Université de Montréal

Mixed-Integer Programming Representation for Symmetrical Partition Function Form Games

par

Justine Pepin

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

Orientation recherche opérationnelle

22 novembre 2022

© Justine Pepin, 2022

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Mixed-Integer Programming Representation for Symmetrical Partition Function Form Games

présenté par

Justine Pepin

a été évalué par un jury composé des personnes suivantes :

Emma Frejinger

(présidente-rapporteuse)

Margarida Carvalho

(directrice de recherche)

Sriram Sankaranarayanan

(codirecteur)

Andrea Lodi

(membre du jury)

Résumé

Dans tout contexte impliquant plusieurs agents (joueurs), il est impératif de déterminer comment les agents coopéreront par la formation de coalitions et comment ils partageront les bénéfices supplémentaires issus de la collaboration. Ceci peut fournir une aide à la décision aux joueurs, ou encore des outils d'analyse pour les responsables en charge de réguler les marchés économiques. De telles situations relèvent de la théorie des jeux coopérative. Un élément crucial de ce domaine est la taille de la représentation de ces jeux : pour chaque partition de joueurs possible, la valeur de chaque coalition qu'on y retrouve doit être donnée.

Les jeux symétriques à fonction de partition (SPFG) appartiennent à une classe de jeux coopératifs possédant deux caractéristiques principales. Premièrement, ils sont sensibles aux externalités, provoquées par n'importe quel groupe de joueurs qui s'allient ou défont leurs alliances, qui sont ressenties par les autres coalitions de joueurs. Deuxièmement, ils considèrent que les joueurs sont indistincts, et donc que seul le nombre de joueurs dans chaque coalition est à retenir pour représenter un SPFG. Par l'utilisation d'outils de programmation mixte en nombres entiers, nous présentons la première représentation de SPFG qui est polynomiale en nombre de joueurs dans le jeu. De surcroît, nous caractérisons la famille des SPFG qu'il est possible de représenter, qui inclut notamment tous les SPFG de cinq joueurs ou moins. De plus, elle dispose d'une approximation compacte pour le cas où, dans un jeu à six joueurs ou plus, le SPFG ne peut pas être représenté de façon exacte. Également, nous introduisons un cadre flexible qui utilise des méthodes visant la stabilité inspirées par la littérature pour identifier, à l'aide de notre représentation, une issue stable qui maximise le bien-être social des joueurs. Nous démontrons la valeur de notre représentation (approximée) compacte et de notre approche pour sélectionner une partition stable et une allocation des profits dans une application de marché compétitif provenant de la littérature.

Mots clés: Théorie des jeux, Théorie des jeux coopératifs, Jeux à fonction de partition, Programmation mixte en nombres entiers.

Abstract

In contexts involving multiple agents (players), determining how they can cooperate through the formation of coalitions and how they can share surplus benefits coming from the collaboration is crucial. This can provide decision-aid to players and analysis tools for policy makers regulating economic markets. Such settings belong to the field of cooperative game theory. A critical element in this area has been the size of the representation of these games: for each possible partition of players, the value of each coalition on it must be provided.

Symmetric partition function form games (SPFGs) belong to a class of cooperative games with two important characteristics. First, they account for externalities provoked by any group of players joining forces or splitting into subsets on the remaining coalitions of players. Second, they consider that players are indistinct, meaning that only the number of players in each coalition is relevant for the SPFG. Using mixed-integer programming, we present the first representation of SPFGs that is polynomial on the number of players in the game. We also characterize the family of SPFGs that we can represent. In particular, the representation is able to encode exactly all SPFGs with five players or less. Furthermore, we provide a compact representation approximating SPFGs when there are six players or more and the SPFG cannot be represented exactly. We also introduce a flexible framework that uses stability methods inspired from the literature to identify a stable social-welfare maximizing game outcome using our representation. We showcase the value of our compact (approximated) representation and approach to determine a stable partition and payoff allocation to a competitive market from the literature.

Keywords: Game theory, Cooperative game theory, Partition function games, Mixed-integer programming.

Contents

List of tables

List of figures

Notation and abbreviations

CFG	Characteristic Function Form Game
CSG	Coalition Structure Generation
MAS	Multi-Agent System
MILP	Mixed-Integer Linear Program
MIP	Mixed-Integer Programming
PDT	Partition Decision Tree
PFG	Partition Function Form Game
SPFG	Symmetric Partition Function Form Game
SW	Social Welfare

Acknowledgements

First of all, I would like to thank my advisors, Margarida and Sriram. Without their care, insights and constant involvement, I don't think I would have been able to accomplish the work you will find in this thesis. I am so, so grateful for the rigor they instilled in me (or tried to, at least!), their generous advice, their communicative love of maths and game theory, their very creative minds full of suggestions that were master-saving in situations of stalling, and their comprehensiveness when I was losing a week of work or two to the problem-solving competitions I always get caught in. You have complementary qualities that make for an awesome duo of advisors, and I know the next person after me will be very lucky, just like I was.

Also, I am sending a warm thought in the direction of my lab mates that were willing to embark on the competition train with me, just for the sake of doing a project together: Carl, Caroline, Federico, Feng, Flore, Warley and William, please keep being as funny, intelligent, hard-working and supporting as you are. Planet Earth is more beautiful (and optimized) because of you!

Finally, I would like to thank IVADO for their financial support through the granting of a MSc Excellence Scholarship.

Introduction

Context. Countries must come together to find effective solutions to global warming. Small independent craftspeople and artists must ally to organize fares to attract customers and achieve more sales. From local to international and from small to big scales, there will always be a category of problems where agents, in a system, want to optimize their own individual payoffs and have the option of collaborating to do so. However, in many cases, these systems are incredibly complex, as there can be various types of agents with various palettes of strategies and end goals without necessarily having the same ideal solution. But above these layers of complexity, we could also add a very natural consequence of being part of a system that is just like a web: the actions of an agent can affect those of the other agents.

Game theory has revealed to be very useful to formalize, solve and analyze such problems. Yet, standard methods either focus on *(i)* the computation of equilibria (individual strategies) to non-cooperative games or *(ii)* the determination of agreements to cooperative games where the side effects of the agents' actions are ignored (e.g., characteristic function form games (CFGs)). In Nagarajan and Sošić [55], they describe the former method as a micro approach, as it concentrates on the details of "what happens" instead of looking at the possible outcomes and studying what stable alliances and payoff splittings would lead there. Without taking agreements between players into consideration, the strategies might be more conservative and lead to lower player benefits due to mistrust and unpredictability. The latter method considering cooperation is indeed pertinent since it captures relevant applications, such as optimal medical team formation and assignation or airline crew scheduling. However, it fails to capture the nuances of externality-driven problems, like the exploitation of international seas or shared audits for firms sharing suppliers, which are common in our contemporary setting.

Partition function form games (PFGs), first defined by Thrall and Lucas [78], are cooperative games where externalities can be considered. In other words, in PFGs, the set of alliances formed (coalitions), i.e., the partition of the players, is important in determining the *value* of each alliance. PFGs generalize CFGs and present interesting applications. Notably, Basso et al. [4] look into the logistic of collaborative transportation of merchandise by firms, with the addition of a case study of the Swedish forestry industry, while Nagarajan and Sošić [54] study the dynamic formation of alliances among symmetric agents that are competing in the same market. In both cases, the possibility of cooperating and externalities are crucial elements on their analysis and results.

Inspired by these works and the potential of the cooperative game theory field to promote and improve collaborations, we decided to study symmetric PFGs (SPFGs), that are PFGs with indistinct (symmetric) agents. As it is often done, we also consider transferable utility, i.e., for any alliance, players on it are able to distribute the alliance's value among themselves, and they can make side payments as well to outsider players, because what they get from the game is not indivisible. In order to solve SPFGs, we first use mixed-integer programming (MIP) to model it. Mixed-integer programming is a field that is devoted to the modeling of decision-making problems and the development of solution methods for combinatorial optimization problems. This is very much in line with solving an SPFG since it involves determining a partition of the players (coalition formation) and utility sharing. Once a problem is modeled with the MIP paradigm, there are many available, powerful and easy to use off-the-shelf MIP solvers, such as Gurobi [33] or CPLEX [22], that one can use to solve the problem. As we will see in our literature review, this fusion of game theory and MIP tools is not uncommon and gives good results.

Motivations. In settings that belong to the field of cooperative game theory, where multiple agents (players) can cooperate through the formation of coalitions, determining the partition that will result as well as how the agents will share the additional gains generated by the collaboration is critical. Moreover, since those settings necessitate, for each possible partition of agents, the value of each coalition it contains, the size and practicability of the representation of these games are at stake. In that respect, a good representation can help to provide decision-aid to players and analysis tools for policy makers regulating economic markets.

Contributions. In this thesis, we provide the following contributions:

(1) The first compact mixed-integer linear program (MILP)-representation for SPFGs;
(2) A flexible framework exploiting the MILP-representation to find a stable partition and a stable payoff allocation vector;
(3) A characterization of the family of SPFGs that we can represent exactly with the new compact representation;
(4) A compact approximated MILP-representation for any SPFG;
(5) Approximation bounds relating the MILP-representation approximation with the stability on the original game of the outcome proposed by our framework;
(6) A computational study validating the value of our framework when we apply it to solve an SPFG from the literature.

