduced resource-player matrix associated to x.

$$
\text{(3.13)} \qquad \begin{aligned} &1) \ \ 0 \leq B^x_{ik} \leq B_{ik} \text{ trivial} \\ &2) \ \ \sum_{k=1}^{n} B^x_{ik} = \sum_{k=1}^{n} \frac{b^x_i}{b_i} B_{ik} = \frac{b^x_i}{b_i} \sum_{k=1}^{n} B_{ik} = \frac{b^x_i}{b_i} b_i = b^x_i = (Ax)_i. \end{aligned}
$$

In the rest of the section, we will assume for any (A, B, c) that B^x is one reduced resource-player matrix of $\mathcal{B}(A, B, x)$.

Basic results

Before going on, we present some technical results that will be useful in the rest of the section.

Lemma 3.2.1 Let $(A, B, c) \in \mathcal{L}$. Then

1. $v_{\max}(A, B, c) = v_{\max}(A, B^x, c)\ \forall\ x \in O_{\max}(A, B, c)$.
2. $v_{\min}(A, B, c) = v_{\min}(A, B^x, c)\ \forall\ x \in O_{\max}(A, B, c)$.
3. $O_{\max}(A, B, c) = \bigcup_{x \in O_{\max}(A,B,c)} O_{\max}(A, B^x, c)$.
4. $O_{\min}(A, B, c) \subset \bigcap_{x \in O_{\max}(A,B,c)} O_{\min}(A, B^x, c)$.

Proof.

1. Trivial, since x is solution to $P^x(N)\ \forall\ x \in O_{\max}(A, B, c)$.
2. Trivial from the Complementary Slackness theorem. Take $xO_{\max}(A, B, c)$. If $y \in O_{\min}(A, B, c)$ one has that $y_i = 0\ \forall\ i : (Ax)_i < b_i$. Therefore,

   $$y_i \neq 0 \Rightarrow b_i^x = b_i \Rightarrow by = b^x y. \tag{3.14}$$

   Since $v_{\max}(A, B, c) = v_{\max}(A, B^x, c)$, the result follows.
3. First consider $\widehat{x} \in O_{\max}(A, B, c)$. Trivially $\widehat{x} \in O_{\max}(A, B^{\widehat{x}}, c)$. Then

   $$\widehat{x} \in \bigcup_{x^* \in O_{\max}(A,B,c)} O_{\max}(A, B^{x^*}, c). \tag{3.15}$$

Now consider $\widehat{x} \in \bigcup_{x^* \in O_{\max}(A,B,c)} O_{\max}(A, B^{x^*}, c)$. Then, there exists x^* such that $\widehat{x} \in O_{\max}(A, B^{x^*}, c)$. Thus

$$\left.\begin{array}{l} A\widehat{x} \leq b^{x^*} = Ax^* \leq b \\ \widehat{x} \geq 0 \\ c\widehat{x} = v_{\max}(A, B^{x^*}, c) = v_{\max}(A, B, c) \end{array}\right\} \Rightarrow \widehat{x} \in O_{\max}(A, B, c). \tag{3.16}$$

4. Let $\widehat{y} \in O_{\min}(A, B, c)$ and $\widehat{x} \in O_{\max}(A, B, c)$. Applying the Complementary Slackness theorem and the Duality theorem, we have that

$$\widehat{y}b^{\widehat{x}} = \widehat{y}b = c\widehat{x} = v_{\max}(A, B, c) = v_{\max}(A, B^{\widehat{x}}, c) = v_{\min}(A, B^{\widehat{x}}, c). \tag{3.17}$$

Trivially $\widehat{y} \in F_{\min}(A, B^{\widehat{x}}, c)$, since problems D and $D^{\widehat{x}}$ have the same constraints. Thus, we conclude that $\widehat{y} \in O_{\min}(A, B^{\widehat{x}}, c)$ and the result follows.

□

It often happens that there is only one optimal production plan for a given LP process (A, B, c), that is, $O_{\max}(A, B, c) = \{x\}$. In the rest of the section we will focus on such subclass of $\mathcal{L}$.

Definition 3.2.2 *$\widehat{\mathcal{L}}$ is the subclass of* $\widehat{\mathcal{L}} = \{(A, B, c) \in \mathcal{L} : O_{\max}(A, B, c) = \{x\}\}$.

The Extended Owen Set

From the previous definitions, a new solution rule for our linear production games is presented. It is based on the idea of not taking into account the surplus generated. Later on we calculate allocations of the (only) reduced game.

Definition 3.2.3 *Let $(A, B, c) \in \widehat{\mathcal{L}}$. The Extended Owen set of (A, B, c) is*

$$EOwen(A, B, c) = \{y^x B^x : y^x \in O_{\min}(A, B^x, c), \{x\} = O_{\max}(A, B, c)\}. \tag{3.18}$$

A direct consequence of this definition gives us an alternative definition of the Extended Owen set.

Proposition 3.2.1 *EOwen*$(A,B,c) = Owen(A,B^x,c)$ for all $(A,B,c) \in \widehat{\mathcal{L}}$, where $\{x\} = O_{\max}(A,B,c)$.

Proof. Let $(A,B,c) \in \widehat{\mathcal{L}}$ and $\{x\} = O_{\max}(A,B,c)$. By the definitions of the Extended Owen set and the Owen set one has

$$EOwen(A,B,c) = \{yB^x : y \in O_{min}(A,B^x,c)\} = Owen(A,B^x,c). \tag{3.19}$$

□

The following proposition proves that the name Extended Owen is meaningful, as the Extended Owen set of our linear production games contains the classical Owen set.

Proposition 3.2.2 Let $(A,B,c) \in \widehat{\mathcal{L}}$. Then

$$Owen(A,B,c) \subset EOwen(A,B,c). \tag{3.20}$$

Proof. Let $(A,B,c) \in \widehat{\mathcal{L}}$ and $\gamma \in Owen(A,B,c)$. Then there exists $\widehat{y} \in O_{\min}(A,B,c)$ such that $\gamma = \widehat{y}B$. By Lemma 3.2.1, we deduce that $\widehat{y} \in O_{\min}(A,B^x,c)$, where $\{x\} = O_{\max}(A,B,c)$.

By the Complementary Slackness theorem, $(Ax)_i < b_i = \sum_{k=1}^n B_{ik} \Rightarrow \widehat{y}_i = 0$. Then, if $\widehat{y}_i \neq 0 \Rightarrow (Ax)_i = b_i = \sum_{k=1}^n B_{ik} \Rightarrow B^x_{ik} = B_{ik}$. Thus we have that

$$\gamma_k = \sum_{i=1}^r \widehat{y}_i B_{ik} = \sum_{i:\widehat{y}_i \neq 0} \widehat{y}_i B_{ik} = \sum_{i:\widehat{y}_i \neq 0} \widehat{y}_i B^x_{ik} = \sum_{i=1}^r \widehat{y}_i B^x_{ik}, \tag{3.21}$$

which implies that $\gamma = \widehat{y}B^x$. Then $\gamma \in EOwen(A,B,c)$. □

The following example shows that in general the Owen set is a "proper" subset of the Extended Owen set, that is, there are allocations in *EOwen* that are not in the Owen set.

Example 3.2.1 *Consider the linear production game (N, v) arising from the linear production process (A, B, c) where*

(3.22)
$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix}, c = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Equation 3.23 shows the linear programming problems corresponding to the game (A, B, c). One can check that $O_{\max}(A, B, c) = \{x = (1, 1)\}$.

(3.23)
$$\begin{array}{ll} \max & 2x_1 + x_2 \\ s.t. & x_1 \leq 1 \\ & x_2 \leq 1 \\ & x_1 + 2x_2 \leq 4 \\ & x_1, x_2 \geq 0 \end{array} \qquad \begin{array}{ll} \min & y_1 + y_2 + 4y_3 \\ s.t.: & y_1 + y_3 \geq 2 \\ & y_2 + 2y_3 \geq 1 \\ & y_1, y_2, y_3 \geq 0 \end{array}$$

Consider now the reduced game (A, B^x, c) associated to x, where

(3.24)
$$B^x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$

It is easy to see that $y^x = (3/2, 0, 1/2) \in O_{\min}(A, B^x, c)$. In Equation 3.25 one can see the linear programming problems associated to (A, B^x, c).

(3.25)
$$\begin{array}{ll} \max & 2x_1 + x_2 \\ s.t. & x_1 \leq 1 \\ & x_2 \leq 1 \\ & x_1 + 2x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{array} \qquad \begin{array}{ll} \min & y_1 + y_2 + 3y_3 \\ s.t.: & y_1 + y_3 \geq 2 \\ & y_2 + 2y_3 \geq 1 \\ & y_1, y_2, y_3 \geq 0 \end{array}$$

From y^x, we deduce that

(3.26)
$$\alpha = y^x B^x = (3/2, 0, 3/2) \in EOwen(A, B, c).$$

One may check that α *cannot be an Owen allocation for* (A,B,c)*, as* $\widehat{y}_3 = 0 \ \forall \ \widehat{y} \in O_{\min}(A,B,c)$ *and, therefore,* $\gamma_3 = 0 \ \forall \ \gamma \in Owen(A,B,c)$*. This concludes that the Owen set is a proper subset of the Extended Owen set.*

The following proposition proves that for the reduced games the Extended Owen Set and the Owen Set coincide.

Proposition 3.2.3 Let $(A,B,c) \in \widehat{\mathcal{L}}$ and $\{x\} = O_{\max}(A,B,c)$. Then one has that

$$EOwen(A,B^x,c) = Owen(A,B^x,c). \tag{3.27}$$

Proof. Let $(A,B,c) \in \widehat{\mathcal{L}}$ and $\{x^*\} = O_{\max}(A,B,c)$. It is easy to see that $O_{\max}(A,B^{x^*},c) = \{x^*\}$ (trivially $x^* \in O_{\max}(A,B^{x^*},c)$ and, if there were more solutions, then they would be in $O_{\max}(A,B^{x^*},c)$, which is a contradiction because $(A,B,c) \in \widehat{\mathcal{L}}$.) Therefore $(A,B^{x^*},c) \in \widehat{\mathcal{L}}$. It is clear that $(B^{x^*})^{x^*} = B^{x^*}$.

Thus, one has that

$$\begin{aligned}
& EOwen(A,B^{x^*},c) \\
= \ & \{\widehat{y}(B^{x^*})^{\widehat{x}} : \widehat{y} \in O_{\min}(A,(B^{x^*})^{\widehat{x}},c), \{\widehat{x}\} = O_{\max}(A,B^{x^*},c)\} \\
= \ & \{\widehat{y}(B^{x^*})^{\widehat{x}} : \widehat{y} \in O_{\min}(A,(B^{x^*})^{x^*},c)\} \\
= \ & \{\widehat{y}B^{x^*} : \widehat{y} \in O_{\min}(A,B^{x^*},c)\} \\
= \ & \{\widehat{y}B^{x^*} : \widehat{y} \in O_{\min}(A,B^{x^*},c)\} = Owen(A,B^{x^*},c).
\end{aligned} \tag{3.28}$$

□

As a corollary to this proposition one can state that in linear production games where all the resources are completely used, the Extended Owen Set coincides with the Owen Set.

The following proposition proves that allocations in the Extended Owen set distribute exactly $v(N)$ among the players.

Proposition 3.2.4 Let $(A,B,c) \in \widehat{\mathcal{L}}$ and $\gamma \in EOwen(A,B,c)$. Then γ is efficient.

Proof. Let $\gamma \in EOwen(A,B,c)$ then there exists $\widehat{y} \in O_{\min}(A,B^x,c)$, where $\{x\} = O_{\max}(A,B,c)$, such that $\gamma_k = \sum_{i=1}^r \widehat{y}_i B^x_{ik} \ \forall \ k = 1,\ldots,n$. We have that

$$\gamma(N) = \sum_{k=1}^{n} \gamma_k = \sum_{k=1}^{n}\sum_{i=1}^{r} \widehat{y}_i B^x_{ik} = \sum_{i=1}^{r} \widehat{y}_i \sum_{k=1}^{n} B^x_{ik} = \sum_{i=1}^{r} \widehat{y}_i b^x_i = \widehat{y} b^x. \tag{3.29}$$

Since $\widehat{y} \in O_{\min}(A, B^x, c)$ and $\{x\} = O_{\max}(A, B, c)$, we know that $\widehat{y}b^x = cx = v(N)$. That concludes that $\gamma(N) = v(N)$. □

The following example shows that allocations in the Owen Set might not be individually rational.

Example 3.2.2 *Consider the LP game (A, B, c) where*

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1.5 & 0 & 0.5 \\ 3 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, c = \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \tag{3.30}$$

The problem $P(N)$ corresponding to (A, B, c) is

$$\begin{aligned} \max \quad & x_1 + 2x_2 \\ s.t.: \quad & x_1 \leq 2 \\ & x_1 + x_2 \leq 4 \\ & x_2 \leq 1 \\ & x_i \geq 0, i = 1, 2, 3 \end{aligned} \tag{3.31}$$

and one may check that $O_{\max}(A, B, c) = x^* = (2, 1)$. From this solution, consider the reduced game (A, B^{x^*}, c) where

$$B^{x^*} = \begin{pmatrix} 1.5 & 0 & 0.5 \\ 2.25 & 0.75 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{3.32}$$

calculated by the proportional rule.

The corresponding linear program $D^{x^*}(N)$ is:

$$\begin{aligned} \min \quad & 2y_1 + 3y_2 + y_3 \\ s.t.: \quad & y_1 + y_2 \geq 1 \\ & y_2 + y_3 \geq 2 \\ & y_i \geq 0, i = 1, 2, 3 \end{aligned} \tag{3.33}$$

One has that $y^{x^*} = (0, 1, 1) \in O_{\min}(A, B^{x^*}, c)$. From this vector, we obtain the allocation

$$\alpha = y^{x^*} B^{x^*} = (3.25, 0.75, 0) \in EOwen(A, B, c). \tag{3.34}$$

On the other hand, it is easy to see that $v(\{1\}) = 3.5$. Then we conclude that $\alpha_1 < v(\{1\})$, and therefore α is not individually rational.

We have given the definition of this new solution set for LP games and some if its properties. The following step will be to provide an axiomatic characterization of such solution set.

3.2.1 Axiomatic characterization

In this section we characterize the Extended Owen set in the class $\widehat{\mathcal{L}}$. Note that for the case in which $B^x = B$, that is to say, linear production games in which all the resources are used in the optimal production plan, *EOwen* and Owen coincide, see Proposition 3.2.3. Therefore we will use the same axioms that characterize the Owen set for LP games in [24]:

Definition 3.2.4 (Axiom 1) *A solution concept φ over $\widehat{\mathcal{L}}$ satisfies* one person efficiency *if $\varphi(A, e_R, c) = \{v_{\min}(A, e_R, c)\}$ for all $(A, e_R, c) \in \widehat{\mathcal{L}}$.*

One person efficiency says that if there is only one agent owning one unit of all resources, then the solution concept assigns to him the maximal profit that can be made from his resource bundle.

The second axiom demands that the solution concept remains invariant if the units in which the resources are measured change.

Definition 3.2.5 (Axiom 2) *A solution concept* φ *on* $\widehat{\mathcal{L}}$ *satisfies rescaling if*

(3.35) $\varphi(HA, HB, c) = \varphi(A, B, c)\ \forall\ H \in \mathbb{R}_+^{r \times r}$ *diagonal with positive diagonal entries,*

for all $(A, B, c) \in \widehat{\mathcal{L}}$.

The following property states that if the resources are shuffled among the agents, then the solution rule changes in the same way.

Definition 3.2.6 (Axiom 3) *A solution concept* φ *on* $\widehat{\mathcal{L}}$ *satisfies the property of* shuffle *if for all* $H \in \mathbb{R}_+{}^{n \times m}$ *with* $He_M = e_N$, *and for all* $(A, b, c) \in \widehat{\mathcal{L}}$, *the following equality holds:*

$$\varphi(A, B, c)H = \varphi(A, BH, c). \tag{3.36}$$

where $\varphi(A, B, c)H = \{\alpha H : \alpha \in \varphi(A, B, c)\}$.

The fourth axiom assumes that the players agree that the profit is divided according to a vector $\alpha \in \varphi(A, I_n, c)$. Afterwards player k takes his payoff α_k and leaves. Suppose that his resource can be used by the other agents for a price α_k per unit. Then, a solution rule satisfies consistency if the restriction of α to agents $N \setminus \{k\}$ is a solution to the reduced linear production game.

Definition 3.2.7 (Axiom 4) *A solution concept satisfies* consistency *if for all* $(A, I_N, c) \in \widehat{\mathcal{L}}$ *with* $r = n \geq 2$ *and for all* $\alpha \in \varphi(A, I_N, c)$ *one has that* $(A_{-k\bullet}, I_{N\setminus k}, \widetilde{c}) \in \widehat{\mathcal{L}}$ *and* $\alpha_{-k} \in \varphi(A_{-k\bullet}, I_{N\setminus i}, \widetilde{c})$ *for all* $k \in N$, *where* $\widetilde{c}_j = c_j - \alpha_k A_{kj}$ *for all* $j \in P$ *and* $A_{-k\bullet}$ *denotes the matrix resulting after deleting row* k *from matrix* A.

The consistency axiom has to do with the special case in which every player owns exactly one unit of exactly one resource, and different players own different resources. Therefore we can identify the set of resources with the set of players in an appropriate way, I_N denoting the identity matrix of dimension $n = r$.

The last axiom we will use to characterize the Extended Owen set is *deletion*. Deletion says that if a production technology is not needed to make the maximal profit $v(N)$, then we can delete this technology, the outcomes of the old situations being also outcomes in the new game.

Definition 3.2.8 (Axiom 5) *A solution concept φ over $\widehat{\mathcal{L}}$ satisfies deletion if for all $(A, I_N, c) \in \widehat{\mathcal{L}}$ and for all $J \subset P$ such that $v_{\min}(A_{\bullet -J}, I_N, c_{-J}) = v_{\min}(A, I_N, c)$, then $\varphi(A, I_N, c) \subset \varphi(A_{\bullet -J}, I_N, c_{-J})$, where $A_{\bullet -J}$ denotes the matrix resulting after deleting column j, $\forall\, j \in J$, from matrix A.*

After those axioms, we characterize the Extended Owen set as follows.

Theorem 3.2.1 If φ satisfies one person efficiency, rescaling, shuffle, consistency and deletion over the reduced game (A, B^x, c), where $\{x\} = O_{\max}(A, B, c)$, then $\varphi(A, B, c) = EOwen(A, B, c)$ for all $(A, B, c) \in \widehat{\mathcal{L}}$.

Proof. Take $(A, B, c) \in \widehat{\mathcal{L}}$, and suppose that $\varphi(A, B, c)$ satisfies the Axiom 1-5 in (A, B^x, c). Then, see [24], $\varphi(A, B, c) = Owen(A, B^x, c)$. Since $EOwen(A, B, c) = Owen(A, B^x, c)$, see Proposition 3.2.1, the result follows. □

3.2.2 Extended Owen set and independent agents

The case in which matrix B is square and diagonal, that is, when each player owns only one resource and different players own different resources, is special. Three main characteristics of the Extended Owen allocations must be underlined for this subclass of linear production games:

1. For all $x \in O_{\max}(A, B, c)$, $\mathcal{B}(A, B, x)$ consists of only one matrix. Such matrix is obtained when all players get rid of their corresponding resource until no more surplus is produced.

2. Allocations in the Owen Set are individually rational.

   Proposition 3.2.5 Let $(A, B, c) \in \widehat{\mathcal{L}}$ such that $B_{ik} = 0\ \forall\ i \neq k$ (each resource is owned by only one player, and each player owns only one resource). The allocations in $EOwen(A, b, c)$ are individually rational.

Proof. Let $(A,B,c) \in \widehat{\mathcal{L}}$ under the conditions of the proposition, and $\gamma \in EOwen(A,B,c)$. We have to prove that $\gamma_k \geq v(\{k\})$. Consider the two possible cases:

- $v(\{k\}) = 0$. The result is trivial.
- $v(\{k\}) > 0$. Then there exists a consumption good j that can be produced only from resource k, that is, $A_{kj} \neq 0$ and $A_{ij} = 0 \ \forall\ i \neq k$. Therefore $v(\{k\}) = \frac{B_{kk}}{A_{kj}} c_j$.

  Since $\gamma \in EOwen(A,B,c)$ there exist $\widehat{y} \in O_{\min}(A, B^{x^*}, c)$, where $\{x^*\} = O_{\max}(A,B,c)$, such that $\gamma_k = \widehat{y}_k B^{x^*}_{kk}$. It is easy to check that $B^{x^*}_{kk} = B_{kk}$ (because there can be no surplus of resource k in an optimal solution, as such resource can always be used to produce product j and generate benefit).

  We have that

  $$\widehat{y}A \geq c \Rightarrow \sum_{i=1}^{n} \widehat{y}_i A_{ij} \geq c_j \Rightarrow \widehat{y}_k A_{kj} \geq c_j \Rightarrow \widehat{y}_k \geq \frac{c_j}{A_{kj}}. \tag{3.37}$$

  On the other hand,

  $$\gamma_k = \widehat{y}_k B^{x^*}_{kk} = \widehat{y}_k B_{kk} \geq \frac{c_j}{A_{kj}} B_{kk} = v(\{k\}). \tag{3.38}$$

□

3. If all resources are necessary to produce each product, we have that allocations in the Extended Owen set are core allocations. This comes from the fact that in such games $v(S) = 0$, and trivially $\alpha \geq 0$ for every Extended Owen allocation α.

3.2.3 Multiple production plans and the Extended Owen solution

It might happen that a linear production problem has more than one optimal production plan. In such case, the definition of the Extended Owen set must be broadened. There are different possibilities to tackle this problem. Consider a linear programming game (A, B, c) and $\{x^1, \ldots, x^w\}$ the extreme optimal solutions to the corresponding $P(N)$.

Extension 1

The first approach to define the Extended Owen solution in the complete class of linear production games is based on the idea of finding allocations belonging to the Owen sets of all reduced games associated to extreme optimal solutions to $P(N)$. Thus, a formal definition of such solution concept is the following:

Definition 3.2.9 *Let $(A, B, c) \in \mathcal{L}$ and $\{x^1, \ldots, x^w\}$ the extreme optimal solutions to $P(N)$. The Extended Owen 1 set of (A, B, c) denoted by $EOwen1(A, B, c)$ is the set*

$$EOwen1(A, B, c) = \bigcap_{t=1}^{w} Owen(A, B^{x^t}, c). \tag{3.39}$$

The first consequence of the definition is that the name given to this solution concept is meaningful, that is to say, allocations belonging to the Owen set of a given linear production game are also allocations in the Extended Owen 1 set.

Proposition 3.2.6 Let $(A, B, c) \in \mathcal{L}$. Then one has that

$$Owen(A, B, c) \subset EOwen1(A, B, c). \tag{3.40}$$

Proof. Let $(A, B, c) \in \mathcal{L}$ and $\{x^1, \ldots, x^w\}$ the extreme optimal solutions to the corresponding $P(N)$. Consider $\alpha \in Owen(A, B, c)$. By definition, we have that

$$\exists\ y \in O_{\min}(A, B, c) : \alpha = yB. \tag{3.41}$$

By Lemma 3.2.1 we deduce that

$$\exists\ y \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c) : \alpha = yB. \tag{3.42}$$

Let x^* be an extreme solution. It must be proven that $\alpha \in Owen(A, B^{x^*}, c)$. Trivially we have that $y \in Owen(A, B^{x^*}, c)$. Applying that $y_i \neq 0$ implies $(Ax^*)_i = b_i$, having this way that $B_{ik}^{x^*} = B_{ik}\ \forall\ k$, one has

$$\alpha_k = \sum_{i=1}^{r} y_i B_{ik} = \sum_{i:y_i \neq 0} y_i B_{ik} = \sum_{i:y_i \neq 0} y_i B_{ik}^{x^*} = \sum_{i=1}^{r} y_i B_{ik}^{x^*}\ \forall\ k = 1, ..., n. \tag{3.43}$$

Therefore, $\alpha = yB^{x^*}$ which implies that $\alpha \in Owen(A, B^{x^*}, c)$ and the result follows. □

The following example shows that in general the Owen set is a proper subset of the Extended Owen 1 set.

Example 3.2.3 *Consider the LP game (A, B, c) with the following data:*

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{3.44}$$

The corresponding problems $P(N)$ and $D(N)$ are

$$\begin{array}{lll} \max & x_1 + x_2 & \\ s.t.: & x_1 \leq 2 & \\ & x_1 + x_2 \leq 3 & P(N) \\ & x_2 \leq 2 & \\ & x_1, x_2 \geq 0 & \end{array} \qquad \begin{array}{lll} \max & 2y_1 + 3y_2 + 2y_3 & \\ s.t.: & y_1 + y_2 \geq 1 & D(N) \\ & y_2 + y_3 \geq 1 & \\ & y_1, y_2, y_3 \geq 0 & \end{array} \tag{3.45}$$

One has that the extreme optimal solutions to $P(N)$ are $\{x^1 = (1,2), x^2 = (2,1)\}$ and that $O_{\min}(A, B, c) = \{y = (0,1,0)\}$. Therefore, $Owen(A, B, c) = yB = \{\alpha = (0,3,0)\}$.

Consider now the two reduced games arising from the optimal solutions to $P(N)$, (A, B^{x^1}, c) and (A, B^{x^2}, c) where B^{x^1} and B^{x^2} are calculated following the proportional rule, that is,

$$B^{x^1} = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad B^{x^2} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix}. \tag{3.46}$$

The corresponding problems $D^{x^t}(N),\ t = 1, 2$ *are*

$$\begin{array}{llcll} \max & y_1 + 3y_2 + 2y_3 & & \max & 2y_1 + 3y_2 + y_3 \\ s.t.: & y_1 + y_2 \geq 1 & D^{x^1}(N) & s.t.: & y_1 + y_2 \geq 1 \quad D^{x^2}(N) \\ & y_2 + y_3 \geq 1 & & & y_2 + y_3 \geq 1 \\ & y_1, y_2, y_3 \geq 0 & & & y_1, y_2, y_3 \geq 0 \end{array} \tag{3.47}$$

It is easy to see that $y^* = (1, 0, 1) \in O_{\min}(A, B^{x^t}, c)\ t = 1, 2$. *Therefore, we have that:*

$$y^* B^{x^1} = (1.5, 0, 1.5) \in Owen(A, B^{x^1}, c),\ y^* B^{x^2} = (1.5, 0, 1.5) \in Owen(A, B^{x^2}, c). \tag{3.48}$$

Then, by definition, $(1.5, 0, 1.5) \in EOwen1(A, B, c)$. *But we know that* $Owen(A, B, c)$ *consists only of the allocation* $(0, 3, 0)$. *Then we conclude that* $Owen(A, B, c) \subsetneq EOwen1(A, B, c)$.

The following result shows that the Extended Owen 1 set and the Extended Owen set given in previous sections coincide on the class $\widehat{\mathcal{L}}$.

Proposition 3.2.7 Let $(A, B, c) \in \widehat{\mathcal{L}}$. Then one has that

$$EOwen1(A, B, c) = EOwen(A, B, c). \tag{3.49}$$

Proof. Let $(A, B, c) \in \widehat{\mathcal{L}}$, then $O_{\max}(A, B, c) = \{x^*\}$. Therefore, applying the definition of EOwen1 and Proposition 3.2.1,

$$EOwen1(A, B, c) = Owen(A, B^{x^*}, c) = EOwen(A, B, c). \tag{3.50}$$

□

Now we prove that allocations in the Extended Owen 1 set are efficient, that is, they distribute all the benefit made by the grand coalition.

Proposition 3.2.8 Let $(A,B,c) \in \mathcal{L}$ and $\alpha \in EOwen1(A,B,c)$. Then α is efficient.

Proof. Let $(A,B,c) \in \mathcal{L}$ and $\alpha \in EOwen1(A,B,c)$. Assume that $\{x^1,\ldots,x^w\}$ are the extreme optimal solutions to $P(N)$. Then we have that $\alpha \in Owen(A,B^{x^t},c)\ \forall\ t = 1,\ldots,w$. In particular, let $x^* \in \{x^1,\ldots,x^w\}$, we have that $\alpha \in Owen(A,B^{x^*},c)$. Therefore, there exists $y \in O_{\min}(A,B^{x^*},c)$ such that $\alpha = yB^{x^*}$. Thus,

(3.51)
$$\alpha(N) = \sum_{k=1}^{n} \alpha_k = \sum_{k=1}^{n}\sum_{i=1}^{r} y_i B_{ik}^{x^*} = \sum_{i=1}^{r} y_i \sum_{k=1}^{n} B_{ik}^{x^*} = \sum_{i=1}^{r} y_i b_i^{x^*} = yb^{x^*} = cx^* = v(N).$$

And the result follows. □

Now another possible way of defining the Extended Owen set for the whole class of linear programming games is presented.

Extension 2

The second extension of the Extended Owen set for general linear programming games to be considered is based on the idea of choosing one optimal solution to $P(N)$ and a vector which is an optimal solution to all problems $D^{x^t}(N)$, $t = 1,\ldots,w$. By Proposition 3.2.1, we know that there exists such vector (at least the solutions to $D(N)$). In the rest of the section we will prove that such solution set strictly contains the Extended Owen 1 set, it is a natural extension of the Extended Owen set on $\widehat{\mathcal{L}}$ and consists of efficient allocations.

Definition 3.2.10 *Let $(A,B,c) \in \mathcal{L}$ and $\{x^1,\ldots,x^w\}$ the extreme optimal solutions to $P(N)$. The Extended Owen 2 solution of (A,B,c) is*

$$EOwen2(A,B,c) = \{\widehat{y}B^{\widehat{x}} : \widehat{y} \in \bigcap_{t=1}^{w} O_{\min}(A,B^{x^t},c),\ \widehat{x} \in O_{\max}(A,B,c)\}. \tag{3.52}$$

The following result shows that the Extended Owen 2 set contains the Extended Owen 1 set.

Proposition 3.2.9 Let $(A, B, c) \in \mathcal{L}$. Then one has that

$$EOwen1(A, B, c) \subset EOwen2(A, B, c). \tag{3.53}$$

Proof. Let $(A, B, c) \in \mathcal{L}$ and $\alpha \in EOwen1(A, B, c)$. Then

(3.54)
$$\alpha \in \bigcap_{t=1}^{w} Owen(A, B^{x^t}, c) \Rightarrow \alpha \in \bigcap_{t=1}^{w} \{y^t B^{x^t} : y^t \in O_{\min}(A, B^{x^t}, c)\}$$
$$\Rightarrow \exists\ y \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c) : \alpha = yB^{x^t}\ \forall\ t = 1, \ldots, w$$
$$\Rightarrow \exists\ y \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c),\ \exists\ x^* \in \{x^1, \ldots, x^w\} : \alpha = yB^{x^*} \Rightarrow \alpha \in EOwen2(A, B, c).$$

□

As a direct consequence of this proposition and Proposition 3.2.6, we have that the Owen set is also a subset of the Extended Owen 2 set.

Corollary 3.2.1 Let $(A, B, c) \in \mathcal{L}$. Then one has that

$$Owen(A, B, c) \subset EOwen2(A, B, c). \tag{3.55}$$

The following example shows that the Extended Owen 2 set and the Extended Owen 1 set do not coincide, that is, the Extended Owen 1 set is a proper subset of the Extended Owen 2 set.

Example 3.2.4 *Consider the LP game (A, B, c) where*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 2 & 1 \\ \frac{3}{2} & \frac{3}{2} & 0 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{3.56}$$

The corresponding problem $P(N)$ is

$$\begin{aligned} \max \quad & x_1 + x_2 + x_3 \\ s.t.: \quad & x_1 + x_2 + x_3 \le 4 \\ & x_1 + x_2 \le 3 \\ & x_1 + x_3 \le 3 \\ & x_2 + x_3 \le 3 \\ & x_i \ge 0 \;\; i = 1,2,3. \end{aligned} \qquad P(N) \tag{3.57}$$

the optimal extreme solutions to $P(N)$ being

$$x^1 = (2,1,1), \;\; x^2 = (1,2,1), \;\; x^3 = (1,1,2). \tag{3.58}$$

The corresponding reduced resource-player matrices B^{x^t}, calculated by the proportional rule, are

$$B^{x^1} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \; B^{x^2} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & \frac{4}{3} & \frac{2}{3} \\ \frac{3}{2} & \frac{3}{2} & 0 \end{pmatrix}, \; B^{x^3} = \begin{pmatrix} 2 & 1 & 1 \\ \frac{2}{3} & 0 & \frac{4}{3} \\ 0 & 2 & 1 \\ \frac{3}{2} & \frac{3}{2} & 0 \end{pmatrix}. \tag{3.59}$$

Let us now analyze the reduced games (A, B^{x^t}, c), $t = 1,2,3$. Their corresponding problems $D^{x^t}(N)$ are

$$\begin{aligned} \min \quad & 4y_1 + 3y_2 + 3y_3 + 2y_4 \\ s.t.: \quad & y_1 + y_2 + y_3 \ge 1 \\ & y_1 + y_2 + y_4 \ge 1 \\ & y_1 + y_3 + y_4 \ge 1 \\ & y_i \ge 0 \;\; i = 1,2,3,4. \end{aligned} \qquad D^{x^1}(N) \tag{3.60}$$

$$
\begin{array}{lll}
\min & 4y_1 + 3y_2 + 2y_3 + 3y_4 & \\
s.t.: & y_1 + y_2 + y_3 \geq 1 & \\
& y_1 + y_2 + y_4 \geq 1 & D^{x^2}(N) \\
& y_1 + y_3 + y_4 \geq 1 & \\
& y_i \geq 0 \; i = 1,2,3,4. &
\end{array} \tag{3.61}
$$

$$
\begin{array}{lll}
\min & 4y_1 + 2y_2 + 3y_3 + 3y_4 & \\
s.t.: & y_1 + y_2 + y_3 \geq 1 & \\
& y_1 + y_2 + y_4 \geq 1 & D^{x^3}(N) \\
& y_1 + y_3 + y_4 \geq 1 & \\
& y_i \geq 0 \; i = 1,2,3,4. &
\end{array} \tag{3.62}
$$

It is easy to see that $O_{\min}(A, B^{x^t}, c)$ *has two extreme points,* $\{(1,0,0,0),(0,\frac{1}{2},\frac{1}{2},\frac{1}{2})\}$ *for any* $t = 1,2,3$. *Therefore we have that the extreme points of the three Owen sets of the reduced games are*

$$
\begin{array}{l}
Owen(A, B^{x^1}, c) = \{y^1 B^{x^1} : y^1 \in O_{\min}(A, B^{x^1}, c)\} = \{(2,1,1),(1,\frac{3}{2},\frac{3}{2})\}. \\
Owen(A, B^{x^2}, c) = \{y^2 B^{x^2} : y^2 \in O_{\min}(A, B^{x^2}, c)\} = \{(2,1,1),(\frac{5}{4},\frac{17}{12},\frac{4}{3})\}. \\
Owen(A, B^{x^3}, c) = \{y^3 B^{x^3} : y^3 \in O_{\min}(A, B^{x^3}, c)\} = \{(2,1,1),(\frac{13}{12},\frac{7}{4},\frac{7}{6})\}.
\end{array} \tag{3.63}
$$

Now let us calculate an allocation in $EOwen2(A,B,c)$. *We know that* $y = (0,\frac{1}{2},\frac{1}{2},\frac{1}{2}) \in \bigcap_{t=1}^{3} O_{\min}(A, B^{x^t}, c)$ *and* x^1 *is an extreme optimal solution to* $P(N)$. *Therefore the allocation* $\alpha = yB^{x^1} = (1,\frac{3}{2},\frac{3}{2}) \in EOwen2(A,B,c)$. *Let us check that* $\alpha \notin EOwen1(A,B,c)$. *Otherwise, we would have that* $\alpha \in Owen(A, B^{x^2}, c)$. *Let us see that such assertion is a contradiction.*

$$
\begin{array}{l}
\alpha \in Owen(A, B^{x^2}, c) \Rightarrow \exists\, \beta \in [0,1] : \beta(2,1,1) + (1-\beta)(\frac{5}{4},\frac{17}{12},\frac{4}{3}) = \alpha \\
\Rightarrow \beta + \frac{4}{3}(1-\beta) = \frac{3}{2} \Rightarrow \beta = -\frac{-1}{2}!!!
\end{array} \tag{3.64}
$$

This concludes that $\alpha \notin Owen(A, B^{x^2}, c)$ *and, consequently,* $\alpha \notin \bigcap_{t=1}^{3} Owen(A, B^{x^t}, c) = EOwen1(A,B,c)$.

The next result states that the Extended Owen 2 set and the Extended Owen set coincide on the class $\widehat{\mathcal{L}}$.

Proposition 3.2.10 Let $(A, B, c) \in \widehat{\mathcal{L}}$. Then

$$EOwen2(A, B, c) = EOwen(A, B, c). \tag{3.65}$$

Proof. Let $(A, B, c) \in \widehat{\mathcal{L}}$. Then $w = 1$ and $O_{\max}(A, B, c) = \{x^1\}$. Therefore,

$$\begin{aligned} EOwen2(A, B, c) &= \{\widehat{y}B^{\widehat{x}} : \widehat{y} \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c),\ \widehat{x} \in O_{\max}(A, B, c)\} \\ &= \{yB^{x^1} : y \in O_{\min}(A, B^{x^1}, c)\} = Owen(A, B^{x^1}, c) = EOwen(A, B, c). \end{aligned} \tag{3.66}$$

□

Following the same reasonings we did for $EOwen1$, now we prove that allocations in the Extended Owen 2 set are efficient.

Proposition 3.2.11 Let $(A, B, c) \in \mathcal{L}$ and $\alpha \in EOwen2(A, B, c)$. Then $\alpha(N)$ is efficient.

Proof. Consider $(A, B, c) \in \mathcal{L}$ and $\alpha \in EOwen2(A, B, c)$. Then there exists $y \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c)$ and $x^* \in \{x^1, \ldots, x^w\}$ such that $\alpha = yB^{x^*}$. Therefore

$$\alpha(N) = \sum_{k=1}^{n} \alpha_k = \sum_{k=1}^{n} \sum_{i=1}^{r} y_i B_{ik}^{x^*} = \sum_{i=1}^{r} y_i \sum_{k=1}^{n} B_{ik}^{x^*} = \sum_{i=1}^{r} y_i b_i^{x^*} = yb^{x^*} = cx^* = v(N). \tag{3.67}$$

□

Extension 3

The third definition for the Extended Owen set in a general linear production game is based on the idea of joining the Owen sets of all reduced games associated to extreme optimal solutions to $P(N)$. Thus, a formal definition of such solution concept is:

Definition 3.2.11 *Let $(A, B, c) \in \mathcal{L}$ and $\{x^1, \dots, x^w\}$ the extreme optimal solutions to $P(N)$. The Extended Owen 3 set of (A, B, c) denoted by $EOwen3(A, B, c)$ is the set*

$$EOwen3(A, B, c) = \bigcup_{t=1}^{w} Owen(A, B^{x^t}, c). \tag{3.68}$$

As we did before, we firstly show that the two solution concepts aforementioned are included in the Extended Owen 3 set.

Proposition 3.2.12 Let $(A, B, c) \in \mathcal{L}$. Then

$$EOwen2(A, B, c) \subset EOwen3(A, B, c). \tag{3.69}$$

Proof. Let $(A, B, c) \in \mathcal{L}$ and $\alpha \in EOwen2(A, B, c)$. Then

$$\begin{aligned} &\exists\ y \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c),\ x^* \in \{x^1, \dots, x^w\} : \alpha = yB^{x^*} \\ &\Rightarrow \exists\ x^* \in \{x^1, \dots, x^w\},\ y \in O_{\min}(A, B^{x^*}, c) : \alpha = yB^{x^*} \\ &\Rightarrow \alpha \in Owen(A, B^{x^*}, c) \Rightarrow \alpha \in EOwen3(A, B, c). \end{aligned} \tag{3.70}$$

□

A trivial corollary to this proposition states that both the Owen set and the Extended Owen 2 set are subsets of the Extended Owen 3 set.

Corollary 3.2.2 Let $(A, B, c) \in \mathcal{L}$. Then

$$Owen(A, B, c) \subseteq EOwen3(A, B, c),\ EOwen1(A, B, c) \subsetneq EOwen3(A, B, c). \tag{3.71}$$

It is also interesting to see that the Extended Owen 2 set and the Extended Owen 3 set do not coincide, as shown in the following example. In other words, the Extended Owen 2 set is a proper subset of the Extended Owen 3 set.

Example 3.2.5 *Consider the LP game (A, B, c) where*

$$A = B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{3.72}$$

The corresponding problem $P(N)$ is

$$\begin{array}{rl} \max & x_1 + x_2 + x_3 \\ s.t.: & x_1 + x_2 + x_3 \leq 3 \\ & x_1 + x_2 \leq 2 \\ & x_2 + x_3 \leq 2 \\ & x_i \geq 0 \; i = 1, 2, 3 \end{array} \tag{3.73}$$

The optimal extreme solutions to $P(N)$ are $\{x^1 = (2, 0, 1), x^2 = (1, 0, 2), x^3 = (1, 1, 1)\}$. From those solutions, we obtain the corresponding reduced resource-player matrices:

$$B^{x^1} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix}, \quad B^{x^2} = \begin{pmatrix} 1 & 1 & 1 \\ 0.5 & 0.5 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad B^{x^3} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \tag{3.74}$$

Problem $D^{x^1}(N)$ is

$$\begin{array}{rl} \min & 3y_1 + 2y_2 + 2y_3 \\ s.t.: & y_1 + y_2 \geq 1 \\ & y_1 + y_2 + y3 \geq 1 \qquad D^{x^1}(N) \\ & y_1 + y_3 \geq 1 \\ & y_i \geq 0, \;\; i = 1, 2, 3. \end{array} \tag{3.75}$$

One has that $y^1 = (0, 1, 1)$ is an optimal solution to $D^{x^1}(N)$. Therefore, $\alpha^1 = y^1 B^{x^1} = (1, 1.5, 1.5) \in Owen(A, B^{x^1}, c) \Rightarrow \alpha^1 \in EOwen3(A, B^{x^1}, c)$.

One can see that problems $D^{x^2}(N)$ and $D^{x^3}(N)$ are

$$\begin{array}{llll} \min & 3y_1 + y_2 + 2y_3 & & \min \ 3y_1 + 2y_2 + 2y_3 \\ s.t.: & y_1 + y_2 \geq 1 & & s.t.: \ y_1 + y_2 \geq 1 \\ & y_1 + y_2 + y3 \geq 1 & D^{x^2}(N) & \quad y_1 + y_2 + y3 \geq 1 \quad D^{x^3}(N) \\ & y_1 + y_3 \geq 1 & & \quad y_1 + y_3 \geq 1 \\ & y_i \geq 0, \ i = 1,2,3. & & \quad y_i \geq 0, \ i = 1,2,3. \end{array} \tag{3.76}$$

The only solution in $\bigcap_{t=1}^{3} O_{\min}(A, B^{x^t}, c)$ is $y = (1,0,0)$, and therefore we have that $EOwen2(A, B, c) = \{(1,1,1)\}$. Then we conclude that $\alpha^1 \notin EOwen2(A, B, c)$ and, as a consequence, $EOwen3(A, B, c) \nsubseteq EOwen2(A, B, c)$.

To follow the same reasonings we did for the Extended Owen 1 and Extended Owen 2 sets, we prove now that the Extended Owen 3 set and the Extended Owen set coincide on $\widehat{\mathcal{L}}$.

Proposition 3.2.13 Let $(A, B, c) \in \widehat{\mathcal{L}}$. Then

$$EOwen(A, B, c) = EOwen3(A, B, c). \tag{3.77}$$

Proof. Let $(A, B, c) \in \widehat{\mathcal{L}}$ and $\alpha \in EOwen3(A, B, c)$. Let x^1 be the only solution to $P(N)$. Then,

$$\begin{aligned} &\exists \ y \in \bigcap_{t=1}^{w} O_{\min}(A, B^{x^t}, c), \ x^* \in \{x^1, \ldots, x^w\} : \alpha = yB^{x^*} \\ &\Leftrightarrow \exists \ y \in O_{\min}(A, B^{x^1}, c), \ x^* \in \{x^1\} : \alpha = yB^{x^*} \\ &\Leftrightarrow \exists \ y \in O_{\min}(A, B^{x^1}, c), : \alpha = yB^{x^1} \Leftrightarrow \alpha \in EOwen(A, B, c). \end{aligned} \tag{3.78}$$

□

Analogously as we did for $EOwen1$ and $EOwen2$ the following result is proven.

Proposition 3.2.14 Allocations in $EOwen3(A, B, c)$ are efficient for all $(A, B, c) \in \mathcal{L}$.

To summarize this section, due to the fairness problems that the Owen allocations might present, three different modifications of the Extended Owen set defined for $\widehat{\mathcal{L}}$ have been proposed for the whole class of linear production games $\mathcal{L}$. Those three solution sets satisfy

- $Owen(A,B,c) \subseteq EOwen1(A,B,c) \subseteq EOwen2(A,B,c) \subseteq EOwen3(A,B,c)$ $\forall\ (A,B,c) \in \mathcal{L}$. This gives us the idea that, if the Owen solution is not acceptable we try to allocate the benefits by using the Extended Owen 1 solution. If this new allocation is not acceptable yet, we allocate following an allocation in the Extended Owen 2 set. If this allocation is not accepted by the players, we calculate a Extended Owen 3 allocation.

- In general one has that
(3.79)
$$Owen(A,B,c) \not\supseteq EOwen1(A,B,c) \not\supseteq EOwen2(A,B,c) \not\supseteq EOwen3(A,B,c).$$

  Meaning that the three new solution concepts presented do not coincide.

- For any $(A,B,c) \in \widehat{\mathcal{L}}$ we have that

$$\begin{aligned} EOwen(A,B,c) &= EOwen1(A,B,c) = EOwen2(A,B,c) \\ &= EOwen3(A,B,c) = Owen(A,B^{x^*},c), \end{aligned} \tag{3.80}$$

  where $\{x^*\} = O_{\max}(A,B,c)$. This properties state that the three new solution sets defined for the whole class of linear production games are extensions of the Extended Owen set defined for the class $\widehat{\mathcal{L}}$.

Once defined the extensions to the Owen set on LP games, let us see now how to apply this solution set to another class of linear production games: Flow Games.

3.3 Application to Flow Games

We begin this section by showing an example of the Owen set in a Flow game to motivate the need to apply the Extended Owen solution to Flow Games.

Example 3.3.1 *Consider the flow game described in Example 2.3.1 The Owen solution, which is a core allocation, results from the optimal solutions to the dual problem D_N, see Problem (2.45). One optimal solution to such problem is:*

$$y = (1, 1, 1, 0, 0, 0). \tag{3.81}$$

Therefore, the corresponding Owen solution is

$$(3, 1, 6), \tag{3.82}$$

where the i^{th} component of this allocation is equal to $yb(i)$.

The interpretation the Owen solution is the following: those variables with optimal value in the dual program equal to 1 give us the minimum cut of the graph, so every player who does not own any of the arcs participating in the minimum cutset shall receive nothing after the Owen allocation. This seems to be unfair since those players having an excess in the capacity of their arcs will have an associated dual variable equal to 0 (this comes from the Complementary Slackness Theorem, see [4]). In the following section the Extended Owen set is applied to Flow Games, trying this way to avoid the above-mentioned problems.

Now we introduce some notation for the rest of the section. Given is a Flow Game (N, v). From the vectors $b(i)$, $i = 1, \ldots, n$, that help us to formulate Flow Games as Linear Programming Games, we define the matrix $B \in \mathbb{R}^{l \times n}$ as the matrix whose columns are the vectors $b(1), \ldots, b(n)$.

Then we can define the characteristic function of a flow game from the linear program

$$P(S) \quad \begin{aligned} \max \quad & p \cdot x \\ \text{s.t.:} \quad & x \leq Be_S \\ & xH = 0 \\ & x \geq 0 \end{aligned} \tag{3.83}$$

where $p \in \mathbb{R}^l$ and $H \in \mathbb{R}^{l \times n}$ are as defined in Section 2.3.2. Analogously, its dual program $D(S)$ is

$$
(3.84) \qquad D(S) \qquad \begin{array}{rl} \min & yBe_S \\ \text{s.t.:} & y + Hz \geq p \\ & y \geq 0 \end{array}
$$

So, a Flow Game can be represented by (p, B, H). We shall denote the set of optimal solutions to $P(N)$ and $D(N)$ by $O_{\max}(p, H, B)$ and $O_{\min}(p, H, B)$ respectively.

Let us remember that each solution to $P(N)$ $x^* \in O_{max}(p, H, B)$ defines an optimal feasible source-to-sink flow, and that each optimal feasible solution to $D(N)$ $y^* \in O_{\min}(p, H, B)$ leads us to an allocation in the core of the game, of the form $\gamma = y^* B$.

The set consisting of all the allocations obtained from optimal solutions of D is the well-known *Owen Set*, defined for Flow Games as follows:

$$
(3.85) \qquad \text{Owen}(p, B, H) = \{\gamma = y^* B \; : \; y^* \in O_{\min}(p, H, B)\}.
$$

Now we follow a similar reasoning as we did for LP games. Let (p, H, B) be a flow game, $x^* \in O_{\max}(p, H, B)$ and $y^* \in O_{\min}(p, H, B)$. By the Complementary Slackness Theorem (see [41]), we know that if c_j, the capacity of the j^{th} arc, is strictly higher than x_j^*, the amount of material running through the arc j, $(c_j > x_j)$, then $y_j^* = 0$. That is, the owner of this arc will not receive anything for this arc in the Owen allocation. That means that, if a player does not own any of the arcs participating in the minimum cut, then he/she will not receive anything. We already know that this allocation is in the core of the game, but is it always an admissible allocation? To overcome this problem, we make use of the Extended Owen solution introduced before for LP games and apply it to Flow games.

The coordinates of x^* define the amount of material that runs through each arc in an optimal source-to-sink flow. We now consider the flow game in which every arc k reduces its capacity until x_k^*. Such game is denoted by (p^*, B^*, H^*). This way, we expect to allow arcs that do not participate in the minimum cutset to have an allocation different from zero, because now their respective dual variables are not forced to be zero.

Let us now interpret for Flow Games the five axioms we used to characterize the Extended Owen set for Linear Production games.

One person efficiency (Axiom 1) means that if there is only one agent owning all arcs and the capacity of each arc is one, then the solution concept assigns to him the maximal amount of units that can be sent from the source to the sink. The second axiom, rescaling, demands that the solution concept remains invariant if the units in which the capacities are measured change. Axiom 3, shuffle, states that if the arcs are shuffled among the agents, then the solution rule changes in the same way.

The fourth axiom, consistency, assumes that the players agree that the profit is divided according to a vector $\alpha \in \varphi(A, I_n, c)$. Afterwards player k takes his payoff α_k and leaves. Suppose that the arcs that player k owned can be used by the other agents for a price α_k per unit of capacity. Then, a solution rule satisfies consistency if the restriction of α to agents $N \setminus \{k\}$ is a solution to the reduced flow game. The consistency axiom has to do with the special case in which every player owns exactly one arc with capacity one, and different players own different arcs. Therefore we can identify the set of arcs with the set of players in an appropriate way. Axiom 5, deletion, says that if an arc is not needed to make the maximal profit $v(N)$, then we can delete this arc from the network, the outcomes of the old situations being also outcomes in the new game.

Therefore the following theorem follows.

Theorem 3.3.1 Let (p, B, H) be a Flow Game, with only one optimal flow from the source to the sink. Let x^* be such optimal source-to-sink flow. If ϕ satisfies one person efficiency, rescaling, shuffle, consistency and deletion over the only reduced flow game (p^*, B^*, H^*), then $\phi(p, B, H) = EOwen(p, B, H)$.

Let us see an example of the Extended Owen set in Flow Games.

Example 3.3.2 *Consider the flow game* (N, v) *as depicted in Figure 3.1 with* $N = \{1, 2, 3\}$. *Over the arcs, two numbers are represented: the first one corresponds to the maximum capacity of that arc, the second one being the player owning the arc.*

Let us list the arcs as follows: arc joining nodes 1 *and* 2 *is arc 1, the one that joins* 1 *and* 3 *is arc 2,* $(2, 3)$ *is named as arc* 3, $(2, 4)$ *is arc* 4 *and* $(3, 4)$ *is arc* 5. *Then, this*

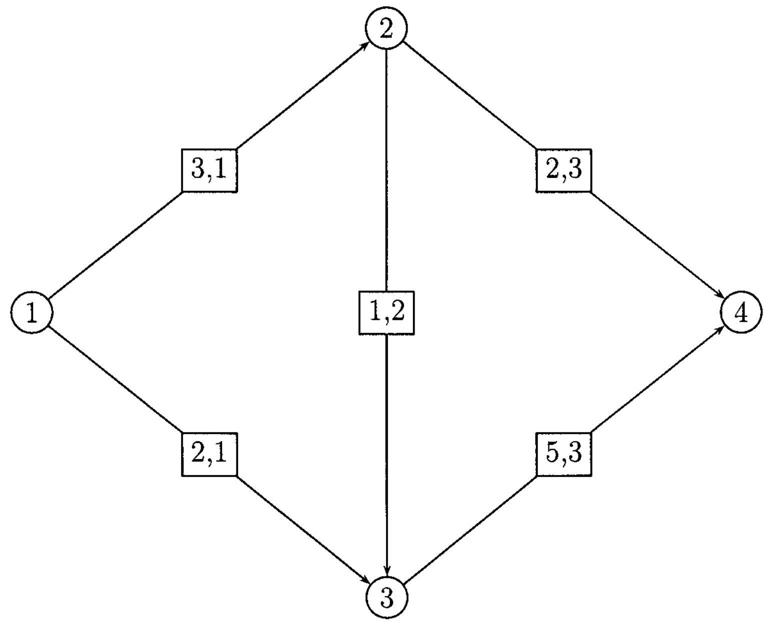

Figure 3.1: Flow Game.

game is generated by:

$$B = \begin{pmatrix} 3 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 5 \end{pmatrix}, \ p = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.86}$$

and H is the corresponding flow conservation matrix.

There is only one optimal flow, $x^* = (3,2,1,2,3)$, where x_i^* denotes the flow running through arc i, $i = 1, ..., 5$. One can see that there are two minimum cutsets separating the source and the sink, arcs $\{1,2\}$ and arcs $\{2,3,4\}$. From that, we conclude that the $O_{\min}(p,H,B)$ is the convex hull of $\{y^1, y^2\} = \{(1,1,0,0,0),(0,1,1,1,0)\}$. Therefore, the Owen set for this game is any convex combination of the allocations

$$\alpha^1 = y^1 B = (5,0,0), \ \alpha^2 = y^2 B = (2,1,3). \tag{3.87}$$

From the only optimal flow x^* we generate the corresponding reduced Flow Game (p, B^{x^*}, H), which is described in figure 3.2. The B^{x^*} matrix of such reduced game is

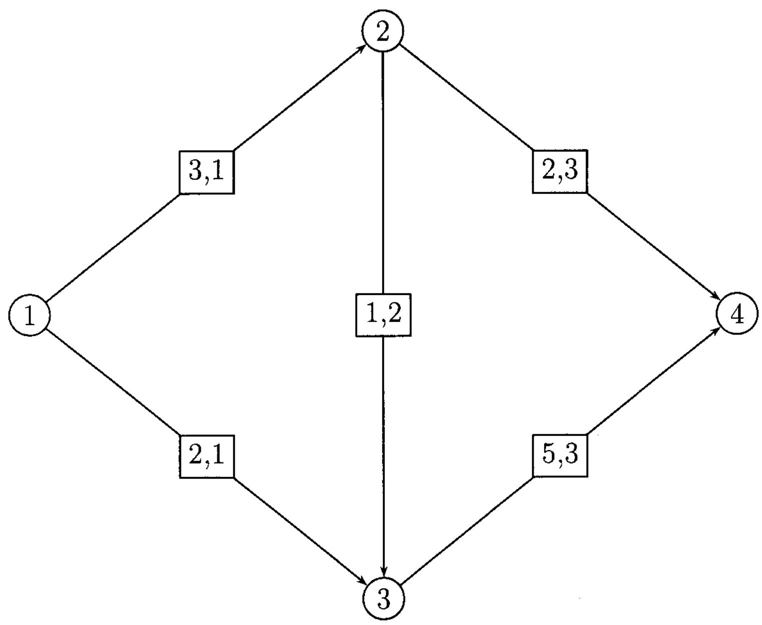

Figure 3.2: Reduced Flow Game.

$$B = \begin{pmatrix} 3 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 3 \end{pmatrix}. \tag{3.88}$$

There are three minimum cutsets for this Flow Problem, $\{1,2\}$, $\{2,3,4\}$ and $\{4,5\}$ (note that, as expected, the two minimum cutsets of the flow problem (p, B, H) are also minimum cutsets of the reduced flow problem (p, B^{x^*}, H).) Therefore, $O_{\min}(p, B^{x^*}, H)$ is the convex hull of the three points:

$$\{y^1, y^2, y^3\} = \{(1,1,0,0,0), (0,1,1,1,0), (0,0,0,1,1)\}, \tag{3.89}$$

having this way that the Extended Owen set of our Flow Game is the any convex combi-

nation of the following three allocations:

$$\{y^1 B^{x^*}, y^2 B^{x^*}, y^3 B^{x^*}\} = \{(5,0,0),(2,1,3),(0,0,5)\}. \tag{3.90}$$

The reader may check that, as we already knew, the Owen set is a subset of the Extended Owen set for this Flow Game.

Chapter 4

Supply Chain Games

In this chapter we study a model of cooperation over a Supply Chain Problem which arises when, over a graph, a group of nodes offers certain commodity, other nodes require it and a third group of nodes does not need this material nor offer it but is strategically relevant to the distribution plan. The delivery of one unit of material to a demand node generates a fixed profit, and the shipping of the material through the arcs has an associated cost. Such problem is defined in the first section of this chapter. We show that in that framework cooperation is beneficial for the different parties. We prove that such a cooperative situation, which will be called a Supply Chain Game, is totally balanced by finding a fair allocation (in the core of an associated cooperative game). The chapter also shows the relation between these cooperative games and other well-known games: Linear Production, Flow, Assignment, Transportation and Shortest Path games. Later on we will introduce some solution concepts specifically applied to our new class of games, to later finish the chapter by showing the natural extension of Supply Chain Games to the multicriteria case.

4.1 Supply Chain Problem: definition and formulation

The classical transportation problem, see Section 1.3, arises when an optimal distribution plan must be determined in order to transport a product on a network, in which there are some nodes offering that product and others that need it. There is an arc joining each node that can produce material with each node demanding it. This problem assumes that the transport of one unit of material from a supply node i to a demand node j gives rise to a profit of b_{ij} monetary units. The goal is to maximize the overall profit generated when covering the total demand.

The problem we now deal with is a generalization of the transportation problem. Let $G = (N, A)$ be a directed network, where $N = \{1, \ldots, n\}$ is the set of nodes and $A \subset N \times N$ is the set of arcs of the graph. Each node $i \in N$ has two scalar numbers, b_i and k_i, associated with it. If b_i is positive we shall say that i is a *supply* node, if it is negative node i is a *demand* node and if $b_i = 0$ node i is a *transfer* node. So, b_i is the amount of offer or demand that node i has. If i is a supply node, k_i is interpreted as the necessary cost for node i to produce one unit of material. On the other hand, if i is a demand node, k_i represents the profit that will be generated at node i if one of its unit of demand is satisfied. If i is a transfer node k_i is set to 0. We shall denote each arc of A by the ordered pair constituted by its initial node and its final one. That is, arc (i, j) joins nodes i and j in this direction. Each arc (i, j) has a scalar number $c_{ij} \in \mathbb{R}$ associated with it, which is interpreted as the necessary cost to transport one unit of the material trough the arc (i, j). Additionally, the capacity of some of the arcs might be bounded from above. Let $h_{ij} \in (0, +\infty]$ be the capacity of arc (i, j). Note that arcs are allowed to have unbounded capacity. Thus, the problem consists of finding a feasible distribution plan maximizing the total benefit. See the following example.

Example 4.1.1 *Let us consider the transportation network given in Figure 4.1. By each node there is a 2-dimensional vector. Its first coordinate is the amount of material that this node can produce or its demand (b_i). The second coordinate is the unitary cost to produce one unit of material at node i, if i is a supply node ($b_i > 0$), or the unitary profit after having one of the node i's units of demand satisfied, if i is a demand node ($b_i < 0$). On every arc there is a vector. The first component of such vectors is the cost of the corresponding arc, its capacity being the second component of the vector. Those data constitute a Supply Chain Problem (SChP for short), the goal being to maximize the general profit. From now on we will omit h_{ij} on those arcs with $h_{ij} = +\infty$.*

Note that in this model it is not necessary to cover all the demand nor to launch all the offer available. The demand of a node j will be satisfied if and only if there is a

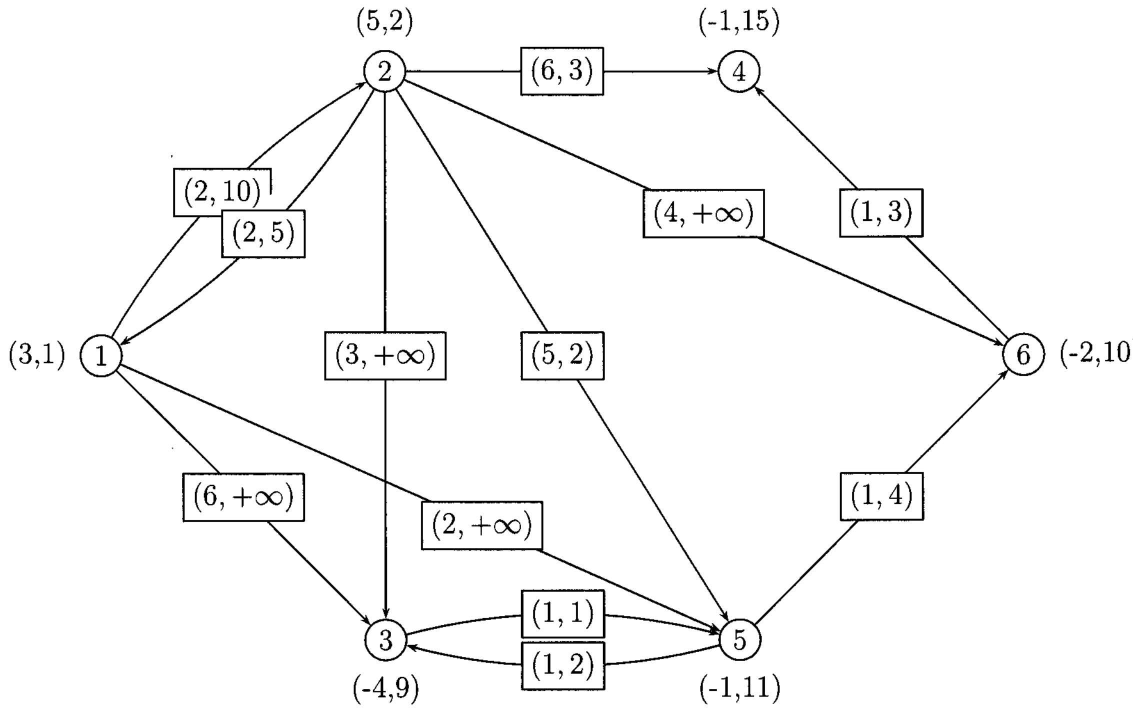

Figure 4.1: Transportation Network

profitable path from a supply node i to a demand node j and node i has some material available.

Now, we define our supply chain problem formally. Consider a directed graph $G = (N, A)$, two vectors $b, k \in \mathbb{R}^n$ and two matrices $C \in \mathbb{R}^{n \times n}$ and $H \in (\mathbb{R}_+ \cup \{+\infty\})^{n \times n}$, where c_{ij} is the unitary cost of transporting one unit from node i to node j and h_{ij} is the capacity of arc (i, j).

Consider the following sets:

$$P := \{i \in N : b_i > 0\}, \quad Q := \{i \in N : b_i < 0\}, \quad R := \{i \in N : b_i = 0\}. \tag{4.1}$$

These sets are called *supply* set, *demand* set and *transfer* set, respectively. It is clear that

$$P \cup Q \cup R = N, \quad P \cap Q = P \cap R = Q \cap R = \emptyset, \tag{4.2}$$

that is, P, Q, R are a partition of N.

Definition 4.1.1 *Given a directed network (N, A), where N is the set of nodes, $|N| = n$, and A is the set of arcs $A \subset N \times N$, two matrices $C \in \mathbb{R}^{n\times n}$ and $H \in (\mathbb{R}_+ \cup \{+\infty\})^{n\times n}$, a supply-and-demand vector $b \in \mathbb{R}^n$ and the cost-profit vector $k \in \mathbb{R}^n$, a Supply Chain Problem is the 6-tuple*

$$(N, A, C, b, k, H). \tag{4.3}$$

Example 4.1.2 *In the transportation network described in Example 4.1.1, the corresponding supply chain problem is defined as*

$$(N, A, C, b, k, H) \tag{4.4}$$

where

$$N = \{1, 2, 3, 4, 5, 6\}, \tag{4.5}$$

$$A = \{(1,2),(1,3),(1,5),(2,1),(2,3),(2,4),(2,5),(2,6),(3,5),(5,3),(5,6),(6,4)\}, \tag{4.6}$$

$$C = \begin{pmatrix} - & 2 & 6 & - & 2 & - \\ 2 & - & 3 & 6 & 5 & 4 \\ - & - & - & - & 1 & - \\ - & - & - & - & - & - \\ - & - & 1 & - & - & 1 \\ - & - & - & 1 & - & - \end{pmatrix}, \quad H = \begin{pmatrix} - & 10 & +\infty & - & +\infty & - \\ 5 & - & +\infty & 3 & 2 & +\infty \\ - & - & - & - & 1 & - \\ - & - & - & - & - & - \\ - & - & 2 & - & - & 4 \\ - & - & - & 3 & - & - \end{pmatrix}, \tag{4.7}$$

$$b = (3, 5, -4, -1, -1, -2), \quad k = (1, 2, 9, 15, 11, 10). \tag{4.8}$$

Now we are going to formulate our SChP as a linear program. Let x_{ij} be the amount of transported material from i to j. A feasible distribution plan must satisfy several conditions:

- Supply nodes cannot create more material than they offer. Thus, the amount of material leaving from each supply node should not be higher than the amount of material that it can offer; that is, the amount of material leaving from a supply node, from now on it will be called *outgoing* flow of material, minus the amount of material that comes to it, from now on *incoming* flow of material, must be less than or equal to what this node can produce. Besides, supply nodes should not keep new material. Thus we have that:

$$0 \leq \sum_{j \in N:(i,j) \in A} x_{ij} - \sum_{j \in N:(j,i) \in A} x_{ji} \leq b_i \quad \forall\, i \in P. \tag{4.9}$$

- Demand nodes must not receive more than what they require: the amount of incoming flow minus the amount of outgoing flow must be less than or equal to their demand. Further, demand nodes do not have the capability to create new material. In terms of x_{ij} one has that:

$$0 \leq \sum_{j \in N:(j,i) \in A} x_{ji} - \sum_{j \in N:(i,j) \in A} x_{ij} \leq -b_i \quad \forall\, i \in Q. \tag{4.10}$$

- The incoming flow and the outgoing flow must be the same for every transfer node, that is, transfer nodes can not neither create material nor keep material. Mathematically we have:

$$\sum_{j \in N:(i,j) \in A} x_{ij} - \sum_{j \in N:(j,i) \in A} x_{ji} = 0 \quad \forall\, i \in R. \tag{4.11}$$

- Besides, we must ask for the flow to be non-negative and respect the capacity constraints

$$0 \leq x_{ij} \leq h_{ij} \;\forall\, (i,j) \in A. \tag{4.12}$$

Now that the constraints that a distribution plan must satisfy to be feasible have been created, let us study the objective function that a feasible distribution plan must maximize to be optimal. We create such a function step by step:

- The general benefit obtained, that is, the total demand covered at each node of

Q times the benefit that is generated in this node, must be maximized. Let us consider $i \in Q$. The total demand of material that is covered after the distribution plan $(x_{ij})_{(i,j)\in A}$ at node i is:

$$\sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij}, \tag{4.13}$$

that is, the amount of incoming material minus the amount of outgoing material at $i \in Q$. Thus, we maximize the total demand covered by the distribution plan times the benefits, that is

$$\sum_{i\in Q} k_i \left(\sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij} \right). \tag{4.14}$$

- The costs of producing material are to be minimized. That is, for each node $i \in P$ we have to minimize the total amount of material that is produced at node i,

$$\sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{j\in N:(j,i)\in A} x_{ji}, \tag{4.15}$$

times its unitary cost. Thus, in the objective function we minimize

$$\sum_{i\in P} k_i \left(\sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{k\in N:(k,i)\in A} x_{ki} \right). \tag{4.16}$$

- The transport through each arc is a cost, so we shall minimize

$$\sum_{(i,j)\in A} c_{ij} x_{ij}. \tag{4.17}$$

Therefore, our objective function is to maximize

$$\begin{aligned}\sum_{i\in Q} k_i\left(\sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij}\right) - \\ \sum_{i\in P} k_i\left(\sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{j\in N:(j,i)\in A} x_{ji}\right) - \sum_{(i,j)\in A} c_{ij}x_{ij}.\end{aligned} \tag{4.18}$$

Expression (4.18) can be summarized as

$$\sum_{i\in P\cup Q} k_i\left(\sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij}\right) - \sum_{(i,j)\in A} c_{ij}x_{ij}. \tag{4.19}$$

Taking into account that $k_i = 0$ for all $i \in R$, our objective function can be expressed as

$$\sum_{i\in N} k_i\left(\sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij}\right) - \sum_{(i,j)\in A} c_{ij}x_{ij}. \tag{4.20}$$

In the expression above we have that:

$$\begin{aligned} & \sum_{i\in N} k_i\left(\sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij}\right) \\ = & \sum_{i\in N} k_i \sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{i\in N} k_i \sum_{j\in N:(i,j)\in A} x_{ij} \\ = & \sum_{i\in N} \sum_{j\in N:(j,i)\in A} k_i x_{ji} - \sum_{i\in N} \sum_{j\in N:(i,j)\in A} k_i x_{ij} = \sum_{(j,i)\in A} k_i x_{ji} - \sum_{(i,j)\in A} k_i x_{ij} \\ = & \sum_{(i,j)\in A} k_j x_{ij} - \sum_{(i,j)\in A} k_i x_{ij} = \sum_{(i,j)\in A} (k_j - k_i) x_{ij}\end{aligned} \tag{4.21}$$

To summarize, given a Supply Chain Problem (N, A, C, b, k, H), its optimal distribution plans are given by optimal solutions to the linear program (4.22).

$$
\begin{aligned}
\max \quad & \sum_{(i,j)\in A} (k_j - k_i - c_{ij})x_{ij} \\
\text{s.t.} \quad & \sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{j\in N:(j,i)\in A} x_{ji} \le b_i \quad \forall\, i \in P \\
& \sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij} \le 0 \quad \forall\, i \in P \\
& \sum_{j\in N:(j,i)\in A} x_{ji} - \sum_{j\in N:(i,j)\in A} x_{ij} \le -b_i \quad \forall\, i \in Q \\
& \sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{j\in N:(j,i)\in A} x_{ji} \le 0 \quad \forall\, i \in Q \\
& \sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{j\in N:(j,i)\in A} x_{ji} = 0 \quad \forall\, i \in R \\
& 0 \le x_{ij} \le h_{ij} \quad \forall\, (i,j) \in A
\end{aligned} \tag{4.22}
$$

Example 4.1.3 *The formulation of the SChP in Example 4.1.1 as a linear programming problem is*

$$
\begin{aligned}
\max \quad & (2-1-2)x_{12} + (9-1-6)x_{13} + (11-1-2)x_{15} + \\
& (1-2-2)x_{21} + (9-2-3)x_{23} + (15-2-6)x_{24} + \\
& (11-2-5)x_{25} + (10-2-4)x_{26} + (11-9-1)x_{35} + \\
& (9-11-1)x_{53} + (10-11-1)x_{56} + (15-10-1)x_{64} \\
s.t.: \quad & 0 \le x_{12} + x_{13} + x_{15} - x_{21} \le 3 \quad (i=1) \\
& 0 \le x_{21} + x_{23} + x_{24} + x_{25} + x_{26} - x_{12} \le 5 \quad (i=2) \\
& 0 \le x_{13} + x_{23} + x_{53} - x_{35} \le 4 \quad (i=3) \\
& 0 \le x_{24} + x_{64} \le 1 \quad (i=4) \\
& 0 \le x_{15} + x_{25} + x_{35} - x_{53} - x_{56} \le 1 \quad (i=5) \\
& 0 \le x_{26} + x_{56} - x_{64} \le 2 \quad (i=6) \\
& x_{12} \le 10,\ x_{21} \le 5,\ x_{24} \le 3,\ x_{25} \le 2 \\
& x_{35} \le 2,\ x_{53} \le 2,\ x_{56} \le 4,\ x_{64} \le 3 \\
& x_{ij} \ge 0 \quad \forall\, (i,j) \in A
\end{aligned} \tag{4.23}
$$

and one can see that an optimal solution is given by the distribution plan:

(4.24) $$x_{15} = 3, x_{23} = 4, x_{26} = 1, x_{56} = 2, x_{64} = 1, \ x_{ij} = 0 \textit{ in any other case.}$$

This distribution plan generates a profit of 41 *monetary units. In Figure 4.2 this optimal distribution plan is shown. Only those arcs that send material are depicted, the respective amount of material running through each arc being expressed on them.*

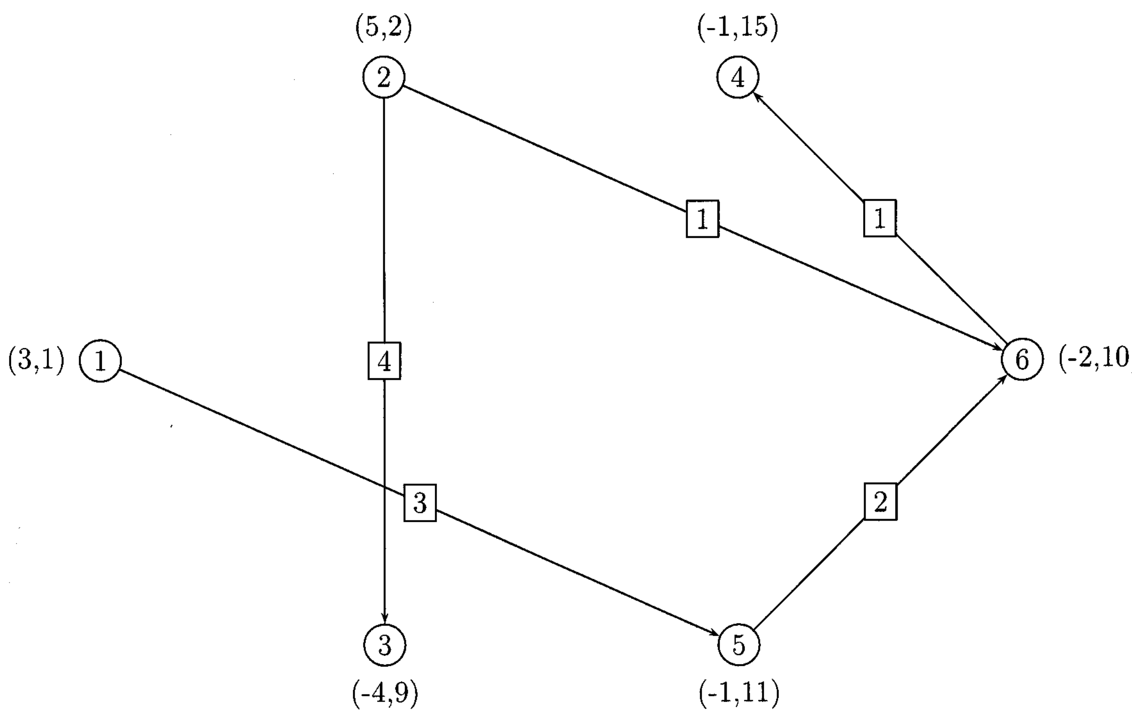

Figure 4.2: Optimal distribution plan.

In the previous example all the demand was covered, but this is not a general characteristic of our problem. There exist Supply Chain Problems where an optimal distribution plan may not satisfy all the demand, or not launch all the material that supply nodes have, or both things, as we can see in the following example.

Example 4.1.4 *One can check that the distribution plan* $x_{12} = 2, x_{23} = 0, x_{13} = 0$ *is optimal in the Supply Chain Problem described in Figure 4.3 (note that all arcs have unbounded capacity). Nevertheless, the supply node* 1 *keeps one surplus unit and the* *demand node* 3 *is still demanding another unit after the optimal distribution plan has been taken. This is because of the fact that the necessary cost to send one unit from node 1 to node 3 plus the cost of* k_1 *is higher than the benefit that this unit would generate,* k_3.

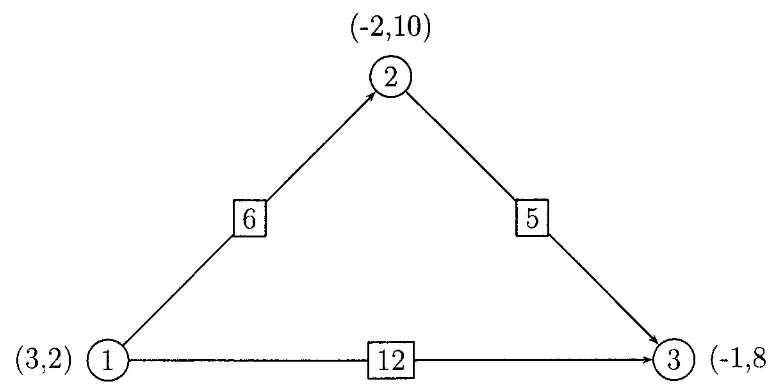

Figure 4.3: Supply Chain Problem.

In the previous example the demand at some nodes in Q was not completely satisfied and there still was some material available and paths to transport it from nodes in P. Situations like this may not arise when the goal of the network is to satisfy basic needs such as medical care, educational centers, etc. That is why we can say that in the Supply Chain problem presented, the active agents are "selfish".

4.2 Supply Chain Games

Once we have defined our Supply Chain Problems, we are interested in studying the possible cooperation that may arise when the nodes are controlled by different agents having conflicting objectives. So, given a Supply Chain Problem (N, A, C, b, k, H) we naturally define the corresponding Supply Chain Game as follows.

The set of players is $N = \{1, \ldots, n\}$, every node is owned by one player and each player is associated with the only node that it owns. Then, to avoid a more complicated notation, we shall denote the player that owns node i by i as well. Thus, we shall have supply players, demand players and transfer players, depending on the kind of node they own. In other words, the set of players N is divided into the set of supply players

$P = \{i \in N : b_i > 0\}$, the set of demand players $Q = \{i \in N : b_i < 0\}$ and the set of transfer players $R = \{i \in N : b_i = 0\}$.

Now we have to define v, the characteristic function of the game,

$$\begin{aligned} v:\ 2^N &\rightarrow \mathbb{R} \\ S &\rightarrow v(S) \end{aligned} \tag{4.25}$$

That is to say, we need to find the maximum profit that each group of players can make by acting on their own, without taking into account what the rest of players do.

For each $S \subset N$ a distribution subnetwork $(S, A_S, C_S, b_S, k_S, H_S)$ can be created, where:

- $A_S = (S \times S) \cap A$, meaning that only those arcs having both endpoints in the coalition S are considered.

- C_S, H_S, b_S and k_S are the restriction of C, H, b and k to S, respectively.

In the same way, the offer, demand and transfer sets of S are defined as:

$$P_S = P \cap S,\ Q_S = Q \cap S,\ R_S = R \cap S. \tag{4.26}$$

The subnetwork above has at least one optimal distribution plan that gives us the maximum profit that the coalition S can generate. Such a value is given by the optimal solutions to the following linear program, which will from now on be called $Pr(S)$.

$$
\begin{array}{llllll}
\max & \sum_{(i,j)\in A_S} (k_j - k_i - c_{ij})x_{ij} := f_S(x) & & & & \\
\text{s.t.} & \sum_{j\in S:(i,j)\in A_S} x_{ij} & - & \sum_{j\in S:(j,i)\in A_S} x_{ji} & \leq b_i & \forall\, i \in P_S \\
& \sum_{j\in S:(j,i)\in A_S} x_{ji} & - & \sum_{j\in S:(i,j)\in A_S} x_{ij} & \leq 0 & \forall\, i \in P_S \\
& \sum_{j\in S:(j,i)\in A_S} x_{ji} & - & \sum_{j\in S:(i,j)\in A_S} x_{ij} & \leq -b_i & \forall\, i \in Q_S \quad (Pr(S)) \\
& \sum_{j\in S:(i,j)\in A_S} x_{ij} & - & \sum_{j\in S:(j,i)\in A_S} x_{ji} & \leq 0 & \forall\, i \in Q_S \\
& \sum_{j\in S:(i,j)\in A_S} x_{ij} & - & \sum_{j\in S:(j,i)\in A_S} x_{ji} & = 0 & \forall\, i \in R_S \\
& 0 \leq x_{ij} \leq h_{ij} \quad \forall\, (i,j) \in A_S & & & &
\end{array}
\tag{4.27}
$$

Then, we define the cooperative game with transferable utility associated with the Supply Chain Problem (N, A, C, b, k, H) as follows:

Definition 4.2.1 *Given is (N, A, C, b, k, H) a Supply Chain Problem. The Supply Chain Game (SChG for short) associated with this problem is the cooperative game with transferable utility (N, v) where $v(S)$ is given by the optimal value of Problem (4.27) for all $S \subset N$.*

The following example shows a SChG.