Thesis organization. In Chapter 1, we review the main theoretical elements that will be used throughout the thesis. In Chapter 2, we survey the literature linked to coalition formation and payoff sharing. Chapter 3 presents the solution concepts used for solving SPFGs. Chapter 4 contains the details on the MIP formulation that we introduce, combined with the solution concepts from Chapter 3, into a flexible framework to solve SPFGs. It contains as well a description of the SPFG family that the formulation can represent exactly and a procedure to approximate SPFG with our representation. Finally, Chapter 5 tests the framework on a symmetric agent market equivalent to the deterministic game presented in Nagarajan and Sošić [54]. Finally, the thesis ends with conclusions and future research directions.

Chapter 1

Background

In this chapter, we will review mathematical definitions that will serve us in the present document. The subjects revolve around Partition Function Form Games (PFG), our main interest, and their mathematical surroundings. In Section 1.1, we will establish a few conventions to enhance comprehension, and subsequent sections will each detail a concept and exemplify it.

If, after consulting this chapter, the reader would like to know more on the matter of PFG, we suggest the related Kóczy textbook [42].

1.1. Conventions

In this thesis, we will follow some conventions for our notation:

- *Sets* will be written in uppercase calligraphic style, such as $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$.
- The *cardinality* of a set $\mathcal{A}$ is denoted $|\mathcal{A}|$.
- *Matrices* will be written in uppercase bold regular style, such as $\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$.
- We will indicate the *dimensions of a matrix* of $n \times m$ real entries as $\mathbf{A} \in \mathbb{R}^{n \times m}$.
- The *entry of a matrix* $\mathbf{A}$ at row i and column j will be written $\mathbf{A}_{ij}$.
- The *i-th row* of a matrix will be written $\mathbf{A}_i$.
- The *j-th column* of a matrix will be written $\mathbf{A}_{*j}$.
- The *rank* of a matrix $\mathbf{A}$ is denoted $\operatorname{rank}(\mathbf{A})$.
- *Vectors* will be written in lowercase normal style and surmounted by an arrow, such as $\vec{a}, \vec{b}, \vec{c}$.
- We will indicate the *dimension of a vector* of n real components as $\vec{a} \in \mathbb{R}^n$.
- The *i-th component of a vector* $\vec{a}$ will be written $\vec{a}_i$. Note that this is an abuse of notation since $\vec{a}_i$ is not a vector; nevertheless, we represent it in this way to make it clear that it is a component of a vector.
- The *set of all subsets* of a set $\mathcal{A}$ will be denoted $2^{\mathcal{A}}$.

1.2. Partitions and related concepts

Now we will define partitions, a concept central to the thesis. A partition is always defined in relation to a set, because it represent a distribution of the elements of the set into subsets.

Definition 1.2.1 (Partition)**.** *Given a set $\mathcal{D}$ of n elements, a partition of $\mathcal{D}$ is a set $\mathcal{P}$ of non-empty, disjointed elements whose union form the set $\mathcal{D}$. We have*

$$(\forall \mathcal{A} \in \mathcal{P}, \mathcal{A} \neq \emptyset) \bigwedge (\forall \mathcal{A}, \mathcal{B} \in \mathcal{P}, \mathcal{A} \neq \mathcal{B} \implies \mathcal{A} \cap \mathcal{B} = \emptyset) \bigwedge \left(\bigcup_{\mathcal{A} \in \mathcal{P}} \mathcal{A} = \mathcal{D} \right).$$

We denote by $\Pi(\mathcal{D})$ the set of all partitions that can be generated given the set $\mathcal{D}$.

A partition $\mathcal{P} \in \Pi(\mathcal{D})$ has its cardinality limited by the cardinality of $\mathcal{D}$ such that $1 \leq |\mathcal{P}| \leq |\mathcal{D}|$.

Example 1.2.1 (Partition $\mathcal{P}$ of set $\mathcal{D}$)**.** Let us consider $\mathcal{D} = \{$ 🟧, 🟦, 🟨, 🟩 $\}$, a set of dinosaurs. If we write down all the partitions $\mathcal{P} \in \Pi(\mathcal{D})$, we will get the set

$\Pi(\mathcal{D}) = \Big\{$

{{🟧, 🟦, 🟨, 🟩}},

{{🟧, 🟦, 🟨}, {🟩}},

{{🟧, 🟦, 🟩}, {🟨}},

{{🟧, 🟨, 🟩}, {🟦}},

{{🟧}, {🟦, 🟨, 🟩}},

{{🟧, 🟦}, {🟨, 🟩}},

{{🟧, 🟨}, {🟦, 🟩}},

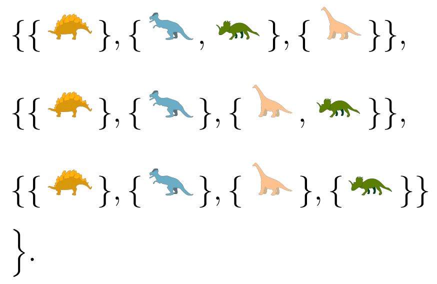
{{🟧, 🟩}, {🟦, 🟨}},

{{🟧, 🟦}, {🟨}, {🟩}},

{{🟧, 🟨}, {🟦}, {🟩}},

{{🟧, 🟩}, {🟦}, {🟨}},

{{🟧}, {🟦, 🟨}, {🟩}},



In this work, since we are in the context of game theory, we will treat partitions of players labeled by a number from 1 to n. Consequently, the set of elements we will manipulate is $\mathcal{N} \coloneqq \{1, 2, \dots, i, \dots, n\}$, of which each element is known respectively as player 1, player 2, ..., player i, ..., player n.

If, beside their labels, players are considered interchangeable, it could be advantageous to try to reduce the amount of information needed to represent them. In such occasions, Kóczy [42] uses numerical partitions.

A numerical partition is a synthesized way of representing a partition of a set of indistinct elements. It substitutes the elements of the ordinary partition by their cardinalities without the loss of any information.

Definition 1.2.2 (Numerical Partition)**.** *Given a set $\mathcal{I}$ of n indistinct elements, a numerical partition $\mathcal{P}^{\text{num}}$ of $\mathcal{I}$ is a set of strictly positive integers whose sum equals n.*

$$\mathcal{P}^{\text{num}} \in \Pi^{\text{num}}(\mathcal{I}) \coloneqq \{c_1, c_2, \dots, c_j, \dots, c_k\} : c_j = |\mathcal{P}_j| \quad \forall \mathcal{P}_j \in \mathcal{P} \in \Pi(\mathcal{I}).$$

We denote by $\Pi^{\text{num}}(\mathcal{I})$ the set of all numerical partitions that can be generated given the set $\mathcal{I}$.

A numerical partition set $\mathcal{P}^{num}$ has the same cardinality as $\mathcal{P}$.

Example 1.2.2 (Numerical Partition $\mathcal{P}^{num}$ of set $\mathcal{I}$)**.** Let us consider $|\mathcal{I}| = 4$. A first numerical partition of $\mathcal{I}$ is $\{4\}$. Another numerical partition of $\mathcal{I}$ is {2,1,1}. The set of all numerical partitions is $\Pi^{num}(\mathcal{I}) = \{\{4\},\{3, 1\},\{2, 2\},\{2, 1, 1\},\{1, 1, 1, 1\}\}$.

When partitioning a set $\mathcal{I}$ where it is possible to consider the elements indistinct, using numerical partitions reduces $|\Pi(\mathcal{I})|$ into $|\Pi^{num}(\mathcal{I})|$, which is a significant concision advantage, especially for big $|\mathcal{I}|$. Example 1.2.3 will highlight better this advantage.

There are two sequences, the **Bell Numbers** and the **Partition Numbers**, that represent the number of ways to partition a set of n **distinct** elements and of n **indistinct** elements, respectively. They both grow exponentially in the size of n, but Bell Numbers grow faster.

Given a set $\mathcal{D}$ of n distinct elements, the respective Bell Number is $\text{Bell}(n) = |\Pi(\mathcal{D})| = \sum_{i=0}^{n-1} \binom{n-1}{i} \text{Bell}(i)$. This recursive formula starts with $\text{Bell}(0) = 1$, and this e-book [32] contains a chapter that explains this formula clearly.

Given a set $\mathcal{I}$ of n indistinct elements, the respective Partition Number is $\mathrm{PN}(n) = |\Pi(\mathcal{I})|$. There is an asymptotic formula for $\mathrm{PN}(n)$ found by Hardy and Ramanujan [34] demonstrating that the sequence follows an exponential growth:

$$\mathrm{PN}(n) \sim \frac{1}{4n\sqrt{3}} \exp\left(\sqrt{\frac{2}{3}}\pi n^{\frac{1}{2}}\right). \tag{1.1}$$

This approximation function is best when n is large: when $n \leq 25$, the relative error is greater than 9%, while when $1000 \leq n \leq 10000$, the relative error is greater than 0.44%. In Li [48], they find experimentally a better approximation function when $n \leq 80$, with a relative error less than 0.004%:

$$\mathrm{PN}(n)_{n\leq 80} \sim \left\lfloor \frac{1}{4(n+\mathrm{C}(n))\sqrt{3}} \exp\left(\sqrt{\frac{2}{3}}\pi n^{\frac{1}{2}}\right)\right\rfloor, \tag{1.2}$$

where $\mathrm{C}(n) = 0.4527092482 \times \sqrt{n + 4.35278} - 0.05498719946$. For $n > 80$, they recommend using instead:

$$\mathrm{PN}(n)_{n>80} \sim \left\lfloor \frac{1}{4(n+a\sqrt{n+c}+b)\sqrt{3}} \exp\left(\sqrt{\frac{2}{3}}\pi n^{\frac{1}{2}}\right)\right\rfloor, \tag{1.3}$$

where $a = 0.4432884566$, $b = 0.1325096085$ and $c = 0.274078$. In this case, the relative error is below 0.001% [48]. [1] If we take Approximations (1.3) and (1.3) minus their respective relative error, we create $\mathrm{PN}_{lb}(n)$, a lower bound on $\mathrm{PN}(n)$:

$$\mathrm{PN}_{lb}(n) = \begin{cases} \left\lfloor \frac{1}{4(n+\mathrm{C}(n))\sqrt{3}} \exp\left(\sqrt{\frac{2}{3}}\pi n^{\frac{1}{2}}\right)\right\rfloor \times 0.996 & n \leq 80 \\ \left\lfloor \frac{1}{4(n+a\sqrt{n+c}+b)\sqrt{3}} \exp\left(\sqrt{\frac{2}{3}}\pi n^{\frac{1}{2}}\right)\right\rfloor \times 0.999 & n > 80, \end{cases} \tag{1.4}$$

where the 0.996 and 0.999 correspond to 1 minus the aforementioned errors.