Example 4.2.1 *Let us consider the Supply Chain Game (N, v) arising from the Supply Chain problem described on Figure 4.4, whose arc capacities are not bounded from above.*

One can check that the values of the characteristic function of the game (N, v) are

(4.28)

S	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
$v(S)$	0	0	0	0	8	3	8

The first result from the definition of Supply Chain Games is that they are well defined.

Proposition 4.2.1 Given is (N, v) the Supply Chain Game defined from the 6-tuple

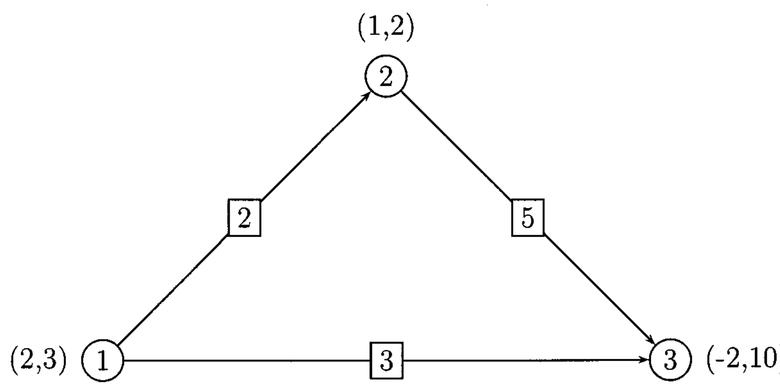

Figure 4.4: Transportation Network

(N, A, C, b, k, H). One has that $\forall\ S \subset N,\ v(S)$ is well defined and nonnegative.

Proof. Given $S \subset N$, let us consider $\overline{x}_{ij} = 0\ \forall\ (i,j) \in A_S$. The distribution plan $\overline{x} = (\overline{x}_{ij})_{(i,j)\in A_S}$ is feasible, since it trivially satisfies the constraints of the corresponding linear program $Pr(S)$. So, we conclude that $v(S)$ is well defined. Besides, the objective function of $Pr(S)$ takes value 0 in $x = \overline{x}$, so $v(S)$ (the optimal value) is greater than or equal to 0. □

In following sections we will study some properties that SChG satisfy as well as their relation with some known games presented before: Flow Games, Transportation Games and Shortest Path Games.

4.3 Properties of Supply Chain Games

In this section certain properties about Supply Chain Games are presented. The first one we are to prove is that they are 0-normalized, see Definition 2.1.1.

Proposition 4.3.1 Supply Chain Games are 0-normalized.

Proof. Let (N, v) be a SChG. Given $i \in N$, it is clear that $A_{\{i\}}$, the arcs that i owns, is the empty set. That is, the objective function of problem $Pr(\{i\})$, see Problem (4.27), is the function identically null. Then we conclude that $v(\{i\}) = 0$. □

Another property of the class of SChG is the superadditivity, (Definition 2.1.2).

Proposition 4.3.2 Every Supply Chain Game is superadditive.

Proof. Let (N, v) be the SChG arising from the SChP (N, A, C, b, k, H). Let $S, T \subset N$ such that $S \cap T = \emptyset$. We have to prove that $v(S) + v(T) \leq v(S \cup T)$.
Let x^S and x^T be two optimal distribution plans for the coalitions S and T, respectively, that is, $f_S(x^S) = v(S)$ and $f_T(x^T) = v(T)$, where f_L denotes the objective function of problem $Pr(L)$, see Equation (4.27), for all $L \subset N$.

We define the following distribution plan for the coalition $S \cup T$:

$$x^* = (x^*_{ij})_{(i,j)} \in A_{S\cup T} \; : \; x^*_{ij} = \begin{cases} x^S_{ij} & \text{if } (i,j) \in A_S \\ x^T_{ij} & \text{if } (i,j) \in A_T \\ 0 & \text{otherwise} \end{cases} \tag{4.29}$$

By definition

$$A_{S\cup T} = A \cap ((S \cup T) \times (S \cup T)). \tag{4.30}$$

Let us see that x^* is feasible for $Pr(S \cup T)$. To do so, we have to prove that:

1. $\displaystyle\sum_{j \in S\cup T:(i,j)\in A} x^*_{ij} - \sum_{j \in S\cup T:(j,i)\in A} x^*_{ji} \leq b_i \quad \forall\, i \in P_{S\cup T}.$

2. $\displaystyle\sum_{j \in S\cup T:(j,i)\in A} x^*_{ji} - \sum_{j \in S\cup T:(i,j)\in A} x^*_{ij} \leq 0 \quad \forall\, i \in P_{S\cup T}.$

3. $\displaystyle\sum_{j \in S\cup T:(j,i)\in A} x^*_{ji} - \sum_{j \in S\cup T:(i,j)\in A} x^*_{ij} \leq -b_i \quad \forall\, i \in Q_{S\cup T}.$

4. $\displaystyle\sum_{j \in S\cup T:(i,j)\in A} x^*_{ij} - \sum_{j \in S\cup T:(j,i)\in A} x^*_{ji} \leq 0 \quad \forall\, i \in Q_{S\cup T}.$

5. $\displaystyle\sum_{j \in S\cup T:(i,j)\in A} x^*_{ij} - \sum_{j \in S\cup T:(j,i)\in A} x^*_{ji} = 0 \quad \forall\, i \in R_{S\cup T}.$

6. $0 \leq x^*_{ij} \leq h_{ij} \quad \forall\, (i,j) \in A_{S\cup T}.$

We prove (1), the proof of (2)-(5) being analogous.

- Let $i \in P_{S \cup T}$.

$$\begin{aligned} &\sum_{j \in S \cup T:(i,j) \in A} x^*_{ij} - \sum_{k \in S \cup T:(k,i) \in A} x^*_{ki} \\ &= \sum_{j \in S:(i,j) \in A} x^*_{ij} - \sum_{k \in S:(k,i) \in A} x^*_{ki} + \sum_{j \in T:(i,j) \in A} x^*_{ij} - \sum_{k \in T:(k,i) \in A} x^*_{ki}. \end{aligned} \tag{4.31}$$

It is clear that $P_{S \cup T} = P_S \cup P_T$ and $P_S \cap P_T = \emptyset$. Let us suppose that $i \in P_S$, the other case is analogous. Then

$$x^*_{ij} = \begin{cases} x^S_{ij} & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases} \tag{4.32}$$

and Equation (4.31) becomes

$$\sum_{j \in S:(i,j) \in A} x^S_{ij} - \sum_{k \in S:(k,i) \in A} x^S_{ki} \leq b_i, \tag{4.33}$$

since x^S is feasible for $Pr(S)$.

Clearly $0 \leq x^*_{ij} \leq h_{ij} \; \forall \; (i,j) \in A_{S \cup T}$.

Thus, x^* is feasible for $Pr(S \cup T)$. By the optimality of $v(S \cup T)$ in $Pr(S \cup T)$ one has that $f_{S \cup T}(x^*) \leq v(S \cup T)$.
Besides, it is easy to check that $f_{S \cup T}(x^*) = f_S(x^S) + f_T(x^T)$. Then we conclude that

$$v(S) + v(T) = f_S(x^S) + f_T(x^T) = f_{S \cup T}(x^*) \leq v(S \cup T). \tag{4.34}$$

□

By joining the nonnegativity property of v with the fact that the game is superadditive, we conclude that Supply Chain Games are monotonic.

Proposition 4.3.3 Let (N, v) be a Supply Chain Game. (N, v) is monotonic.

Another desirable property of cooperative games is the convexity, see Definition 2.1.5. Unfortunately Supply Chain Games are not convex in general, as shown in the following example.

Example 4.3.1 *Let us consider the Supply Chain Problem described in Figure 4.5 and its corresponding Supply Chain Game* (N, v)*,* $N = \{1, 2, 3\}$*.*

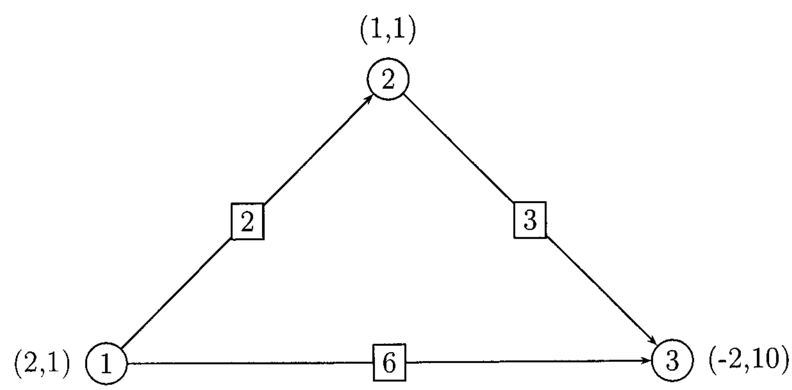

Figure 4.5: Transportation Network

The values of the characteristic function are:

(4.35)

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$v(S)$	0	0	0	0	6	6	10

Taking $S = \{3\}$*,* $T = \{1, 3\}$*,* $i = 2$*, we get that*

$$v(S \cup \{i\}) - v(S) = 6 - 0 > 4 = 10 - 6 = v(T \cup \{i\}) - v(T). \tag{4.36}$$

Since $S \subset T$*, we conclude that the game* (N, v) *is not convex.*

The following step is to prove that SChG have non-empty core. We will prove that in two different ways. The first one is by means of the Shapley-Bondareva's theorem, that is, by proving that the class of SChG is totally balanced. An alternative proof, in following sections, will give us a core allocation for any SChG.

Theorem 4.3.1 Supply Chain Games are balanced.

Proof. Let (N, A, C, b, k, H) be a Supply Chain Problem and (N, v) its associated Supply Chain Game. Let $\{S_1, \ldots, S_r\}$ be a balanced collection and let $\{y_1, \ldots, y_r\}$ be its balancing weights (that is, $\sum_{j:i \in S_j} y_j = 1 \; \forall \; i \in N$). In order to prove the balancedness of (N, v) we have to prove that $\sum_{l=1}^{r} y_l v(S_l) \leq v(N)$.

Given x^l an optimal distribution plan for S^l, $l = 1, \ldots, r$, we have that

$$v(S_l) = \sum_{(i,j) \in A_{S_l}} (k_j - k_i - c_{ij}) x^l_{ij}. \tag{4.37}$$

Let us consider the map $\delta^l : A \to \{0, 1\}$ such that:

$$\delta^l(i,j) = \begin{cases} y_l & \text{if } (i,j) \in A_{S_l} \\ 0 & \text{otherwise} \end{cases} \tag{4.38}$$

$\forall \; l = 1, \ldots, r$.

Let x^* be a distribution plan for the complete network, defined as follows:

$$x^*_{ij} = \sum_{l=1}^{r} \delta^l(i,j) x^l_{ij}. \tag{4.39}$$

We shall see that x^* is feasible for $Pr(N)$, that is to say, that x^* is feasible for the whole network. To do that, it must be checked:

1. $x^*_{ij} \geq 0$. Clearly by its definition. Besides,

$$x^*_{ij} = \sum_{l=1}^{r} \delta^l(i,j) x^l_{ij} \leq h_{ij} \sum_{l=1}^{r} \delta^l(i,j) = h_{ij} \sum_{l:(i,j) \in A_{S_l}} y^l \leq h_{ij} \sum_{l:i \in S_l} y^l = h_{ij}. \tag{4.40}$$

   Note that $\{l : (i,j) \in A_{S_l}\} \subset \{l : i \in S_l\}$.

2. $\sum_{j\in N:(i,j)\in A} x^*_{ij} - \sum_{j\in N:(j,i)\in A} x^*_{ji} \leq b_i \quad \forall\, i \in P$. To prove so, let $i \in P$.

(4.41)

$$\begin{aligned}
&\sum_{j\in N:(i,j)\in A} x^*_{ij} - \sum_{j\in N:(j,i)\in A} x^*_{ji} = \sum_{j\in N:(i,j)\in A}\sum_{l=1}^{r} \delta^l(i,j)x^l_{ij} - \sum_{j\in N:(j,i)\in A}\sum_{l=1}^{r} \delta^l(j,i)x^l_{ji}\\
&= \sum_{l=1}^{r}\Big(\sum_{j\in N:(i,j)\in A} \delta^l(i,j)x^l_{ij} - \sum_{j\in N:(j,i)\in A} \delta^l(j,i)x^l_{ji}\Big)\\
&= \sum_{l:\ i\in S_l}\Big(\sum_{j\in S_l:(i,j)\in A_{S_l}} y_l x^l_{ij} - \sum_{j\in S_l:(j,i)\in A_{S_l}} y_l x^l_{ji}\Big)\\
&= \sum_{l:\ i\in S_l} y_l\Big(\sum_{j\in S_l:(i,j)\in A_{S_l}} x^l_{ij} - \sum_{j\in S_l:(j,i)\in A_{S_l}} x^l_{ji}\Big) \leq \sum_{l:\ i\in S_l} y_l b_i = b_i \sum_{l:\ i\in S_l} y_l = b_i.
\end{aligned}$$

And we conclude that the inequality holds.

3. Analogously to (2) we prove that

$$\begin{aligned}
&\sum_{j\in N:(j,i)\in A} x^*_{ji} - \sum_{j\in N:(i,j)\in A} x^*_{ij} \leq 0 \quad \forall\, i \in P.\\
&\sum_{j\in N:(j,i)\in A} x^*_{ji} - \sum_{j\in N:(i,j)\in A} x^*_{ij} \leq -b_i \quad \forall\, i \in Q.\\
&\sum_{j\in N:(i,j)\in A} x^*_{ij} - \sum_{j\in N:(j,i)\in A} x^*_{ji} \leq 0 \quad \forall\, i \in Q.\\
&\sum_{j\in N:(i,j)\in A} x^*_{ij} - \sum_{j\in N:(j,i)\in A} x^*_{ji} = 0 \quad \forall\, i \in R.
\end{aligned} \tag{4.42}$$

From (1), (2) and (3) we conclude that the distribution plan x^* is feasible in the linear program $Pr(N)$ that defines the value $v(N)$.

On the other hand,

$$\begin{aligned}
&\sum_{l=1}^{r} y_l v(S_l) = \sum_{l=1}^{r} y_l \sum_{(i,j)\in A_{S_l}} (k_j - k_i - c_{ij})x^l_{ij} = \sum_{(i,j)\in A_{S_l}}\sum_{l=1}^{r}(k_j - k_i - c_{ij})x^l_{ij}y_l\\
&= \sum_{(i,j)\in A}\sum_{l=1}^{r}(k_j - k_i - c_{ij})x^l_{ij}\delta(i,j)^l = \sum_{(i,j)\in A}(k_j - k_i - c_{ij})\sum_{l=1}^{r} x^l_{ij}\delta(i,j)^l\\
&= \sum_{(i,j)\in A}(k_j - k_i - c_{ij})x^*_{ij} = f_N(x^*) \underbrace{\leq}_{x^* \text{ is feasible in Pr(N)}} v(N).
\end{aligned} \tag{4.43}$$

We have proven that $\sum_{l=1}^{r} y_l v(S_l) \leq v(N)$ for any balanced collection $\{S_1, \ldots, S_r\}$ with balancing weights $(y_1, \ldots, y_r)$. Thus, we can assure that the game (N, v) is balanced. □

Since every subgame of a Supply Chain Game is also a Supply Chain Game, we can state that every Supply Chain Game is totally balanced.

Theorem 4.3.2 Supply Chain Games are totally balanced.

As a consequence of the theorem above, and applying the Bondareva-Shapley theorem, see Theorem 2.2.1, the main result of this section follows:

Theorem 4.3.3 Supply Chain Games have non-empty core.

To finish this section the structure of the class SChG is studied, in terms of additivity and multiplication by a scalar.

It can be shown that, if (N, v) and (N, w) are two Supply Chain Games, $(N, v+w)$ is not, in general, a SChG. One can find an example in [59], page 51. There it is shown that the addition of two Transportation Games is not, in general, a TG. Since $TG \subset SChG$, as will be shown in Section 4.4.2, the result follows.

But it can be proven that the multiplication by a non-negative scalar is a closed operation for SChG.

Proposition 4.3.4 Given is a Supply Chain Game (N, v). One has that for every $\lambda \in [0, \infty)$ the game $(N, \lambda v) \in SChG$.

Proof. Since (N, v) is a SChG, there must be a Supply Chain Problem (N, A, C, b, k, H) such that (N, v) is the associated game to (N, A, C, b, k, H). It is not difficult to see that the Supply Chain Problem $(N, A, \lambda C, b, \lambda k, H)$ generates the game $(N, \lambda v)$, which proves the proposition. □

4.4 Supply Chain Games and the state of the art

In this section the class of SChG is compared with other well-studied cooperative games: Maximum Flow Games, Linear Programming Games, Transportation Games, Assignment Games and Shortest Path Games.

4.4.1 Flow Games, Linear Programming Games and SChG

Another interesting conclusion from the balancedness of Supply Chain Games is that SChG are both Flow Games (FG for short) and Linear Programming Games (LPrG for short). This is due to the fact both the class of FG and the class of LPrG have been proven to be equivalent to the class of totally balanced games, see [34] and [12]. This leads to the following result.

Proposition 4.4.1 $SChG \subset FG$ and $SChG \subset LPrG$.

One would like the reverse inclusion to be true, having this way that the class of SChG is equivalent to the class of totally balanced games. This not true, see the following example.

Example 4.4.1 *Consider the game* (N, v) *such that* $N = \{1, 2\}$ *and* $v(i) = 1$ $i = 1, 2, v(N) = 3$. *Trivially this game is totally balanced, take the core allocation* $(1.5, 1.5)$ *as instance.*

On the other hand, this game satisfies that $v(i) > 0$ *for all players. Then* $(N, v) \notin$ *SChG, since SChG have been proven to be 0-normalized, see 4.3.1, that is,* $v(i) = 0 \ \forall \ i$.

To conclude this section, remark that $SChG \subsetneq FG$ and $SChG \subsetneq LPrG$.

4.4.2 Transportation Games and SChG

Now we are going to see that the class of Transportation Games (TG for short) is included in the class of Supply Chain Games.

Proposition 4.4.2 Transportation Games are Supply Chain Games.

Proof. Let (N, v) be a TG defined from the Transportation Problem (P, Q, B, p, q). If we consider the following data:
(4.44)
$R = \emptyset,\ C = -B,\ k = 0,\ b = ((p_i)_{i \in P}, (-q_j)_{j \in Q}),\ A = P \times Q,\ N = P \cup Q, H = (+\infty)^{n \times n}.$

then the Supply Chain Game (N', v') defined from the Supply Chain Problem (N, A, C, b, k, H) is equivalent to the transportation game (N, v) arising from the transportation problem (P, Q, B, p, q) as defined in Section 2.3.4. Let us prove it.

- $N' = P \cup Q \cup R = P \cup Q = N.$

- The value of $v'(S)$ is given when we maximize the following function

  $$\sum_{(i,j) \in A_S} (k_j - k_i - c_{ij}) x_{ij} \tag{4.45}$$

  subject to the constraints

  $$\sum_{j \in S : (i,j) \in A_S} x_{ij} - \sum_{j \in S : (j,i) \in A_S} x_{ji} \leq b_i \quad \forall\, i \in P_S \tag{4.46}$$

  $$\sum_{j \in S : (j,i) \in A_S} x_{ji} - \sum_{j \in S : (i,j) \in A_S} x_{ij} \leq 0 \quad \forall\, i \in P_S \tag{4.47}$$

  $$\sum_{j \in S : (j,i) \in A_S} x_{ji} - \sum_{j \in S : (i,j) \in A_S} x_{ij} \leq -b_i \quad \forall\, i \in Q_S \tag{4.48}$$

  $$\sum_{j \in S : (i,j) \in A_S} x_{ij} - \sum_{j \in S : (j,i) \in A_S} x_{ji} \leq 0 \quad \forall\, i \in Q_S \tag{4.49}$$

  $$\sum_{j \in S : (i,j) \in A_S} x_{ij} - \sum_{j \in S : (j,i) \in A_S} x_{ji} = 0 \quad \forall\, i \in R_S \tag{4.50}$$

$$x_{ij} \geq 0 \ \forall \ (i,j) \in A_S \tag{4.51}$$

Taking

$$k_i = 0 \ \forall \ i, \ c_{ij} = -b_{ij}, \ A = P \times Q \tag{4.52}$$

the objective function (4.45) above changes into

$$\max \sum_{i \in P_S} \sum_{j \in Q_S} b_{ij} x_{ij}. \tag{4.53}$$

For each $i \in P_S$ one has that:

$$\{j \in S : (i,j) \in A_S\} \equiv \{j \in Q\} \tag{4.54}$$

$$\{j \in S : (j,i) \in A_S\} = \emptyset. \tag{4.55}$$

Then, constraints (4.46) become

$$\sum_{j \in Q_S} x_{ij} \leq p_i \quad \forall \ i \in P_S. \tag{4.56}$$

Analogously inequalities (4.48) change into

$$\sum_{i \in P_S} x_{ij} \leq q_j \quad \forall \ j \in Q_S, \tag{4.57}$$

and due to $A_S = P_S \times Q_S$ we get that Equation (4.51) becomes

$$x_{ij} \geq 0 \quad \forall \ i \in P_S, \ \forall \ j \in Q_S. \tag{4.58}$$

Constraints (4.47) and (4.49) are redundant, and restriction (4.50) disappears as $R = \emptyset$.

So, the problem we have to solve in order to find the value $v'(S)$ is

$$
\begin{aligned}
\max \quad & \sum_{i\in P_S}\sum_{j\in Q_S} b_{ij}x_{ij} \\
\text{s.t.:} \quad & \sum_{j\in Q_S} x_{ij} \le p_i \quad \forall\, i\in P_S \\
& \sum_{i\in P_S} x_{ij} \le q_j \quad \forall\, j\in Q_S \\
& x_{ij}\ge 0 \quad \forall\, i\in P_S,\ \forall\, j\in Q_S
\end{aligned}
\tag{4.59}
$$

which is the problem we have to solve to calculate $v(S)$. So, we conclude that $v'(S)=v(S)$ for all $S\subset N$.

□

The reciprocity of the previous theorem is not true, that is, there are Supply Chain Games that are not Transportation Games. Let us see an example.

Example 4.4.2 *Let us consider the SChG as depicted in Figure 4.6.*

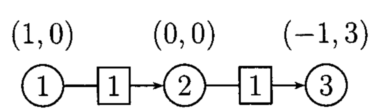

Figure 4.6: Supply Chain Game

One can see that $v(N)=1$, and $v(S)=0\ \forall\ S\neq N$. It is obvious that in a Transportation Game with $v(N)>0$, there exists a pair $\{i,j\}$, $i\in P$, $j\in Q$ such that $v(\{i,j\})>0$. This does not happen in this example, so we conclude that (N,v) is not a Transportation Game and, therefore, $SChG \not\subseteq TG$.

Since Assignment Games (AG for short) are a particular case of Transportation Games, we conclude that Assignment Games are Supply Chain Games too.

Then, it can be concluded that $TG \subsetneq SChG$ and $AG \subsetneq SChG$.

4.4.3 Shortest Path Games and SChG

This section studies the relationship between Supply Chain Games and Shortest Path Games. Since SChG have been proven monotonic, and in [23] it is shown that the class of SPG coincides with the class of monotonic games (MO for short), we conclude the first implication:

Proposition 4.4.3 SChG $\subset$ SPG.

Proof. Immediate from the fact that $MO \equiv SPG$ and SChG are monotonic. □

Unfortunately the converse is not true, as we can see from the following example borrowed from [23].

Example 4.4.3 *Consider the shortest path game* σ *given by* $N = \{1, 2\}$, $g = 6$ *and* Σ *and* o *as depicted in Figure 4.7.*

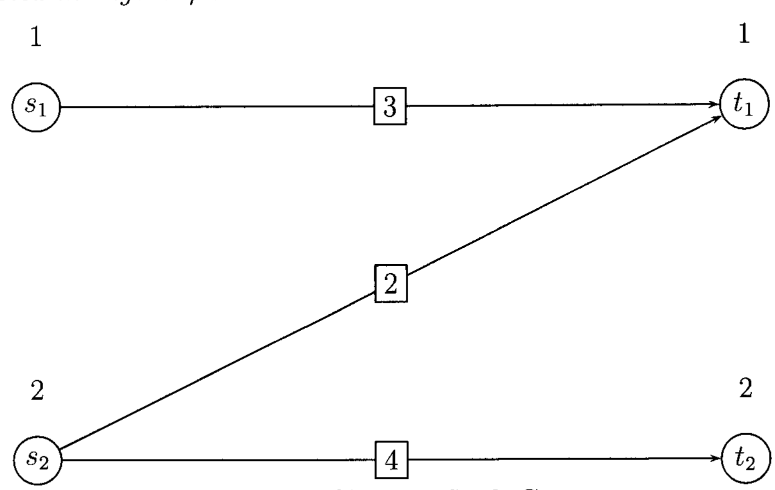

Figure 4.7: Shortest Path Game

The characteristic function v *for this game is*

$$v(\{1\}) = 3, \ v(\{2\}) = 2, \ v(\{1,2\}) = 4. \tag{4.60}$$

Clearly $C(N, v) = \emptyset$, *so* (N, v) *cannot be a Supply Chain Game, since SChG are balanced.*

Thus, the following result can be stated:

Proposition 4.4.4 *SPG* $\not\subseteq$ *SChG.*

Nevertheless, a subclass of SPG can be included in the class of SChG.

Proposition 4.4.5 Consider Shortest Path Game $\sigma = (\Sigma, N, o, g)$ where $\Sigma = (N, A, L, s, t)$ and $o(i) = i \ \forall \ i \in N$. If $|s| = 1$ or $|t| = 1$ then σ is a Supply Chain Game.

Proof. It is clear that, if $|s| = 1$, the objective for any profitable coalition is to send the only unit of commodity from s to the nearest $j \in t$. Analogously, if $|t| = 1$ then the optimal transportation plan would be to find the closest $i \in s$ to t and ship the unit that i offers to t. In both cases, by definition, that is the same objective as in SChG, and the result follows. □

In [23] some conditions for SPG to have non-empty core are given. From Proposition 4.4.5 we have the following result.

Theorem 4.4.1 If $|s| = 1$ or $|t| = 1$, then the corresponding SPG is balanced.

Proof. Immediate from the fact that those SPG are SChG, which are balanced. □

So far in this chapter a new class of games arising from a particular Supply Chain Problem has been presented, named Supply Chain Games. This class of games proved monotonic, superadditive and totally balanced. We also proved that $SChG \subsetneq FG$, $SChG \subsetneq LPrG$, $AG \subsetneq SChG$, $TG \subsetneq SChG$ and $SChG \subsetneq SPG$.

The following two sections keep analyzing SChG. The first one, Section 4.5, is devoted to study solution concepts in SChG. The second one, Section 4.6, analyzes the multicriteria extension of our Supply Chain Games.

4.5 Solution Concepts in Supply Chain Games

In this section several solution concepts for SChG are presented. We start by giving allocations based on duality properties: the Owen set and the Extended Owen set. Afterwards we shall study the Shapley value in SChG and its properties. Later on two new contributions are presented: another solution concept based on the proportional division of the general profit and a way to sequentially allocate the benefits generated. To finish the section, some experimental results are shown.

4.5.1 The Owen set

Since SChG have been proven to be Linear Programming Games (see Section 4.4.1), it makes sense to apply the Owen solution to SChG, as defined in Chapter 3.

In SChG the characteristic function is given by the optimal value of the linear problem (4.27). One can check that the dual problem of (4.27), which shall be called from now on $Dr(S)\ \forall\ S \subset N$, is the following linear program:

$$
\begin{array}{lll}
\min & \sum_{i \in P_S} b_i u_i - \sum_{j \in Q_S} b_j u_j + \sum_{(i,j) \in A_S^{\mathbb{R}}} v_{ij} h_{ij} & \\
\text{s.t.:} & (u_i - t_i) - (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in P_S \times P_S \\
& (u_i - t_i) + (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in P_S \times Q_S \\
& (u_i - t_i) - u_j + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in P_S \times R_S \\
& -(u_i - t_i) - (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in Q_S \times P_S \\
& -(u_i - t_i) + (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in Q_S \times Q_S \\
& -(u_i - t_i) - u_j + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in Q_S \times R_S \\
& u_i - (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in R_S \times P_S \\
& u_i + (u_j - t_j) + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in R_S \times Q_S \\
& u_i - u_j + v_{ij} \geq k_j - k_i - c_{ij} & \forall\ \{i,j\} \in R_S \times R_S \\
& u_i, t_i \geq 0\ \forall\ i \in P_S \cup Q_S & \\
& v_{ij} \geq 0\ \forall\ (i,j) \in A_S^{\mathbb{R}} &
\end{array}
\tag{4.61}
$$

where $A_S^{\mathbb{R}} = \{(i,j) \in A_S : h_{ij} < +\infty\}$. Note that variables v_{ij} do not make sense when $h_{ij} = +\infty$ and that the above constraints are only valid for those pairs $\{i,j\}$ such that

$(i,j) \in A$.

Let us consider Problem (4.61) with $S = N$, and let $((u^*)_i, (t^*)_i, (v^*)_{ij})$ be an optimal feasible solution to it. We shall see that the allocation $\alpha = (\alpha_1, \dots, \alpha_n) \in C(N, v)$, where

$$\alpha_i = |b_i| u_i^* + \frac{1}{2}\{ \sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* + \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji} v_{ji}^* \} \quad \forall\, i \in N. \tag{4.62}$$

1. First we see that such allocation is efficient.

(4.63)

$$\begin{aligned}
\alpha(N) &= \sum_{i\in N} \alpha_i = \sum_{i\in N} (|b_i| u_i^* + \frac{1}{2}\{ \sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* + \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji} v_{ji}^* \}) \\
&= \sum_{i\in P} b_i u_i^* + \sum_{j\in Q} (-b_j) u_j^* + \frac{1}{2} \sum_{i\in N} \sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* + \frac{1}{2} \sum_{i\in N} \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji} v_{ji}^* \\
&= \sum_{i\in P} b_i u_i^* - \sum_{j\in Q} b_j u_j^* + \frac{1}{2} \sum_{(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* + \frac{1}{2} \sum_{(j,i)\in A^{\mathbb{R}}} h_{ji} v_{ji}^* \\
&= \sum_{i\in P} b_i u_i^* - \sum_{j\in Q} b_j u_j^* + \sum_{(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* = v(N).
\end{aligned}$$

2. Now it will be proven that no coalition can improve the payoff they receive from α by acting by themselves.

$$\alpha(S) = \sum_{i\in S} \alpha_i = \sum_{i\in S} |b_i| u_i^* + \frac{1}{2} \sum_{i\in S} \sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij} v_{ij}^* + \frac{1}{2} \sum_{i\in S} \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji} v_{ji}^*. \tag{4.64}$$

Given $i \in S$, one has that $\{j : (i,j) \in A_S^{\mathbb{R}}\} \subset \{j : (i,j) \in A^{\mathbb{R}}\}$ and $\{j : (j,i) \in A_S^{\mathbb{R}}\} \subset \{j : (j,i) \in A^{\mathbb{R}}\}$. It is also clear that $h_{ij} v_{ij}^* \geq 0 \ \forall\, (i,j) \in A^{\mathbb{R}} \supseteq A_S^{\mathbb{R}}$. Thus, we have that (4.64) is greater than or equal to

$$\begin{aligned}
&\sum_{i\in S} |b_i| u_i^* + \frac{1}{2} \sum_{i\in S} \sum_{j:(i,j)\in A_S^{\mathbb{R}}} h_{ij} v_{ij}^* + \frac{1}{2} \sum_{i\in S} \sum_{j:(j,i)\in A_S^{\mathbb{R}}} h_{ji} v_{ji}^* \\
&= \sum_{i\in P_S} b_i u_i^* - \sum_{i\in Q_S} b_i u_i^* + \frac{1}{2} \sum_{(i,j)\in A_S^{\mathbb{R}}} h_{ij} v_{ij}^* + \frac{1}{2} \sum_{(j,i)\in A_S^{\mathbb{R}}} h_{ji} v_{ji}^* \\
&= \sum_{i\in P_S} b_i u_i^* - \sum_{i\in Q_S} b_i u_i^* + \sum_{(i,j)\in A_S^{\mathbb{R}}} h_{ij} v_{ij}^*.
\end{aligned} \tag{4.65}$$

Let (u^{*S}, t^{*S}, v^{*S}) be an optimal solution to $Dr(S)$. We obviously have that (u^*, t^*, v^*)

is feasible for $Dr(S)$. Then we deduce that (4.65) is greater than or equal to

$$\sum_{i\in P_S} b_i u_i^{*S} - \sum_{i\in Q_S} b_i u_i^{*S} + \sum_{(i,j)\in A_S^{\mathbb{R}}} h_{ij} v_{ij}^{*S} = v(S). \tag{4.66}$$

Thus, we have proven that

$$\alpha(S) \geq v(S) \ \forall\ S \subset N. \tag{4.67}$$

By joining (1) and (2) we conclude that α is in the core of the game. Let us see an example of how to calculate the such allocation for Supply Chain games.

Example 4.5.1 *Consider the SChP in Figure 4.8. One can check that the corresponding*

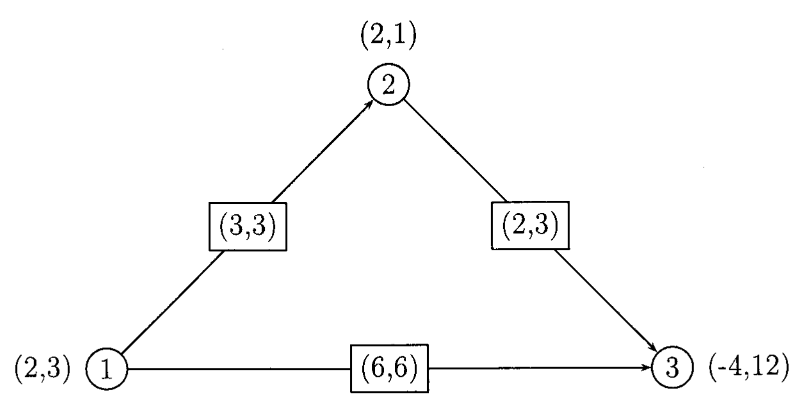

Figure 4.8: Supply Chain Problem with Capacity Constraints

problem $Dr(N)$ is:

$$\begin{array}{ll} \min & 2u_1 + 2u_2 + 4u_3 + 3v_{12} + 6v_{13} + 3v_{23} \\ s.t.: & (u_1 - t_1) - (u_2 - t_2) + v_{12} >= 1 - 3 - 3 \\ & (u_1 - t_1) + (u_3 - t_3) + v_{13} >= 12 - 3 - 6 \\ & (u_2 - t_2) + (u_3 - t_3) + v_{23} >= 12 - 1 - 2 \\ & u_i, t_i \geq 0,\ v_{ij} \geq 0 \end{array} \tag{4.68}$$

and one optimal feasible solution is:

$$u_1^* = 3,\ u_2^* = 8,\ u_3^* = 0,\ t_1^* = t_2^* = t_3^* = 0,\ v_{12}^* = 0,\ v_{13}^* = 0,\ v_{23}^* = 1. \tag{4.69}$$

The following core allocation derives from that optimal feasible solution

$$\begin{aligned} \alpha_1 &= 3 \cdot 2 = 6 \\ \alpha_2 &= 8 \cdot 2 + \tfrac{1}{2}(1 \cdot 3) = 17.5 \\ \alpha_3 &= 0 \cdot 4 + \tfrac{1}{2}(1 \cdot 3) = 1.5 \end{aligned} \tag{4.70}$$

Another interesting property of the Owen allocations is their computational efficiency. Since (4.61) can be solved in polynomial time, (see [70]), we have a procedure that provides a core allocation in polynomial time.

Theorem 4.5.1 An allocation in the core of Supply Chain Games can be computed in polynomial time.

Proof. The above argument. □

The set consisting of every allocation obtained from an optimal solution of $Dr(N)$ following the process above is the well-known Owen Set, see [47], that was studied for linear production games in Section 3.1. Let us now present a formal definition of the Owen set for the class of Supply Chain games. We will use the following notation:

Let (N, A, C, b, k, H) be a SChG.

- $O_{\min}(N, A, C, b, k, H)$ is the set of optimal solutions to the corresponding problem $Dr(N)$.
- $O_{\max}(N, A, C, b, k, H)$ is the set if optimal solutions to the corresponding problem $Pr(N)$.

The Owen set of a Supply Chain Game is:

$$Owen(N, A, C, b, k, H) = \{\alpha^{y,t,v} = (\alpha_1^{y,t,v}, \ldots, \alpha_n^{y,t,v}) : (y, t, v) \in O_{\min}(N, A, C, b, k, H)\} \tag{4.71}$$

where $\alpha_i^{y,t,v} = y_i b_i + \frac{1}{2}\{\sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij}v_{ij} + \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji}v_{ji}\}$. The Owen set is characterized for the class of linear production games as the only solution concept satisfying the following axioms, see [24]:

1. *One person efficiency.* It says that if there is only one agent owning one unit of all resources, then a solution concept satisfying one person efficiency assigns to the player the maximal profit that can be made from his resource bundle.

2. *Rescaling.* It means that the solution rule is independent of the units in which the resources are measured.

3. *Shuffle.* This axiom says that if the resources are shuffled among the agents, then the solution rule changes in the same way.

4. *Consistency.* This property has to do with the following situation: suppose that the solution rule divides the profit according to a vector y, then agent i takes y_i monetary units and leaves. Afterwards his resource can be used by other players at a cost of y_i per unit. A solution concept satisfies consistency if the restriction of y to the remaining agents is a solution to the reduced linear production process.

5. *Deletion.* It says that if a production technology is not needed to make the maximal profit $v(N)$, this technology can be omitted and the solutions to the old situation are also solutions to the new game.

which are the same axioms we needed to characterize the Extended Owen set for linear programming games. From the fact that the class of LP games coincide with the class of totally balanced games, and SChG are totally balanced, it is clear that $SChG \subset LP$ games. Then, it is concluded that the Owen set for SChG satisfies all those properties.

Nevertheless, Owen allocations sometimes presents similar fairness problems as it does in LP games, such as giving to a veto player, see Definition 2.1.6, a payoff equal to zero. See the following example:

Example 4.5.2 *Let us consider the Supply Chain Game given by the Supply Chain Problem in Figure 4.9. The characteristic function of this game is:*

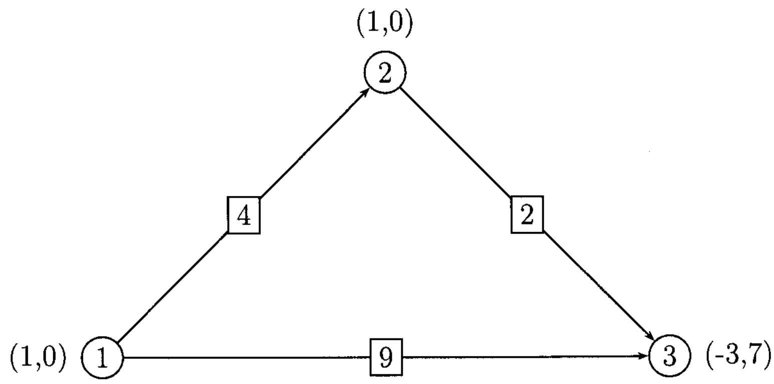

Figure 4.9: Transportation Network

(4.72)

S	*{1}*	*{2}*	*{3}*	*{1,2}*	*{1,3}*	*{2,3}*	*{1,2,3}*
v(S)	*0*	*0*	*0*	*0*	*0*	5	6

One can check that the dual linear program $Dr(N)$ for this game is:

$$\begin{aligned} \max \quad & u_1 + u_2 + 3u_3 \\ s.t.: \quad & (u_1 - t_1) - (u_2 - t_2) \geq -4 \\ & (u_1 - t_1) + (u_3 - t_3) \geq -2 \\ & (u_2 - t_2) + (u_3 - t_3) \geq 5 \\ & u_i, t_i \geq 0 \;\; i = 1, 2, 3 \end{aligned} \tag{4.73}$$

The only optimal solution to this problem is

$$u^* = (1, 5, 0), \;\; t^* = (0, 0, 0) \tag{4.74}$$

and the corresponding Owen allocation is

$$\gamma = (1, 5, 0). \tag{4.75}$$

Clearly γ is an allocation in the core of the game. But let us note that, even though player 3 is a veto player ($v(\{1,2\}) = 0$), it receives nothing after this allocation ($\gamma_3 = 0$).