If the reader wants more information on those sequences and how to generate them, they can consult [60, 61].

1.3. Partition function form game

In cooperative game theory, players are allowed to cooperate, i.e., players can establish binding agreements. Once a subset of players have agreed to cooperate, we call it a coalition: a cluster of players who have decided to coordinate their actions in order to favor their group. When the outcome of a coalition depends on its context, the partition, we need to embed their definitions into a single representative entity.

Definition 1.3.1 (Embedded Coalition)**.** *Given a set of players $\mathcal{N}$, an embedded coalition is a pair $(\mathcal{C}, \mathcal{P})$ composed of a coalition $\mathcal{C}$ and a partition $\mathcal{P}$ of $\mathcal{N}$, where the coalition $\mathcal{C} \in \mathcal{P}$. The set of all embedded coalitions is denoted by $\mathcal{E}(\mathcal{N})$.*

[1] We verified experimentally the approximations provided by Li [48].

Example 1.2.3 (Comparing the Bell and Partition Numbers sequences with small n)**.**

n	Bell(n)	PN(n)
0	1	1
1	1	1
2	2	2
3	5	3
4	15	5
5	52	7
6	203	11
7	877	15
8	4140	22
9	21147	30
10	115975	42
11	678570	56
12	4213597	77
13	27644437	101
14	190899322	135
15	1382958545	176

Table 1.1. Bell and Partition Numbers.

Example 1.3.1 (Embedded Coalitions $(\mathcal{C}, \mathcal{P})$ of set $\mathcal{N}$)**.** Let us consider $\mathcal{N} = \{1, 2, 3\}$. An embedded coalition example could be ({3},{{1,2},{3}}). If we write down all the embedded coalitions $(\mathcal{C}, \mathcal{P})$ of $\mathcal{N}$, we obtain the set

$$
\begin{aligned}
&\mathcal{E}(\mathcal{N}) = \Big\{ \\
&(\{1,2,3\}, \{\{1,2,3\}\}), \\
&(\{1,2\}, \{\{1,2\}, \{3\}\}), \\
&(\{3\}, \{\{1,2\}, \{3\}\}), \\
&(\{1,3\}, \{\{1,3\}, \{2\}\}), \\
&(\{2\}, \{\{1,3\}, \{2\}\}), \\
&(\{1\}, \{\{1\}, \{2,3\}\}), \\
&(\{2,3\}, \{\{1\}, \{2,3\}\}), \\
&(\{1\}, \{\{1\}, \{2\}, \{3\}\}), \\
&(\{2\}, \{\{1\}, \{2\}, \{3\}\}), \\
&(\{3\}, \{\{1\}, \{2\}, \{3\}\}), \\
&\Big\}.
\end{aligned}
$$

In terms of size, keeping in mind that $|\Pi(\mathcal{N})| = \text{Bell}(|\mathcal{N}|)$, we are left with a set that is towering: $|\mathcal{E}(\mathcal{N})| = \sum_{\mathcal{P}\in\Pi(\mathcal{N})} |\mathcal{P}|$.

In the eventuality where, from the game's point of view, the players in the set $\mathcal{N}$ are indiscernible, they can still form coalitions. Moreover, the writing of the coalitions can be synthesized using embedded numerical coalitions in a manner similar to that we previously demonstrated with numerical partitions.

Definition 1.3.2 (Embedded Numerical Coalition)**.** *Given a set of indiscernible players $\mathcal{N}$ of cardinality n, an embedded numerical coalition is a pair $(c, \mathcal{P}^{\text{num}})$ composed of a strictly positive integer $c \in \mathbb{I}_{>0}$, $c \leq n$, and a numerical partition $\mathcal{P}^{\text{num}}$ of $\mathcal{N}$, where the coalition $c \in \mathcal{P}^{\text{num}}$. The set of all embedded numerical coalitions is denoted by $\mathcal{E}^{\text{num}}(\mathcal{N})$.*

Example 1.3.2 (Embedded Numerical Coalitions $(c, \mathcal{P}^{num})$ of set $\mathcal{N}$)**.** Let us consider $|\mathcal{N}| = 3$. If we write down all the embedded coalitions $(c, \mathcal{P}^{num})$ of $\mathcal{N}$, we will get the set

$$
\begin{aligned}
&\mathcal{E}^{num}(\mathcal{N}) = \Big\{\\
&(3, \{3\}),\\
&(2, \{2{,}1\}),\\
&(1, \{2{,}1\}),\\
&(1, \{1{,}1{,}1\})\\
&\Big\}.
\end{aligned}
$$

With embedded numerical coalitions, we still have $|\mathcal{E}(\mathcal{N})| = \sum_{\mathcal{P}\in\Pi(\mathcal{N})} |\mathcal{P}|$, however, this time $|\Pi(\mathcal{N})| = \text{PN}(|\mathcal{N}|)$, which makes it sensibly smaller than the embedded coalition set for the same number of players. We have $|\mathcal{E}^{num}(\mathcal{N})|$ that coincides with $\text{CN}(|\mathcal{N}| - 1)$, where $\text{CN}(n)$ **is the number of embedded numerical coalitions of a set possessing** $n + 1$ **elements** [59].

Once the players in $\mathcal{N}$ have formed a partition $\mathcal{P}$, we are interested in determining how they split the value of each coalition in $\mathcal{P}$ among them. In simple words, the players in a coalition will collaborate if they deem it worthwhile in terms of their individual interests. Hence, we provide the following definition:

Definition 1.3.3 (Utility allocation vector)**.** *Given a set of players $\mathcal{N}$ of cardinality n, a utility allocation vector, also called payoff allocation vector, is a vector $\vec{y} \in \mathbb{R}^n$ where the i-th entry $\vec{y}_i$ indicates the allocation received by player i.*

Building on top of the previous definitions, we are now ready to define *Partition Function Form Game* (PFG). The minimal information required to define a PFG is the *worth* (payoff) of a coalition, which may depend on the context in which it is formed, i.e., how the players partition themselves into coalitions. As a result, the definition of a PFG links an embedded coalition to a payoff.

Definition 1.3.4 (Partition Function Form Game (PFG))**.** *A game in partition function form is given by a pair $(\mathcal{N}, V)$ where $\mathcal{N}$ is the set of players and $V : \mathcal{E}(\mathcal{N}) \to \mathbb{R}$ is the partition function assigning a real number to every embedded coalition.*

Essentially, we can model the gain of an embedded coalition through the definition of a partition function. Therefore, the value $V(\mathcal{C}, \mathcal{P})$ will be referred as the utility of the coalition $\mathcal{C} \in \mathcal{P}$.

Example 1.3.3 (Partition Function Form Game (PFG))**.** Let us define a PFG $(\mathcal{N}, V)$ where $\mathcal{N} = \{1,2,3\}$ and

$$V = \Big\{$$

$$\begin{aligned}
(\{1,2,3\}, \{\{1,2,3\}\}) &\mapsto 9\\
(\{1,2\}, \{\{1,2\}, \{3\}\}) &\mapsto 5\\
(\{3\}, \{\{1,2\}, \{3\}\}) &\mapsto 2\\
(\{1,3\}, \{\{1,3\}, \{2\}\}) &\mapsto 7\\
(\{2\}, \{\{1,3\}, \{2\}\}) &\mapsto 3\\
(\{1\}, \{\{1\}, \{2,3\}\}) &\mapsto 1\\
(\{2,3\}, \{\{1\}, \{2,3\}\}) &\mapsto 5\\
(\{1\}, \{\{1\}, \{2\}, \{3\}\}) &\mapsto 4\\
(\{2\}, \{\{1\}, \{2\}, \{3\}\}) &\mapsto 3\\
(\{3\}, \{\{1\}, \{2\}, \{3\}\}) &\mapsto 3
\end{aligned}$$

$$\Big\}.$$

When solving a PFG, we aim at answering two questions:

(1) Which partition $\tilde{\mathcal{P}}$ of $\mathcal{N}$ will form?

(2) What will be the payoff allocation $\vec{y} \in \mathbb{R}^n$ received by each player in $\mathcal{N}$ such that $\sum_{i=1}^n \vec{y}_i = \sum_{\mathcal{C} \in \tilde{\mathcal{P}}} V(\mathcal{C}, \tilde{\mathcal{P}})$?