One could be tempted to think that all the possible allocations in the core of SChG can be obtained from an optimal solution to the dual problem $Dr(N)$. The following example shows that this supposition is not true.

Example 4.5.3 *Consider the SChG* (N, v) *in 4.5.2. It is easy to check that* $C(N, v)$ *is the convex hull of* $\{(0,6,0),(0,0,6),(1,0,5),(1,5,0)\}$. *Take the core allocation* $(0,6,0)$. *If this allocation came from an optimal solution to* $Dr(N)$, *this solution would be* $(\overline{u}, \overline{t})$ *with* $\overline{u} = (0,6,0)$. *The vector* $(\overline{u}, \overline{t})$ *is not even a feasible point in the dual problem for any* $\overline{t} \in \mathbb{R}^3_+$.

So, we deduce that there are core allocations that cannot be obtained from the optimal solutions of the dual problem (4.61) for $S = N$, that is, $Owen(N, A, C, b, k, H) \neq C(N, A, C, b, k, H)$.

Computational Complexity

Now we prove that the computational cost to calculate one Owen allocation is polynomial with respect to the number of players.

Theorem 4.5.2 Let (N, v) be a SChG. Then the computational complexity for calculating one Owen solution is polynomial with respect to the number of players, and given by

$$O(n^2). \tag{4.76}$$

Proof. To calculate one Owen allocation we have to solve Problem (4.61) with $S = N$. This problem has $(n+q)$ variables, where $q = |A|$. That means that the computational cost of calculating one solution to this dual program is $O[(n+q)]$, see Predictor-Corrector algorithm. Afterwards we have to assign to each player their allocation, which can be done in $O(n)$.

If we suppose that q, the number of arcs, is $O(n^2)$, the computational complexity of the Owen solution can be stated only in terms of the number of players, exactly $O(n^2)$. □

In the following section we apply to SChG a solution concept that tries to solve the problems of the Owen set expressed in Example 4.5.2, the Extended Owen set as introduced in Section 3.2. Such solution concept was developed with the idea of solving the problem arising when players do not send all the material they can offer (or do not receive all the material they require) and their arcs do not use all their capacities. Note that a player i in this situation receives nothing from Owen allocations, as the corresponding variables u_i, v_{ij}, v_{ji} are null in any optimal solution to $Dr(N)$.

4.5.2 The Extended Owen set

In this section, the Extended Owen solution is adapted to our Supply Chain Games. We first start by defining the associated games to the optimal solutions of $Pr(N)$.

Definition 4.5.1 *Let (N, A, C, b, k, H) be a SChP and x^* an optimal distribution plan. We define the reduced game of (N, A, C, b, k, H) associated to x^* by $(N, A, C, b^{x^*}, k, H^{x^*})$, where*

$$\begin{aligned} b_i^{x^*} &= \textstyle\sum_{j:(i,j)\in A} x_{ij}^* - \sum_{j:(j,i)\in A} x_{ji}^* \ \forall\ i\ \in P \\ b_i^{x^*} &= \textstyle\sum_{j:(j,i)\in A} x_{ji}^* - \sum_{j:(i,j)\in A} x_{ij}^* \ \forall\ i \in Q \\ h_{ij}^* &= x_{ij}^* \ \forall\ (i,j)\ \in A \end{aligned} \tag{4.77}$$

Note that in each of those games, players reduce their demand or offer until the optimal solution. They also make agreements to reduce the capacity of the arcs. This way, the corresponding dual variables in $Dr(N)$ are not forced to be zero.

Given those games, the Extended Owen set can be applied to the class of Supply Chain Games. We will only introduce the Extended Owen set for SChG whose corresponding SChP has only one optimal distribution plan. The extension to general SChG can be done analogously to the line followed for linear production games.

Definition 4.5.2 *Given a SChG (N, v) arising from a SChP (N, A, C, b, k, H), the Extended Owen set for such game is denoted by $EOwen(N, A, C, b, k, H)$ and defined via*

$$\{\alpha^{u,t,v,x^*} = (u, t, v) \in O_{\min}(N, A, C, b^{x^*}, k, H^{x^*}), x^* = O_{\max}(N, A, C, b, k, H)\} \tag{4.78}$$

where

$$\alpha_i^{u,t,v,x^*} = u_i b_i^{x^*} + \frac{1}{2}\{ \sum_{j:(i,j)\in A^{\mathbb{R}}} h_{ij}^{x^*} v_{ij} + \sum_{j:(j,i)\in A^{\mathbb{R}}} h_{ji}^{x^*} v_{ji}\}. \tag{4.79}$$

Just like for LP games, it is easy to see that

$$EOwen(N, A, C, b, k, H) = Owen(N, A, C, b^{x^*}, k, H^{x^*}). \tag{4.80}$$

Therefore, to calculate allocations in the Extended Owen set, one just needs to find allocations in the Owen set of the corresponding reduced game. Let us see an example.

Example 4.5.4 *Consider the Supply Chain Game in Example 4.5.2. The optimal distribution plan of the corresponding Supply Chain Problem is* $x^*_{12} = 1, x^*_{13} = 0, x^*_{23} = 1$. *Therefore the corresponding reduced game associated to this distribution plan is given by the SChP in Figure 4.10.*

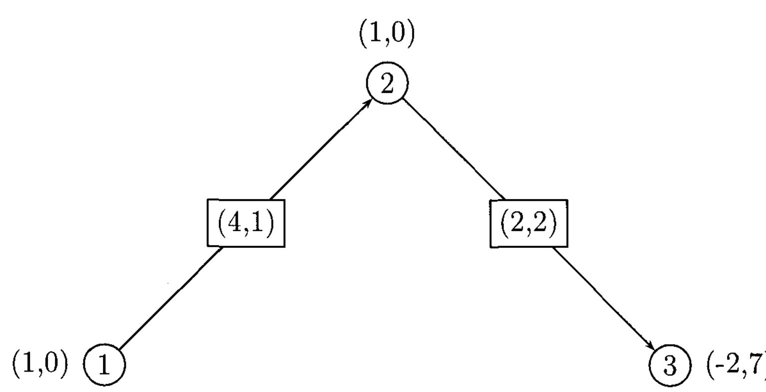

Figure 4.10: Transportation Network

Note that now arcs $(1, 2)$ *and* $(2, 3)$ *have reduced their capacities until* x^*_{12} *and* x^*_{23}, *respectively, and that arc* $(1, 3)$ *does not exist any more (or it does exist but having capacity equal to zero.) Note as well that node 3 only demands 2 units of material. The* *corresponding dual problem of this reduced games is:*

$$\begin{array}{ll} \max & u_1 + u_2 + 2u_3 \\ s.t.: & (u_1 - t_1) - (u_2 - t_2) + v_{12} \geq -4 \\ & (u_2 - t_2) + (u_3 - t_3) + v_{23} \geq 5 \\ & u_i, t_i, v_{ij} \geq 0. \end{array} \tag{4.81}$$

One optimal solution to this problem is $u_1^* = 0.5, u_2^* = 4.5, u_3^* = 0.5$, *the other variables being zero. The Extended Owen solution arising from this dual solution is* $(0.5, 4.5, 1)$, *which gives a positive payoff to all players.*

In the following section another allocation for SChG is explored: the Shapley value.

4.5.3 The Shapley value

In previous sections we have discussed about the core of SChG, which is, in general, not a unique point but a set. An alternative way to allocate the payoff generated by the cooperation of the grand coalition is by an allocation rule, i.e., a criterion which assigns to every SChG an allocation of the benefit to the players. One of the most important allocation rules treated by the game theory literature is the Shapley value, see [22].

The Shapley value, denoted by ϕ from now on, is a solution concept in cooperative games that always is a preimputation ($\phi(N) = v(N)$), if the game is monotonic it is also an imputation ($\phi_i \geq v(\{i\})\ \forall\ i$) and if the game is convex then it is an allocation in the core of the game ($\phi(S) \geq v(S)\ \forall\ S$). SChG have proved monotonic but not convex, thus it can be concluded that ϕ is always an imputation of SChG. But in general it cannot be stated that the Shapley value is a core allocation for SChG. A counterexample will show that in general $\phi \notin C(N, v)$ for every $(N, v) \in SChG$.

We shall now provide an axiomatic characterization of the Shapley value for the class of Supply Chain Games based on some axioms related to the Supply Chain Problem that gives rise to the game. Both these axioms and the proof of the characterization are similar to those that Myerson used to characterize the Shapley value for games with graph-restricted communication, see [42] and [43], and to those in [23].

Axiomatic Characterization

Let us see first some previous definitions.

Given a Supply Chain Problem (N, A, C, b, k, H), we say that:

- $i \in N$ and $j \in N$ are adjacent in $B \subset A$ if and only if $(i, j) \in B$.
- $i \in N$ and $j \in N$ are connected in $B \subset A$ if and only if there exists $\{i_0, \ldots, i_r\} \subset N$ such that $(i_{k-1}, i_k) \in B \; \forall \; k = 1, \ldots, r, \quad i_0 = i, \; i_r = j$.
- $i \in N$ and $j \in N$ are unconnected if they are not connected.
- A node is isolated in B if it is not an endnode of any arc in B.
- $\forall \; i \in N, \; \forall \; B \subset A, \; B_{-i}$ is the set of arcs of B which do not have i as an endnode, i.e.:

$$B_{-i} = \{a = (j, k) \in B \; / \; i \neq j, \text{and } i \neq k\}. \tag{4.82}$$

- (N, v_B) is the Supply Chain Game (N, B, C_B, b, k, H_B), where C_B and H_B are the restrictions of C and H to B.

That is, (N, v_B) is the Supply Chain Game where only the arcs in B are available.

An allocation rule over the class SChG is a map $\Phi \; : 2^A \longrightarrow \mathbb{R}^n$ that assigns to every SChG an allocation of the general benefit. For every $B \subset A$, $\Phi_i(B)$ is the allocation proposed by Φ for player i in the game (N, v_B). The following axioms are considered:

1. Efficiency (EFF). Φ is said to satisfy EFF if,

$$\sum_{i \in N} \Phi_i(B) = v_B(N) \quad \forall B \subset A. \tag{4.83}$$

   Efficiency means that the allocation rule allocates all the benefit generated by the cooperation of all players.

2. Irrelevant Player Property (IPP). Φ is said to satisfy IPP if, for all $B \subset A$ the following equality

$$\Phi_i(B) = 0 \tag{4.84}$$

holds $\forall\ i \in N$ isolated in B. The irrelevant players property states that irrelevant players (those that are isolated nodes) should not receive anything.

3. Fairness for Unconnected Players (FUP). Φ is said to satisfy FUP if, for all $B \subset A$ and for all $i, j \in N$ unconnected in B, then

$$\Phi_i(B) - \Phi_i(B_{-j}) = \Phi_j(B) - \Phi_j(B_{-i}). \tag{4.85}$$

Fair treatment for unconnected players states that, for any pair i and j unconnected in B, i must gain or loose when j leaves the same as j gains or looses when i leaves.

4. Fairness for Adjacent Players (FAP). Φ is said to satisfy FAP if, for all $B \subset A$ and for all $a = (i, j) \in B$ the next equality

$$\Phi_i(B) - \Phi_i(B \setminus a) = \Phi_j(B) - \Phi_j(B \setminus a) \tag{4.86}$$

holds. Fairness for adjacent players means that, when an arc is added or deleted, the players owning its endnodes benefit or suffer equally.

The following result provides an axiomatic characterization of the Shapley value from the previous four axioms.

Theorem 4.5.3 In Supply Chain Games, there exists a unique allocation rule Ψ satisfying EFF, IPP, FUP and FAP. It is given by

$$\Psi_i(B) = \phi_i(v_B) \tag{4.87}$$

for all $B \subset A$ and all $i \in N$, where ϕ denotes the Shapley value.

Proof. Let ϕ be the Shapley value. Let (N, A, C, b, k) be a Supply Chain Problem and (N, v) its corresponding game. The proof consists of two parts. In the first one we shall prove that the Shapley value satisfies the four axioms above. In the second part we shall prove that there can be only one allocation rule satisfying those axioms.

1. Let us first see that the Shapley value satisfies those four properties.

   - ϕ satisfies EFF and IPP because it is always efficient and satisfies the dummy player property.
   - Let us see that the Shapley value satisfies FUP. Let $B \subset A$, $i, j \in N$ disconnected in B. We define the TU-game (N, w) with $w = v_B - v_{B_{-i}} - v_{B_{-j}}$. For all $S \subset N \setminus \{i, j\}$ one has that

     $$w(S \cup \{i\}) = v_B(S \cup \{i\}) - v_{B_{-i}}(S \cup \{i\}) - v_{B_{-j}}(S \cup \{i\}). \tag{4.88}$$

     It is true that

     - $v_{B_{-i}}(S \cup \{i\}) = v_B(S)$, since in both cases the coalition S acts without the arcs of i, due to the fact that $i \notin S$.
     - $v_{B_{-j}}(S \cup \{i\}) = v_B(S \cup \{i\})$, since $i \notin S \cup \{j\}$.

     So, (4.88) becomes

     $$v_B(S \cup \{i\}) - v_B(S) - v_B(S \cup \{i\}) = -v_B(S). \tag{4.89}$$

     Analogously we deduce that $w(S \cup \{j\}) = -v_B(S)$.
     Hence, players i and j are symmetric in w and, by the symmetry of the Shapley value, $\phi_i(w) = \phi_j(w)$. Thus, from the linearity of ϕ we have that

     $$\phi_i(v_B) - \phi_i(v_{B_{-i}}) - \phi_i(v_{B_{-j}}) = \phi_j(v_B) - \phi_j(v_{B_{-i}}) - \phi_j(v_{B_{-j}}) \tag{4.90}$$

     and, since $\phi_i(v_{B_{-i}}) = \phi_j(v_{B_{-j}}) = 0$, because the Shapley value satisfies the irrelevant player property, we get that

     $$\phi_i(v_B) - \phi_i(v_{B_{-j}}) = \phi_j(v_B) - \phi_j(v_{B_{-i}}), \tag{4.91}$$

     what proves that ϕ satisfies FUP.

- Let us see that ϕ satisfies FAP. Consider $B \subset A$, $a = (i, j) \in B$. We have to prove (4.86).

  (a) If $i = j$ the result is obvious.

  (b) Suppose that $i \neq j$ and define the TU-Game (N, w) where

  $$w = v_B - v_{B \setminus a}. \tag{4.92}$$

  If $S \subset N$ and $i \notin S$ or $j \notin S$, clearly

  $$v_B(S) = v_{B \setminus a}(S) \Rightarrow w(S) = 0, \tag{4.93}$$

  (since S cannot use arc a in any case). Hence, players i and j are symmetric in (N, w). From the symmetry of the Shapley value, we get that $\phi_i(w) = \phi_j(w)$. From the linearity of the Shapley value one has that

  $$\phi_i(v_B) - \phi_i(v_{B \setminus a}) = \phi_i(v_B - v_{B \setminus a}) = \phi_i(w) \quad \forall\, i \in N. \tag{4.94}$$

  Joining these equalities we get that

  $$\phi_i(v_B) - \phi_i(v_{B \setminus a}) = \phi_i(w) = \phi_j(w) = \phi_j(v_B) - \phi_j(v_{B \setminus a}), \tag{4.95}$$

  as we wanted to prove. So Equation (4.86) holds and we conclude that ϕ satisfies the Fairness of Adjacent Players property.

2. Suppose that there exist R^1 and R^2 two different allocation rules satisfying EFF, IPP, FUP and FAP. Let $B \subset A$ the minimal set such that $R^1(B) \neq R^2(B)$. Let M be the set

$$M = \{i \in N \;/\; i \text{ is not isolated in } B\}. \tag{4.96}$$

Let us prove that M has at least two elements.

- Since R^1 and R^2 satisfy IPP we have that

$$R^1_i(B) = R^2_i(B) = 0 \quad \forall\, i \in N \setminus M. \tag{4.97}$$

Hence, if $M = \emptyset$ then

$$R_i^1(B) = R_i^2(B)\ \forall\ i \in N \Rightarrow R^1(B) = R^2(B)\ !!! \tag{4.98}$$

- Also, if $M = \{i\}$, one has that $R_j^1(B) = R_j^2(B) = 0\ \forall\ j \neq i$, since R^1 and R^2 satisfy IPP. Due to the fact that R^1 and R^2 satisfy EFF, one gets that (4.99)
$$R_i^1(B) = \sum_{j \in N} R_j^1(B) = v_B(N) = \sum_{j \in N} R_j^2(B) = R_i^2(B) \Rightarrow R^1(B) = R^2(B)\ !!!$$

So, we have proven that $|M| \geq 2$.

Let $i, j \in M$. It can happen that:

- i, j are adjacent in $B \Rightarrow \exists\ a \in B\ :\ a = (i, j)$. Hence, since R^1 and R^2 satisfy FAP,

$$\left.\begin{array}{rl} R_i^1(B) - R_i^1(B \setminus a) = R_j^1(B) - R_j^1(B \setminus a) \\ R_i^2(B) - R_i^2(B \setminus a) = R_j^2(B) - R_j^2(B \setminus a) \\ \text{B is minimal } \Rightarrow \begin{array}{r} R_i^1(B \setminus a) = R_i^2(B \setminus a) \\ R_j^1(B \setminus a) = R_j^2(B \setminus a) \end{array} \end{array}\right\} \tag{4.100}$$

This concludes that $R_i^1(B) - R_i^2(B) = R_j^1(B) - R_j^2(B)$.

- i, j are not adjacent but they are connected in B, $\exists\ \{i_0, \ldots, i_q\}\ :\ (i_{k-1}, i_k) \in B\ \forall\ k = 1, \ldots, q,\ i = i_0,\ j = i_q$. If we repeat the previous step q times, we obtain

$$R_i^1(B) - R_i^2(B) = R_j^1(B) - R_j^2(B). \tag{4.101}$$

- i, j are not connected in B, since R^1 and R^2 satisfy FUP we have that:

$$\left.\begin{array}{l} \left.\begin{array}{r} R_i^1(B) - R_i^1(B_{-j}) = R_j^1(B) - R_j^1(B_{-i}) \\ R_i^2(B) - R_i^2(B_{-j}) = R_j^2(B) - R_j^2(B_{-i}) \end{array}\right\} \\ \text{B is minimal } \Rightarrow \left.\begin{array}{r} R_i^1(B_{-j}) = R_i^2(B_{-j}) \\ R_j^1(B_{-i}) = R_j^2(B_{-i}) \end{array}\right\} \end{array}\right\} \tag{4.102}$$

which implies that $R_i^1(B) - R_i^2(B) = R_j^1(B) - R_j^2(B)$.

Then $R_i^1(B) - R_i^2(B) = R_j^1(B) - R_j^2(B) = c \in \mathbb{R}$ for all $i, j \in M$. We have that:

(4.103)
$$\left.\begin{array}{l}\sum_{k\in N} R_k^1(B) = v_B(N) \\ \\ \sum_{k\in N} R_k^2(B) = v_B(N)\end{array}\right\} \Rightarrow 0 = \sum_{k\in N}(R_k^1(B) - R_k^2(B)) = \sum_{k\in M}(R_k^1(B) - R_k^2(B)) = c|M|.$$

It follows that

$$c = 0 \Rightarrow R_i^1(B) = R_i^2(B) \ \forall\, i \in M, \tag{4.104}$$

and we already knew that $R_i^1(B) = R_i^2(B) \ \forall\, i \in N \setminus M$, so we conclude that

$$R^1(B) = R^2(B)!!! \tag{4.105}$$

By joining (1) and (2) the proof is finished. □

Supply Chain Games are monotonic, so we assure that the Shapley value is an imputation of the game, but they are not, in general, convex, so there is no theoretical reason to assure that $\phi \in C(N, v)$. Example 4.5.5 shows a SChG in which ϕ is not a core allocation. Nevertheless, due to the properties satisfied by the Shapley value, it is a well considered solution concept.

Example 4.5.5 *Consider the Supply Chain Problem described in Figure 4.11. One can see that the distribution plan*

$$x_{25} = 4, \ x_{31} = 1, \ x_{32} = 3 \tag{4.106}$$

is optimal and gives a profit of 14 units. The Shapley value is

$$\phi = (2, 5.166, 3.25, 0.0833, 3.5). \tag{4.107}$$

In this case, $\phi \notin C(N, v)$, since

$$v(\{1, 2, 3, 5\}) = 14 \quad x(\{1, 2, 3, 5\}) = 13.9166. \tag{4.108}$$

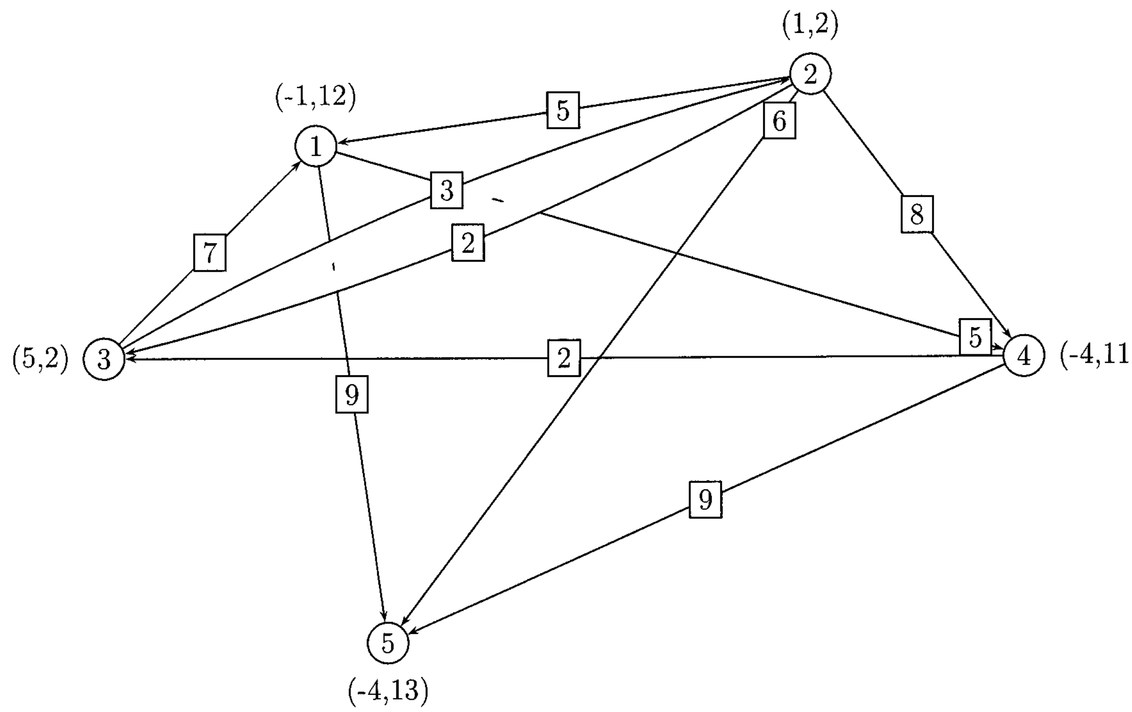

Figure 4.11: Transportation Network

As a remark, note that the previous example gives a second SChG that is not convex. This comes from the fact that in convex games the Shapley value is always a core allocation.

Computational Complexity

We can state the following result:

Theorem 4.5.4 Let (N, v) be a SChG. Then the computational complexity for calculating Shapley value is exponential with respect to the number of players, and given by

$$O(n^2 2^n). \tag{4.109}$$

Proof. In order to calculate the Shapley value one has to know the characteristic function of the game, that is to say, one has to solve $O(2^n)$ linear programming problems. Calculating $v(S)$ implies solving a linear programming problem with j constraints and q_j

variables, where $j = |S|$ and $q_j = |A_S|$. Again supposing that $q_j = O(j^2)$, one has that the effort to calculate v is given by:

$$\sum_{j=1}^{n} \binom{n}{j} O(j^2) = O(n^2 2^n). \tag{4.110}$$

Note that Predictor-Corrector algorithm complexity has been used to develop that formula.

After calculating v we have to allocate to each player their payoff, according to the formula of the Shapley value. But this operation does not increase the theoretical complexity of the final calculation. □

Besides not giving stable allocations in general, the main problem with the Shapley value is the necessary computation time, since we have to calculate the value of the characteristic function for every coalition. In our special case of supply chain game, that means solving a number of linear programs which is exponential with respect to the number of players.

Thus, we need to look for other solution concepts of easier calculation. At the same time such solutions must satisfy good properties. In the following section we shall explore an allocation rule based on optimal distribution plans over the complete graph of the Supply Chain Problem associated to the game.

4.5.4 Arc-Proportional solution

Due to the fact that the Owen solution, provided in previous sections from the dual program, gives rise to "fairness" problems, even though it is an allocation in the core, and that in order to work out the Shapley value we need to know the complete characteristic function of the game, which is not always computationally possible, we are going to propose in this section a new solution concept: the Arc-Proportional solution. This solution is a generalization of the *Arc-Equality Solution* given in [59] for Transportation Games.

We start by giving its definition.

Definition 4.5.3 *Let (N, A, C, b, k, H) be a Supply Chain Problem, and let x^* be an optimal distribution plan. Let (N, v) be the associated game. The allocation $\gamma(x^*)$ of the* *game (N, v) is:*

$$\gamma_i(x^*) = \frac{1}{2}\sum_{j:(i,j)\in A}\left(\frac{L(x^*)}{T(x^*)} - c_{ij}\right)x^*_{ij} + \frac{1}{2}\sum_{j:(j,i)\in A}\left(\frac{L(x^*)}{T(x^*)} - c_{ji}\right)x^*_{ji},\ \forall\ i \in N \tag{4.111}$$

where

$$L(x^*) = \sum_{(i,j)\in A}(k_j - k_i)x^*_{ij}, \quad T(x^*) = \sum_{(i,j)\in A} x^*_{ij}. \tag{4.112}$$

The Arc-Proportional set is:

$$\Omega = \{\gamma(x^*)\ :\ x^* \textit{ is optimal for } Pr(N)\}. \tag{4.113}$$

$L(x^*)$ and $T(x^*)$ are interpreted as the benefit without taking into account the transportation costs after the distribution plan x^* and the total amount of material transported between nodes, respectively. Note that both in $L(x^*)$ and $T(x^*)$ the material might be counted more than once, if it flows through different arcs from a supply node to a demand node. From now on, we may refer to γ, L and T instead of $\gamma(x^*)$, $L(x^*)$ and $T(x^*)$ as long as it is clear which the optimal distribution plan x^* is.

Let us see now that the Arc-Proportional set as defined above is a generalization of the Arc-Equality set for Transportation Games as proposed in [59].

Given a Transportation Game (P, Q, B, p, q), the Arc-Equality set for this game is:

$$AI(P, Q, B, p, q) = \{\frac{1}{2}\sum_{j\in Q} b_{1j}\mu_{1j}, \dots, \frac{1}{2}\sum_{j\in Q} b_{nj}\mu_{nj}; \frac{1}{2}\sum_{i\in P} b_{i1}\mu_{i1}, \dots, \frac{1}{2}\sum_{i\in P} b_{im}\mu_{im}\ :\ \mu \in \Omega\} \tag{4.114}$$

where Ω is the set of optimal transport plans for the transportation problem associated to the game.

In previous sections we saw that the transportation game (P, Q, B, p, q) coincides with the Supply Chain Game $(P \cup Q, P \times Q, -B, (p, q), 0, H)$ where $h_{ij} = +\infty\ \forall\ i \in$

$P, j \in Q$. Since

$$\{j : (i,j) \in P \times Q\} \equiv \{j \in Q\},\ \{j : (j,i) \in P \times Q\} \equiv \emptyset \quad \forall\, i \in P \tag{4.115}$$

and $L = 0$, the Arc-Proportional solution results, for $i \in P$,

$$\delta_i = \frac{1}{2} \sum_{j:(i,j)\in P\times Q} \left(\frac{L}{T} - c_{ij}\right) x^*_{ij} + \frac{1}{2} \sum_{j:(j,i)\in P\times Q} \left(\frac{L}{T} - c_{ji}\right) x^*_{ji} = -\frac{1}{2}\sum_{j\in Q} x^*_{ij}c_{ij} = \frac{1}{2}\sum_{j\in Q} x^*_{ij}b_{ij}. \tag{4.116}$$

Analogously, for every $j \in Q$ we have

$$\delta_j = \frac{1}{2} \sum_{i:(i,j)\in P\times Q} \left(\frac{L}{T} - c_{ij}\right) x^*_{ij} + \frac{1}{2} \sum_{i:(j,i)\in P\times Q} \left(\frac{L}{T} - c_{ji}\right) x^*_{ji} = -\frac{1}{2}\sum_{i\in Q} x^*_{ij}c_{ij} = \frac{1}{2}\sum_{i\in Q} x^*_{ij}b_{ij}. \tag{4.117}$$

Since both solutions coincide, we conclude that the Arc-Proportional solution proposed for the class of Supply Chain Games is a generalization of the Arc-Equality solution proposed for the class of Transportation Games in [59].

Proposition 4.5.1 The Arc-Proportional solution proposed for SChG is an extension of the Arc-Equality solution proposed for Transportation Games.

Proof. The discussion above. □

Now we are going to see some properties that the Arc-Proportional solution satisfies.

Properties

In this section we explore some properties for the Arc-Proportional solution. It will be shown that it is an efficient allocation, that is, a preimputation. We shall also see that, under certain conditions, it is an imputation, that is, an allocation satisfying the individual rationality property. To finish this section we shall give some conditions for the Arc-Proportional solution to be in the core.

Proposition 4.5.2 (Efficiency) Given is (N, v) a Supply Chain Game. $\forall\, \gamma \in \Omega(N, v)$,

γ is efficient.

Proof. Let (N, A, C, b, k, H) be a Supply Chain Problem, and (N, v) its associated game. Let x^* be an optimal distribution plan for the problem (N, A, C, b, k, H) and γ its associated Arc-Proportional allocation. We have to prove that $\sum_{i=1}^{n} \gamma_i = v(N)$.

$$(4.118)\quad 2\sum_{i=1}^{n}\gamma_i = \sum_{i=1}^{n}\sum_{j:(i,j)\in A}\frac{L}{T}x^*_{ij} - \sum_{i=1}^{n}\sum_{j:(i,j)\in A}c_{ij}x^*_{ij} + \sum_{i=1}^{n}\sum_{j:(j,i)\in A}\frac{L}{T}x^*_{ji} - \sum_{i=1}^{n}\sum_{j:(j,i)\in A}c_{ji}x^*_{ji}$$
$$= \frac{L}{T}\sum_{(i,j)\in A}x^*_{ij} - \sum_{(i,j)\in A}c_{ij}x^*_{ij} + \frac{L}{T}\sum_{(j,i)\in A}x^*_{ji} - \sum_{(j,i)\in A}c_{ji}x^*_{ji} = 2(\frac{L}{T}\sum_{(i,j)\in A}x^*_{ij} - \sum_{(i,j)\in A}c_{ij}x^*_{ij})$$

Since $\sum_{(i,j)\in A} x^*_{ij} = T$, (4.118) becomes

$$(4.119)\quad 2L - 2\sum_{(i,j)\in A}x^*_{ij}c_{ij} = 2\sum_{(i,j)\in A}(k_j - k_i)x^*_{ij} - 2\sum_{(i,j)\in A}c_{ij}x^*_{ij} = 2\sum_{(i,j)\in A}(k_j - k_i - c_{ij})x^*_{ij}$$

From the optimality of x^* we conclude that

$$(4.120)\qquad \sum_{i=1}^{n}\gamma_i = v(N).$$

□

Unfortunately, this solution concept does not always satisfy the individual rationality property, as we can see in the following example.

Example 4.5.6 *Consider the Supply Chain Problem described in Figure 4.12 and let (N, v) be its associated Supply Chain Game. One has that an optimal transportation of this Supply Chain Problem is*

$$(4.121)\qquad x^*_{35} = 5,\ x^*_{42} = 1,\ x^*_{13} = 5,\ x^*_{21} = 5,\ x^*_{ij} = 0 \text{ otherwise.}$$

The maximum profit generated by the grand coalition is 25 units.

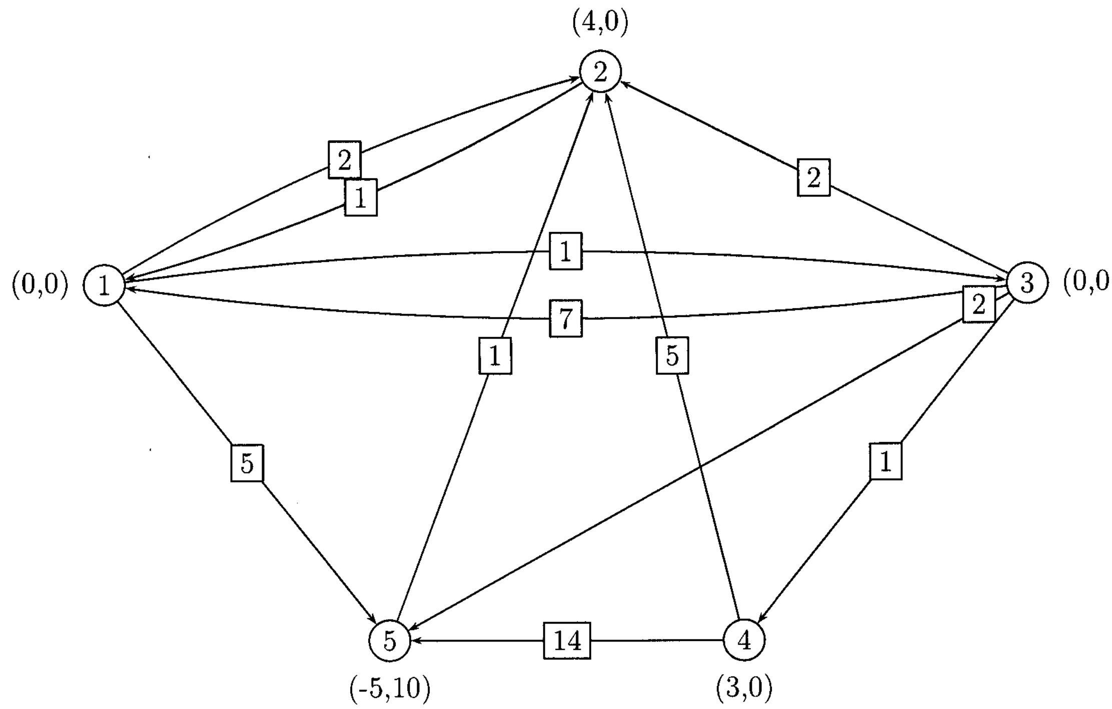

Figure 4.12: Supply Chain Game

Let us calculate $\gamma_4(x^*)$.

$$\gamma_4 = \frac{1}{2}(\frac{L}{T} - c_{42})x^*_{42} \tag{4.122}$$

The values of L and T are

$$\begin{aligned} L = \;& x^*_{35}(k_5 - k_3) + x^*_{42}(k_2 - k_4) + x^*_{13}(k_3 - k_1) + x^*_{21}(k_1 - k_2) \\ & = 5(10 - 0) + 1(0 - 0) + 5(0 - 0) + 5(0 - 0) = 50 \\ T = \;& 5 + 1 + 5 + 5 = 16. \end{aligned} \tag{4.123}$$

Thus,

$$\gamma_4 = \frac{1}{2}(50/16 - 5)1 = \frac{-30}{32} = -0.9375 < v(\{1\}), \tag{4.124}$$

and therefore this arc-proportional solution does not satisfy the individual rationality principle.

The following result assures that if the material does not have "to take a long way" from the supply nodes to the demand nodes upon the optimal distribution plan, then the associated Arc-Proportional solution satisfies the individual rationality principle. In other words, if there are too many transfer nodes from the supply nodes to the demand nodes, the Arc-Proportional solution might be individually irrational.

Proposition 4.5.3 Let (N, v) be the Supply Chain Game defined from the Supply Chain Problem (N, A, C, b, k, H), and let x^* be an optimal distribution plan. If the following condition

$$\forall\ (i,j) \in A\ :\ x^*_{ij} \neq 0 \Rightarrow \frac{L(x^*)}{T(x^*)} \geq c_{ij} \tag{4.125}$$

holds, then the associated Arc-Proportional solution $\gamma(x^*)$ satisfies the individual rationality property.

Proof.

$$\gamma_i(x^*) = \frac{1}{2} \sum_{j:(i,j)\in A} x^*_{ij} \left(\frac{L(x^*)}{T(x^*)} - c_{ij} \right) + \frac{1}{2} \sum_{j:(j,i)\in A} x^*_{ji} \left(\frac{L(x^*)}{T(x^*)} - c_{ji} \right). \tag{4.126}$$

Since $x^*_{ij} \geq 0$ and $\frac{L(x^*)}{T(x^*)} - c_{ji} \geq 0\ \forall\ (i,j)$, we conclude that

$$\gamma_i(x^*) \geq 0 = v(\{i\})\ \forall\ i \in N. \tag{4.127}$$

□

The Example 4.5.6 does not contradict this result, since $c_{42} = 5 > 3.125 = \frac{L(x^*)}{T(x^*)}$.

The previous proposition gives us a sufficient condition for the Arc-Proportional solution to satisfy the individual rationality property, but this condition is not necessary, as we see in the following example.

Example 4.5.7 *Consider the transportation network as depicted in Figure 4.13, where* $k = 5$. *One has that the unique optimal distribution plan is* $x^*_{12} = x^*_{23} = x^*_{34} = 1$. *Thus, we have that*

$$L(x^*) = 5,\ T(x^*) = 3 \Rightarrow \frac{L(x^*)}{T(x^*)} = \frac{5}{3}. \tag{4.128}$$

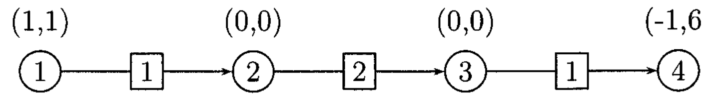

Figure 4.13: Transportation network

For instance, $c_{23} = 2$*, then* $c_{23} > \frac{L(x^*)}{T(x^*)}$ *and the condition mentioned in Proposition 4.5.3 does not hold. Nevertheless, the Arc-Proportional solution associated to this transportation is:*

$$\gamma(x^*) = (\frac{1}{3}, \frac{1}{6}, \frac{1}{6}, \frac{1}{3}). \tag{4.129}$$

Obviously we have that $\gamma_i(x^*) > v(\{i\}) = 0$ *for all* $i \in N$*. Then, we deduce that the previously given condition is not a necessary condition.*

The Arc-Proportional solution is not unique, since there could be more than one optimal distribution plan. If we are asked to give a unique allocation, any convex combination of all possible Arc-Proportional solutions is a logical approach. See the following example.

Example 4.5.8 *Consider the supply chain game described in Figure 4.14.*

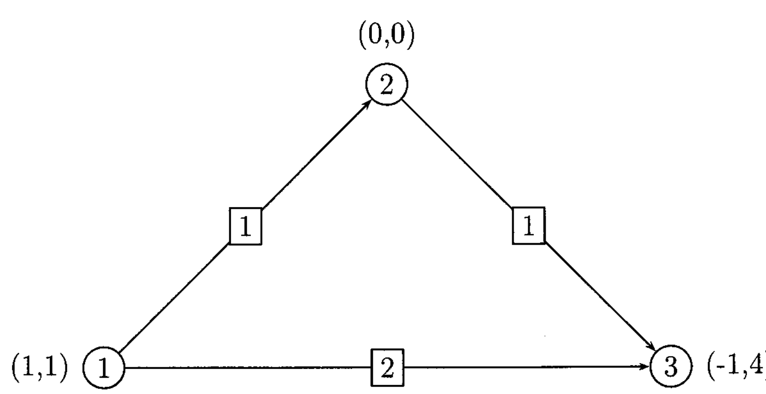

Figure 4.14: Supply Chain Problem

One can see that the distribution plan

$$x'_{12} = 1, \; x'_{23} = 1 \tag{4.130}$$

is optimal and gives rise to the Arc-Proportional allocation

$$\gamma' = (0.25, 0.5, 0.25). \tag{4.131}$$

On the other hand, the distribution plan

$$x^*_{13} = 1 \tag{4.132}$$

is also optimal for the described Supply Chain Problem, and generates the Arc-Proportional allocation

$$\gamma^*(0.5, 0, 0.5). \tag{4.133}$$

So, an allocation rule could be, for example, just the average between the two Arc-Proportional solutions we have found, that is

$$\gamma = \frac{1}{2}(\gamma' + \gamma^*) = (0.375, 0.25, 0.375). \tag{4.134}$$

Another property that Arc-Proportional solutions could be desirable to have is the symmetry. That is, equal players should receive equal payoffs.

Definition 4.5.4 *A solution concept Ψ for SChG is said to satisfy the property of symmetry if $\forall\ i, j \in N$ such that $b_i = b_j$, $k_i = k_j$ and $c_{ki} = c_{kj}$ and $c_{ik} = c_{jk}$ for all $k \in N$, we have that*

$$x \in \Psi(N, v) \Rightarrow \exists\ \overline{x}\ :\ \overline{x}_r = x_r\ \forall\ r \in N \setminus \{i, j\},\ \overline{x}_i = x_j,\ \overline{x}_j = x_i. \tag{4.135}$$

Proposition 4.5.4 The Arc-Proportional solution satisfies the symmetry property.

Proof. Clear from the definition of the Arc-Proportional solution. □

So, we have that if two players are equal in the Supply Chain Problem, then the Arc-Proportional solution gives them the same opportunities.

It is also trivial to prove that Arc-Proportional solutions satisfy the property of irrelevant players (IPP). That is, $\gamma(i) = 0 \ \forall \ \gamma \ \in AP$.

Proposition 4.5.5 Let (N, v) be a SChG and $\gamma \in \Omega(N, v)$. Then γ satisfies IPP.

Another desirable property for AP is the symmetry between groups, that is, P and Q should receive the same.

Definition 4.5.5 *A solution concept on SChG, Ψ, is said to satisfy the property of* equal treatment between groups *if for all $(N, v) \in SChG$ and for all $x \in \Psi(N, v)$, one has that $x(P) = x(Q)$.*

Unfortunately this desirable property is not true in general for the class of Supply Chain Games, as we can see in the following example.

Example 4.5.9 *In Example 4.5.6, one Arc-Proportional solution is*

$$\gamma = [10.625, 4.375, 8.125, -0.9375, 2.8125], \tag{4.136}$$

thus

$$\gamma(P) = 4.375 + (-0.9375) \neq 2.8125 = \gamma(Q). \tag{4.137}$$

One could be interested in checking if AP satisfies the property of Fairness for Adjacent Players, see (4.86). The following example shows that this is not true in general.

Example 4.5.10 *Let us consider the Supply Chain Game arising from the Supply Chain Problem in Figure 4.15.*

There exists only one optimal transportation, $x_{13} = x_{32} = 1$, and the associated arc-proportional solution is:

$$\gamma(A) = (2, 2, 4). \tag{4.138}$$

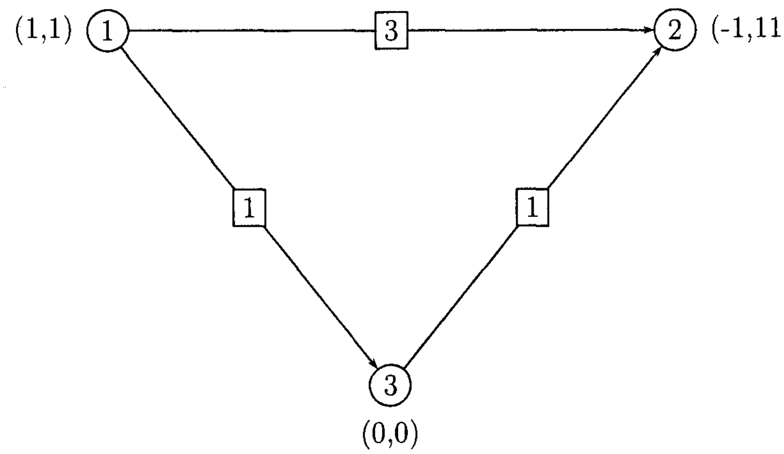

Figure 4.15: Supply Chain Problem

If we deleted arc $(3,2)$ *from the picture,* $x_{13} = 1$ *would be the only optimal transportation, and the arc-proportional solution* γ *in this case would have the form:*

$$\gamma(A \setminus (3,2)) = (3.5, 3.5, 0). \tag{4.139}$$

Therefore, we have that:

$$\begin{aligned} \gamma_2(A) - \gamma_2(A \setminus (3,2)) &= 2 - 3.5 = -1.5 \\ \gamma_3(A) - \gamma_3(A \setminus (3,2)) &= 4 - 0 = 4 \end{aligned} \tag{4.140}$$

Thus, the Arc-Proportional solution does not satisfy the property of fairness for adjacent players, unlike the Shapley value.

Another property that the Shapley value has is the fairness for unconnected players, FUP see Equation (4.85). Unfortunately this is not satisfied by the Arc-Proportional solution, as we see in the following example.

Example 4.5.11 *Consider the example in Figure 4.16. We have that players* 1 *and* 3 *are not connected.*

From this problem, the arc-proportional solution results

$$\gamma(A) = (0, 8, 16, 8) \tag{4.141}$$

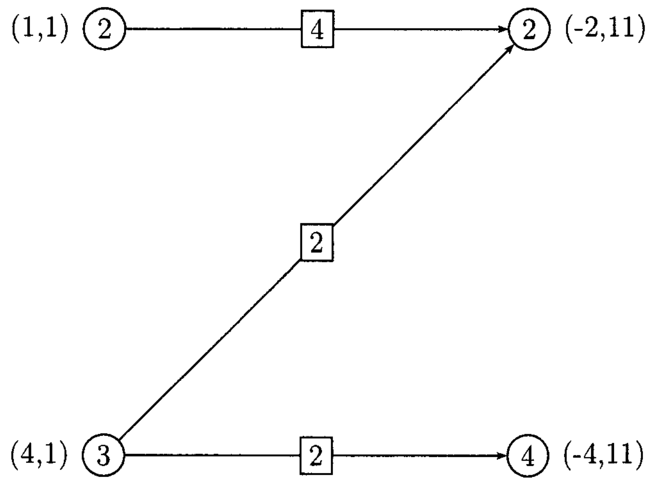

Figure 4.16: Supply Chain Problem

from the only optimal transportation $x_{32} = x_{34} = 2$. *So, we have that*

$$\gamma_1(A) = 0, \ \gamma_3(A) = 16. \tag{4.142}$$

If we remove players 1 *or* 3 *from the game, we can check that*

$$\gamma_3(A_{-1}) = 16, \ \gamma_1(A_{-3}) = 3. \tag{4.143}$$

Thus, we have that

$$\begin{aligned} \gamma_1(A) - \gamma_1(A_{-3}) &= 0 - 3 = -3. \\ \gamma_3(A) - \gamma_3(A_{-1}) &= 16 - 16 = 0. \end{aligned} \tag{4.144}$$

So, we conclude that FUP *is not true for* AP *in the class of Supply Chain Games.*

Another interesting property is the following:

Definition 4.5.6 *A solution concept* Ψ, *for SChG, is said to satisfy the* standard property for two players *if for all* $(N,v) \in SChG$ *such that* $|N| = 2$, $\Psi(N,v)$ *is a unique point and* $\Psi_i(N,v) = \frac{v(\{i,j\})}{2}$ *for all* $i \in \{1,2\}$.

Proposition 4.5.6 The Arc-Proportional solution satisfies the standard property for two players.

Proof. Immediate from the Arc-Proportional solution definition. □

In the following section we shall see several conditions for the Arc-Proportional solution to be a core allocation.

Stability conditions

We have seen that, in general, the Arc-Proportional set is not in the core. Nevertheless we will show that, under several conditions, there are Supply Chain Games for which the AP set is a subset of the core.

In some supply chain situations where we have a certain number of supply nodes, transfer nodes and demand nodes, it is common to find that any supply node is directly connected to any demand node and that the transportation costs are quite similar from one arc to another. It is also common to find that the costs of production are similar in every supply node and the profits are similar in the demand nodes as well. Under these circumstances several results about the stability of the Arc-Proportional solution hold.

The following theorem proves that, under five conditions, the arc-proportional solution is a core allocation for Supply Chain Games. Condition 1 says that each supply node is directly connected by one arc to every demand node. Condition 2 states that the costs of sending one unit of material from a supply node to a demand node are constant. The third condition assures that shipping units via transfer nodes is not optimal. The forth condition says that the total amount of demand equals the total amount of material available. Condition 5 states that the costs of producing material and the benefits after receiving them are not dependent upon the nodes.

Theorem 4.5.5 Let (N, A, C, b, k, H) be a Supply Chain Problem satisfying:

1. $\{(i,j) : i \in P, j \in Q\} \subset A$.

2. $c_{ij} = c \; \forall \; i \in P, \; j \in Q$.

3. $c_{ij} > \frac{c}{2} \; \forall \; (i,j) \in A \setminus \{(i,j) : i \in P, j \in Q\}$.

4. $\sum_{i=1}^{n} b_i = 0$

5. $k_i = k_1 \; \forall \; i \in P, \; k_j = k_2 \; \forall \; j \in Q$.

Let (N, v) be the associated Supply Chain Game. Then we have

$$\Omega(N,v) \subset C(N,v). \tag{4.145}$$

Proof. Under the conditions above, it is easy to see that supply nodes send all their material directly to demand nodes without passing through transfer nodes in any optimal distribution plan. Thus, γ (the arc-proportional solution associated to the optimal transportation x^*) is:

$$\gamma_i = \frac{1}{2}\{\sum_{j \in N:(i,j) \in A} x^*_{ij}(\frac{L}{T} - c_{ij}) + \sum_{j \in N:(j,i) \in A} x^*_{ji}(\frac{L}{T} - c_{ji})\} \; \forall \; i \in N. \tag{4.146}$$

We have that

$$\left.\begin{array}{l} T = \sum_{(i,j) \in A} x^*_{ij} = \sum_{i \in P} b_i = -\sum_{j \in Q} b_j \\ L = \sum_{(i,j) \in A} x^*_{ij}(k_j - k_i) = \sum_{i \in P}\sum_{j \in Q} x^*_{ij} \underbrace{(k_2 - k_1)}_{=k} = k \sum_{i \in P} b_i \end{array}\right\} \Rightarrow \frac{L}{T} = k. \tag{4.147}$$

Thus, if $i \in P$, (4.146) results

$$\frac{1}{2} \sum_{j \in Q:(i,j) \in A} x^*_{ij}(k - c) = \frac{1}{2}(k - c) \underbrace{\sum_{j \in Q:(i,j) \in A} x^*_{ij}}_{=b_i} = \frac{1}{2}(k - c) b_i, \tag{4.148}$$

and if $i \in Q$ (4.146) becomes

$$\frac{1}{2}\sum_{j\in P:(j,i)\in A} x^*_{ji}(k-c) = \frac{1}{2}(k-c)\underbrace{\sum_{j\in P:(j,i)\in A} x^*_{ij}}_{=-b_i} = -\frac{1}{2}(k-c)b_i. \tag{4.149}$$

So, we conclude that

$$\gamma_i = \frac{1}{2}(k-c)|b_i| \ \forall\ i \in N. \tag{4.150}$$

Now we shall see that γ is in the core of the game.

- Obviously $\sum_{i=1}^{n} \gamma_i = v(N)$, because the Arc-Proportional solution is always efficient.
- Let S be a coalition of N. On the one hand, since the members of S send as much as they can from P_S to Q_S, we have that

$$v(S) = (k-c)\min\{\sum_{i\in P_S} b_i, -\sum_{j\in Q_S} b_j\}. \tag{4.151}$$

On the other hand

(4.152)

$$\begin{aligned}
\gamma(S) &= \sum_{i\in S}\gamma_i = \sum_{i\in P_S}\gamma_i + \sum_{j\in Q_S}\gamma_j + \underbrace{\sum_{k\in R_S}\gamma_k}_{=0} = \sum_{i\in P_S}\frac{1}{2}(k-c)b_i + \sum_{j\in Q_S} -\frac{1}{2}(k-c)b_j \\
&= \frac{1}{2}(k-c)\sum_{i\in P_S} b_i + \frac{1}{2}(k-c)(-\sum_{j\in Q_S} b_j) \\
&\geq \frac{1}{2}(k-c)\min\{\sum_{i\in P_S} b_i, -\sum_{j\in Q_S} b_j\} + \frac{1}{2}(k-c)\min\{\sum_{i\in P_S} b_i, -\sum_{j\in Q_S} b_j\} \\
&= (k-c)\min\{\sum_{i\in P_S} b_i, -\sum_{j\in Q_S} b_j\} = v(S),
\end{aligned}$$

and that concludes the proof.

□

Now, we see that the second condition of the theorem above is necessary.

Example 4.5.12 *Let us consider the Supply Chain Game associated with the Supply Chain Problem described in Figure 4.17. Then, one Arc-Proportional solution is*

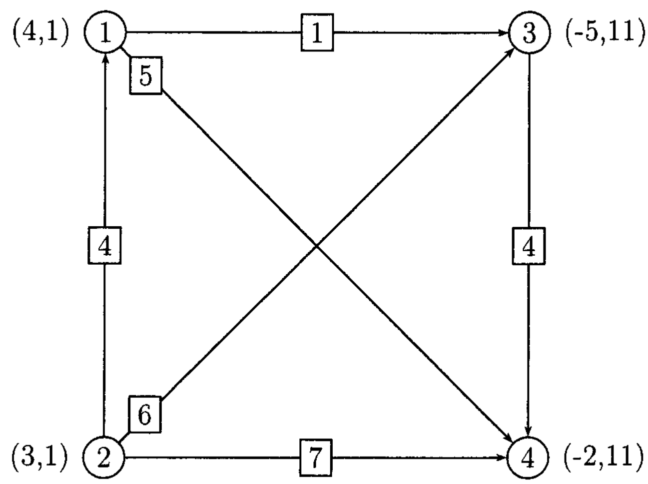

Figure 4.17: Supply Chain Problem

(4.153) $$\gamma = (21.75, 4.125, 19.375, 1.75),$$

and it is not in the core of the game, since

(4.154) $$\gamma(\{2,4\}) = 5.875 < 6 = v(\{2,4\}).$$

But this does not contradict the theorem before, since in this example the condition (2) does not hold (one can check that (1), (3), (4) and (5) hold).

The conditions in Theorem 4.5.5 can be relaxed and the arc-proportional solution keeps being a core allocation, as we see in the following theorem.

Theorem 4.5.6 Let (N, A, C, b, k, H) be a Supply Chain Problem and (N, v) its associated game. Suppose that the following conditions are satisfied:

1. $\max_{P \times Q}\{k - c_{ij}\} \leq (1 + \varepsilon) \min_{P \times Q}\{k - c_{ij}\}$, where $k = k_2 - k_1$.

2. $2+2\varepsilon = \min_{\substack{S \subsetneq P\cup Q \\ P_S, Q_S \neq \emptyset}} \left\{ \frac{\sum_{i\in P_S} b_i - \sum_{j\in Q_S} b_j}{\min\{\sum_{i\in P_S} b_i, -\sum_{j\in Q_S} b_j\}} \right\}.$

3. $\sum_{i=1}^{n} b_i = 0.$

4. $P \times Q \subset A.$

5. $c_{kl} > \max_H\{c_{ij}\} - \min_H\{c_{ij}\}\ \forall\ (k,l) \in (P\times P)\cup(Q\times Q).$

6. $c_{kl} > \frac{1}{2}\max_H\{c_{ij}\}\ \forall\ (k,l) : k \in R \text{ or } l \in R.$

Then

$$\Omega(N,v) \subset C(N,v). \tag{4.155}$$

Proof. Let (N, A, C, b, k, H) be a SChP and (N, v) its associated SChG.

Let $\gamma \in AP(N,v)$, $c^+ = \max_{P\times Q}\{k - c_{ij}\}$, $c^- = \min_{P\times Q}\{k - c_{ij}\}$.

From conditions (5), (6), we have that the optimal distribution plan consists of shipping all the material directly from the nodes in P to the nodes in Q. Then, for all $S \subset N$ one has that

$$\gamma(S) = \gamma(P_S) + \gamma(Q_S) \geq \sum_{i\in P_S} \frac{b_i}{2}c^- - \sum_{j\in Q_S} \frac{b_j}{2}c^-. \tag{4.156}$$

That comes from the fact that

$$\gamma_i = \frac{1}{2}\sum_{j:(i,j)\in A}(k - c_{ij})b_i \geq \frac{1}{2}c^- b_i\ \forall\ i \in P, \tag{4.157}$$

and analogously

$$\gamma_j \geq -\frac{1}{2}c^- b_j\ \forall\ j \in Q. \tag{4.158}$$

Besides, from (4), (5) and (6), we have that

$$v(S) \leq c^+ \min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\}. \tag{4.159}$$

Let S be a coalition of N. If $P_S = \emptyset$ or $Q_S = \emptyset$, then

$$v(S) = 0 \leq \gamma(S), \tag{4.160}$$

since the conditions for γ to have the individual rationality property hold, then $\gamma(S) \geq 0 \; \forall S \in 2^N$. Let us study the case in which P_S and Q_S are not empty.

Let S be a coalition such that $S \cap P \neq \emptyset$ and $S \cap Q \neq \emptyset$. We know that

$$v(S) \leq c^+ \min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\} \leq (1+\varepsilon)c^- \min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\}. \tag{4.161}$$

From condition 2, we have that
(4.162)

$$\varepsilon = -1 + \frac{1}{2} \min_{\substack{S \subsetneq P \cup Q \\ P_S, Q_S \neq \emptyset}} \left\{ \frac{\sum_{i \in P_S} b_i - \sum_{j \in Q_S} b_j}{\min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\}} \right\} \leq -1 + \frac{1}{2} \left\{ \frac{\sum_{i \in P_S} b_i - \sum_{j \in Q_S} b_j}{\min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\}} \right\},$$

then, (4.161) is less than or equal to

$$\begin{aligned} &\leq \left(1 - 1 + \tfrac{1}{2} \left\{ \frac{\sum_{i \in P_S} b_i - \sum_{j \in Q_S} b_j}{\min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\}} \right\} \right) c^- \min\{\textstyle\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\} \\ &= \frac{1}{2} c^- (\sum_{i \in P_S} b_i - \sum_{j \in Q_S} b_j) \leq \gamma(S). \end{aligned} \tag{4.163}$$

□

Condition (5) assures that the cost of sending material from a node in P to a node in Q via another node in P or Q is higher than sending it directly from P to Q. From condition

(6) we know that it is cheaper to send directly from P to Q than to send via any node of R.

To finish this section, see in the following example that the fifth condition is necessary.

Example 4.5.13 *Consider the Supply Chain Problem as depicted in Figure 4.18.*

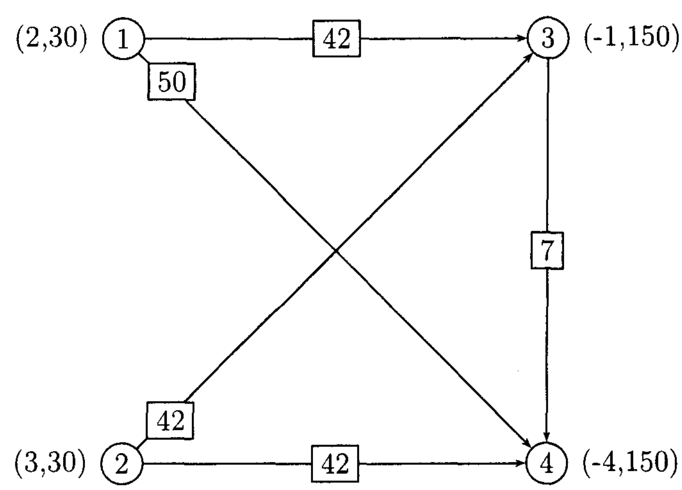

Figure 4.18: Supply Chain Problem

We have that:

$$\min_{\substack{S \subsetneq P \cup Q \\ P_S, Q_S \neq \emptyset}} \left\{ \frac{\sum_{i \in P_S} b_i - \sum_{j \in Q_S} b_j}{\min\{\sum_{i \in P_S} b_i, -\sum_{j \in Q_S} b_j\}} \right\} = \min \left\{ \frac{3}{1}, \frac{6}{2}, \frac{7}{2}, \frac{6}{1}, \frac{9}{4}, \frac{4}{1}, \frac{7}{3}, \frac{8}{3} \right\} = \frac{9}{4}. \tag{4.164}$$

So, we get that

$$2 + 2\varepsilon = \frac{9}{4} \Rightarrow \varepsilon = \frac{1}{8}. \tag{4.165}$$

Let us see which of the conditions of Theorem 4.5.6 are satisfied:

- (4.166)
$$\left.\begin{array}{l}\max_{P\times Q}\{k-c_{ij}\}=120-42=78\\ \min_{P\times Q}\{k-c_{ij}\}=120-50=70\end{array}\right\}\Rightarrow(1+\varepsilon)70=78.75\geq 78$$

  and we conclude that (1) holds.

- *Obviously,* $\sum_{i\in N} b_i = 0$, *so we have that (3) holds.*
- $P \times Q \subset A$, *then we have condition (4).*
- *But (5) does not hold, because*

  (4.167)
$$\max_{P\times Q}\{c_{ij}\}-\min_{P\times Q}\{c_{ij}\}=50-42=8\not< c_{34}.$$

- *(6) is trivially satisfied because there is no transfer node in this example.*

So, the only condition of Theorem 4.5.6 that is not satisfied is (5). One can see that the Arc-Proportional solution of this example is

(4.168)
$$\gamma=(58,87,104.5,133.5),$$

which does not belong to the core, since

(4.169)
$$v(\{2,4\})=234>220.5=\gamma(\{2,4\}).$$

Computational Complexity

To finish this section we show the computational complexity to calculate one Arc-Proportional solution.

Theorem 4.5.7 Let (N, v) be a SChG. Then the computational complexity for calculating one Arc-Proportional solution is polynomial with respect to the number of players, and given by

(4.170)
$$O(n^2).$$

Proof. To calculate one Arc-Proportional solution we first have to generate an optimal distribution plan, which implies solving a linear programming problem with $q = |A|$ variables and n constraints. The effort in this step is $O[(n + q)]$, where n is the number of players (nodes) and q is the number of arcs of the corresponding SChP. After that, we have to calculate the payoff of each player, which is done in $O(n^2)$ steps. Again considering that $q = O(n^2)$, the result follows. □

4.5.5 Sequential Solutions

In previous sections we saw an "easy" procedure to build an allocation in the core by solving the dual linear program. It is known that, if a node remains with an excess of material after an optimal distribution plan has been taken, then its associated dual variable is null in the optimum, and that means that this player receives nothing from the Owen allocation. Analogously, any player that does not get all its demand satisfied, would not receive anything after the allocation taken from the dual program. To avoid that, each supply node could choose to offer less units of material so that they can send all their supply, and analogously each demand node could require less material so that none of their demand is unsatisfied. Then, we propose the following way of allocating costs.

Given is a Supply Chain Problem (N, A, C, b, k, H) and (N, v) its associated game. Let $z = \min\{|b_i| : b_i \neq 0\}$. We first consider the game G^1 only changing that every supply node offers z units and every demand node asks for z units. After solving this problem, the vector b and the matrix H are updated, decreasing the offers, demands and capacities according to the optimal distribution plan that has been chosen in the previous step. So, it might happen that, after the optimal distribution plan has been done, some supply nodes become transfer nodes (since their original supplies are over) and some demand nodes become transfer nodes (since their original demands are completely satisfied). It could also happen that some arcs have their capacity reduced. Then the new z is calculated and this process is repeated until there is not more profitable material to transport.

Notice that, if in every game G^k an allocation rule is calculated, we can propose as a final allocation the sum of all these allocations, as we will see in following examples.

But first, let us see in a formal way how this sequential process works.

Given is $G^0 = (N, A, C, b, k, H)$ a Supply Chain Problem and (N, v) its associated game. Set $l = 1$ and $y \in \mathbb{R}^n$, $y = 0$ and go to 1.

1. Let $z^l = \max\{|b_i| : b_i \neq 0\}$. Consider the SChP $G^l = (N, A, C, b^l, k, H)$ where:
   - If $b_i > 0$ then $b_i^l = z^l$.
   - If $b_i < 0$ then $b_j^l = -z^l$.
   - If $b_i = 0$ then $b_i^l = 0$.

   Consider (N, v^l) the associated game to G^l.

   - If $v^l(N) = 0$, STOP.
   - If $v^l(N) > 0$ then we calculate an optimal distribution plan x^l for G^l and an allocation of (N, v^l), denoted by y^l. Then we update b and H:
     - For all $i \in N$ do,
       if $b_i > 0$, then $b_i = b_i - (\sum_{j:(i,j)\in A} x_{ij}^l - \sum_{j:(j,i)\in A} x_{ji}^l)$.
       if $b_i < 0$, then $b_i = b_i + (\sum_{j:(j,i)\in A} x_{ji}^l - \sum_{j:(i,j)\in A} x_{ij}^l)$.
       * For all $j \in N$, if $h_{ij} \neq +\infty$ then $h_{ij} = h_{ij} - x_{ij}^l$.

     Set $y = y + y^l$ and go to step 1.

   It is easy to find examples in which the final allocation, y, does not allocate all the possible benefit that the grand coalition could have obtained in the original game (N, v). To avoid such drawback we normalize and propose as final allocation $y = \frac{v(N)}{\sum_{i\in N} y_i} y$.

Note that, depending on the allocation calculated at each step, y^l, we obtain different allocations. So, there can be defined:

- *Sequential Owen Solution*, if an Owen allocation is calculated in each step.
- *Sequential Extended Owen Solution*, if an Extended Owen allocation is calculated in each step.

- *Sequential Arc-Proportional solution*, if an AP allocation is calculated in each step.

- *Sequential Shapley Value*, if the Shapley value is calculated in each step.

Example 4.5.14 *Let us consider the Supply Chain Problem in Figure 4.19. The char-*

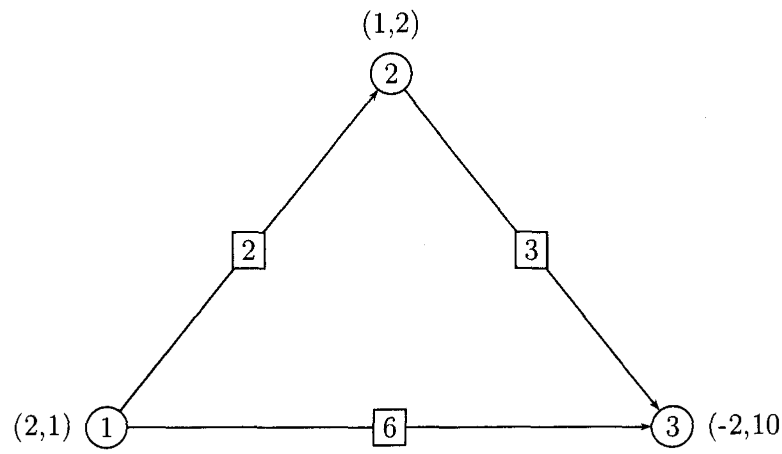

Figure 4.19:
Supply Chain Problem

acteristic function of the associated Supply Chain Game is $v(\{1,3\}) = 6$, $v(\{2,3\}) = 5$, $v(\{1,2,3\}) = 9$ *and null for any other coalition. One Owen allocation for this game is*

$$(0,1,8).$$

Let us find a Sequential Owen solution and a Sequential Arc-Proportional solution. $z^1 = 1$ *and the game* G^1 *is depicted in Figure 4.20. We have that the characteristic function for this game is* $v^1(\{1,3\}) = 3$, $v^1(\{2,3\}) = 5$, $v^1(\{1,2,3\}) = 5$ *and zero otherwise. One Owen solution* y^1 *for this game is*

$$(0,1,4)$$

and one Arc-Proportional solution γ^1 *is*

$$(0,2.5,2.5).$$

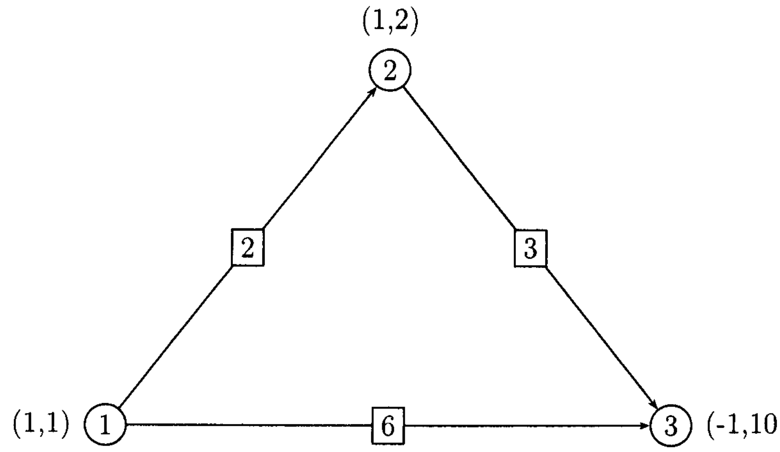

Figure 4.20: Supply Chain Problem

The optimal transportation is $x^1_{23} = 1$ *and zero for the other arcs. Then, we set* $b_2 = 1 - 1 = 0$ *and* $b_3 = -2 + 1 = -1$, *leaving* b_1 *as it was.*

Then, $z^2 = 1$ *and our new problem,* G^2, *is shown in Figure 4.21. The characteristic*

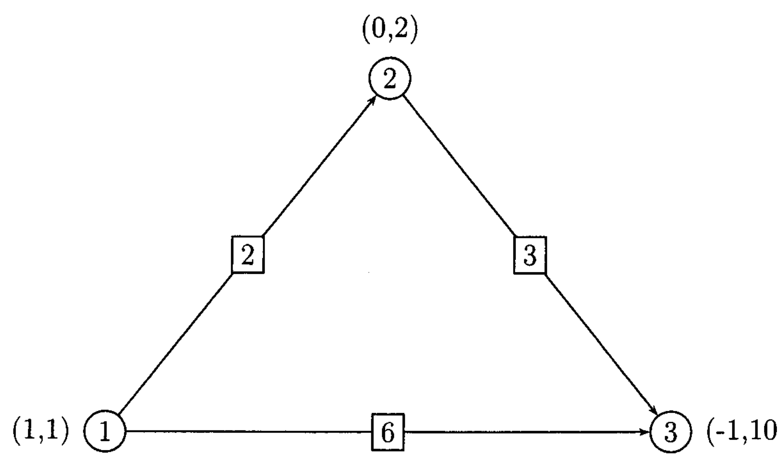

Figure 4.21: Supply Chain Problem

function of G^2 *is* $v^2(\{1,3\}) = 3$, $v^2(\{1,2,3\}) = 4$ *and zero otherwise.*

We get that

$$y^2 = (4, 0, 0), \quad \gamma^2 = (1.25, 2, 0.75).$$

The optimal transportation is $x^2_{13} = 1$, *so now we set* $b_1 = 2 - 1 = 1$ *and* $b_3 = -1 + 1 = 0$. *One can check that the problem* G^3 *is not profitable, so the algorithm is over. Then, the proposed allocations after this process are:*

$$y = y^1 + y^2 = (4, 1, 4), \quad \gamma = \gamma^1 + \gamma^2 = (1.25, 4.5, 3.25).$$

Since in this case $y(N) = \gamma(N) = v(N)$ *we do not have to normalize the allocations* y *and* γ.

Unfortunately, the sequential process explained above does not always keep the properties of the allocations that originate it. See for instance in the following example that a Sequential Owen Solution might not be included in the core of the corresponding SChG, even though the Owen set is always included in the Core of SChG.

Example 4.5.15 *Consider the SChG* (N, v) *arising from the SChP in Figure 4.22. One*

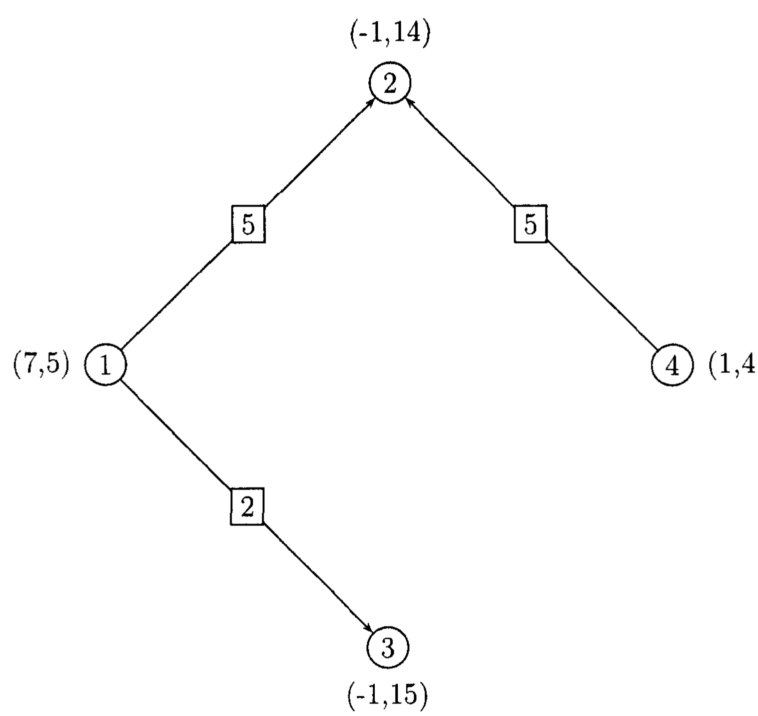

Figure 4.22: Supply Chain Problem

can see that the dual program of the game resulting in the first step of the process to build the Sequential Owen set is

$$\begin{aligned} \min\ & y_1 + y_2 + y_3 + y_4 \\ s.t.:\ & y_1 + y_2 \geq 4 \\ & y_1 + y_3 \geq 8 \\ & y_4 + y_2 \geq 5 \\ & y_i \geq 0\ \forall\ i. \end{aligned} \tag{4.171}$$

One feasible optimal solution to this problem is the vector $y^1 = (6, 0, 2, 5)$. *It is not difficult to check that after this step the process is over, resulting in the allocation*

$$\gamma = (6, 0, 2, 5). \tag{4.172}$$

Take coalition $S = \{1, 2, 3\}$. *One has that* $v(S) = 12$, *but* $\gamma(S) = 8$. *We conclude that* $\gamma \notin C(N, v)$ *and prove that, in general, the Sequential Owen set is not a subset of the core of SChG.*

4.5.6 Experimental results

The computational complexities calculated before only reflect the performance of the corresponding algorithm in the most pathological instances. In order to compare the time expenses to calculate those three allocations it is more appropriate to perform some experiments, as shown in this section.

Some experimental results on the speed and quality of the solution concepts proposed for SChG are presented. The experiments were performed by building random Supply Chain Problems (N, A, C, b, k, H). The number of players n varied from 3 to 10, 100 instances of each case being performed. To build each SChP, random numbers were generated as follows:

- For each pair of nodes (i, j), there is an arc joining i and j in this direction with probability $\frac{2}{3}$.

- For each arc (i, j), its associated cost, c_{ij}, follows a uniform random variable between

1 and 5.

- For each arc (i, j), its associated upper bound, h_{ij}, follows a uniform random variable between 6 and 10.

- For each node i, its demand or supply b_i is randomly generated and varies from -10 to 10.

- For each node $i \in Q$, its unit profit k_i follows a uniform random variable between 6 and 10.

- For each node $i \in P$, its unit cost of production k_i follows a uniform random variable between 1 and 5.

For each SChP, the corresponding SChG was generated and the three allocations presented in this work (Shapley value, Owen solution and Arc-Proportional solution) were calculated. For each of those allocations their computational time was calculated and it was checked if they belonged to the core of the corresponding game. It goes without saying that the Owen solution always belongs to the core of the game.

Those experiments were run by using the mathematical software Maple 7 in a Pentium IV computer (frequency 3.06 GHz). Table 4.1 summarizes the results obtained. The

Players	Time Shapley	Time Owen	Time AP	Core Shapley	Core AP
3	0.03353	0.00848	0.00854	99%	93%
4	0.12226	0.01683	0.01217	86%	71%
5	0.39523	0.03048	0.02207	66%	57%
6	1.27179	0.05362	0.04142	81%	35%
7	3.62249	0.08243	0.06557	72%	29%
8	10.95464	0.13888	0.10615	67%	29%
9	27.63470	0.19014	0.13953	74%	13%
10	86.98020	0.29885	0.23429	46%	20%

Table 4.1: Experimental results.

first column of Table 4.1 shows the number of players of the random games considered. The second, third and fourth ones are the average time in seconds spent to calculate the Shapley value, one Owen solution and one Arc-Proportional solution, respectively. The last two columns show the frequency of Shapley values and Arc-Proportional solutions that belong to the core of the corresponding games.

Figure 4.23 shows the comparison between the average times spent by the three allocations calculated.

Figures 4.24 show the evolution of the computational time needed to calculate each allocation. Due to the high differences between the computational times of the Shapley value and the other two solutions calculated, we show two different charts. The first one regards the Shapley value and shows the evolution of the average time needed to find it depending on the number of players. The second one compares the increase in time of one Owen solution (the upper line) and one Arc-Proportional solution (the lower line). In Figure 4.25 it is shown the frequency of times in which the Shapley value (the upper line) and the Arc-Proportional solution (the lower line) belong to the core depending on the number of players of the corresponding game. From the experimental results we can say that the computational time to calculate the Shapley value is much higher (not comparable with) than the necessary time to calculate the Owen solution or the Arc-Proportional solution. The Arc-Proportional solution compared favorably with the Owen solution in terms of computational costs. Regarding the quality of the allocations, the Shapley value proved to be more often a core allocation than the Arc-Proportional solution. It is well known that the Owen allocations always belong to the core of the corresponding game.

Due to the fact that there is no best allocation in the two criteria studied simultaneously (speed and balancedness), there is no worst allocation in the two criteria either, the decision of which allocation to choose remains open and would depend on our priorities: speed or fairness.

4.6 Multicriteria Supply Chain Games

In this section the class SChG is extended to the multicriteria case. To begin with, a short review of a multicriteria cooperative game studied in the literature is given in Section 4.6.1. Afterwards in Section 4.6.2 the multicriteria version of our supply chain games is presented. Later on, the multicriteria versions of the transportation games and assignment games is presented in Section 4.6.3. To finish the section, an application of the multicriteria minimum cost spanning tree game to a class of stochastic spanning tree games is shown.

4.6.1 Multi-criteria minimum cost spanning tree games

This section presents the multi-criteria minimum cost spanning tree games as introduced in [16]. Such games are a natural extension of the classical minimum cost spanning tree (MCST) games as presented in Section 2.3.6. The natural difference with MCST games is that the cost associated to an arc is a vector instead of a single number. Let $G = (N_0, A)$ be the complete graph with set of nodes $N_0 = N \cup \{0\}$ and set of arcs denoted by A. Each arc has an associated vector of costs. Let $a^{ij} = (a_1^{ij}, \dots, a_k^{ij})$ denote the vector cost of using arc $(i,j) \in A$. A Pareto-minimum cost spanning tree for a given connected graph is a spanning tree that has Pareto-minimum costs among all possible spanning trees [14]. Here a formal definition:

Definition 4.6.1 *Let $G = (N_0, A)$ be a complete graph. The associated Pareto-minimum cost spanning tree game is the pair (N, V) where N is the set of players and V is the characteristic function, given by*

$$V(S) = \min_{T_{S_0}:\ spanning\ tree} \sum_{(i,j) \in A} e^{ij} \tag{4.173}$$

for every non-empty coalition $S \subset N$ and $V(\emptyset) = 0$, where $E_{T_{S_0}}$ is the set of arcs of the spanning tree T_{S_0}, which contains $S_0 = S \cup 0$, and min *stands for the Pareto-minimization.*

Example 4.6.1 *Consider the multi-criteria MCST game (N, V), borrowed from [16], arising from the graph in Figure 4.26. It is not difficult to check that the characteristic function of this game is:*

S	$\{1\}$	$\{2\}$	$\{3\}, \{2,3\}$	$\{1,2\}$	$\{1,3\}$	N
$V(S)$	$\begin{matrix}1\\3\end{matrix}$	$\begin{matrix}1\\2\end{matrix}$	$\begin{matrix}1\\5\end{matrix}$	$\begin{matrix}2\\3\end{matrix}$	$\begin{matrix}3\\5\end{matrix}, \begin{matrix}2\\6\end{matrix}$	$\begin{matrix}2\\6\end{matrix}, \begin{matrix}4\\5\end{matrix}$

There are two Pareto-minimum cost spanning trees: one corresponding with $(2,6)^t \in V(N)$, which consists of the arcs $(0,2), (1,2)$ and $(2,3)$; and the one corresponding to $(4,5)^t \in V(N)$, which consists of the arcs $(0,2), (1,2), (1,3)$.