The two questions are tightly related. On one hand, knowing which partition will be formed gives us the total amount of utility the players can share. On the other hand, knowing the payoff that each player will get in every partition influences the formation of a partition. Depending on the method used, they can be answered sequentially (in any order) or simultaneously. Together, their answers constitute the outcome of the game.

Definition 1.3.5 (Outcome of a PFG)**.** *An outcome for a partition function form game $(\mathcal{N}, V)$ is a pair $(\tilde{\mathcal{P}}, \vec{y})$ where $\tilde{\mathcal{P}} \in \Pi(\mathcal{N})$ and $\vec{y} \in \mathbb{R}^n$ is a payoff allocation vector such that $\sum_{i=1}^n \vec{y}_i = \sum_{\mathcal{C} \in \tilde{\mathcal{P}}} V(\mathcal{C}, \tilde{\mathcal{P}})$.*

When studying PFGs, a few partitions are worth noting because of their relation with solution concepts for these games. They will reveal of interest for our work.

- The *grand coalition (GC)* is the partition of coarser granularity where all players are teamed up together. Thus, the grand coalition is simply when $\mathcal{N} \in \mathcal{P}$.
- The *singleton partition (SP)* is the partition of finer granularity where all the players are alone in their coalition. This is given by $\{\{1\},\{2\},\ldots,\{n\}\}$, or, more generally, when $\forall j \in \mathcal{N}, \{j\} \in \mathcal{P}$.
- The γ*-partitions* are the partitions where at most one coalition has more than one player. It is any partition of structure such that $\Big(\exists! \mathcal{S} \in \mathcal{P} : |\mathcal{S}| > 1\Big) \vee \Big(|\mathcal{S}| = 1 \ \ \forall \mathcal{S} \in \mathcal{P}\Big)$, which includes the GC and the SP.

Example 1.3.4 (GC, SP and γ-partitions of $\mathcal{N}$)**.** When $\mathcal{N} = \{1,2,3,4\}$, we have

$$\begin{aligned}
&GC = \{\{1,2,3,4\}\},\\
&SP = \{\{1\},\{2\},\{3\},\{4\}\} \text{ and}\\
&\gamma\text{-partitions} = \Big\{\\
&\{\{1\},\{2\},\{3\},\{4\}\},\\
&\{\{1,2\},\{3\},\{4\}\},\\
&\{\{1,3\},\{2\},\{4\}\},\\
&\{\{1,4\},\{2\},\{3\}\},\\
&\{\{1\},\{2,3\},\{4\}\},\\
&\{\{1\},\{2,4\},\{3\}\},\\
&\{\{1\},\{2\},\{3,4\}\},\\
&\{\{1,2,3\},\{4\}\},\\
&\{\{1,2,4\},\{3\}\},\\
&\{\{1,3,4\},\{2\}\},\\
&\{\{1\},\{2,3,4\}\},\\
&\{\{1,2,3,4\}\}\\
&\Big\}.
\end{aligned}$$

Those special partitions also have their numerical version.

Example 1.3.5 (Numerical GC, SP and γ-partitions of $\mathcal{N}$)**.** When $\mathcal{N} = \{1,2,3,4\}$, we have

$$GC = \{4\},$$

$SP = \{1,1,1,1\}$ and

γ-partitions $= \Big\{$

$\{1,1,1,1\}$,

$\{2,1,1\}$,

$\{3,1\}$,

$\{4\}$

$\Big\}$.

1.3.1. Symmetric partition function form game

A *Symmetric Partition Function Form Game* (SPFG) is a special case of a PFG where players are clones of each other. We mean by using this term they possess the same initiative levels, strategies, objectives, perceived payoff value, risk tolerance, etc. This creates symmetry in the game, in the sense that if you were to swap any two players in any partition, the permutation would not generate any perturbation to the game at the exception of the two players now receiving the payoff of the other player, and continuing the game from there.

Giving the utility in terms of embedded coalitions becomes redundant, since the important parameter under this paradigm is the number of players per coalition in the partition. Thus, the partition function links an embedded numerical coalition to a payoff.

Definition 1.3.6 (Symmetric Partition Function Form Game (SPFG))**.** *A symmetric partition function form game is given by a pair* $(\mathcal{N}, W)$ *where* $\mathcal{N}$ *is a set of indiscernible players and* $W : \mathcal{E}^{\text{num}}(\mathcal{N}) \to \mathbb{R}$ *is the partition function assigning a real number to every embedded numerical coalition.*

Example 1.3.6 (Symmetric Partition Function Form Game (SPFG) of $\mathcal{N}$)**.** Let us define an SPFG $(\mathcal{N}, W)$ where $|\mathcal{N}| = 3$ and

$W = \Big\{$

$(\{3\}, \{3\}) \mapsto 9,$

$(\{2\}, \{2, 1\}) \mapsto 6,$

$(\{1\}, \{2, 1\}) \mapsto 1,$

$(\{1\}, \{1, 1, 1\}) \mapsto 2$

$\Big\}$.

The study of an SPFG is very advantageous as symmetry greatly simplifies the game. Not only does it reduces the size of its representation, it also collapses its solution concepts for fairness study. We will see in Chapter 3 that such solution concepts in symmetric games become vain since the distribution of utility between identical players is trivial.

1.3.2. Characteristic function form game

A characteristic function form game (CFG) is a game of type similar to the PFG's, but where the partition function (here known as the characteristic function) maps a coalition to a payoff. Hence, the players of a coalition are not impacted by the actions of the remaining players.

Definition 1.3.7 (Characteristic Function Form Game (CFG))**.** *A characteristic function form game is given by a pair* $(\mathcal{N}, U)$ *where* $\mathcal{N}$ *is the set of players and* $U : 2^{\mathcal{N}} \to \mathbb{R}$ *is the characteristic function assigning a real number to every subset of* $\mathcal{N}$*. We have* $U(\emptyset) = 0$*.*

Note that a PFG $(\mathcal{N}, V)$ where the partition function satisfies the following equation for each $\mathcal{C} \subseteq \mathcal{N}$:

$$V(\mathcal{C},\mathcal{P}) = V(\mathcal{C},\mathcal{P}') \qquad \forall \mathcal{P} \in \Pi(\mathcal{N}) : \mathcal{C} \in \mathcal{P}$$

is a CFG.

Example 1.3.7 (Characteristic Function Form Game (CFG) of $\mathcal{N}$)**.** Let us define a CFG $(\mathcal{N}, U)$ where $\mathcal{N} = \{1,2,3\}$ and

$$U = \Big\{$$
$$\{1,2,3\} \mapsto 9,$$
$$\{1,2\} \mapsto 5,$$
$$\{1,3\} \mapsto 6,$$
$$\{2,3\} \mapsto 4,$$
$$\{1\} \mapsto 2,$$
$$\{2\} \mapsto 3,$$
$$\{3\} \mapsto 1,$$
$$\Big\}.$$

When solving a CFG, we aim at answering two questions:

(1) Which partition $\tilde{\mathcal{P}}$ of $\mathcal{N}$ will form? We know that the payoffs do not depend on the partition, but two coalitions figuring a common player cannot be formed simultaneously.

(2) What will be the payoff allocation $\vec{\tilde{y}} \in \mathbb{R}^n$ received by each player of $\mathcal{N}$ such that $\sum_{i=1}^{n} \vec{\tilde{y}}_i = \sum_{\mathcal{C} \in \tilde{\mathcal{P}}} U(\mathcal{C})$?

The two questions above are the same as the ones posed for PFGs. However, as we will discuss in Chapter 2, the assumptions and reasoning usually employed to solve CFGs do not hold when tackling PFGs. Indeed, the class of PFGs is a more general class of games than the class CFGs. Thus, the study in this thesis for PFGs will hold for CFGs.

1.3.3. Games with positive and negative externalities

Externalities in a game are the quantifiable impact of the formation of a coalition on the other coalitions. In a CFG, there are no externalities, since the partition function value of a coalition only depends on the players in the coalition itself.

In the general case of PFGs, we can have externalities. Whenever a coalition changes, the context, or partition in which the players are currently distributed, also switches. Thus, when a coalition benefits (resp. does not benefit) from the distribution into coalitions of the other players, we say that this variation in utility is the fruit of positive (resp. negative) externalities on the coalition.

If, in a game, all coalition formations are beneficial to all coalitions, we say that the game is with *positive externalities*. On the contrary, if all coalition formations are detrimental to all coalitions, we say that the game is with *negative externalities*.

Many games will not be limited to only positive or negative externalities, but will have a mixture of both. Therefore, coalition formations will be sometimes beneficial, sometimes detrimental, depending on the context and on the observed impacted coalition.

Definition 1.3.8 (PFG with positive externalities)**.** *A PFG* $(\mathcal{N}, V)$ *is with positive externalities if for each mutually disjoint coalitions* $\mathcal{R}$, $\mathcal{S}$, $\mathcal{T} \subseteq \mathcal{N}$ *and partition* $\mathcal{P} \in \Pi(\mathcal{N} \setminus (\mathcal{R} \cup \mathcal{S} \cup \mathcal{T}))$, *the following inequality holds:*

$$V(\mathcal{R}, (\mathcal{R}, \mathcal{S} \cup \mathcal{T}, \mathcal{P})) \geq V(\mathcal{R}, (\mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{P})) \ \ \forall \mathcal{R}, \mathcal{S}, \mathcal{T}.$$

Definition 1.3.9 (PFG with negative externalities)**.** *A PFG* $(\mathcal{N}, V)$ *is with negative externalities if for each mutually disjoint coalitions* $\mathcal{R}$, $\mathcal{S}$, $\mathcal{T} \subseteq \mathcal{N}$ *and partition* $\mathcal{P} \in \Pi(\mathcal{N} \setminus (\mathcal{R} \cup \mathcal{S} \cup \mathcal{T}))$, *the following inequality holds:*

$$V(\mathcal{R}, (\mathcal{R}, \mathcal{S} \cup \mathcal{T}, \mathcal{P})) \leq V(\mathcal{R}, (\mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{P})) \ \ \forall \mathcal{R}, \mathcal{S}, \mathcal{T}.$$

Example 1.3.8 (PFG of $\mathcal{N}$ with positive externalities)**.** Let us define a PFG $(\mathcal{N}, V)$ where $\mathcal{N} = \{1,2,3\}$ and

$$V = \Big\{$$
$$(\{1,2,3\}, \{\{1,2,3\}\}) \mapsto 300,$$
$$(\{1,2\}, \{\{1,2\}, \{3\}\}) \mapsto 250,$$

$(\{3\},\{\{1,2\},\{3\}\}) \mapsto 40,$

$(\{1,3\},\{\{1,3\},\{2\}\}) \mapsto 260,$

$(\{2\},\{\{1,3\},\{2\}\}) \mapsto 30,$

$(\{1\},\{\{1\},\{2,3\}\}) \mapsto 20,$

$(\{2,3\},\{\{1\},\{2,3\}\}) \mapsto 270,$

$(\{1\},\{\{1\},\{2\},\{3\}\}) \mapsto 2,$

$(\{2\},\{\{1\},\{2\},\{3\}\}) \mapsto 3,$

$(\{3\},\{\{1\},\{2\},\{3\}\}) \mapsto 4$

$\Big\}.$

Example 1.3.9 (PFG of $\mathcal{N}$ with negative externalities)**.** Let us define a PFG $(\mathcal{N}, V)$ where $\mathcal{N} = \{1,2,3\}$ and

$V = \Big\{$

$(\{1,2,3\},\{\{1,2,3\}\}) \mapsto 1,$

$(\{1,2\},\{\{1,2\},\{3\}\}) \mapsto 20,$

$(\{3\},\{\{1,2\},\{3\}\}) \mapsto 100,$

$(\{1,3\},\{\{1,3\},\{2\}\}) \mapsto 30,$

$(\{2\},\{\{1,3\},\{2\}\}) \mapsto 120,$

$(\{1\},\{\{1\},\{2,3\}\}) \mapsto 140,$

$(\{2,3\},\{\{1\},\{2,3\}\}) \mapsto 40,$

$(\{1\},\{\{1\},\{2\},\{3\}\}) \mapsto 200,$

$(\{2\},\{\{1\},\{2\},\{3\}\}) \mapsto 250,$

$(\{3\},\{\{1\},\{2\},\{3\}\}) \mapsto 300$

$\Big\}.$

As we will show in Chapter 3, if we know that a PFG is with positive or negative externalities, this simplifies the existence of a solution satisfying stability conditions inspired by strategical behavior.

1.4. Mixed-integer programming

The flexible framework of mixed-integer programming (MIP) allows us to model several real-world decision-making problems. Concretely, a *mixed-integer linear program* (MILP) is given by

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & \vec{c}^T \vec{x} & \\ s.t. \quad & \mathbf{A}\vec{x} \leq \vec{b} & \\ & \vec{x}_i \in \mathbb{Z} & i = 1, \ldots, n', \end{aligned}$$

where $n' < n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\vec{c}$ and $\vec{b}$ are vectors of dimension n and m, respectively. An MILP models the problem of determining the decision vector $\vec{x}$ maximizing the linear objective function $\vec{c}^T \vec{x}$, while respecting the m linear constraints and the integer requirements.

MILPs have their usefulness well-proven in many domains such as supply chains and health care, although they are classified as NP-hard and thus are theoretically intractable. Nonetheless, motivated by their undeniable modeling flexibility and the existence of powerful solvers to tackle them, MILPs applied to cooperative game theory will be used and discussed in this work. For further details about integer programming, we refer the interested reader to the textbooks by Conforti et al. [18], Wolsey [84].

Chapter 2

Literature review

We stated in Chapter 1 the central questions to cooperative game theory: "What will be the partition $\tilde{\mathcal{P}}$ of the set of players $\mathcal{N}$ to be formed?" and "What will be the payoff allocation $\vec{\tilde{y}} \in \mathbb{R}^n$ received by each player of $\mathcal{N}$?" This literature review will revolve around existent work seeking to solve one or both of those questions. We observe that solution concepts in cooperative game theory are usually defined first for CFGs, which is a game class attributed to von Neumann and Morgenstern [82]. They are then frequently extended to PFGs, a game class first presented by Thrall and Lucas [78] considering the case with externalities. We defined both game classes in Section 1.3.

Before diving into the literature related to cooperative game theory, in Section 2.1, we briefly describe a classic solution to non-cooperative games. Our goal is to emphasize that in both cases, cooperative and non-cooperative games, it is strategic behavior that is mostly in the base of the existent solution concepts.[1] In Section 2.2, we review works discussing multiple solution concepts for the distribution of the payoff between the players. Then, in Section 2.3, we discuss coalition structure formation in Multi-Agent Systems (MAS), which are settings with more than one decision maker (agent). We divide this MAS literature for coalition structure formation into two categories, based on whether the agents are benevolent and willing to accept the agenda of a central planner coordinating their efforts to reach a common objective, or whether the agents are self-interested and open to cooperation strictly because it might enhance their individual payoffs. Since within coalition structure formation, some classical payoff allocation concepts are used, this explains why we first provide a review of the literature on them.

[1]It must be noted that in cooperative game theory, players can still behave strategically as in non-cooperative games. The main difference is in the fact that in cooperative games players can form coalitions and thus, for their mathematical formalization, we only need a partition function. In non-cooperative games, we must know each player payoff function and set of feasible strategies.

2.1. Non-cooperative game theory

The most well-known solution concept of a game is the Nash equilibrium. It defines a *stable* solution for non-cooperative games with two or more players, i.e., games where players cannot form binding agreements. The Nobel laureate John Nash proved that there is a Nash equilibrium (solution) for every finite game [56]. An outcome of a non-cooperative game (i.e., strategies selected by the players) where no player has incentive to unilaterally deviate is a Nash equilibrium. This definition makes it clear why we can see an equilibrium as a concept of stability and that it defines a *neighbourhood* around the equilibrium (set of unilateral deviations). This concept is widely used, sometimes alongside MIP tools, to solve various games such as in the context of power markets [27], (symmetric) network congestion games [66], knapsack and competitive lot-sizing games [13] and the games generated by the test suite GAMUT [58, 72]. These works show us that applying MIP techniques to games can be an effective tool to solve them. However, in the context of cooperative games, we have to look further than Nash Equilibria in order to find a solution concept for PFGs or CFGs, although stability stays an important component of what a solution concept attempts to capture.

What would constitute a satisfactory solution concept for a game where players are open to cooperating, but have conflicting interests about what the best partition and the best payoff allocation are? Certain philosophies put emphasis on *stability*, i.e., attempt at predicting in which state the game would naturally fall; by state we mean partition of the set of players and payoff allocations. The following sections present a discussion on the main payoff allocation and partition formation philosophies together with well-known solution concepts for cooperative games.

2.2. Payoff sharing

As far as the allocation of payoffs is concerned, stability must reflect the fact that no set of players has an interest in breaking away from their respective coalitions to form a new one where their payoffs would be higher. Beyond stability, one could argue that we must represent the interests of all players, so what is needed is *fairness*. For instance, the payoff allocation should reward players according to their contribution to the coalition they are in. Thus, a good solution concept will try to mathematically capture fairness, i.e., to treat the players impartially, like any good referee in a game. Stability is not antagonist to fairness, and a solution concept could arguably present both of these characteristics. Just like in real life, whether something is stable or fair resides primarily in the eye of the observer and in the context of the game. One thing is for sure: within the limits imposed by the rules of the game, each player deserves the same chance to maximize their winnings. Depending on one's philosophy, the ideal solution concept for payoff allocation could resemble the core [28], the nucleolus [73] or the Shapley value [74], to name a few.

Core concepts. The core was first introduced by Gillies [28]. It is a solution concept based on efficiency and on subset rationality, originally defined given a game $(\mathcal{N}, U)$ where there are no externalities, i.e., U is a characteristic function. First, we have *efficiency*:

$$\sum_{i=1}^{n} \vec{y}_i = \sum_{\mathcal{C} \in \tilde{\mathcal{P}}} U(\mathcal{C}). \tag{2.1}$$

Second, we have *coalitional rationality*:

$$\sum_{i \in \mathcal{C}} \vec{y}_i \geq U(\mathcal{C}) \quad \forall \mathcal{C} \subseteq \mathcal{N}. \tag{2.2}$$

According to the core, if we can find a payoff allocation $\vec{y}$ such that Constraints (2.1)-(2.2) hold, the chosen partition $\tilde{\mathcal{P}}$ is stable. This is because respecting Constraints (2.1)-(2.2) ensures that no set of players has incentive to change from $\tilde{\mathcal{P}}$ if each player i receives the respective $\vec{y}_i$, as they would not be able to do better on their own [23]. Originally defined for CFGs, the core assumes the GC will form and thus, an allocation $\vec{y}$ is in the core if it satisfies Constraints (2.1)-(2.2) with $\tilde{\mathcal{P}} = \{\mathcal{N}\}$. Since in a CFG, the formation of a coalition does not impact the value of the remaining coalitions, it is possible to consider that the GC is always forming. In fact, we can consider the existence of a central planner who finds the social welfare (SW)-maximizing partition, and then, we assume that the determined maximum SW is the characteristic value of GC. This scheme allows players to always dispose of the greatest total payoff that the game could offer to share. In addition, it systematizes the solving of the problem, as the GC will always be assumed to be formed, reducing the problem to the allocation of payoffs among the players in $\mathcal{N}$. The core, when it is non-empty, furnishes a stable set of payoff allocation vectors. To name one common core usage, Anupindi et al. [1] employ it for their cooperative surplus allocation step in a two-stage decentralized distribution system.