Bird rule, see [5] and Section 2.3.6, can be extended to the multi-criteria MCST game by allocating to each player the cost vector of the edge incident upon it on the unique path between 0 and the player's node, in the corresponding Pareto-minimum cost spanning tree.

Regarding the existence of dominance core allocations, in [16] the following result is proved:

Proposition 4.6.1 Let T_N be a Pareto-minimum cost spanning tree of a complete graph with associated cost vector $z^N \in V(N)$. Then the corresponding vectorial Bird's cost allocation is in the dominance core of the game.

This automatically implies that the dominance core of multi-criteria MCST games is non-empty.

But sometimes, players may want to accept allocations if they pay less than any of the worths given by the characteristic set. In such conditions, players would only accept allocations in the preference core of the game, see Section 2.4. Due to the fact that the order that defines the preference core is stronger than the order that defines the dominance core, it is not assured in general that multi-criteria MCST games have non-empty preference core.

In order to characterize those multi-criteria MCST games with non-empty preference core, consider a vector $\overline{z} \in \mathbb{R}^k$ and the scalar games defined as follows:

Definition 4.6.2 *The scalar l-component minimum cost spanning tree game* ($l = 1, 2, \ldots, k$) *associated to* $\overline{z}$ *is a pair* $(N, v_l^{\overline{z}})$ *where* N *is the set of players and* $v_l^{\overline{z}}$ *is the characteristic function defined by*

1. $v_l^{\overline{z}}(\emptyset) = 0$.
2. *For each non-empty coalition* $S \subset N$,

$$v_l^{\overline{z}}(S) = \min_{T_{S_0}:\ \text{spanning tree}} \sum_{(i,j)\in E_{T_{S_0}}} e_l^{ij}, \tag{4.174}$$

where $E_{T_{S_0}}$ *is the set of arcs of the spanning tree* T_{S_0} *that contains the set of nodes* $S_0 = S \cup 0$.

3. $v_l^{\bar{z}}(N) = \bar{z}$.

A necessary and sufficient condition for the non-emptiness of the preference core of multi-criteria MCST games is given in the following theorem.

Theorem 4.6.1 The preference core is non-empty if and only if there exists at least one $z^N \in V(N)$ such that all the scalar l-component games $(N, v_l^{z^N})$ are balanced.

Proof. See [16]. □

4.6.2 Multicriteria Supply Chain Games

Suppose that we are given a set of players $N = \{1, \ldots, n\}$ that represent the nodes of a graph $G = (N, A)$, where A is the set of arcs. The philosophy of this games is the same as in the Supply Chain Games presented in Chapter 4 with two differences:

- The shipping of one unit along the arc (i, j) has an associated vector of costs $(c_{ij}^1, \ldots, c_{ij}^l)$, not as in SChG where every cost is an scalar. Let C denote the $l \times |A|$-matrix of all the k-dimensional cost vectors.

- Analogously, each unit of satisfied demand after a transportation generates not a scalar profit k_j but a vector profit $(k_j^1, \ldots, k_j^l)$ for each $j \in Q$, and the production of one unit of the good has an associated cost vector $(k_i^1, \ldots, k_i^l)\ \forall\ i \in P$.

The rest is as in SChG, that is, each player owns one node in which there is an associated scalar b_i, which is i player's available supply of the item (if $b_i > 0$) or i player's required demand for the item (if $b_1 < 0$). If $b_i = 0$, then player i does not have any supply of the item and he does not require it either, we call it a transfer player. This situation gives rise to a *Multicriteria Supply Chain Game* (MSChG for short).

The formulation of this kind of multicriteria games is as follows. For every coalition $S \subset N$, let A_S denote the set of arcs in the partial graph $G_S = (S, A_S)$, $A_S = A \cap (S \times S)$. Coalition S induces a multicriteria supply chain problem, namely:

$$\text{(4.175)}\quad \begin{array}{ll} \max & (f_S^1(x),\ldots,f_S^l(x)) \\ \text{s.t.:} & \sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{k\in S:(k,i)\in A_S} x_{ki} \le b_i \quad \forall\, i\in P_S \\ & \sum_{k\in S:(k,i)\in A_S} x_{ki} - \sum_{j\in S:(i,j)\in A_S} x_{ij} \le 0 \quad \forall\, i\in P_S \\ & \sum_{k\in S:(k,i)\in A_S} x_{ki} - \sum_{j\in S:(i,j)\in A_S} x_{ij} \le -b_i \quad \forall\, i\in Q_S \\ & \sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{k\in S:(k,i)\in A_S} x_{ki} \le 0 \quad \forall\, i\in Q_S \\ & \sum_{j\in S:(i,j)\in A_S} x_{ij} - \sum_{k\in S:(k,i)\in A_S} x_{ki} = 0 \quad \forall\, i\in R_S \\ & 0\le x_{ij}\le h_{ij} \quad \forall\,(i,j)\in A_S \end{array} \quad (MPr(S))$$

where $f_S^t(x) = \sum_{(i,j)\in A_S} (k_j^t - k_i^t - c_{ij}^t)x_{ij}\ \forall\, t = 1,\ldots,l$. Problem (4.175) will be denoted as MPr_S.

The meaning of the constraints of the problem is the same as in SChG. They indicate that the total outgoing flow of a supply node can not be more than the incoming in the node flow plus the amount available, and in a demand node the total incoming flow cannot be more than the demand required plus the outgoing flow. For any transfer node, the outgoing flow must be equal to the incoming flow. This problem is always feasible, since the transportation $x = 0$ satisfies the constraints, as in the unidimensional SChP.

Problem MP_S can be written as:

$$\text{(4.176)}\quad \begin{array}{ll} \max & Q_S x_S \\ \text{s.t.:} & H_S x_S \le b(S) \\ & x_{ij}\ge 0 \end{array}$$

where x_S represents the flow through the arcs $(i,j)\in A_S$. Matrices Q_S and H_S and vector $b(S)$ are chosen so that they express F_S and the constraints in (4.175).

If x_S is an efficient solution to MP_S, $z(x_S) = Q_S x_S$ denotes the corresponding vector of results in the criteria space. Let $\varepsilon(MP_S)$ denote the set of efficient solutions of problem

MP_S, and consider the whole set of non-dominated outcomes in the criteria space

$$Z(MP_S) = \{z(x_S) \in \mathbb{R}^k \ : \ z(x_S) = Q_S x_S, \ x_S \in \varepsilon(MP_S)\}. \tag{4.177}$$

Since (4.175) is always feasible, we have that $Z(MP_S) \neq \emptyset$ for all $S \subset N$.

It is now possible to introduce the multicriteria supply chain game (N, A, C, b, K), where $N = \{1, \ldots, n\}$, $A \subset N \times N$, $C \in \mathbb{R}^{|A| \times l}$, $b \in \mathbb{R}^n$, $K \in \mathbb{R}^{n \times l}$, as a combinatorial multicriteria game with characteristic function:

$$V(S) = Z(MP_S) \ \forall \ S \subset N. \tag{4.178}$$

Example 4.6.2 *Let us consider the multicriteria network in Figure 4.27. The costs associated to the arcs are represented as a bi-dimensional vector on them. The 2-dimensional vector k_i represents the two-objective unitary benefit or cost, respectively. Thus, the formulation of this problem as a multiobjective linear program is:*

(4.179)

$$\begin{array}{lll}
\max & (-2x_{12} + 2x_{13} + 6x_{23} + 3x_{24} - 3x_{34}, -3x_{12} + 2x_{13} + x_{23} + x_{24} - x_{34}) & \\
s.t.: & 0 \leq x_{12} + x_{13} \leq 4 & \\
& 0 \leq x_{23} + x_{24} - x_{12} \leq 2 & MP_N \\
& 0 \leq x_{13} + x_{23} - x_{34} \leq 1 & \\
& 0 \leq x_{24} + x_{34} \leq 5 & \\
& x_{ij} \geq 0 \quad \forall \ (i,j) \in A &
\end{array}$$

To calculate the characteristic function of the game, one has to work out the efficient solutions for the multicriteria linear programs associated to each coalition $S \subset N$. Such characteristic function is given by the non-dominated solutions of the corresponding MP_S problems.

We are now interested in obtaining allocations in the core of the multicriteria supply chain game. To this end, we propose an specific approach for the model that we are analyzing in this section, which is based on the special structure of our supply chain problem. It also takes advantage of Isermann's results about duality in multiple objective linear programming [33]. We will prove that allocations in the dominance core can be obtained from the dual of the supply chain problem for the grand coalition.

By introducing n slack variables, problem MP_S can be represented as:

$$\begin{array}{rl} \max & Q_S x_S \\ \text{s.t:} & \overline{H}_S = b(S) \\ & x_{ij} \geq 0 \end{array} \tag{4.180}$$

where $\overline{H}_S$ differs from H_S in the columns associated to the slack variables. Isermann's dual problem for (4.180) is

$$\begin{array}{rl} \min & D_S(U_S) = U_S b(S) \\ \text{s.t.:} & U_S \in J_S \end{array} \tag{4.181}$$

where

$$J_S = \{U_S \in \mathbb{R}^{l \times |S|} \ / \ U_S \overline{H}_S w \leq Q_S w \text{ for no } w \in \mathbb{R}^{|A_S|+n}_{\geqq}\}. \tag{4.182}$$

Theorem 4.6.2 Let x^* be an efficient solution to the multicriteria supply chain problem for the grand coalition MP_N. There exists a solution of the dual problem U^* such that $z(x^*) = D_N(U^*)$. In addition, matrix $Y \in \mathbb{R}^{l \times n}$, $Y_{ri} = U^*_{ri} b_i$, $r = 1, \ldots, l$, $i \in N$, is an allocation in the dominance core of the multicriteria combinatorial game (N, A, C, b, K).

Proof. We have to prove that Y obtained as explained in the theorem has the efficiency property and is coalitionally rational.

1. Let us prove that Y is efficient. Given an efficient solution x^* of the supply chain problem for the grand coalition, MP_N, Proposition 5 in [33] guarantees that there exist a feasible solution for its dual, $U^* \in \mathbb{R}^{l \times n}$ such that $Qx^* = U^* b$. Therefore, Y allocates $Qx^* \in V(N)$. So, the efficiency of Y is proven.

2. Now, we are going to prove that matrix Y is an allocation in the dominance core of the multicriteria supply chain game. First, we will prove that $\forall \ S \in 2^N$, U^*_S is dual feasible for MP_S. As U^* is dual feasible for MP_N, $\nexists \ w \in \mathbb{R}^{|A_S|+n}$ such that $U^* \overline{H} w \leq Qw$. Consider U^*_S and suppose that it is not dual feasible for (4.180). Hence, $\exists \ w^S \in \mathbb{R}^{|A_S|+n}$ such that $U^*_S \overline{H}_S w^S \leq Q_S w^S$. Now, consider w such that

$w_{ij} = w_{ij}^S$, for $i, j \in S$, and 0 otherwise. $U^* \overline{H} w \leq Qw$ holds, what contradicts the fact that U^* is dual feasible for (4.180).

Now, if U_S^* is a feasible solution of the dual problem for coalition S, it follows from Lemma 3 in [33], that $U_S^* b(S) \leq Q_S x_S^0$ does not hold for any feasible x_S^0, in particular for the elements in $V(S)$.

□

As a consequence of this result, it is possible to obtain allocations of elements of $V(N)$ in the dominance core of the supply chain game since they correspond to feasible basic solutions of MD_N. Remark that if a maximizing multicriteria linear problem is bounded from above, it has at least one feasible basic solution which is efficient.

Another way of obtaining allocations in the dominance core of the game is by scalarization. See the following example.

Example 4.6.3 *To calculate one allocation in the dominance core of the multicriteria SChG arising from Example 4.6.2, we will make use of the construction explained in Theorem 2.4.1. It can be checked that the vector* $z^N = (8, 4) \in V(N)$, *i.e., it is a non-dominated feasible solution to Problem (4.179). Consider the scalarization that gives equal importance to both criteria, that is,* $\lambda = (0.5, 0.5)$. *From this vector a SChG* (N, v^λ) *arises. It can be seen that the corresponding problem* $Pr(N)$ *of this game is:*

$$
\begin{array}{rl}
\max & -2.5x_{12} + 2x_{13} + 3.5x_{23} + 2x_{24} - 2x_{34} \\
s.t.: & 0 \leq x_{12} + x_{13} \leq 4 \\
& 0 \leq x_{23} + x_{24} - x_{12} \leq 2 \\
& 0 \leq x_{13} + x_{23} - x_{34} \leq 1 \\
& 0 \leq x_{24} + x_{34} \leq 5 \\
& x_{ij} \geq 0 \quad \forall\ (i,j) \in A
\end{array}
\tag{4.183}
$$

The core allocation of the game (N, v^λ) *we need to build the dominance core allocation of the game* (N, V) *will be one Owen allocation. To do so the dual problem of the above-expressed LP problem is needed. Such problem is:*

$$
\begin{aligned}
\min \quad & 4y_1 + 2y_2 + 2y_3 + 4y_4 \\
s.t.: \quad & (y_1 - t_1) - (y_2 - t_2) \geq -2.5 \\
& (y_1 - t_1) + (y_3 - t_3) \geq 2 \\
& (y_2 - t_2) + (y_3 - t_3) \geq 3.5 \\
& (y_2 - t_2) + (y_4 - t_4) \geq 2 \\
& -(y_3 - t_3) + (y_4 - t_4) \geq -2 \\
& y_i \geq 0 \quad \forall\, i
\end{aligned}
\tag{4.184}
$$

One optimal feasible solution to Problem (4.184) is

$$y^* = (0, 2, 2, 0). \tag{4.185}$$

This solution leads to the Owen allocation

$$\gamma = (0, 4, 2, 0). \tag{4.186}$$

It goes without saying that $\gamma \in Core(N, v^\lambda)$. Then, we build the vectors

$$X^i = \frac{\gamma_i}{\lambda z^N} z^N, \tag{4.187}$$

which are meant to be the rows of the final allocation. Thus, the dominance core allocation results

$$X = \begin{pmatrix} 0 & 0 \\ 16/3 & 8/3 \\ 8/3 & 4/3 \\ 0 & 0 \end{pmatrix} \in C(N, V; \not\leq). \tag{4.188}$$

4.6.3 Multicriteria Transportation Games

A particular instance of the class MSChG is the multicriteria transportation game induced by the multicriteria transportation problem. Moreover, when the offers and demands are 1 and -1 for each of the nodes of the two partition sets, respectively, one gets the multicriteria assignment game.

As in the transportation game, in the multicriteria transportation game the set of players, $N = \{1, \ldots, n\}$ is partitioned into two disjoint subsets P and Q containing each one m and m' players, respectively. Here, the members of P will be called *origin players*, and the members of Q *destination players.* Each origin player $i \in P$ has a positive integer number of units of a certain indivisible good, p_i, and each destination player $j \in Q$ demands a positive integer number of units of this good, q_j. Sending of one unit from origin player i to destination player j produces a l-dimensional profit $(c_{ij}^1, c_{ij}^2, \ldots, c_{ij}^l)'$. Let $C \in \mathbb{R}^{l \times mm'}$ denote the array of all the l-dimensional profit vectors.

For every coalition $S \subset N = P \cup Q$, consider $P_S = S \cap P$ and $Q_S = S \cap Q$. For each $S \subset N$ we define the multicriteria transportation problem, T_S given by:

$$
\begin{aligned}
\max \quad & \left(\sum_{i \in P_S, j \in Q_S} c_{ij}^1 x_{ij} \ , \ \ldots \ , \sum_{i \in P_S, j \in Q_S} c_{ij}^l x_{ij} \right) \\
\text{s.t.:} \quad & \sum_{j \in Q_S} x_{ij} \leq p_i \quad \forall \, i \in P_S \\
& \sum_{i \in P_S} x_{ij} \geq q_j \quad \forall \, j \in Q_S \\
& x_{ij} \geq 0
\end{aligned}
\tag{4.189}
$$

It is easy to see that T_S is a particular case of MP_S, (just take $k_i = 0$ and $C = B$), and therefore our results on supply chain games can be applied.

An assignment game is a particular case of the transportation game. To model the assignment problem, the set of players, $N = \{1, \ldots, n\}$, is also partitioned into two disjoint subsets M and M' containing each one m and m' players and the supplies and demands are 1 and -1, respectively, for each of the nodes of the two partition sets. Now, the values of the variables are $x_{ij} \in \{0, 1\}$, and represent whether player $i \in M$ is assigned to player $j \in M'$.

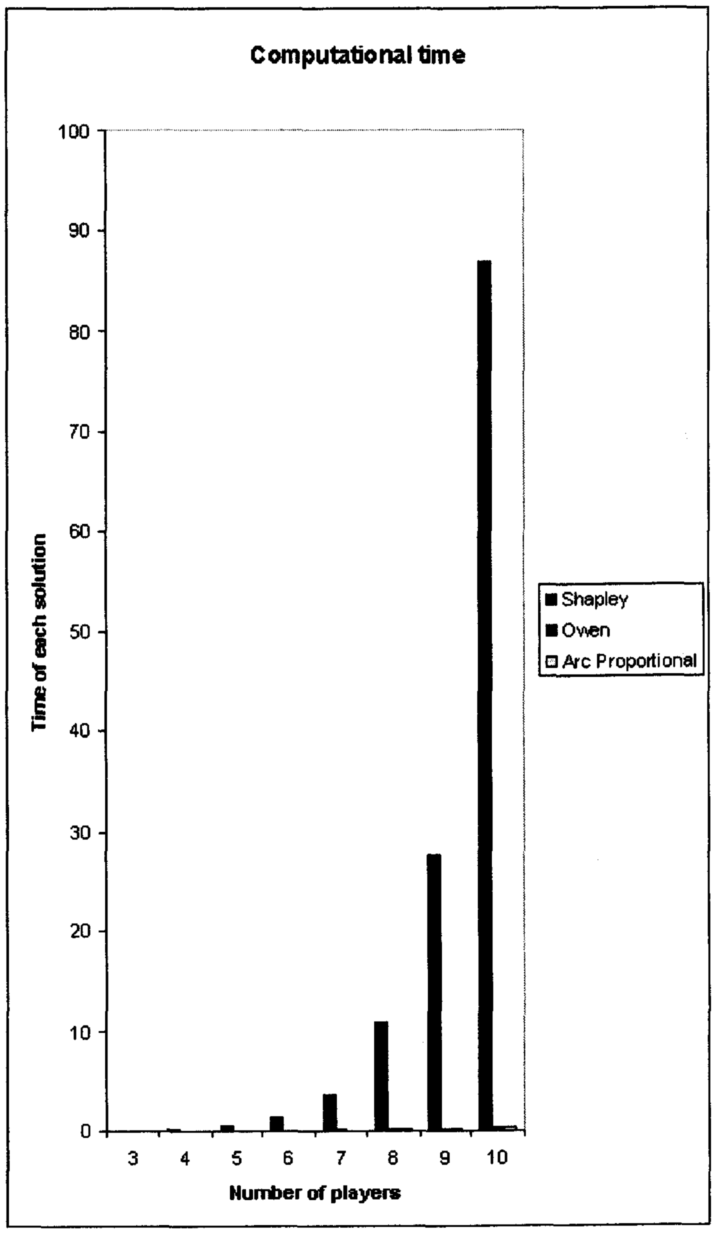

Figure 4.23: Computational time.

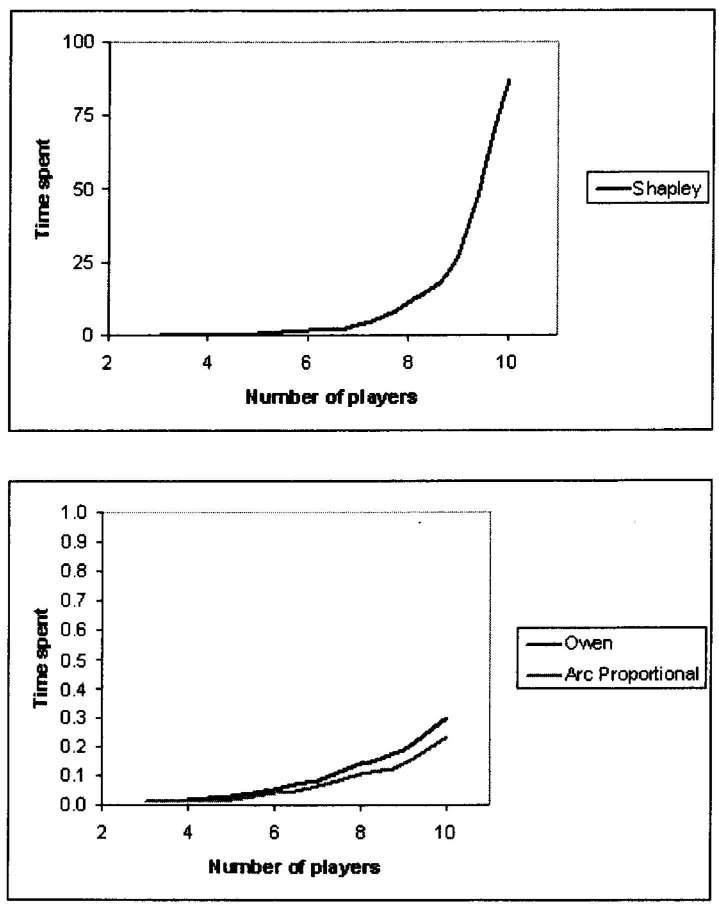

Figure 4.24: Evolution in time.

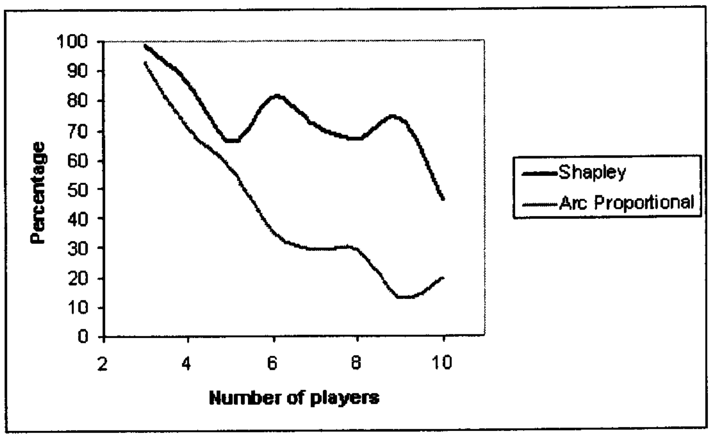

Figure 4.25: Frequency in the core.

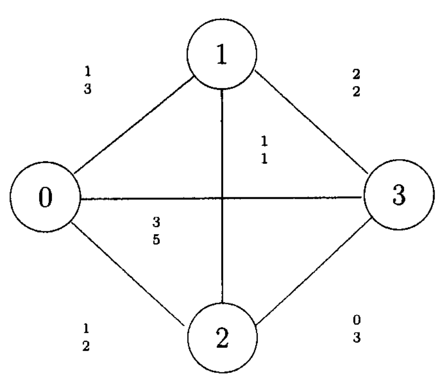

Figure 4.26: The graph G.

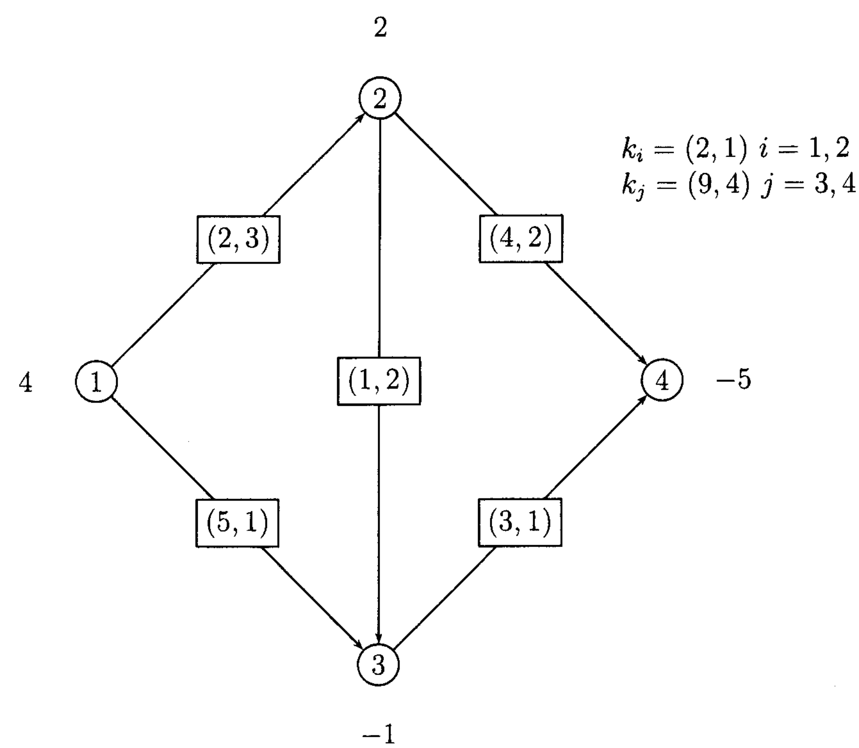

Figure 4.27: Multicriteria Supply Chain Problem

Chapter 5

Diameter Games

In this chapter a new class of cooperative games is presented, named *Diameter Games*. It is proved that diameter games are balanced. Two well-known solution concepts, the Shapley value and the nucleolus, are studied for this class of games. In general there is no efficient way of calculating neither the Shapley value nor the nucleolus. However, we show that for our special class of games both solution concepts can be computed in polynomial time. Additionally, we also prove that the nucleolus of a diameter game can be obtained by solving a continuous linear programming problem.

5.1 The game

Given a tree graph $G = (N \cup v_0, A)$, where v_0 is a distinguished node called the root of the tree we define the characteristic function v over all possible coalitions of N via:

$$v(S) := d(S \cup \{v_0\}) \ \forall \ S \subset N, \tag{5.1}$$

where $d(L)$ denotes the diameter of the set L over the tree G, for all $L \subset N \cup v_0$. The diameter of a set of nodes L over a tree is the maximum distance between two nodes of L. Note that $v(\emptyset) = 0$.

Once we have defined the set N and the characteristic function v over N, we define the diameter game $\Gamma = (N, v)$. Such a game satisfies the following properties:

- Γ is well defined, since $v(S) = d(S \cup \{v_0\})$ is well defined over the tree G.
- Trivially $v(S) \geq 0$ for all $S \subset N$ and $v(\emptyset) = 0$.
- Since the function diameter is submodular on a tree, see [67], the game Γ is convex, that is, $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$.
- A direct consequence of the convexity is that the game is subaditive, that is, for all $S, T \subset N$ such that $S \cap T = \emptyset$ one has $v(S \cup T) \leq v(S) + v(T)$.

In the following section it is shown that the core of diameter games is always non-empty.

5.2 The core

Given the game $\Gamma = (N, v)$, which is a cost game, the core of Γ is defined as:

$$C(N,v) := \{x \in \mathbb{R}^n \;:\; x(S) \leq v(S) \;\forall\, S \subset N,\; v(N) = \sum_{i \in N} x_i\}, \tag{5.2}$$

where $x(S) = \sum_{i \in S} x_i$. Let us see that $C(N,v) \neq \emptyset$ for every diameter game (N, v).

Theorem 5.2.1 Given is a tree graph G and (N, v) the diameter game defined over G. Then, $C(N,v) \neq \emptyset$.

Proof. Let (N, v) be a diameter game. In Corollary 3 of [19], it is proved that if the function v is submodular and weakly increasing, then $C(N, \overline{v}) \neq \emptyset$, where

$$\overline{v}(S) := \min\{\sum_j v(E_j) | E_j \in 2^N \;\forall\, j,\; E_j's \text{ partition } S\} \;\forall\, S \subset N. \tag{5.3}$$

In [19] it is also proved that $v = \overline{v}$ if and only if v is subaditive.

Since v is submodular and weakly increasing, one has that $C(N, \overline{v}) \neq \emptyset$. From the fact that v is subaditive, we deduce that $\overline{v} = v$ and consequently $C(N, v) \neq \emptyset$. $\square$

Now we present a core allocation for diameter games, based on the idea that the total cost of the grand coalition may be payed by the players that give the diameter of the tree.

Given a diameter game (N, v), the following events could happen:

1. If there exists $a \in N$ such that $v(N) = d(a, v_0) = d$, where $d(i, j)$ is the distance from i to j over the tree G, for all $i, j \in N \cup v_0$, then the following allocation

$$(5.4) \qquad x = (x_i)_{i \in N}, \quad x_i = \begin{cases} 0 & i \neq a \\ v(N) & i = a \end{cases}$$

is an allocation in the core of the game.

   (a) $x(N) = \sum_{i \in N} x_i = d = v(N)$.

   (b) Given $S \subset N$,

   i) If $a \in S$, then $x(S) = d = d(a, v_0) \leq d(S \cup \{v_0\}) = v(S)$.

   ii) If $a \notin S$ then $x(S) = 0 \leq v(S)$.

   In both cases we have that $x(S) \leq v(S)$.

From 1) and 2) we conclude that $x \in C(N, v)$.

2. If there exist $a, b \in N$ such that $v(N) = d(a, b) = d$, the following allocation

$$(5.5) \qquad x = (x_i)_{i \in N}, \quad x_i = \begin{cases} 0 & i \neq a, b \\ \frac{d(a,v_0)}{d(a,v_0)+d(b,v_0)}\, d & i = a \\ \frac{d(b,v_0)}{d(a,v_0)+d(b,v_0)}\, d & i = b \end{cases}$$

is a core allocation.

   (a) $x(N) = \frac{d(a,v_0)}{d(a,v_0)+d(b,v_0)}\, d + \frac{d(b,v_0)}{d(a,v_0)+d(b,v_0)}\, d = d = v(N)$.

   (b) Given $S \subset N$

   i) If both a and b are in S, then $x(S) = d(a, b) \leq d(S \cup v_0) = v(S)$.

   ii) If $a \in S$ and $b \notin S$, $x(S) = \frac{d(a,v_0)}{d(a,v_0)+d(b,v_0)}\, d(a, b) \leq \frac{d(a,v_0)}{d(a,v_0)+d(b,v_0)}\, (d(a, v_0) + d(v_0, b)) = d(a, v_0) \leq d(S \cup v_0) = v(S)$.

and therefore $x \in C(N, v)$.

Besides, it is interesting to underline that the calculation of the previous core allocations is extremely fast, if we compare with the calculation of a core allocation for a general game, for which we need to solve a linear programming problem with n variables and $O(2^n)$ constraints.

Therefore, we have the following theorem:

Theorem 5.2.2 An allocation in the core of Diameter Games can be obtained in linear time.

Proof. The result can be proved by taking into account that the complexity to find the diameter of a graph tree of n nodes is $O(n)$, see [39], and that the allocation, after the diameter is found, is done in constant time. □

But not only finding allocations in the core of a game is interesting. In general, checking if a given allocation is in the core is a NP-hard problem. Nowadays, several classes of games have been proven to have special characteristic: checking membership in the core is polynomial. Due to the special structure of the class of diameter games, one can state the following result:

Theorem 5.2.3 Checking membership in the core of diameter games can be done in polynomial time.

Proof. Given a diameter game (N, v) and an allocation of this game x, the proof of the result is trivial taking into account that one only has to check if $x(T(i,j)) \leq v(T(i,j))$ for all $i, j \in N$, where $T(i,j)$ is the minimum subtree containing i and j. Note that for all $S \subset T(i,j)$, such that $i, j \in S$, one has that $v(S) = v(T(i,j))$, and $x(S) \leq x(T(i,j))$ and, therefore, all those coalitions need not be checked. □

5.3 The Shapley value

Given that v is submodular, one has that the game (N, v) is convex. Thus the Shapley value is always an allocation in the core of the game. Recall that the Shapley value is the allocation $\phi = (\phi_1, \dots, \phi_n)$ given by

$$\phi_i = \sum_{S \subset N - \{i\}} \frac{s!(n-s-1)!}{n!}(v(S \cup \{i\}) - v(S)) \qquad \forall\, i \in N, \tag{5.6}$$

where $s = |S|$. In principle, assuming that the characteristic function is already known, to calculate ϕ it is necessary an exponential number of basic operations with respect to the number of players. In the rest of the section we show that, for diameter games, ϕ can be calculated in polynomial time.

Firstly we will provide two polynomial algorithms to calculate the Shapley value of a diameter game on a general tree. Later on we will show that such calculation can be done faster when the tree that generates our diameter game is linear.

General tree

Note that for each possible value of $v(S \cup \{i\}) - v(S)$ there can be more than one combination of coalitions and players giving us this value. Given $i \in N$, let us distinguish the following cases depending on the coalition $S \subset N - \{i\}$:

- $S = \emptyset$, then $v(S \cup \{i\}) = d(i, v_0),\ v(S) = 0$.
- $v(S \cup \{i\}) = d(v_0, i) \Rightarrow \exists\, j \in S\ :\ v(S) = d(j, v_0)$.
- $v(S \cup \{i\}) = d(i, j),\ j \in S$. Now we can have two possibilities:
  1. $v(S) = d(j, v_0),\ j \in S$.
  2. $v(S) = d(j, k),\ j, k \in S$.
- $v(S \cup \{i\}) = d(j,k)\ j,k \in S \cup \{v_0\} \Rightarrow v(S) = d(j,k) \Rightarrow v(S \cup \{i\}) - v(S) = 0$, and we do not have to take this case into consideration to do the calculation of the Shapley value.

Note that in order to enumerate the previous cases, we made use of the fact that if $v(S \cup \{i\}) = d(i,j), j \in S \cup \{v_0\}$, then $v(S)$ is given by the distance from j to another point of $S \cup \{v_0\}$. Making use of the previous properties, the following algorithm to calculate the Shapley value is proposed.

Algorithm 1

Given is $i \in N$.

1. Set $\phi_i := \frac{1}{n}\, d(v_0, i)$ (the marginal contribution of player i to the empty coalition times its coefficient in the Shapley Value.) This step can be done in constant time.

2. For all $j \in N - \{i\}$, do:

   (a) if $v(\{i,j\}) = d(v_0, i)$ let S_i^j be the set

   $$(5.7) \quad S_i^j = \{k \in N - \{i,j\} \;:\; v(\{i,j,k\}) = d(i, v_0) \text{ and } v(\{j,k\}) = d(j, v_0)\}.$$

   Clearly, $\forall\, L \subset S_i^j$ we have that

   $$(5.8) \qquad v(L \cup \{j\} \cup \{i\}) = d(v_0, i) \text{ and } v(L \cup \{j\}) = d(v_0, j).$$

   That means that all marginal contributions of i to coalitions $L \cup \{j\}$ are equal to $d(v_0, i) - d(v_0, j)$. Thus the coefficients of the Shapley value for all $L \cup \{j\}$ such that $L \subset S_i^j$ only depend on their cardinality.
   Then, we set

   $$(5.9) \qquad \phi_i := \phi_i + \sum_{t=0}^{|S_i^j|} \binom{|S_i^j|}{t} \frac{(t+1)!(n-t-2)!}{n!} \,(d(i, v_0) - d(j, v_0)).$$

   To build S_i^j we need $O(n)$ steps.

   (b) if $v(\{i,j\}) = d(i,j)$, let S_i^j be

   $$(5.10) \quad S_i^j = \{k \in N - \{i,j\} \;:\; v(\{i,j,k\}) = d(i,j) \text{ and } v(\{j,k\}) = d(j, v_0)\}.$$

Following the same reasoning as in (a), we set

$$\phi_i := \phi_i + \sum_{t=0}^{|S_i^j|} \binom{|S_i^j|}{t} \frac{(t+1)!(n-t-2)!}{n!} (d(i,j) - d(j, v_0)). \tag{5.11}$$

S_i^j can be built with complexity, as in case (a), $O(n)$.

(c) For all $\{k\} \subset N - \{i,j\}$ do:

if $v(\{i,j,k\}) = d(i,j)$ and $v(\{j,k)\} = d(j,k)$, let $S_i^{j,k}$ be the set

(5.12)

$S_i^{j,k} = \{l \in N - \{i,j,k\} \; : \; v(\{i,j,k,l\}) = d(i,j) \text{ and } v(\{j,k,l\}) = d(j,k)\}.$

For the same reason as in the two previous steps, we set

$$\phi_i := \phi_i + \sum_{t=0}^{|S_i^{j,k}|} \binom{|S_i^{j,k}|}{t} \frac{(t+2)!(n-t-3)!}{n!} (d(i,j) - d(j,k)). \tag{5.13}$$

S_i^{jk} takes $O(n^2)$ operations.

Since steps (a), (b), (c) have to be done for all $j \in N - \{i\}$, the complexity of step 2 is $O(n^3)$.

Note that in all other possible coalitions, $v(S \cup \{i\}) = v(S)$, and therefore they are not relevant for the Shapley value.

Since the effort to calculate ϕ_i is $O(n^3)$ for each $i = 1, \ldots, n$, the total complexity of the proposed algorithm to find the Shapley allocation is $O(n^4)$.

Let us see an example of how to calculate the Shapley value of a diameter game by means of the previously described algorithm.

Example 5.3.1 *Find in Figure 5.1 a tree graph with a root, v_0, and four other nodes. The distances between each other are represented over the corresponding arcs. From this tree the diameter game (N, v) can be built, where $N = \{1, 2, 3, 4\}$ and the values of v are expressed in the following tables:*

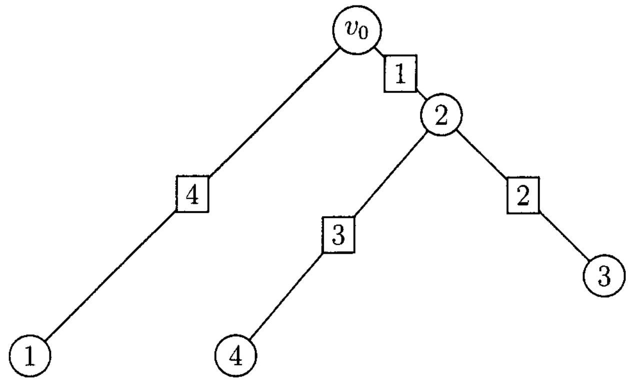

Figure 5.1: 4-node tree.

S	$v(S)$	S	$v(S)$	S	$v(S)$
{1}	4	{1, 3}	7	{1, 2, 3}	7
{2}	1	{1, 4}	8	{1, 2, 4}	8
{3}	3	{2, 3}	3	{1, 3, 4}	8
{4}	4	{2, 4}	4	{2, 3, 4}	5
{1, 2}	5	{3, 4}	5	{1, 2, 3, 4}	8

Let us calculate the Shapley value corresponding to the first player.

1. $d(v_0, 1) = 4,\ \phi_i = 1.$

2. *Now we analyze how player* $j = 2$ *contributes to* ϕ_i.

(5.14) $$v(1,2) = 5,\ d(1,2) = 5 \Rightarrow \textit{Case 2.b.}$$

It can be checked that $S_1^2 = \emptyset$. *Then*

(5.15) $$\phi_i = \phi_i + \frac{2!}{4!}(d(1,2) - d(v_0,2)) = 1 + \frac{1}{3} = \frac{4}{3}.$$

- *There is no* $k \in N - \{1,2\}$ *such that* $d(1,2,k) = d(1,2)$ *and* $v(2,k) = d(2,k)$.

3. *Now we analyze how player* $j = 3$ *contributes to* ϕ_i.

(5.16) $$v(1,3) = 7,\ d(1,3) = 7 \Rightarrow \textit{Case 2.b.}$$

It is easy to see that $S_1^3 = \{2\}$. *Therefore*

$$\phi_i = \phi_i + (\frac{2!}{4!} + \frac{2!}{4!})(d(1,3) - d(v_0,3)) = \frac{4}{3} + \frac{2}{3} = 2. \tag{5.17}$$

- *There is no* $k \in N - \{1,3\}$ *such that* $d(1,3,k) = d(1,3)$ *and* $v(3,k) = d(3,k)$.

4. *Now we analyze how player* $j = 4$ *contributes to* ϕ_i.

$$v(1,4) = 8,\ d(1,4) = 8 \Rightarrow \textit{Case 2.b.} \tag{5.18}$$

It can be checked that $S_1^4 = \{2\}$. *Thus*

$$\phi_i = \phi_i + (\frac{2!}{4!} + \frac{2!}{4!})(d(1,4) - d(v_0,4)) = 2 + \frac{4}{6} = \frac{8}{3}. \tag{5.19}$$

- *We have that* $d(1,4,3) = d(1,4)$ *and* $v(4,3) = d(4,3)$, *and* $k = 2$ *does not satisfy such conditions. It is easy to see that* $S_1^{43} = \{2\}$. *Therefore,*

$$\phi_i = \phi_i + (\frac{2!1!}{4!} + \frac{3!0!}{4!})(d(1,4) - d(4,3)) = ... = \frac{11}{3}. \tag{5.20}$$

Then we conclude that $\phi_1 = \frac{11}{3}$. *The other components of the allocation are calculated analogously. The Shapley value results:*

$$\phi = (\frac{11}{3}, \frac{1}{3}, \frac{3}{2}, \frac{5}{2}). \tag{5.21}$$

Algorithm 2

The second algorithm we propose calculates the possible values of the diameter and enumerate all coalitions whose characteristic function take each of those values. The values that the diameter can take depend on each possible pair of nodes i, j.

In the first phase of the algorithm $v(\{i,j\})$ (i and j not necessarily distinct) is calculated for each pair of nodes $i, j \in N$. The cost of this step is $O(n^2)$. Let $T(i,j)$ be the minimal subtree containing i, j and v_0 and let $N(i,j)$ be the number of nodes different from i and j in $T(i,j)$. Now we calculate $v(\{i,j,k\})$ for all $k \notin T(i,j)$. Therefore the complexity of the algorithm is increased to $O(n^3)$.

Afterwards the contribution of each value of the diameter to the Shapley value is computed. To do so we count the number of times in which node k is part of the coalitions that give such value of the diameter. For each pair i, j such value is denoted by $C(i,j)$ and given by the formula

$$C(i,j) = \begin{cases} \displaystyle\sum_{r=0}^{N(i,j)} \binom{N(i,j)}{r} \frac{(r+2)!(n-(r+2)-1)!}{n!} & i \neq j \\ \displaystyle\sum_{r=0}^{N(i,i)} \binom{N(i,i)}{r} \frac{(r+1)!(n-(r+1)-1)!}{n!} & i = j \end{cases} \tag{5.22}$$

This calculation is done in linear time. Thus so far the complexity of the algorithm is $O(n^3)$.

After having calculated $C(i,j)$ for each pair of nodes of the tree, the Shapley value of a given player k is:

$$\phi_k = \sum_{i,j \in N, k \notin T(i,j)} C(i,j)(v(\{i,j,k\} - v(\{i,j\})). \tag{5.23}$$

The complexity of this step is $O(n)$, and therefore the complexity of the algorithm remains $O(n^3)$.

5.3.1 Linear trees

The special case of a linear tree is that when every node (including the root) has at most one son. That means that for every $i \in N$ we only have to perform steps 1 and 2(a), the latter only for the ancestors of node i. The calculation of the Shapley value following the previous algorithm leads to this formula:

(5.24)

$$\phi_i = \sum_{k=1}^{i-1} \sum_{s=0}^{k-1} \binom{k-1}{s} \frac{(s+1)!(n-s-2)!}{n!} (d(i,v_0) - d(k,v_0)) + \frac{1}{n}\, d(i,v_0), \quad \forall i \in N.$$

Take into account that we have ordered the graph in the following way: $(v_0, 1, 2, \ldots, n)$, where 1 is the only son of v_0 and k is the only son of $k-1$, for k from 2 to n. The following example shows how to compute the Shapley value for a diameter game when the tree under consideration is linear.

Example 5.3.2 *In Figure 5.2 it is depicted an instance of a linear tree. The calculations of the Shapley value for the corresponding diameter game, according to Formula 5.24, result in:*

$$\phi = (1, 2, 3, 4). \tag{5.25}$$

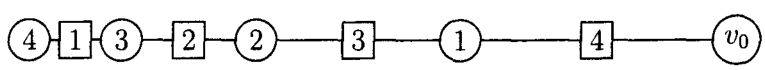

Figure 5.2: 4-node linear tree.

5.4 The nucleolus

Another well-known solution concept is the nucleolus, see [22] or [48].

From its definition, to calculate this allocation we have to solve the following linear program:

$$\begin{aligned} \min \quad & \alpha \\ \text{s.t.} \quad & x(S) - \alpha \le v(S) \quad \forall S \subsetneq N \\ & x(N) = v(N) \\ & x_i \ge 0 \quad \forall i \end{aligned} \tag{5.26}$$

In this problem there are $O(2^n)$ constraints and $n + 1$ variables. That means that the computational effort to calculate the nucleolus is exponential with respect to the number of players. But for diameter games the number of constraints can significantly be reduced to $O(n^2)$.

Given the tree $G = (N \cup v_0, A)$, there exist at most $\binom{n+1}{2}$ different values of the characteristic function v, that is, the possible pairs we can build from the nodes of G. Note that $\binom{n+1}{2} = \frac{n(n+1)}{2} \sim O(n^2)$.

For all $S_1 \subset S_2 \subset N$ such that $i, j \in S_1$, and $v(S_1) = v(S_2) = d(i,j)$, it is easy to see that

$$x(S_2) - \alpha \leq v(S_2) \Rightarrow x(S_1) - \alpha \leq v(S_1). \tag{5.27}$$

That assertion comes from the fact that $v(S_1) = v(S_2)$ and $x_i \geq 0 \ \forall i$.

Thus, given a pair $i, j \in N \cup \{v_0\}$, we look for $S^{i,j}$ the biggest coalition of N satisfying that $v(S) = d(i,j)$ and $i, j \in S \cup \{v_0\}$, keeping the constraint $x(S^{i,j}) - \alpha \leq v(S^{i,j})$ and deleting all the constrains related to the subsets of $S^{i,j}$. Therefore, we have to solve this new and smaller linear program:

$$\begin{array}{ll} \min & \alpha \\ \text{s.t.} & x(S^{i,j}) - \alpha \leq v(S^{i,j}) \quad \forall (i,j) \in N \cup \{v_0\} \\ & x(N) = v(N) \\ & x_i \geq 0 \quad \forall\, i \end{array} \tag{5.28}$$

There is no theoretical reason to ensure that the solution to the previous program is unique. So, in general we have as a solution a value α^1 and a set of allocations X^1. For the optimal value α^1 we have a set of coalitions, $\mathcal{B}^1$, satisfying that $x(S) - \alpha^1 = v(S)$ for all $S \in \mathcal{B}^1$. In this case we would proceed by solving the following linear program:

$$\begin{array}{ll} \min & \alpha \\ \text{s.t.} & x(S^{i,j}) - \alpha \leq v(S^{i,j}) \quad \forall (i,j) \in N \cup \{v_0\} \text{ and } S^{i,j} \notin \mathcal{B}^1 \\ & x(N) = v(N) \\ & x \in X^1 \end{array} \tag{5.29}$$

If the solution of this problem, (α^2, X^2) is such that X^2 is only one point, then this point is the nucleolus. Otherwise we proceed in an analogous way as we did before, until the we find a unique solution. This process is assured to be finished in, at most, n steps.

To summarize, and taking into account that the complexity for solving linear programming problems can be bounded by a linear function on the number of variables, the complexity of this algorithm can be bounded by $O(\frac{n(n+1)}{2} \cdot (n-1) \cdot n \cdot (n+1)) = O(n^6)$.

Let us now describe the algorithm in a more structured way:

Algorithm

Set $L = \emptyset$. For i from 1 to n do.

1. Look for every k such that $v(\{i,k\}) = d(i, v_0)$. Let S^i be the set $S^i = \{k \in N - \{i\} : v(\{i,k\}) = d(i, v_0)\} \cup \{i\}$.

2. For j from $i+1$ to n do:
if $v(\{i,j\}) = d(i,j)$, then $L = L \cup \{(i,j)\}$ and $S^{i,j} = \{k \in N - \{i,j\} : v(\{i,j,k\}) = d(i,j)\} \cup \{i,j\}$.

Solve the linear problem:

$$\begin{aligned} \max\ & \alpha \\ \text{s.t.:}\ & x(S^i) - \alpha \le d(i, v_0) \quad \forall\, i \in N \\ & x(S^{i,j}) - \alpha \le v(S^{i,j}) \quad \forall\, (i,j) \in L \\ & x(N) = v(N) \\ & x_i \ge 0 \quad \forall\, i \end{aligned} \tag{5.30}$$

If that problem has unique optimal solution then this is the nucleolus. Otherwise, we solve the previous problem over the set of optimal solutions and repeat the process until we get a unique optimal solution. This process is ensured to be finished in, at most, n steps. □

Let us now see an example of how to calculate the nucleolus of a diameter game.

Example 5.4.1 *Consider the diameter game in Example 5.3.1. Let us compute the nucleolus of such game by means of the algorithm just described.*

- $i = 1$, $d(1, v_0) = 4$. *Now we have to perform steps 1 and 2.*

1.

$$\left.\begin{array}{l} k=2,\ v(1,2)=5\neq d(1,v_0)\Rightarrow 2\notin S^1 \\ k=3,\ v(1,3)=7\neq d(1,v_0)\Rightarrow 3\notin S^1 \\ k=4,\ v(1,4)=8\neq d(1,v_0)\Rightarrow 4\notin S^1 \end{array}\right\}\Rightarrow S^1=\{1\}. \tag{5.31}$$

2. Now for all j from 2 to 4:

$$\begin{array}{l} j=2 \\ \qquad \left.\begin{array}{l} v(1,2)=5 \\ d(1,2)=5 \end{array}\right\}\Rightarrow \{1,2\}\in L \\ \qquad \left.\begin{array}{l} k=3,\ v(1,2,3)=7\neq d(1,2) \\ k=4,\ v(1,2,4)=8\neq d(1,2) \end{array}\right\}\Rightarrow S^{1,2}=\{1,2\} \\ j=3 \\ \qquad \left.\begin{array}{l} v(1,3)=7 \\ d(1,3)=7 \end{array}\right\}\Rightarrow \{1,3\}\in L \\ \qquad \left.\begin{array}{l} k=2,\ v(1,2,3)=7=d(1,3) \\ k=4,\ v(1,3,4)=8\neq d(1,3) \end{array}\right\}\Rightarrow S^{1,3}=\{1,2,3\} \\ j=4 \\ \qquad \left.\begin{array}{l} v(1,4)=8 \\ d(1,4)=8 \end{array}\right\}\Rightarrow \{1,4\}\in L \\ \qquad \left.\begin{array}{l} k=2,\ v(1,2,4)=8=d(1,4) \\ k=3,\ v(1,3,4)=8=d(1,4) \end{array}\right\}\Rightarrow S^{1,4}=\{1,2,3,4\} \end{array} \tag{5.32}$$

- $i=2$, $d(2,v_0)=1$. *Now we have to perform steps 1 and 2.*

1.

$$\left.\begin{array}{l} k=1,\ v(1,2)=5\neq d(2,v_0)\Rightarrow 1\notin S^2 \\ k=3,\ v(2,3)=3\neq d(2,v_0)\Rightarrow 3\notin S^2 \\ k=4,\ v(2,4)=4\neq d(2,v_0)\Rightarrow 4\notin S^2 \end{array}\right\}\Rightarrow S^2=\{2\}. \tag{5.33}$$

2. Now for $j = 3, 4$:

$$
\begin{array}{l}
j = 3 \\
\quad \left.\begin{array}{l} v(2,3) = 3 \\ d(2,3) = 2 \end{array}\right\} \Rightarrow \{1,2\} \notin L \\
\quad \left.\begin{array}{l} k = 1,\ v(1,2,3) = 7 \neq d(2,3) \\ k = 4,\ v(2,3,4) = 5 \neq d(2,3) \end{array}\right\} \Rightarrow S^{2,3} = \{2,3\} \\
j = 4 \\
\quad \left.\begin{array}{l} v(2,4) = 4 \\ d(2,4) = 3 \end{array}\right\} \Rightarrow \{2,4\} \notin L \\
\quad \left.\begin{array}{l} k = 1,\ v(1,2,4) = 8 \neq d(2,4) \\ k = 3,\ v(2,3,4) = 5 \neq d(2,4) \end{array}\right\} \Rightarrow S^{2,4} = \{2,4\}
\end{array}
\tag{5.34}
$$

- $i = 3$, $d(3, v_0) = 3$. *Now we have to perform steps 1 and 2.*

1.

$$
\left.\begin{array}{l}
k = 1,\ v(1,3) = 7 \neq d(3,v_0) \Rightarrow 1 \notin S^3 \\
k = 2,\ v(2,3) = 3 = d(3,v_0) \Rightarrow 2 \in S^3 \\
k = 4,\ v(3,4) = 5 \neq d(3,v_0) \Rightarrow 4 \notin S^3
\end{array}\right\} \Rightarrow S^3 = \{2,3\}.
\tag{5.35}
$$

2. Now we only have to check $j = 4$:

$$
\begin{array}{l}
j = 4 \\
\quad \left.\begin{array}{l} v(3,4) = 5 \\ d(3,4) = 5 \end{array}\right\} \Rightarrow \{3,4\} \in L \\
\quad \left.\begin{array}{l} k = 1,\ v(1,3,4) = 8 \neq d(3,4) \\ k = 2,\ v(2,3,4) = 5 = d(2,3) \end{array}\right\} \Rightarrow S^{3,4} = \{2,3,4\}
\end{array}
\tag{5.36}
$$

- $i = 4$, $d(4, v_0) = 4$. *For $i = 4$ we only have to perform step 1.*

1.

$$
\left.\begin{array}{l}
k = 1,\ v(1,4) = 8 \neq d(4,v_0) \Rightarrow 1 \notin S^4 \\
k = 2,\ v(2,4) = 3 = d(4,v_0) \Rightarrow 2 \in S^4 \\
k = 3,\ v(3,4) = 5 \neq d(4,v_0) \Rightarrow 3 \notin S^4
\end{array}\right\} \Rightarrow S^4 = \{2,4\}.
\tag{5.37}
$$

Therefore, the linear programming problem to solve in order to calculate the nucleolus of our diameter game is:

$$
(5.38)\quad
\begin{array}{lll}
\min & \alpha & \\
s.t.: & x_1 - \alpha \leq 4 & (S^1 = \{1\}) \\
 & x_2 - \alpha \leq 1 & (S^2 = \{2\}) \\
 & x_2 + x_3 - \alpha \leq 3 & (S^3 = \{2,3\}) \\
 & x_2 + x_4 - \alpha \leq 4 & (S^4 = \{2,4\}) \\
 & x_1 + x_2 - \alpha \leq 5 & (S^{1,2} = \{1,2\}) \\
 & x_1 + x_2 + x_3 - \alpha \leq 7 & (S^{1,3} = \{1,2,3\}) \\
 & x_1 + x_2 + x_3 + x_4 - \alpha \leq 8 & (S^{1,4} = \{1,2,3,4\}) \\
 & x_2 + x_3 + x_4 - \alpha \leq 5 & (S^{3,4} = \{2,3,4\}) \\
 & x_1 + x_2 + x_3 + x_4 = 8 & \\
 & x_i \geq 0 \; i = 1,2,3,4. &
\end{array}
$$

The only optimal solution to this linear program is

$$
(5.39)\qquad \alpha^* = 0, x_1^* = 4, x_2^* = 1, x_3^* = 2, x_4^* = 1.
$$

Therefore, the nucleolus of our diameter game is $(4,1,2,1)$.

Besides, we can state that finding the nucleolus of a diameter game can be done just by solving one linear programming problem.

Theorem 5.4.1 The nucleolus of a diameter game can be calculated by solving a linear programming problem with $O(n^4)$ variables and constraints.

Proof. The Nucleolus (α^*, x^*) corresponds to the lexicographical minimization of excesses. Therefore, there exists a permutation σ, which is denoted by $(\bullet)$, such that (α^*, x^*) is the lexicographical minimum with respect to $(\bullet)$ on the α-variables (excesses).

First of all, we prove that for the permutation $(\bullet)$, (α^*, x^*) is the unique minimum of $(1, \delta, \delta^2, \ldots, \delta^{n^2-1}, \theta)(\alpha, x)^t$ on $P = \{(\alpha, x) \in (\mathbb{R}^{n^2}, \mathbb{R}^n) : x(S) - \alpha_S \leq v(S), S \subset N\}$.

Take $z \in ext(P) - \{(\alpha^*, x^*)\}$, where $ext(P)$ denotes the set of extreme points of set P, and let $r \in \{1, 2, \ldots, n^2\}$ be such that $\alpha_k^* = z_k$ for $k < r$ and $\alpha_k^* > z_k$ for $k \geq r$. For any $\delta > 0$ we have that

(5.40)

$$\begin{aligned}(1,\delta,\delta^2,\ldots,\delta^{n^2-1},\theta)[(\alpha^*,x^*)-z]^t &= \delta^{r-1}(\alpha_r^*-z_r)+\sum_{k=r}^{n^2-1}\delta^k(\alpha_{k+1}-z_{k+1})\\ &= \delta^{r-1}[(\alpha_r^*-z_r)+\sum_{k=r}^{n^2-1}\delta^{k-r+1}(\alpha_{k+1}-z_{k+1})]\\ &= \delta^{r-1}\mathcal{K}(\delta)\end{aligned}$$

Note that $\mathcal{K}(\delta) \to (\alpha_r^* - z_r) > 0$ when $\delta \to 0$. This implies that the above scalar product is positive for all $\delta < \delta(z)$ ($\delta(z)$ depends on the extreme point z). Consider $\delta^* = \min\{\delta(z) : z \in ext(P) - \{(\alpha^*, x^*)\}\}$. Hence, for all $\delta < \delta^*$ one has that

$$(1,\delta,\delta^2,\ldots,\delta^{n^2-1},\theta)[(\alpha^*,x^*)-z]^t > 0 \quad \forall\, z \in ext(P)-\{(\alpha^*,x^*)\}. \tag{5.41}$$

However, for $z = (\alpha^*, x^*)$ it takes null value. Thus,

$$\begin{aligned}(\alpha^*,x^*) &= argmin\{(1,\delta,\delta^2,\ldots,\delta^{n^2-1},\theta)z^t : z \in ext(P)\}\\ &= argmin\{(1,\delta,\delta^2,\ldots,\delta^{n^2-1},\theta)(\alpha,x) : (\alpha,x) \in ext(P)\}\ \forall\, \delta < \delta^*.\end{aligned} \tag{5.42}$$

Now we have to prove that the problem

$$\begin{aligned}\min\ & \sum_{i=0}^{n^2-1}\delta^i\alpha_{(i)}\\ \text{s.t.:}\ & \alpha_{(0)} \geq \alpha_{(1)} \geq \cdots \geq \alpha_{(n^2-1)}\\ & (\alpha,x) \in P\end{aligned} \tag{5.43}$$

can be written as a linear programming problem so as to apply the above argument.

Consider the following linear programming problem

$$\begin{aligned}\min\ & \sum_{i=0}^{n^2-1}(\delta^i-\delta^{i+1})(it_i+\sum_{k=0}^{n^2-1}d_{ki})\\ \text{s.t.:}\ & d_{ki} \geq \alpha_k - t_i \quad \forall\, i,k.\\ & x(T_k)-\alpha_k \leq d(T_k) \quad \forall\, k.\\ & x(N) = d(T).\end{aligned} \tag{5.44}$$

where T_k is the coalition corresponding to α_k. The objective function and the first group of constraints represent the ordered weight sum of the values $\sum_{i=0}^{n^2-1}\delta^i\alpha_{(i)}$, where $\alpha_{(0)} \geq \alpha_{(1)} \geq \cdots \geq \alpha_{(n^2-1)}$. Notice that this tool is the formulation in [46]. It is applicable to here because we consider the convex case of the weighted ordered average, i.e

$\delta^0 \geq \delta^1 \geq \cdots \geq \delta^{n^2-1}$. This formulation, together with the fact that for the permutation ($\bullet$), (α^*, x^*) is the unique minimum of $(1, \delta, \delta^2, \ldots, \delta^{n^2-1}, \theta)(\alpha, x)^t$ on P, proves that computing the nucleolus of a diameter game is equivalent to a continuous linear program with $O(n^4)$ variables and constraints. □

To conclude the chapter, note that the extension of diameter games to general graphs can be done in a natural way, and the algorithm proposed for calculating the Shapley value and the nucleolus would still be valid.

Chapter 6

Multidimensional Assignment games

In this chapter a new class of cooperative games, named Multidimensional Assignment games (MDA games), is presented. Such class of games arises from the combinatorial optimization problem known as Multidimensional Assignment problem. The class of MDA games is, in general, non-balanced. Nevertheless, totally balanced subclasses of MDA games are found. Due to the NP-hardness of the MDA problem, we make use of approximation algorithms to give allocations in MDA games.

6.1 Multidimensional assignment problem

A w-dimensional assignment problem consists of w pairwise disjoint sets, named $N^1, N^2, \ldots, N^w$, of the form

$$N^k = \{i_1^k, \ldots, i_m^k\} \quad k = 1, \ldots, w. \tag{6.1}$$

The assignment of agents $\{i^1, \ldots, i^w\}$, where $i^k \in N^k \ \forall\ k$, results in a benefit equal to $a_{i^1,\ldots,i^w}$ units. In such a case we say that agents $\{i^1, \ldots, i^w\}$ are *associated*. The problem that arises when we want to associate the elements of $N^1, N^2, \ldots, N^w$ so that the total benefit obtained is maximized is a MDA problem of dimension w, or just a w-dimensional assignment problem.

This situation can be described by a linear programming problem. Let us consider the variables $x_{i^1\ldots i^w} \in \{0,1\}$ $\forall$ $i^k \in N^k$, $j = 1,\ldots,w$, $x_{i^1\ldots i^w} = 1$ if the assignment $(i^1,\ldots,i^w)$ is made and zero otherwise. The linear program that solves the MDA problem is:

$$
\begin{array}{lll}
\max & \sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} a_{i^1\ldots i^w} x_{i^1\ldots i^w} & \\
\text{s.t.:} & \sum_{i^2 \in N^2} \cdots \sum_{i^w \in N^w} x_{i^1\ldots i^w} \leq 1 & \forall\, i^1 \in N^1 \\
 & \sum_{i^1 \in N^1} \cdots \sum_{i^{k-1} \in N^{k-1}} \sum_{i^{k+1} \in N^{k+1}} \cdots \sum_{i^w \in N^w} x_{i^1\ldots i^w} \leq 1 & \forall\, i^k \in N^k,\ 1 < k < w \\
 & \sum_{i^1 \in N^1} \cdots \sum_{i^{w-1} \in N^{w-1}} x_{i^1\ldots i^w} \leq 1 & \forall\, i^w \in N^w \\
 & x_{i^1\ldots i^w} \in \{0,1\} & \forall\, (i^1,\ldots,i^w)
\end{array}
\tag{6.2}
$$

To summarize, a w-dimensional assignment problem is denoted by its sets of agents and the vector of benefits

$$(N^1,\ldots,N^w;a). \tag{6.3}$$

Let us see an example of such a situation:

Example 6.1.1 *Suppose that we have two factories, two warehouses and two shops. We know that if factory i, warehouse j and shop k are associated, they together produce a benefit of a_{ijk}. We also know that both factories, warehouses and shops can be associated with only one of the others, that is, only one factory with only one warehouse with only one shop. The benefits a_{ijk} are shown in Table 6.1*

a_{111}	a_{112}	a_{121}	a_{122}	a_{211}	a_{212}	a_{221}	a_{222}
3	4	2	5	2	6	5	4

Table 6.1: Table of benefits.

So, the formulation of this problem as a linear program is:

$$
(6.4) \quad
\begin{array}{ll}
\max & 3x_{111} + 4x_{112} + 2x_{121} + 5x_{122} + 2x_{211} + 6x_{212} + 5x_{221} + 4x_{222} \\
s.t.: & x_{111} + x_{112} + x_{121} + x_{122} \leq 1 \\
& x_{211} + x_{212} + x_{221} + x_{222} \leq 1 \\
& x_{111} + x_{112} + x_{211} + x_{212} \leq 1 \\
& x_{121} + x_{122} + x_{221} + x_{222} \leq 1 \\
& x_{111} + x_{121} + x_{211} + x_{221} \leq 1 \\
& x_{112} + x_{122} + x_{212} + x_{222} \leq 1 \\
& x_{ijk} \in \{0,1\} \ \forall \ i,j,k = 1,2
\end{array}
$$

One optimal feasible solution to the relaxed problem of (6.4) is the vector

$$
(6.5) \qquad [\frac{1}{2}, 0, 0, \frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0]
$$

which leads to a value in the objective function equal to $\frac{19}{2}$.

If we solve the problem taking into account the integer constraints, we obtain the solution

$$
(6.6) \qquad x_{112} = x_{221} = 1, \quad x_{ijk} = 0 \text{ otherwise}
$$

which produces an objective function value equal to 9.

Thus, their optimal associations are:

- *Factory* 1, *warehouse* 1 *and shop* 2.
- *Factory* 2, *warehouse* 2 *and shop* 1.

MDA problems are known to be **NP**-hard problems except in the trivial case $w = 1$ and the classical assignment problem $w = 2$. This fact will make us use approximation algorithms to efficiently allocate benefits in the arising MDA games, as we explain in following sections.

6.2 Multidimensional Assignment Games

Let us consider a w-dimensional assignment problem $(N^1, \ldots, N^w; a)$. Suppose that the agents interacting in the MDA problem have conflicting objectives but, at the same time, they all want to maximize their respective benefits. Thus, a cooperative game, $G = (N, v)$, naturally arises.

The set of players N is obtained as

$$N = N^1 \cup \cdots \cup N^w. \tag{6.7}$$

In order to calculate the characteristic function of this game, the maximum benefit that each coalition can make by themselves has to be calculated. For each $S \subset N$ we define the following sets:

$$N_S^k = N^k \cap S \ \forall \ k = 1, \ldots, w. \tag{6.8}$$

Thus, $v(S)$ is defined as the value of the following linear program:

$$\begin{array}{lll}
\max & \displaystyle\sum_{i^1 \in N_S^1} \cdots \sum_{i^w \in N_S^w} a_{i^1 \ldots i^w} x_{i^1 \ldots i^w} & \\
\text{s.t.:} & \displaystyle\sum_{i^2 \in N_S^2} \cdots \sum_{i^w \in N_S^w} x_{i^1 \ldots i^w} \leq 1 & \forall \ i^1 \in N_S^1 \\
& \displaystyle\sum_{i^1 \in N_S^1} \cdots \sum_{i^{k-1} \in N_S^{k-1}} \sum_{i^{k+1} \in N_S^{k+1}} \cdots \sum_{i^w \in N_S^w} x_{i^1 \ldots i^w} \leq 1 & \begin{array}{l} \forall \ i^k \in N_S^k, \\ 2 < k < w-1 \end{array} \\
& \displaystyle\sum_{i^1 \in N_S^1} \cdots \sum_{i^{w-1} \in N_S^{w-1}} x_{i^1 \ldots i^w} \leq 1 & \forall \ i^w \in N_S^w \\
& x_{i^1 \ldots i^w} \in \{0, 1\} &
\end{array} \quad (P_S) \tag{6.9}$$

Definition 6.2.1 *Let (N, v) be such that $N = N^1 \cup \cdots \cup N^w$ with $N^i \cap N^j = \emptyset \ \forall \ i \neq j$ and $v(S)$ is obtained from (6.9). Then (N, v) is a w-dimensional assignment game.*

The first conclusions we deduce from the definition of MDA games are:

Proposition 6.2.1 Let (N, v) be a multidimensional assignment game. Then v is well defined and nonnegative.

Proof. Obvious from its definition. □

The following property shows that, in MDA games players are forced to make agreements.

Proposition 6.2.2 MDA games are 0-normalized by (w-1)-groups, where w is the size of the associated MDA problem. That means that $v(S) = 0 \;\forall\; S : |S| < w$.

Proof. Obvious. □

Proposition 6.2.3 MDA games are monotonic.

Proof. Let $S \subset T \subset N$. Since the vector x^* that gives the maximum value of Problem 6.9 for coalition S is also feasible in the corresponding problem for coalition T, we have that $v(S) \leq v(T)$. □

Proposition 6.2.4 MDA games are superadditive.

Proof. Let $S, T \subset N$ such that $S \cap T = \emptyset$. We have to prove that $v(S) + v(T) \leq v(S \cup T)$.

Let x^S and x^T an optimal vector for problems P_S and P_T, respectively. Define the variable vector $x^{S \cup T}$ as follows:

$$x^{S\cup T}_{i^1 \ldots i^w} = \begin{cases} x^S_{i^1 \ldots i^w} & \text{if } (i^1 \ldots i^w) \in N^1_S \times \cdots \times N^w_S \\ x^T_{i^1 \ldots i^w} & \text{if } (i^1 \ldots i^w) \in N^1_T \times \cdots \times N^w_T \\ 0 & \text{otherwise.} \end{cases} \tag{6.10}$$

It is easy to check that variables $x^{S \cup T}$ are feasible for problem $P_{S \cup T}$. Since $v(N \cup T)$ is the optimal value of problem $P_{S\cup T}$, one has that

$$
\begin{aligned}
v(S\cup T) &\geq \sum_{(i^1,\ldots,i^w)\in N^1_{S\cup T}\times\cdots\times N^w_{S\cup T}} a_{i^1\ldots i^w} x^{S\cup T}_{i^1\ldots i^w} \\
&= \sum_{(i^1,\ldots,i^w)\in N^1_S\times\cdots\times N^w_S} a_{i^1\ldots i^w} x^{S\cup T}_{i^1\ldots i^w} + \sum_{(i^1,\ldots,i^w)\in N^1_T\times\cdots\times N^w_T} a_{i^1\ldots i^w} x^{S\cup T}_{i^1\ldots i^w} \\
&= \sum_{(i^1,\ldots,i^w)\in N^1_S\times\cdots\times N^w_S} a_{i^1\ldots i^w} x^{S}_{i^1\ldots i^w} + \sum_{(i^1,\ldots,i^w)\in N^1_T\times\cdots\times N^w_T} a_{i^1\ldots i^w} x^{T}_{i^1\ldots i^w} \\
&= v(S)+v(T)
\end{aligned} \tag{6.11}
$$

This concludes the proof. □

Now we show an example of a MDA game to clarify concepts.

Example 6.2.1 *Let us consider the Example 6.1.1. The corresponding MDA game is constructed as follows. Players 1 and 2 are factories 1 and 2 respectively. Players 3 and 4 are warehouses 1 and 2 and players 5 and 6 are shops 1 and 2. A way of allocating the profit among the players could be by considering the dual problem of the program* (P_N), *that is, building the so called* Owen set, *see [47]. Such a linear problem is:*

$$
\begin{aligned}
\min\ & y_1+y_2+y_3+y_4+y_5+y_6 \\
s.t.:\ & y_1+y_3+y_5\geq 3 \\
& y_1+y_3+y_6\geq 4 \\
& y_1+y_4+y_5\geq 2 \\
& y_1+y_4+y_6\geq 5 \\
& y_2+y_3+y_5\geq 2 \\
& y_2+y_3+y_6\geq 6 \\
& y_2+y_4+y_5\geq 5 \\
& y_2+y_4+y_6\geq 4
\end{aligned} \tag{6.12}
$$

One optimal feasible solution to (6.12) is the vector

$$[\frac{3}{2},3,0,\frac{1}{2},\frac{3}{2},3] \tag{6.13}$$

which leads to a value of the objective function equal to $\frac{19}{2}$. *So, the allocation arising from this solution cannot be taken since it is not feasible (it allocates* 9.5 *units while the great coalition only gets* 9 *units). One could think of normalizing this allocation so that it is* *feasible, multiplying it in this case by the factor* $\frac{9}{9.5}$. *This leads to the allocation*

$$[1.42105, 2.84211, 0, 0.473684, 1.42105, 2.84211]. \tag{6.14}$$

Unfortunately this allocation is not in the core of the game, since the coalition $\{2,3,6\}$ *gets a payoff equal to* 5.68 *units from the allocation above, while if they act on their own they can get* 6 *units.*

If we add the integer constraints, the solution to (6.12) is

$$[1, 3, 0, 1, 2, 3] \tag{6.15}$$

with an objective value equal to 10, *strictly higher than* $v(N)$. *Following a similar reasoning as before, we check that the normalization of this allocation does not produce an allocation in the core.*

As the Duality Theorem, see [4], does not satisfy for integer linear programming, we cannot apply the Owen solution to MDA games.

6.3 Balancedness of Multidimensional Assignment Games

In this section we show that MDA games can have empty core. First we need the following lemma.

Lemma 6.3.1 Let (N, v) be a w-dimensional assignment game. The following two problems are equivalent.

$$\begin{aligned} \min \quad & x(N) \\ \text{s.t.:} \quad & x(S) \geq v(S) \ \forall\ S \subset N \end{aligned} \qquad (P_a)$$

(6.16)

$$\begin{aligned} \min \quad & x(N) \\ \text{s.t.:} \quad & x_{i^1} + \cdots + x_{i^w} \geq a_{i^1,\cdots,i^w} \ \forall\ (i^1, \cdots, i^w) \in N^1 \times \cdots \times N^w \quad (P_b) \\ & x_i \geq 0 \ \forall i \in N \end{aligned}$$

Proof. As the objective function of both problems is the same, it is enough to prove that for any $x \in \mathbb{R}^n$ the following implication holds:

$$x \text{ feasible for } P_a \Leftrightarrow x \text{ feasible for } P_b \tag{6.17}$$

1. $\Rightarrow$ Trivial.

2. $\Leftarrow$ Let x be feasible for P_b, that is, $x_{i^1} + \cdots + x_{i^w} \geq a_{i^1,\cdots,i^w}$ $\forall$ $(i^1, \cdots, i^w) \in N^1 \times \cdots \times N^w$ and all its components are nonnegative. For all $S \subset N$, $v(S) = \sum_{j=1}^{q} a_{i_j^1 \ldots i_j^w}$ where i_j^k are distinct members of S for all $k = 1, \ldots, w, j = 1, \ldots, q$. Then

$$\begin{aligned} v(S) = & \sum_{j=1}^{q} a_{i_j^1 \ldots i_j^w} \leq \sum_{j=1}^{q} x_{i_j^1} + \ldots x_{i_j^w} \\ & \leq \sum_{i \in S} x_i = x(S) \end{aligned} \tag{6.18}$$

This concludes that x is feasible for P_a and the result follows.

□

The following example shows an MDA game with empty core.

Example 6.3.1 *It is well-known that for every cooperative game (N, v), the optimal value of Problem (6.19) is less than or equal to $v(N)$ if and only if $C(N, v) = \emptyset$.*

$$\begin{aligned} \min \quad & x(N) \\ s.t.: \quad & x(S) \geq v(S) \ \forall S \subset N, \ S \neq N \end{aligned} \tag{6.19}$$

Applying Lemma 6.3.1, the corresponding Problem (6.19) to the MDA game in Example 6.2.1 is the following linear program:

$$
\begin{array}{llllllll}
\min & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & \\
s.t.: & x_1 & & +x_3 & & +x_5 & & >= 3 \\
& x_1 & & +x_3 & & & +x_6 & >= 4 \\
& x_1 & & & +x_4 & +x_5 & & >= 2 \\
& x_1 & & & +x_4 & & +x_6 & >= 5 \\
& & x_2 & +x_3 & & +x_5 & & >= 2 \\
& & x_2 & +x_3 & & & +x_6 & >= 6 \\
& & x_2 & & +x_4 & +x_5 & & >= 5 \\
& & x_2 & & +x_4 & & +x_6 & >= 4
\end{array} \tag{6.20}
$$

An optimal solution to (6.20) is

$$x_1 = 3, x_2 = \frac{9}{2}, x_3 = 0, x_4 = \frac{1}{2}, x_5 = 0, x_6 = \frac{3}{2} \tag{6.21}$$

with an optimal value equal to $\frac{19}{2}$, which is strictly higher than $v(N) = 9$. Then we conclude that this game has an empty core.

6.3.1 A balanced subclass

Although we have seen that MDA games can have empty core, it is possible to define subclasses of these games that contain only balanced games.

Definition 6.3.1 *An MDA problem $(N^1, \ldots, N^w, A)$ is said to be separable if for all $(i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w$ there exist $a^k_{i^k i^{k+1}}$ $\forall\ k = 1, \ldots, w-1$ such that $a_{i^1,\ldots,i^w} = \sum_{k=1}^{w-1} a^k_{i^k i^{k+1}}$.*

The definition of separable MDA games follows naturally.

Definition 6.3.2 *Let (N, v) be a MDA game. We say that (N, v) is separable if the underlying MDA problem is separable.*

A direct consequence of the definition of separable MDA games is that they can be divided into classical assignment games. The following lemma proves that assertion.

Lemma 6.3.2 Let $G = (N^1, \dots, N^w; v)$ be a separable w-dimensional assignment game. From G, $w-1$ classical assignment games can be created, $G_1 = (N^1 \cup N^2, v^1)$, ... , $G_{w-1} = (N^{w-1} \cup N^w, v^{w-1})$, satisfying that

$$\sum_{k=1}^{w-1} v^k(N^k \cup N^{k+1}) = v(N). \tag{6.22}$$

Proof. For each game G_k, v^k is defined as the classical assignment game with benefits $a^k_{i^k i^{k+1}}$ $\forall\ i^k \in N^k, i^{k+1} \in N^{k+1}$, $\forall\ k = 1, \dots, w-1$.

The value of $v^k(N^k \cup N^{k+1})$ is obtained from the optimal value of the following linear programming problem:

$$(P^k) \quad \begin{array}{ll} \max & \displaystyle\sum_{i \in N^k} \sum_{j \in N^{k+1}} a^k_{ij} x^k_{ij} \\ \text{s.t.:} & \displaystyle\sum_{j \in N^{k+1}} x^k_{ij} \leq 1\ \forall\ i \in N^k \\ & \displaystyle\sum_{i \in N^k} x^k_{ij} \leq 1\ \forall\ j \in N^{k+1} \\ & x^k_{ij} \in \{0,1\}\ \forall\ i \in N^k, j \in N^{k+1} \end{array} \tag{6.23}$$

On the other hand, the value of $v(N)$ is computed after solving the following linear programming problem:

$$(P) \quad \begin{array}{ll} \max & \displaystyle\sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} a_{i^1 \dots i^w} x_{i^1 \dots i^w} \\ \text{s.t.:} & \displaystyle\sum_{i^2 \in N^2} \cdots \sum_{i^w \in N^w} x_{i^1 \dots i^w} \leq 1\ \forall\ i^1 \in N^1, \\ & \displaystyle\sum_{i^1 \in N^1} \cdots \sum_{i^{k-1} \in N^{k-1}} \sum_{i^{k+1} \in N^{k+1}} \cdots \sum_{i^w \in N^w} x_{i^1 \dots i^w} \leq 1\ \forall\ i^k \in N^k,\ 1 < k < w \\ & \displaystyle\sum_{i^1 \in N^1} \cdots \sum_{i^{w-1} \in N^{w-1}} x_{i^1 \dots i^w} \leq 1\ \forall\ i^w \in N^w \\ & x_{i^1 \dots i^w} \in \{0,1\}\ \forall\ (i^1, \dots, i^w) \in N^1 \times \cdots \times N^w \end{array} \tag{6.24}$$

Let $(\overline{x}^k_{i^k i^{k+1}})_{(i^k, i^{k+1}) \in N^k \times N^{k+1}}$ be an optimal solution to Problem (6.23) $\forall\, k = 1, \ldots, w-1$. For all $(i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w$, define

$$\overline{x}_{i^1 \ldots i^w} = \prod_{k=1}^{w-1} \overline{x}^k_{i^k i^{k+1}}. \tag{6.25}$$

Let us see that $(\overline{x}_{i^1 \ldots i^w})_{(i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w}$ is an optimal solution to Problem (6.24).