Numerous works aim at tempering the limitations of this solution concept, notably by extending its definition to a broader class of games such as PFGs. In PFGs, the GC scheme cannot be utilized to simplify solving a game: since there are externalities, we cannot redefine the partition function value for the GC as the total value of the SW-maximizing partition. Most concepts are based on reducing PFGs to CFGs, and then applying the core definition to the obtained CFG. For this reduction, authors make different assumptions on how players outside of a coalition react to it. A first example is the α-core, by Aumann [2], that reads just like the core, but replaces in a game $(\mathcal{N}, W)$ the partition function W by W^{α}. This new function attributes to every coalition $\mathcal{C} \subseteq \mathcal{N}$ the lowest value it can obtain:

$$W^{\alpha}(\mathcal{C}) = \min \left\{ W(\mathcal{C}, \mathcal{P}) : \mathcal{P} \in \{\mathcal{C}, \mathcal{S}\}, \mathcal{S} \in \Pi(\mathcal{N} \setminus \mathcal{C}) \right\}.$$

This pessimistic approach allows to apply the core to a game with externalities by transforming it into the CFG $(\mathcal{N}, W^{\alpha})$. In the same fashion, but following an optimistic approach, there is the ω-core, by Shenoy [75]. It replaces in a PFG $(\mathcal{N}, W)$ the partition function W by W^{ω}, attributing to every coalition $\mathcal{C} \subseteq \mathcal{N}$ the largest value it can obtain:

$$W^{\omega}(\mathcal{C}) = \max \left\{ W(\mathcal{C}, \mathcal{P}) : \mathcal{P} \in \{\mathcal{C}, \mathcal{S}\}, \mathcal{S} \in \Pi(\mathcal{N} \setminus \mathcal{C}) \right\}.$$

In Chander and Tulkens [16], they focus on a model where the deviating coalition expects the other players to split into singletons, in order to prevent free-riding by any deviating coalition. Basically, it replaces in the PFG $(\mathcal{N}, W)$, the partition function W by W^{γ}, attributing to every coalition $\mathcal{C} \subseteq \mathcal{N}$ the value it obtains when it is the coalition of greatest cardinality in the γ-partition:

$$W^{\gamma}(\mathcal{C}) = W(\mathcal{C}, \{\mathcal{C}, \mathcal{S}\}) \text{ such that } \mathcal{S} \in \Pi(\mathcal{N} \setminus \mathcal{C}), |\mathcal{T}| = 1 \; \forall \mathcal{T} \in \mathcal{S}.$$

In Hart and Kurz [36], it is assumed that players will not react to a partition deviation: W of $(\mathcal{N}, W)$ becomes W^{δ}, where the new partition after $\mathcal{C}$ deviates from the coalition structure $\mathcal{P} \ni \mathcal{C}$ is simply $\mathcal{P} \setminus \mathcal{C}$:

$$W^{\delta}(\mathcal{C}) = W(\mathcal{C}, \{\mathcal{C}, \mathcal{P} \setminus \mathcal{C}\}) \text{ such that } \forall \mathcal{S} \in \mathcal{P}, (\mathcal{S} - (\mathcal{S} \cap \mathcal{C})) \in \mathcal{P} \setminus \mathcal{C}.$$

For a given game $(\mathcal{N}, W)$, let $\mathcal{F}^{(\alpha)}(W)$, $\mathcal{F}^{(\gamma)}(W)$ and $\mathcal{F}^{(\delta)}(W)$ represent respectively the set of α-, γ- and δ-stable coalition structures, i.e., the set of players' partitions such that the core of the obtained CFG is non-empty. In other words, a partition $\tilde{\mathcal{P}}$ is in $\mathcal{F}^{(\alpha)}(W)$, $\mathcal{F}^{(\gamma)}(W)$ and $\mathcal{F}^{(\delta)}(W)$, if there is a payoff allocation $\vec{y}$ such that Constraints (2.1)-(2.2) hold for $U = W^{\alpha}$, $U = W^{\gamma}$ and $U = W^{\delta}$, respectively. Then, according to these definitions, we have $\left(\mathcal{F}^{(\delta)}(W) \cup \mathcal{F}^{(\gamma)}(W)\right) \subset \mathcal{F}^{(\alpha)}(W)$ [36]. In a similar vein, there are works by Kóczy [40, 41, 43], who models games with various levels of externalities using the recursive core, a solution concept that, when a deviation occurs, allows the non-deviating players to react and form a core-stable partition. It is also relevant to mention the strong-core by Chander [15], a solution concept for PFGs that reduces to the core when the PFG is also a CFG. A feasible payoff vector belongs to the strong-core if, when compared to the SP, no coalition is strictly better-off in the SP, and if, when compared to the remaining partitions, there exists no partition such that every non-singleton coalition is strictly better-off in the partition. For PFGs with positive externalities, they also show that δ-core $\subset$ strong-core $\subset$ γ-core, and for PFGs with negative externalities, γ-core $\subset$ strong-core $\subset$ δ-core. Bloch and van den Nouweland [10] analyze different definitions of cores and find that the projection core (which is equivalent to the δ-core proposed by Hart and Kurz [36]), because it satisfies subset consistency, and the pessimistic core (which is equivalent to the α-core proposed by Aumann [2]), because it preserves superadditivity, are the most interesting to study PFGs.

In Gopalakrishnan and Sankaranarayanan [29], the authors define an agreeable allocation for cost sharing of interdependent risks that is the average of the extreme core allocations. This custom version of the core is applied to tackle network-wide cooperative security in the food manufacturing sector. Finally, it is important to mention the work by Zhao [87], linking the Nash equilibrium concept to the core through the definition of hybrid solutions that consist in the responses of the players when there is coexistence of cooperation (within a coalition) and competition (between the coalitions). We are in presence of a Nash equilibrium without hybridization when the SP forms, and in presence of a core without hybridization when the GC forms. The present thesis uses a custom version of the core applied to PFGs that is presented in Section 3.3.

Nucleolus. While the definition of core defines a set of payoff allocations, the nucleolus solution concept for CFGs is unique, lies in the core if the core is non-empty, and it is guaranteed to exist in games with individually rational payoff allocations, i.e., allocations where all the players can get at least their SP characteristic function value, as it aims at returning the most stable of them by lexicographically minimizing the dissatisfaction of all coalitions. Introduced by Schmeidler [73], the nucleolus is generally considered a computationally demanding problem for many classes of games, as only a few special classes are guaranteed to have polynomially bounded running times for the computation of the nucleolus [23]. Multiple works seek to extend or refine this solution concept. In Benedek et al. [5], the computation of the nucleolus is democratized to moderately large cooperative games by the introduction of a new, reasonably-sized set of necessary and sufficient conditions to be in the nucleolus, as well as a new lexicographical descent algorithm for finding the nucleolus. While the nucleolus is defined for CFGs, the work by Tripathi and Amit [79] tackles the issue of adapting it to games with externalities (PFGs). In this thesis, we consider SPFGs, so the payoff sharing is more straightforward. Nonetheless, in future work, it could be interesting to investigate the computation of the nucleolus using our MILP-representation (Chapter 4).

Shapley value. Among the popular solution concepts for the allocation of payoffs, we can also include the Shapley value [74]. This one is peculiar as it makes two assumptions from the get-go: *(i)* every coalition of the same size is equally likely to form, and *(ii)* every player is equally likely to join any coalition of the same size. For a CFG $(\mathcal{N}, U)$, it results in a payoff distribution scheme where each player receives $\phi_i(U)$, a weighted average of their marginal contributions:

$$\phi_i(U) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{S}|!(n - |\mathcal{S}| - 1)!}{n!} (U(\mathcal{S} \cup \{i\}) - U(\mathcal{S})).$$

The Shapley value is usually considered as a fair solution concept: its design does not force stability, but it rather puts emphasis on giving to a player the fair share of the payoff according to their contribution to any coalition. Just like with other solution concepts, research has been made with the intention of extending the Shapley value to other types of games, e.g., Norde and Pham Do [57] extended the concept to PFGs. Let *Permutations*($\mathcal{N}$) be the set of all possible permutations of the players, i.e., the set of all possible player orders. For a given $\sigma \in \mathit{Permutations}(\mathcal{N})$ and $i \in \{1, 2, \ldots, n\}$, let γ_i^σ be the γ-partition where coalition $\mathcal{C}_i^\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(i)\}$ is the coalition of greatest cardinality, with $\mathcal{C}_0^\sigma = \mathrm{SP}$. The marginal vectors of a PFG $(\mathcal{N}, W)$ are defined as follows:

$$\vec{m}^\sigma_{\sigma(1)}(W) = W(\{\sigma(1)\}, SP) = W(\mathcal{C}_1^\sigma, \gamma_1^\sigma),$$
$$\vec{m}^\sigma_{\sigma(k>1)}(W) = W(\mathcal{C}_k^\sigma, \gamma_k^\sigma) - W(\mathcal{C}_{k-1}^\sigma, \gamma_{k-1}^\sigma).$$

The Shapley value vector $\vec{\Omega}(W)$ of the PFG $(\mathcal{N}, W)$ is defined as the average of the $n!$ marginal vectors,

$$\vec{\Omega}(W) = \frac{1}{n!} \sum_{\sigma \in \mathit{Permutations}(\mathcal{N})} \vec{m}^\sigma(W).$$

In Bartholdi and Kemahlioğlu-Ziya [3] and Kemahlıoğlu-Ziya and Bartholdi [38], the authors are interested in using the Shapley value for a fair distribution of the excess profit generated by inventory pooling in supply chains, that might not be immediately stable, but can be considered stable in a farsighted sense. In Granot and Sošić [30], they present a 3-stage decentralized cooperative system of inventory ordering and inventory residuals sharing. In this system, the Shapley value can lead to the best result for given inventory decisions, but it does not coincide with core allocation rules, and thus, may result in lower total additional profit from the pooling of residual inventories in the third stage. Again, because we focus on SPFGs in this thesis, the computation of the Shapley value is arguably excessive in our case, because it is clear that to reach *fairness*, the payoff sharing should be equal for all the players.

2.3. Coalition structure formation

In multi-agent systems, when the system is a game, agents are also called players[2]. When agents are allowed to cooperate by forming coalitions, according to Chalkiadakis et al. [14], there are two main classes of problems: *(i) coalition structure generation* (CSG) is the case where agents are not selfish, but rather benevolent, and are willing to implement the solution dictated by a central planner whose goal is often to maximize the SW (i.e., the sum of the values of all coalitions formed), and *(ii) coalition formation activities by selfish rational agents* is the case where the agents want to maximize their individual payoff when participating in coalition formations. Real-world situations where agents share the same intentions and may request help from a central planner to coordinate include problems

[2]Usually, the term "players" is reserved for cases where agents are exclusively self-interested, i.e., they behave strategically.

in the context of wide-area surveillance with multi-sensor networks [46] and airline crew scheduling [77]. Inventory pooling [38], demand pooling for bulk buying, distributed vehicle routing coalitions for delivery/transportation cost sharing [4, 81], audit/demand information sharing [24, 47], multi-country disaster prevention [71] and vaccine distribution [83] are, for their part, examples of situations where agents may be tempted to strategically team up to lower their individual operation costs.

Coalition structure generation. There is a great body of works on the subject of CSG. They mainly focus on finding an efficient representation and/or an efficient algorithm to deal with such large combinatorial problems (remember Table 1.1 depicting the input size of a game). In the realm of efficient representations, there is a trade-off between succinctness, computational tractability and full expressivity, as remarked by Skibski et al. [76] in their work introducing partition decision trees (PDTs). PDTs are a rule-based concise representation of PFGs that does not take a fixed amount of space, but can express every PFG fully and can be exponentially more concise than the conventional representation of a partition function. They also develop algorithms using PDTs that compute direct extensions of the Shapley value in polynomial time, which is of equal time complexity to what a partition function form conventional representation can achieve for the same operations. In Zha et al. [86], PDTs are used to solve the CSG problem along with two methods: a depth-first branch-and-bound algorithm and a MaxSAT encoding. In Ueda et al. [80], they examine three rule-based fully-expressive concise representations of CFGs: *(i)* marginal contribution nets [37] (with a variation, embedded marginal contribution nets (embedded MC-nets) [51], that can be applied on PFGs), *(ii)* synergy coalition groups [20], and *(iii)* synergy coalition groups in multi-issue domains [19]. They develop MIP formulations to solve the CSG problem and state that the best-performing representation scheme depends on the application. Since embedded MC-nets can be applied on PFGs, Skibski et al. [76] compare their PDTs to them and found that embedded MC-nets can be exponentially more concise than PDTs, and at most polynomially less concise than PDTs. However, neither of the two representations is guaranteed to be systematically concise, i.e., to take space polynomial on the number of players in the PFG.

In this thesis, we take the opposite stance and seek a well-determined compact representation at the potential expense of only approximating a PFG. Präntare and Heintz [68] and Olariu et al. [62] also take the branch-and-bound approach to solve the CSG problem, but their perhaps more holistic views on the role of the CSG encourage them to extend or transform the CFG/PFG definition, and hence to use different representations. In the former work, coalitions are formed in order to perform a task. Thus, partitions become ordered sets such that each coalition is assigned to a task, and they can perform simultaneous coalition structure generation and assignment with an anytime algorithm for the case without externalities (most commonly modeled by CFG). In the latter work, the CSG problem is

modeled by a graph and solved using mixed integer linear/quadratic programming. Graphs are another representation on which CSG is studied with many methods being proposed such as the one by Kong et al. [44] based on min-k-cuts. They are not equivalent to CFGs: using graphs allows to give more context that is useful to the problem solving than what a CFG can by adding structure to the set of agents [81]. For instance, when the agents of the MAS are embedded in a network (e.g., computer or social network), there can be situations where two agents are disconnected and, without the presence of intermediaries, cannot cooperate. There are other works using various methods to solve the CSG problem, such as mixed integer programming tools as in Caprara and Letchford [11], Rahwan et al. [69] and Bistaffa et al. [6]. There are also authors developing metaheuristics. For example Contreras et al. [21] employ a genetic algorithm to solve the CSG problem, stating that the major issue faced in this context is to choose a proper encoding of the game that enables efficient computation.
Coalition formation activities by selfish rational agents. Stability is important with regards to solution concepts that aim to anticipate how players will partition themselves in a game. In this context, we immediately think of two algorithmic designs when talking about stability: stability constraint set design, such as in Chwe [17], and process-based designs, such as in Konishi and Ray [45]. Both of these solution concepts are used by Nagarajan and Sošić [54] in an attempt to propose a solution that captures farsightedness to a symmetric Bertrand price problem. In the following, we first discuss the literature primarily on stability constraint set design and then on process-based design.

When we talk about selfish rational agents who are allowed to cooperate (i.e., enter into binding agreements), but who will only do so if it will improve their individual payoffs, the body of literature is slightly more modest. We speculate here that it may be because when agents are no longer guided by a central planner, there is no longer a consensus on what constitutes a good solution to the game, which brings an additional layer of nuance and difficulty. Moreover, an additional question is in order: is collaboration even fruitful for the players, or is it better to stick to the SP? Fang and Cho [24] study the effects of externalities on the players' incentives to cooperate in the context of firm audits, while Basso et al. [4] attempt to justify why cooperation, albeit rewarding for the players in theory, is not common in reality, by analyzing different firms competing in multiple markets in a Cournot fashion. Prior to competing, the firms have the option of signing collaboration agreements that must conform with antitrust authorities' requirements. There are plenty of reasons why collaboration is desirable: better offer of complementary products, diminishing of the waste generated by perishable products, etc. However, as collaborative agreements are rarely seen in reality, the authors underline three possible explanations: *(i)* forming coalitions might be unprofitable, *(ii)* coalition formations might be refused by antitrust authorities and *(iii)* firms (or a central planner) might fail to compute an optimal partition. To navigate this delicate situation, they designed a few custom stability constraints in order to solve their coalition formation problem modeled with a PFG. Custom stability constraints allow to adapt the solution concept (stability definition) precisely to the context of a problem. They target a few key rational deviations that are plausible in their context and ensure they do not compromise the stability of a partition. Basso et al. share the same philosophy as in Ray and Vohra [70]: in terms of deviations, they are more concerned by coalition splitting than merging. We will discuss the constraints presented by Basso et al. [4] in Section 3.2, as we decided to follow this route and use an algorithm with custom stability constraints for the coalition structure formation. Similarly to Basso et al., Nagarajan and Sošić [54] study the behaviour of self-interested agents competing in a Cournot-like market, where signing agreements to enter coalitions could reveal beneficial to the agents. This time however, agents are symmetric and the authors are interested in finding a farsighted outcome. In Carraro and Marchiori [12] just like in Nagarajan and Sošić [54], the coalition structures analyzed are restricted to γ-partitions, since there are many situations where a single agreement is proposed and agents have to choose between signing it or not signing it. In Anupindi et al. [1], the authors analyze decentralized distribution systems where, in a two-stage game, the retailers first cooperatively decide on the allocation rules to share surplus profits, and then competitively decide on the inventory levels.

A work by Yi [85] considers the case where symmetric and self-interested agents can cooperate, and it focuses on how coalitions form (e.g., open membership or unanimity) and the effect that it has on the stable coalition structures of games with positive/negative externalities. Using a two-stage PFG model, Pintassilgo and Lindroos [67] study cooperation between symmetric players in the context of high seas fisheries. They find that prospects of cooperation are low because of countries free-riding cooperative agreements.