Suppose that there exist $\widehat{x}_{i^1 \ldots i^w}$ feasible variables for Problem (6.24) such that

$$\sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} a_{i^1 \ldots i^w} \widehat{x}_{i^1 \ldots i^w} > \sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} a_{i^1 \ldots i^w} \overline{x}_{i^1 \ldots i^w}. \tag{6.26}$$

Due to the separability of (N, v), we have that

$$\sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} a_{i^1 \ldots i^w} \overline{x}_{i^1 \ldots i^w} = \sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} \left(\sum_{k=1}^{w-1} a^k_{i^k i^{k+1}}\right) \overline{x}_{i^1 \ldots i^w}. \tag{6.27}$$

Since $\overline{x}_{i'^1 \ldots i'^w} = 1 \Leftrightarrow \overline{x}^k_{i'^k i'^{k+1}} = 1\ \forall\, k$, one has that the equation above is equal to

$$\sum_{k=1}^{w-1} \sum_{i^k \in N^k} \sum_{i^{k+1} \in N^{k+1}} a^k_{i^k i^{k+1}} \overline{x}^k_{i^k i^{k+1}}. \tag{6.28}$$

Define the following variables for each k and $\forall\, i^k \in N^k, i^{k+1} \in N^{k+1}$:

$$\widehat{x}^k_{i^k i^{k+1}} = \begin{cases} 1 & \text{if } \exists\, (i^1, \ldots, i^{k-1}, i^{k+1}, \ldots, i^w) : \widehat{x}_{i^1 \ldots i^k i^{k+1} \ldots i^w} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.29}$$

One can check that, for all k, these variables are feasible for P_k. Following an analogous reasoning as in (6.28), one has that

$$\sum_{i^1 \in N^1} \cdots \sum_{i^w \in N^w} a_{i^1 \ldots i^w} \widehat{x}_{i^1 \ldots i^w} = \sum_{k=1}^{w-1} \sum_{i^k \in N^k} \sum_{i^{k+1} \in N^{k+1}} a^k_{i^k i^{k+1}} \widehat{x}^k_{i^k i^{k+1}} \tag{6.30}$$

This implies that

$$(6.31)\qquad \sum_{k=1}^{w-1}\sum_{i^k\in N^k}\sum_{i^{k+1}\in N^{k+1}} a^k_{i^k i^{k+1}}\widehat{x}^k_{i^k i^{k+1}} > \sum_{k=1}^{w-1}\sum_{i^k\in N^k}\sum_{i^{k+1}\in N^{k+1}} a^k_{i^k i^{k+1}}\overline{x}^k_{i^k i^{k+1}}.$$

From Equation (6.31), we deduce that there exists $k^* \in \{1,\ldots,w-1\}$ such that

$$(6.32)\qquad \sum_{i^{k^*}\in N^k}\sum_{i^{k^*+1}\in N^{k^*+1}} a^{k^*}_{i^{k^*} i^{k^*+1}}\widehat{x}^{k^*}_{i^{k^*} i^{k^*+1}} > \sum_{i^{k^*}\in N^{k^*}}\sum_{i^{k^*+1}\in N^{k^*+1}} a^{k^*}_{i^{k^*} i^{k^*+1}}\overline{x}^{k^*}_{i^{k^*} i^{k+1}},$$

which is a contradiction because variables $\overline{x}^{k^*}_{i^{k^*} i^{k^*+1}}$ are optimal for P_{k^*}. This contradiction proves that variables $(\overline{x}_{i^1\ldots i^w})_{(i^1,\ldots,i^w)\in N^1\times\cdots\times N^w}$ are optimal for problem P.

Thus, one has that

$$(6.33)\qquad \begin{aligned} v(N) &= \sum_{i^1\in N^1}\cdots\sum_{i^w\in N^w} a_{i^1\ldots i^w}\overline{x}_{i^1\ldots i^w} = \sum_{i^1\in N^1}\cdots\sum_{i^w\in N^w}\Big(\sum_{k=1}^{w-1} a^k_{i^k i^{k+1}}\Big)\overline{x}_{i^1\ldots i^w} \\ &= \sum_{k=1}^{w-1}\sum_{i^k\in N^k}\sum_{i^{k+1}\in N^{k+1}} a^k_{i^k i^{k+1}}\overline{x}^k_{i^k i^{k+1}} = \sum_{k=1}^{w-1} v^k(N^k\cup N^{k+1}), \end{aligned}$$

which concludes the proof.

□

Now we give two different proofs to show that separable MDA games are balanced.

Theorem 6.3.1 Let (N,v) be a separable MDA game. One has that (N,v) is totally balanced.

Proof. Let (N,v) be the separable game arising from the separable MDA problem $(N^1,\ldots,N^w;a)$. Consider the games $G_1,\ldots,G_{w-1}$ as defined in Lemma 6.3.2.

Let $k\in\{1,\ldots,w-1\}$. Since G_k is a balanced game, applying Lemma 6.3.1, we have that the value of Problem (6.34) is less than or equal to $v^k(N^k\cup N^{k+1})$. Consider the following linear programming problem:

(6.34)
$$\begin{array}{lll} \min & \sum_{i \in N^k \cup N^{k+1}} x_i & \\ \text{s.t.:} & x_{i^k} + x_{i^{k+1}} \geq a^k_{i^k i^{k+1}} & \forall\ (i^k, i^{k+1}) \in (N^k \times N^{k+1}) \quad (P_k) \\ & x_i \geq 0 & \forall\ i \in N^k \cup N^{k+1} \end{array}$$

Rename the variables of problems P_k as follows:

- Problem G_1, $x_i = x_i^2\ \forall\ i \in N^2$ and let the others unchanged.
- Problem G_k, $x_i = x_i^1\ \forall\ i \in N^k$ and $x_i = x_i^2\ \forall\ i \in N^{k+1}$, for all $k = 2, \ldots, w-2$.
- Problem G_{w-1}, $x_i = x_i^1\ \forall\ i \in N^{w-1}$ and let the variables belonging to players in N^w unchanged.

Thus, problems $P_1, \ldots, P_{w-1}$ can be expressed as:

(6.35)
$$\begin{array}{lll} \min & \sum_{i \in N^1} x_i + \sum_{i \in N^2} x_i^2 & \\ \text{s.t.:} & x_{i^1} + x^2_{i^2} \geq a^1_{i^1 i^2} & \forall\ (i^1, i^2) \in (N^1 \times N^2) \qquad (P_1) \\ & x_i \geq 0 & \forall\ i \in N^1 \\ & x_i^2 \geq 0 & \forall\ i \in N^2 \\ & & \vdots \\ \min & \sum_{i \in N^k} x_i^1 + \sum_{i \in N^{k+1}} x_i^2 & \\ \text{s.t.:} & x^1_{i^k} + x^2_{i^{k+1}} \geq a^k_{i^k i^{k+1}} & \forall\ (i^k, i^{k+1}) \in (N^k \times N^{k+1}) \qquad (P_k) \\ & x_i^1 \geq 0 & \forall\ i \in N^k \\ & x_i^2 \geq 0 & \forall\ i \in N^{k+1} \\ & & \vdots \\ \min & \sum_{i \in N^{w-1}} x_i^1 + \sum_{i \in N^w} x_i & \\ \text{s.t.:} & x^1_{i^{w-1}} + x_{i^w} \geq a^{w-1}_{i^{w-1} i^w} & \forall\ (i^{w-1}, i^w) \in (N^{w-1} \times N^w) \quad (P_{w-1}) \\ & x_i^1 \geq 0 & \forall\ i \in N^{w-1} \\ & x_i \geq 0 & \forall\ i \in N^w \end{array}$$

The above-mentioned problems can be combined in one problem of the form:

$$
(6.36) \qquad \begin{array}{ll}
\min & \sum_{i^1 \in N^1} x_{i^1} + \cdots + \sum_{i^k \in N^k} x^2_{i^k} + x^1_{i^k} + \cdots + \sum_{i^w \in N^w} x_i \\
\text{s.t.:} & x_{i^1} + x^2_{i^2} \geq a^1_{i^1 i^2} \ \forall \ (i^1, i^2) \in (N^1 \times N^2) \\
& x^1_{i^k} + x^2_{i^{k+1}} \geq a^k_{i^k i^{k+1}} \ \forall \ (i^k, i^{k+1}) \in (N^k \times N^{k+1}) \ \forall \ k = 2, \ldots, w-1 \\
& x^1_{i^{w-1}} + x_{i^w} \geq a^{w-1}_{i^{w-1} i^w} \ \forall \ (i^{w-1}, i^w) \in (N^{w-1} \times N^w) \\
& x_i \geq 0 \ \forall \ i \in N^1, \ x^2_i \geq 0 \ \forall \ i \in N^2 \\
& x^1_i \geq 0 \ \forall \ i \in N^k, \ x^2_i \geq 0 \ \forall \ i \in N^{k+1} \ \forall \ k = 2, \ldots, w-1 \\
& x^1_i \geq 0 \ \forall \ i \in N^{w-1}, \ x_i \geq 0 \ \forall \ i \in N^w
\end{array}
$$

The value of Problem (6.36) is equal to the sum of the values of problems P_k. After a bit of algebra, the set of constraints of Problem (6.36) can be expressed as:

$$
(6.37) \qquad x_{i^1} + x^2_{i^2} + x^1_{i^2} + x^2_{i^3} + \cdots + x^1_{i^{w-1}} + x_{i^w} \geq a^1_{i^1 i^2} + a^2_{i^2 i^3} + \cdots + a^{w-1}_{i^{w-1} i^w}
$$

for all $(i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w$, plus the nonnegativity constraints of the objective variables. Taking into account that

$$
(6.38) \qquad a^1_{i^1 i^2} + a^2_{i^2 i^3} + \cdots + a^{w-1}_{i^{w-1} i^w} = a_{i^1, \ldots, i^w} \ \forall \ (i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w
$$

and making $x_{i^k} = x^1_{i^k} + x^2_{i^k}$, we get that the value of Problem (6.36) is equal to the value of the linear program:

$$
(6.39) \qquad \begin{array}{ll}
\min & \sum_{i \in N} x_i \\
\text{s.t.:} & x_{i^1} + \cdots + x_{i^w} \geq a_{i^1, \ldots, i^w} \ \forall (i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w \ (P) \\
& x_i \geq 0 \ \forall i \in N
\end{array}
$$

For every linear programming problem Q, let us denote its optimal value by $l(Q)$. We have that:

$$
(6.40) \qquad l(P) = \sum_{k=1}^{w-1} l(P_k) \leq \sum_{k=1}^{w-1} v^k(N^k \cup N^{k+1}) = v(N)
$$

Then we conclude that $l(P) \leq v(N)$. This together with Lemma 6.3.1 concludes that the core of (N, v) is non-empty. □

Now a second theorem shows the balancedness of separable MDA games. Besides, the proof of this theorem provides us with a procedure to calculate core allocations.

Theorem 6.3.2 Let (N, v) be a separable w-dimensional assignment game. (N, v) is balanced.

Proof. Let $G_k = (N^k \cup N^{k+1}, v^k), k = 1, \ldots, w-1$ be the corresponding assignment games in which G can be divided, see Lemma 6.3.2. One has that, for each k, the corresponding linear programming problem that gives the value of $v^k(N^k \cup N^{k+1})$ is:

$$
\begin{array}{ll}
\max & \sum_{i^k \in N^k} \sum_{i^{k+1} \in N^{k+1}} a^k_{i^k i^{k+1}} x^k_{i^k i^{k+1}} \\
\text{s.t.:} & \sum_{i^{k+1} \in N^{k+1}} x^k_{i^k i^{k+1}} \leq 1 \ \forall\ i^k \in N^k \\
& \sum_{i^k \in N^k} x^k_{i^k i^{k+1}} \leq 1 \ \forall\ i^{k+1} \in N^{k+1} \\
& x^k_{i^k i^{k+1}} \geq 0 \ \forall\ i^k \in N^k, i^{k+1} \in N^{k+1}
\end{array} \qquad (P_k) \tag{6.41}
$$

Consider the dual of the Program (6.41), which is

$$
\begin{array}{ll}
\min & \sum_{i^k \in N^k} y^1_{i^k} \sum_{i^{k+1} \in N^{k+1}} y^2_{i^{k+1}} \\
\text{s.t.:} & y^1_{i^k} + y^2_{i^{k+1}} \geq a^k_{i^k i^{k+1}} \ \forall\ (i^k, i^{k+1}) \in N^k \times N^{k+1}
\end{array} \qquad (D_k) \tag{6.42}
$$

Generally, there is no nonnegativity restriction on the dual variables. It is not difficult to see that any minimizing vector for (6.42) with all components nonnegative is a core allocation for the game G_k, for all $k = 1, \ldots, w-1$, see [48]. Let $(\overline{y}^1_{i^k}, \overline{y}^2_{i^{k+1}})_{(i^k, i^{k+1}) \in N^k \times N^{k+1}}$ be such a vector. Denote by $l(D_k)$ the value of Problem (6.42). By duality $l(D_k) = v^k(N^k \cup N^{k+1})$.

Now consider the linear programming problem arising after summing up the objectives functions of problems G_k and joining their sets of constraints. Such a problem is:

$$\begin{aligned}
(6.43) \qquad & \min \sum_{i^1 \in N^1} y^1_{i^1} + \sum_{k=2}^{w-1} \sum_{i^k \in N^k} (y^1_{i^k} + y^2_{i^k}) + \sum_{i^w \in N^w} y^2_{i^w} \\
& \text{s.t.:} \; y^1_{i^k} + y^2_{i^{k1}} \geq a^k_{i^k i^{k+1}} \; \forall \, (i^k, i^{k+1}) \in N^k \times N^{k+1} \; \forall \, k = 1, \ldots, w-1
\end{aligned}$$

The value of this problem is equal to $\sum_{k+1}^{w-1} l(D_k)$. One optimal solution is the vector constituted by all optimal solutions to problems D_k for all k, that is, $(\overline{y}^1_{i^k}, \overline{y}^2_{i^{k+1}})_{(i^k, i^{k+1}), k=1,\ldots,w}$.

Taking into account that $\sum_{k=1}^{w-1} a^k_{i^k i^{k+1}} = a_{i^1 \ldots i^w}$, and after a bit of algebra, Problem (6.43) can be transformed into:

$$\begin{aligned}
(6.44) \qquad & \min \sum_{i^1 \in N^1} y^1_{i^1} + \sum_{k=2}^{w-1} \sum_{i^k \in N^k} (y^1_{i^k} + y^2_{i^k}) + \sum_{i^w \in N^w} y^2_{i^w} \\
& \text{s.t.:} \; y^1_{i^1} + \sum_{k=2}^{w-1} (y^1_{i^k} + y^2_{i^k}) + y^2_{i^w} \geq a_{i^1 \ldots i^w} \; \forall \, (i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w
\end{aligned}$$

Consider the following variables:

$$(6.45) \qquad y_i = \begin{cases} y^1_i & \text{if } i \in N^1 \\ y^1_i + y^2_i & \text{if } i \in N^k, \; k = 2, \ldots, w-1 \\ y^2_i & \text{if } i \in N^w \end{cases}$$

Then, Problem (6.44) can be expressed as:

$$\begin{aligned}
(6.46) \qquad & \min \sum_{k=1}^{w} \sum_{i^k \in N^k} y_{i^k} \\
& \text{s.t.:} \; \sum_{k=1}^{w} y_{i^k} \geq a_{i^1 \ldots i^w} \; \forall \, (i^1, \ldots, i^w) \in N^1 \times \cdots \times N^w
\end{aligned}$$

Consider the vector $\overline{y}$ whose coordinates are defined as follows:

$$\begin{aligned} \overline{y}_{i^1} &= \overline{y}^1_{i^1} && \forall\, i^1 \in N^1 \\ \overline{y}_{i^k} &= \overline{y}^1_{i^k} + \overline{y}^2_{i^k} && \forall\, i^k \in N^k,\ k = 2, \ldots, w-1 \\ \overline{y}_{i^w} &= \overline{y}^2_{i^w} && \forall i^w \in N^w \end{aligned} \tag{6.47}$$

From the reasoning above one deduces that the vector $\overline{y}$ is optimal for Problem (6.46). That means that $\sum_{i\in N} \overline{y}_i = v(N)$. Let us see that this vector also defines an allocation in the core.

For any $S \subset N$, we have

$$v(S) = a_{i^1_1 \ldots i^w_1} + \cdots + a_{i^1_q \ldots i^w_q}, \tag{6.48}$$

where $i^1_1 \ldots i^w_1, \ldots, i^1_q \ldots i^w_q$ are distinct members of S. Then

$$\begin{aligned} \sum_{i\in S} \overline{y}_i &\geq \overline{y}_{i^1_1} + \cdots + \overline{y}_{i^w_1} + \cdots + \overline{y}_{i^1_q} + \cdots + \overline{y}_{i^w_q} \\ &\geq a_{i^1_1 \ldots i^w_1} + \cdots + a_{i^1_q \ldots i^w_q} = v(S) \end{aligned} \tag{6.49}$$

and we conclude that $(\overline{y}) \in Core(N, v)$. □

Now we show an example of a separable MDA game.

Example 6.3.2 *Consider a 3-dimensional assignment game where* $N^1 = \{1, 2\}, N^2 = \{3, 4\}, N^3 = \{5, 6\}$. *The benefits of each possible association are shown in the following table:*

(6.50)

(i^1, i^2, i^3)	$\{1,3,5\}$	$\{1,3,6\}$	$\{1,4,5\}$	$\{1,4,6\}$
$a_{i^1 i^2 i^3}$	3	5	4	4

(i^1, i^2, i^3)	$\{2,3,5\}$	$\{2,3,6\}$	$\{2,4,5\}$	$\{2,4,6\}$
$a_{i^1 i^2 i^3}$	4	6	3	3

In Figure 6.1 a picture of this game is shown.

The maximum benefit of the grand coalition is 10 units, and it is reached when the

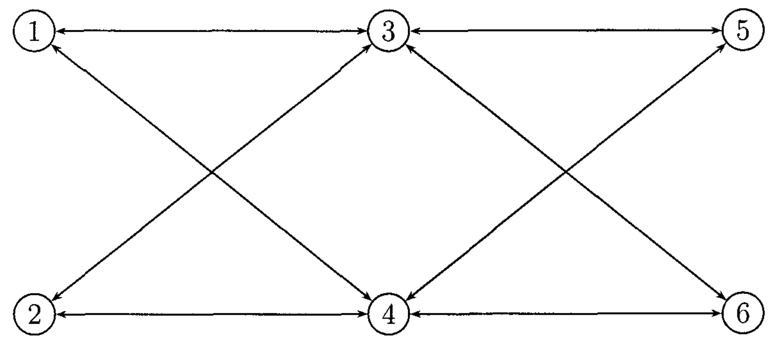

Figure 6.1: Separable 3-dimensional assignment game, G.

associations $(1,4,5)$ *and* $(2,3,6)$ *are made.*

One can see that this 3-dimensional assignment game is separable. The corresponding coefficients $a^k_{i^k i^{k+1}}$ $k=1,2$ *are shown in the following table:*

(6.51)

(i^1,i^2)	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$
$a^1_{i^1 i^2}$	1	3	2	2

(i^2,i^3)	$\{3,5\}$	$\{3,6\}$	$\{4,5\}$	$\{4,6\}$
$a^2_{i^2 i^3}$	2	4	1	1

Figure 6.2 shows the two assignment games in which our game can be divided. Two core allocations for games G_1 *and* G_2 *obtained from their respective dual problems are:*

$$\gamma^1 = (\gamma^1_1, \gamma^1_2, \gamma^1_3, \gamma^1_4) = (3,2,0,0), \ \gamma^2 = (\gamma^2_3, \gamma^2_4, \gamma^2_5, \gamma^2_6) = (4,1,0,0). \tag{6.52}$$

Making $\gamma_i = \gamma^1_i$ $i=1,2,$ $\gamma_i = \gamma^1_i + \gamma^2_i$ $i=3,4$ *and* $\gamma_i = \gamma^2_i$ $i=5,6$ *(the same reasoning as in the proof of Theorem 6.3.2), those two allocations result in the following allocation for the original 3-dimensional assignment game:*

$$\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_6) = (3,2,4,1,0,0). \tag{6.53}$$

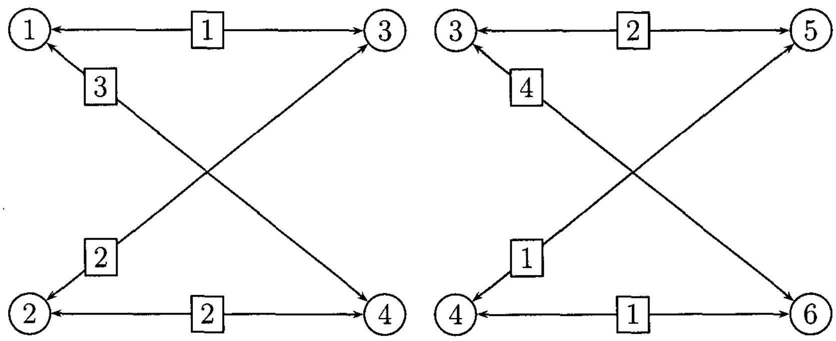

Figure 6.2: 2-dimensional assignment games, G_1 and G_2.

It can be checked that, indeed, $x \in Core(G)$.

One could be tempted to think that all balanced MDA games are separable. This assertion is not true. As an example take the game in the example above with only one change, make $a_{135} = 2$. It is easy to prove that this is not a separable game and, on the other hand, the allocation $(4, 3, 3, 0, 0, 0)$ is in the core of the game.

6.4 Approximation algorithms

Due to the fact that in general MDA games are not balanced, the search for other allocations with good properties such as the Shapley Value, see [62], becomes more important than in games where we can efficiently find core allocations. But in games arising from LP problems that cannot be optimally solved in real time, just like MDA problems, the calculation of the Shapley value becomes extremely hard. Note that to compute such a value we need to know the characteristic function, that is, we have to solve Problem (6.9) for every coalition, in other words, we have to solve $O(2^n)$ MDA problems (each of them is **NP**-hard.) This is enough argumentation to look for other techniques to allocate the benefits generated after an MDA game among the players. To avoid the **NP**-hardness of the underlying LP problems, we will make use of approximation algorithms. For a detailed description on approximation algorithms see [70].

In the following sections we propose a mechanism to allocate benefits, that is, we give a vector $(x_1, \ldots, x_n)$ such that x_i is the payoff of player i, $\forall\ i \in N$.

Given a MDA problem, we define its set of associations as the set

$$S = \{s : s = \{i^1, \ldots, i^w\}, i^k \in N^k\ \forall\ k = 1, \ldots, w\}. \tag{6.54}$$

Let us define the benefit of association $s \in S$, $a(s)$, to be the benefit generated after the association of the players in s, that is, $a(s) = a_{i^1,\ldots,i^w}$. So, the MDA Problem (6.2) can be formulated as:

$$\begin{aligned} \max\ & \sum_{s\in S} y_s a(s) \\ \text{s.t.:}\ & \sum_{s\ :\ i\in s} y_s \leq 1 \quad \forall\ i \in N \\ & y_s \in \{0,1\} \quad\ \ \forall\ s \in S \end{aligned} \tag{6.55}$$

6.4.1 The greedy allocation

The greedy strategy naturally applies for allocating benefits in multidimensional assignment games. The procedure runs as follows. Consider (N, v) a MDA game. Iteratively select the most profitable association and remove the members of such association. Repeat this process until all players have been assigned or there are no more possible associations.When an association $s \in S$ is chosen, its benefit is distributed equally among the players that constitute s. Thus, we set $x_i = a(s)/w\ \forall\ i \in s$ for every association s selected in the greedy algorithm. A pseudocode of this allocation procedure is:

Greedy allocation.

1. $x = 0, C = N$
2. Repeat.
   Find the most profitable association in C, say s.
   Let $a(s)$ be the benefit of association s.
   For each $i \in s$ set $x_i = a(s)/w$.
   $C \leftarrow C \setminus s$.
   Until no more associations are found.
3. Output x.

For a fixed w-dimensional assignment problem, the value of the solution computed by the process above approximates the optimal solution within a factor of w, that is,

$$\text{V(OPT)} \leq w\text{V(GREEDY)}, \tag{6.56}$$

where V(OPT) denotes the value of the optimal solution and V(GREEDY) denotes the value of the solution returned by the greedy algorithm. The proof of this result can be found in [9]. Besides, in [53] it is shown that this approximation factor cannot be improved by a better analysis of the game. Let us see the example that proves it.

Example 6.4.1 *Consider a d-dimensional assignment problem in which the only associations that produce a positive profit are shown in the following table:*

Association	*Benefit*
(1, 2, ... ,d)	*1+ ε*
(1, d+2, ... ,2d)	*1*
(2d + 1, 2, ... ,3d)	*1*
...	...
(dd + 1, dd + 2, ... , dd + d)	*1*

Denote $W(G)$ and $W(OPT)$ the values of the solutions generated by the Greedy algorithm and the optimal one respectively. It is easy to check that $W(G) = 1+\varepsilon$, $W(OPT) = d$. Then, one has that:

$$W(OPT) = \frac{d}{1+\varepsilon} W(G). \tag{6.57}$$

Letting $\varepsilon \longrightarrow 0$ concludes that the given upper bound cannot be improved.

The complexity of the greedy algorithm for MDA problems has been proven to be $O(n \log n)$, see [9]. After those arguments, the following result follows.

Theorem 6.4.1 Let $x = (x_1, \dots, x_n)$ be the greedy allocation for a MDA game (N, v) arising from a w-dimensional assignment problem. The following assertions hold:

1. x is a preimputation of (N, v).
2. x satisfies the individual rationality principle.
3. $\frac{v(N)}{w} \leq x(N) \leq v(N)$.
4. x can be computed in polynomial time.

Example 6.4.2 *Suppose that we have a data set corresponding with the possible associations between three factories and three shops (for the sake of brevity we consider a 2-dimensional assignment problem). The three factories will be players $1, 2$ and 3 respectively, and the three shops will be players $4, 5$ and 6. We assume that the benefits generated by all the possible associations are those shown in Table 6.2. The first coordinate of the vectors represents the factory and the second one represents the shop.*

Association	(1,4)	(1,5)	(1,6)	(2,4)	(2,5)	(2,6)	(3,4)	(3,5)	(3,6)
Benefit	17	16	2	6	1	2	2	3	4

Table 6.2: Possible associations and their benefits.

The greedy allocation would divide the benefits as follows:

- *It firstly takes the association that gives the highest profit. This association is* $(1, 4)$ *with a value of 17 units. Then we set* $x_1 = x_4 = \frac{17}{2}$.

- *Secondly, it takes the most profitable association that does not include neither player 1 nor player 4. This association is* $(3, 6)$, *which generates 4 units. Then we set* $x_3 = x_6 = 2$.

- *To finish, the association* $(2, 5)$ *is made. Then the benefits of players 2 and 5 are* $x_2 = x_5 = \frac{1}{2}$.

Note that this mechanism has not allocated the maximum profit that could be generated by the grand coalition, as it is based on an approximation algorithm.

6.4.2 A generalization of the greedy allocation

In this section we present another way of allocating benefits in MDA games. It is based on the polynomial time heuristic algorithm $K - SGTS$ as presented in [52]. We need a previous definition.

Definition 6.4.1 *Let* (N, v) *be a MDA game. We say that a group of* m *associations* $\{s_1, \ldots, s_m\}$ *is feasible if no player* i *belongs to more than one of them.*

The approach to allocate the overall benefit will be as follows. We iteratively pick the feasible group of $K \in \mathbb{N}$ associations $\{s_1^*, \ldots, s_K^*\}$ that maximizes the sum of benefits. Whenever a set of K associations is picked, we set the payoff of every player included in one of the associations as the proportional part of the allocation to which it belongs. Then we remove $\{s_1^*, \ldots, s_K^*\}$ from the set A as well as every association that contains one of the players in the set $\{s_1^*, \ldots, s_K^*\}$. We repeat the process until one of the following events happens:

1. S is the empty set

2. we cannot find a feasible group of K associations and A is not the empty set.

In the latter case, we reduce the size of the groups to $K - 1$ and repeat the process until S is the empty set.

The procedure given above has a lot of similarities with the classical greedy algorithm. The main difference is that in this case we do not pick the best association but the group of K best pairwise disjoint associations. The greedy algorithm guarantees an approximation factor of $O(\log n)$, see [70]. Although so far we have not succeeded in finding an approximation factor for this algorithm, experimental results given in [52] suggest that the quality of the solution given for the algorithm when $K > 1$ is better than the algorithm in the case $K = 1$, that is, than the classical greedy approach.

Example 6.4.3 *Consider the data in Example 6.4.2. The semi-greedy procedure described in this section, with $K = 2$, would proceed as follows:*

1. *It first takes the pair of associations that maximizes the sum of benefits and does not have any player in common (one factory cannot be in both associations, nor can one shop). It can checked that such a pair is $\{(1,5),(2,4)\}$. So, we set $x_1 = x_5 = \frac{16}{2} = 8,\ x_2 = x_4 = \frac{6}{2} = 3$.*

2. *After having performed step 1, there are no pairs of associations having empty intersection with the two associations previously chosen. So we reduce to $K - 1 = 1$. There is only one association eligible to be part of the same solution as $\{(1,5),(2,4)\}$. It is $(3,6)$ with a value of 4. Then we set $x_3 = \frac{4}{2} = 2,\ x_6 = \frac{4}{2} = 2$.*

One can check that this procedure allocates more benefit among the players than the procedure based on the classical greedy algorithm. Nevertheless, this cannot be guaranteed in general, that is, there are examples of MDA games in which the SGTS solution gives a higher value than the 2-SGTS value.

Analogously as in Theorem 6.4.1, we can state the following result for the K-greedy allocation as described in this section.

Theorem 6.4.2 Let $K \in \mathbb{N},\ K > 1$ and let $x = (x_1, \ldots, x_n)$ be the K-greedy allocation for a MDA game (N, v) arising from a w-dimensional assignment problem. The following assertions hold:

1. x is a preimputation of (N, v).

2. x satisfies the individual rationality principle.

3. x can be computed in polynomial time.

In this chapter a new class of cooperative games extending the classical assignment games has been presented. Such class of games, named MDA games, proved totally balanced. Since the integer linear programming problems that give rise to this class of games are mathematically termed NP-hard, the use of approximation algorithms is needed. A class of approximation algorithms, named K-SGTS, had successfully been used to solve MDA problems and, based on them, two allocations have been introduced for this class of games which can be computed in polynomial time.

Index

Bibliography

[1] Ahuja, R.K., Orlin, J.B. and Tarjan, R.E. *Improved time bounds for the maximum flow problem.* SIAM Journal of Computing 18:939-954 (1989).

[2] Anupindi, R., Bassok, Y. and Zemel, E. *A general framework for the study of decentralized distribution systems* Manufacturing & Service Operations Management 3:349-368 (2001).

[3] Bar-Shalom, Y. and Xiao-Rong Li, "Multitarget-Multisensor Tracking: Principles and Techniques" Storrs, CT: YBS Publishing (1995).

[4] Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D. "Linear programming and network flows" John Wiley & Sons (1990).

[5] Bird C.G. *On Cost Allocation for a Spanning Tree: A Game Theoretic Approach.* Networks 6:335-350 (1976).

[6] Bondareva, O. *Certain Applications of the methods of linear programming to the theory of cooperative games.* (In Russian). Problemy Kibernetiky, 10:119-139 (1963).

[7] Borm, P.E.M., Hamers, H. and Hendrickx, R. *Operations Research Games: A Survey.* TOP 9:139-216 (2001).

[8] Brewer, A.M., Button, K.J., Hensher, D.A. "Handbook of Logistics and Supply Chain Management" Elsevier Science Ltd (2001).

[9] Capponi, A. *Polinomial Time Algorithm for Data Association Problem in Multitarget Tracking.* IEEE Transactions on Aerospace and Electronic Systems, 40:1398-1410 (2004).

[10] Cheriyan, J. and Maheshwari, S.N. *Analysis of preflow push algorithms for maximum network flow*, SIAM Journal on Computing 18:1057-1086 (1989).

[11] Corley, H.W. *Efficient spanning trees.* Journal of Optimization Theory and Applications 45:481-485 (1985).

[12] Curiel, I. "Cooperative Game Theory and applications: Cooperative games arising from combinatorial optimization problems" Kluwer Academic Publishers (1997).

[13] Dijkstra, E. W. *A note on Two Problems in Connection with Graphs.* Numerische Mathematik 1:269-271 (1959).

[14] Ehrgott, M. "Multicriteria Optimization. Lectures Notes in Economic and Mathematical Systems 491" Springer (2000).

[15] Eppen, G.D. *Effect of Centralization on Expected Cost in a Multi-location Newsboy Problem*, Management Science 25:498-501 (1979).

[16] Fernández, F.R., Hinojosa, M.A. and Puerto, J. *Multi-criteria Minimum Cost Spanning Tree Games.* European Journal of Operations Research 158:399-408 (2004).

[17] Fernández, F.R., Hinojosa, M.A. and Puerto, J. *Set valued TU Games.* European Journal of Operations Research 159:181-195 (2004).

[18] Fernández, F.R. and Puerto, J. "Teoría de Juegos Multiobjetivo". ISBN: 84-689-6702-5. Depósito Legal MA-102-2006. Málaga (2006) (In Spanish).

[19] Faigle, U. and Kern, W. *On the core of ordered submodular cost games.* Mathematical Programming 87: 483-499 (2000).

[20] Figueira, J., Greco, S. and Ehrgott, M. "Multiple Criteria Decision Analysis: State of the Art Surveys" Springer's International Series (2005).

[21] Ford, L.R. and Fulkerson, D.R. "Flows in Networks" Princeton University Press (1962).

[22] Forgó, F., Szép, J. and Szidarovsky, F. "Introduction to the Theory of Games" Kluwer Academic Publishers (1999).

[23] Fragnelli, V., García-Jurado, I. and Méndez-Naya, L. *On Shortest Path Games.* Mathematical Methods of Operations Research 52:251-264 (2000).

[24] van Gellekom, J.R.G., Potters, J.A.M., Reijnierse, J.H., Engel, M.C. and Tijs, S.H *Characterization of the Owen Set of Linear Production Processes.* Games and Economic Behavior 32:139-156 (2000).

[25] Gerchak, Y. and Gupta, D. *On Apportioning Costs to Customers in Centralized Continuous Review Inventory Systems.* Journal of Operations Management 10:546-551 (1991).

[26] Goldberg, A.V., *A New Max-Flow Algorithm*, Technical Report MIT/LCS/TM-291, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts (1985).

[27] Goldberg, A.V. and Tarjan, R.E., *A new approach to the maximum flow problem*, Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (1986).

[28] Goldberg, A.V. and Tarjan, R.E., *A new approach to the maximum-flow problem*, Journal of the Association for Computing Machinery 35:921-940 (1988).

[29] Hartman, B.C. and Dror, M. *Cost allocation in continuous review inventory models*, Naval Research Logistics 43:549-561 (1996).

[30] Hartman, B.C. and Dror, M. *Optimizing centralized inventory operations in a cooperative game theory setting* IIE Transactions on Operations Engineering 35:243-257 (2003).

[31] Hartman, B.C. and Dror, M. *Allocation of gains from inventory centralization in newsvendor environments* IIE Transactions on Scheduling and Logistics 37:93-107 (2005).

[32] Hartman, B.C., Dror, M. and Shaked, M. *Cores of inventory centralization games* Games and Economic Behavior 31:26-49 (2000).

[33] Isermann, H. *On some relations between a dual pair of multiple objective linear programs.* Operations Research 22:33-41 (1978).

[34] Kalai, E. and Zemel, E. *Totally Balanced Games and Games of Flow.* Mathematics of Operations Research 7:476-478 (1982).

[35] J. B. Kruskal. *On the shortest spanning subtree and the traveling salesman problem.* Proceedings of the American Mathematical Society 7:48-50 (1956).

[36] Markakis, E., Amin, S. *On the core of the multicommodity flow game.* Decision Support Systems 39:3-10 (2005).

[37] Meca, A., Timmer, J., García-Jurado, I. and Borm, P.E.M. *Inventory Games.* European Journal of Operational Research 156:127-139 (2004).

[38] Meca, A., García-Jurado, I. and Borm, P.E.M. *Cooperation and competition in Inventory Games.* Mathematics Methods of Operation Research 57:481-493 (2003).

[39] Megiddo, N. *Computational complexity of the game theory approach to cost allocation for a tree.* Math. Oper. Res. 3:189-196 (1978).

[40] Minner, S. *Bargaining for cooperative economic ordering.* Decision Support Systems, in press, available online (2005).

[41] Murty, K.G. "Linear and combinatorial programming" John Wiley and Sons (1976).

[42] Myerson, R. *Graphs and cooperation in games.* Mathematics of Operation Research 2:225-229 (1977).

[43] Myerson, R. *Conference structures and fair allocation rules.* International Journal of Game Theory 9:169-182 (1980).

[44] Müller, A., Scarsini, M. and Shaked, M. *The Newsvendor Game Has a Nonempty Core* Games and Economic Behavior 38:118-126 (2002).

[45] Neumann, J. and Morgenstern, O. "Theory of Games and Economic Behaviour" University Press, Princeton, New Jersey (1944).

[46] Ogryczak, W. and Tamir, A. *Minimizing the sum of the k largest functions in linear time.* Information Processing Letters, 85.3:117-122 (2003).

[47] Owen, G. *On the Core of Linear Production Games.* Mathematical Programming 9:358-370 (1975).

[48] Owen, G. "Game Theory" Academic Press, San Diego (1995).

[49] Peleg, B and Tijs, S. "Handbook of Game Theory with Economic Applications, Volume I" Handbooks in Economics. 11. Amsterdam (1992).

[50] Pattipati, K.R., Bar-Shalom, Y. and Washburn, R.B. *Passive multisensor data association using a new relaxation algorithm.* Proc. American Control Conf, Pittsburg, 3:2617-2624 (1989).

[51] Pattipati, K.R., Bar-Shalom, Y. and Washburn, R.B. *Passive multisensor data association using a new relaxation algorithm.* Multitarget-Multisensor Tracking: Advanced Applications 1:187-218 (1990).

[52] Perea, F. and de Waard, H.W. *A new class of Heuristic Polinomial Time algorithms for solving the Multidimensional Assignment Problem.* Proceedings of Information Fusion 2006, Florence, Italy (2006).

[53] Perea, F. *A survey on Greedy-based Algorithms for the Multidimensional Assignment Problem.* Euro Summer Institute, Wittembert Lutherstad, Germany (2006).

[54] W. Pierskalla. *The multidimensional assignment problem.* Operations Research 16:422-431 (1968).

[55] Poore, A.B. *Multidimensional Assignment Formulation of Data Association Problems Arising from Multitarget and Multisensor Tracking.* Computational Optimization and Applications, 3:27-57 (1994).

[56] Poore, A.B., Rijavec, N. and Barker, T.N. *Data association for track initiation and extension using multiscan windows.* Signal and Data Processing of Small Targets, SPIE. 1698:432-441 (1992).

[57] Prim., R.C. *Shortest connection networks and some generalisations.* Bell System Technical Journal, 36:1389-1401 (1957).

[58] Puerto, J., García-Jurado, I. and Fernández, F.R. *On the core of a class of location games.* Mathematical Methods of Operations Research 54:373-385 (2001).

[59] Sánchez, J. *El problema del transporte. Una aproximación desde la Teoría de Juegos.* PhD Thesis. Murcia (1998).

[60] Sánchez-Soriano, J., López, Marco A. and García-Jurado, I. *On the core of transportation games* Mathematical Social Sciences 41:215-225 (2001).

[61] Schmeidler, D. *The nucleolus of a characteristic function game*, SIAM J. Appl. Math. 17:1163-1170 (1969).

[62] Shapley, L.S. *A value for n-person games,* Kuhn, H.W. and Tucker, A.W., editors, Contributions to the Theory of Games II, Princeton University Press, Princeton, 307-317 (1953).

[63] Shapley, L. *On balanced sets and cores.* Naval Research Logistics Quarterly 14:453-460 (1967).

[64] Slikker, M., Fransoo, J. and Wouters, M. *Cooperation between multiple news-vendors with transshipments.* European Journal of Operational Research 167:370-380 (2006).

[65] Steuer, R.E. "Multiple Criteria Optimization: Theory, Computation and Application" John Wiley, New York (1986).

[66] Syslo, M.M. "Discrete Optimization Algorithms" Prentice-Hall (1983).

[67] Tamir, A. *A unifying location model on tree graphs based on submodularity properties.* Discrete Applied Mathematics 47:275-283 (1993).

[68] Tijs, S.H., Meca, A. and López, M.A. *Benefit sharing in holding situations.* European Journal of Operational Research 162:251-269 (2005).

[69] Tunçel, L. *On the complexity of preflow-push algorithms for the maximum-flow problem,* Algorithmica 11:353-359 (1994)

[70] Vazirani, V.V. "Approximation Algorithms" Springer-Verlag (2001).

[71] Wilson, R.J. and Watkins, J.J. "GRAPHS: An Introductory Approach" Library of Congress Cataloging-in-Publication Data, Singapore (1990).

[72] Vanderbei, R.J. "Linear programming: foundations and extensions" Kluwer´s International Series (1996).

[73] Voorneveld, M. and Grahn, S. *Cost allocation in shortest path games.* Mathematical Methods of Operations Research 56:323-340 (2002).

[74] Wosley, L.A. "Integer Programming" John Wiley & Sons (1988).

[75] Özen, U., Norde, H. and Slikker, M. *On the convexity of newsvendor games.* Center Discussion Paper (Tilburg University) (2005).

UNIVERSIDAD DE SEVILLA

Reunido el tribunal en el día de la fecha, integrado por los abajo firmantes, para evaluar la tesis doctoral de D. Federico Perea Rojas - Marcos
tiulada Analisis cooperativo de Cadenas de Distribución
acordó otorgarle la calificación de SOBRESALIENTE CUM LAUDE

Sevilla, a 29 de Mayo de 2007

Vocal,

Vocal,

Vocal,

Presidente,

Secretario,

Doctorando,