In selfish MAS, bargaining is also studied in multiple contexts, such as in supply chains, because it is often a natural activity that occurs when seeking cooperation. Also in the constext of self-interested agents, Klusch and Gerber [39] design and implement a simulation-based dynamic coalition formation model that let agents negotiate rationally their participation to coalitions, a method crafted for the applications of ubiquitous and mobile computing, including mobile commerce in wireless environments. In Nagarajan and Bassok [53], using the Nash bargaining concept, they find that suppliers do not form any coalition when the assembler (that buys from the suppliers) has a strong negotiation power, but form coalitions of coarser granularity as the assembler's negotiation power weakens. On the subject of environmental and social responsibility of companies, the article by Feng et al. [25] uses a combination of bargaining and Shapley value to find the gain allocation across a network. Another instance of Nash bargaining in action is by Mu et al. [52], where it is applied to pulses imports in India. In Feng et al. [26], they decide to compare the very common Nash bargaining solution to the less-known Kalai-Smorodinsky solution, that captures different elements of answer, notably negotiation power shifts. We do not use bargaining for a model simplification purpose in this thesis, but the value function of a PFG can be seen as the aftermaths of any negotiation process between the agents. In Granot and Sošić [31], three firms with products of a certain degree of substitutability have the option of creating an alliance to share development and operation costs. The authors seek to study the impacts of the marketplace joint venture on the (joined or remained independent) firms and their suppliers. Since any two firms can communicate and coordinate to deviate, they select the largest consistent set [17] to identify stable coalition structures.

In Liao et al. [49], they tackle the two same problems we do this thesis: coalition formation and payoff sharing. However, they make different design choices. On one hand, their payoff division employs the Shapley value, as they aim to enforce individual agents to get a payoff corresponding to their contribution to a coalition. On the other hand, their coalition formation employs a Markov process, because they notice that the process of individual agents morphing into different coalition structures fits naturally a Markov process method. We contemplated using a dynamic process to find stability in the coalition formation step, but ultimately chose a deterministic method due to its simplicity and suitability.

This thesis is in line with the research on coalition formation with rational and selfish agents, in contrast with most literature that is devoted to CSG. Moreover, it focuses on a less studied and more general class of games, the PFGs, and on building a framework that welcomes customized stability constraints. In addition, we propose an MILP game representation providing a novel compact modeling for SPFGs. Our compact representation is not guaranteed to achieve full expressiveness for all SPFGs, but it can be used to approximate games while being of polynomial size in the number of players.

Chapter 3

Coalition structure formation and payoff sharing

Summarizing our setting so far, we have a set of self-interested symmetric players that are open to coalition formation if it can benefit them individually, which we will model using an SPFG. This chapter will describe our stability-oriented solution concept for coalition formation and payoff sharing.

First of all, let us discuss the impact of having symmetric players on the design of a solution concept. We would like to point out that relying exclusively on fairness to select the outcome of an SPFG gives a trivial solution. No matter from which player's perspective the game is studied, it will always be exactly the same, since the players are copies of each other. Thus, the natural path to follow is to select the partition that generates the greatest total payoff and to split it equally among the players. This solution is convenient, as it answers both the questions raised in Section 1.3: "What partition will form?" and "What will be the payoff allocation received by each player?". In conclusion, symmetry simplifies both the encoding and the solving of a PFG when we are aiming for fairness. But, what if we are instead interested in stability? After all, we are in the presence of self-interested players, and fairness alone will not prevent the players from deviating. Therefore, the consideration of stability is crucial if we seek to consider strategical behavior, i.e., self-interested players.

3.1. Preliminaries

In our setting, the solution to our problem must be a *stable outcome* of the game. Recall Definition 1.3.5 formalizing the notion of game outcomes, which composed of two parts: *(i)* the partition that will form $\tilde{\mathcal{P}}$, and *(ii)* the payoff allocation vector $\vec{\tilde{y}}$. To determine if an outcome is stable or not, we need to find what threatens its stability, and if it will be acted upon (this is, the plausibility of the threat). In cooperative game theory with self-interested players, threats to stability are often deviations, i.e., players leaving or joining coalitions in the hope of a better individual payoff. In this thesis, we use the term *neighbourhood* to designate the set of deviations that are considered as threats to stability according to a solution concept. We can trace a parallel with competitive game theory, where deviations are a change in strategy from one or more players, and the Nash Equilibrium is the set of strategies selected by the players such that none of them has incentive to unilaterally deviate. In this case, the neighbourhood is the set of unilateral deviations, and a unilateral deviation will be carried out if the player who commits it gains utility by doing so.

Consider a PFG $(\mathcal{N}, V)$ and its outcome $(\tilde{\mathcal{P}}, \vec{\tilde{y}})$. The outcome is stable if there is no incentive to deviate to other outcomes we want to prevent deviations to. Going forward, we will refer to the other outcomes we want to prevent deviations to as the *neighbourhood*. By extension, we call the partition $\tilde{\mathcal{P}}$ (resp. vector $\vec{\tilde{y}}$) of a stable outcome a stable partition (resp. stable payoff allocation vector).

In the following Section 3.2, we propose and explain a few partition stability concepts. Moreover, we formalize our definition of stable partition for general PFGs, and we restrict our attention to those maximizing SW for the SPFG case. Then, in Section 3.3, we present a simple payoff sharing methodology, this time taking full advantage of the fact that the focus of this work are games with symmetric players.

3.2. Stability constraints for partitions

Being interested in stability, we define a deviation neighbourhood for our embedded coalitions. Given a partition, what are the plausible or threatening partition transitions? Before discussing plausible transitions between any two partitions, we determined that a “natural” state in the game is the state without agreement, the singleton partition (SP). Coalitions have to be worked for to be formed. Thus, every stable partition candidate should improve upon the SP, that will *de facto* belong to their neighbourhood.

Basso et al. [4] define some concepts of stability that they find relevant in the context of competing firms that have the option of cooperating for transport and distribution of (timber-related) products. Those stability concepts are expressed through constraints. For the studied PFG, they introduce $\mathcal{F}_1$ (see Equation (3.1)), a set of constraints stipulating that the induced stable partition should have a total utility larger or equal than the sum of utilities of the players in the SP. They even go further on this notion by stating, with their set of constraints $\mathcal{F}_3$ (see Equation (3.3)), that each coalition in a stable partition should receive a payoff greater or equal to what its members would make if their payoffs from the SP were summed. Together, $\mathcal{F}_1$ and $\mathcal{F}_3$ are there to certify that coarser partitions are worth the extra work of making agreements.

But being worthy of making agreements is not enough for ensuring reasonable stability, in most cases. Players will also seek the maximization of their individual utilities, and so we could witness all sorts of deviations: a subset of players splitting from a coalition and creating a new coalition together, coalitions joining their forces, etc. In [4], they define $\mathcal{F}_2$ (see Equation (3.2)), a constraint set accounting for a coalition dividing itself into two parts of varying sizes: the payoff of the coalition should be greater or equal than the sum of the utilities of the two parts for it to be considered stable.

Applying $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ to partitions of a PFG constitutes an example of a partition neighbourhood modeling stability notions. We devised one additional kind of transition (deviation), also simple enough to be fairly plausible, that we denote by $\mathcal{F}_4$ (see (3.4)). We can say it is antagonist to $\mathcal{F}_2$, because $\mathcal{F}_4$ concerns the merging of two coalitions.

These are the four types of stability constraints used in our methodology, particularly, with the goal of showcasing later the benefits of the MIP representation of SPFGs that we propose in Chapter 4. Please note that these stability conditions can be applied for general PFGs, since we have not used the fact that players are symmetric neither their plausibility depends on this property. In this way, in the remainder of this thesis, a partition $\mathcal{P}$ of a PFG $(\mathcal{N},W)$ is called *stable* if the intersection of the following sets is non-empty:

$$\mathcal{F}_1(\mathcal{P}) : \sum_{\mathcal{C}\in\mathcal{P}} W(\mathcal{C},\mathcal{P}) \geq \sum_{i\in\mathcal{N}} W(i,SP) \tag{3.1}$$

$$\begin{aligned}\mathcal{F}_2(\mathcal{P}) : W(\mathcal{C},\mathcal{P}) \geq W(\mathcal{S},(\mathcal{P}-\mathcal{C},\mathcal{S},\mathcal{T})) + W(\mathcal{T},(\mathcal{P}-\mathcal{C},\mathcal{S},\mathcal{T}))\\ \forall \mathcal{C}\in\mathcal{P}, \forall \mathcal{Q}\in\Pi(\mathcal{C}) : |\mathcal{Q}|=2, \{\mathcal{S},\mathcal{T}\}=\mathcal{Q}.\end{aligned} \tag{3.2}$$

$$\mathcal{F}_3(\mathcal{P}) : W(\mathcal{C},\mathcal{P}) \geq \sum_{i\in\mathcal{C}} W(i,SP)\ \forall \mathcal{C}\in\mathcal{P} \tag{3.3}$$

$$\mathcal{F}_4(\mathcal{P}) : W(\mathcal{C},\mathcal{P}) + W(\mathcal{S},\mathcal{P}) \geq W(\mathcal{C}\cup\mathcal{S},(\mathcal{P}-\mathcal{C}-\mathcal{S},\mathcal{C}\cup\mathcal{S}))\ \forall \mathcal{C},\mathcal{S}\in\mathcal{P} : \mathcal{C}\neq\mathcal{S}. \tag{3.4}$$

For SPFGs, given a numerical partition, the number of Constraints (3.1)-(3.4) is $O(n^2)$ which is polynomial on the number of players. Moreover, for the case of SPFGs as discussed in Chapter 4, we design our algorithm to return the stable partition with maximum SW

$$\tilde{\mathcal{P}} = \max_{\mathcal{P}\in\mathcal{F}} \sum_{\mathcal{C}\in\mathcal{P}} W(\mathcal{C},\mathcal{P}), \tag{3.5}$$

where $\mathcal{F} := \Big\{\mathcal{P}\in\Pi(\mathcal{N}) : \mathcal{F}_1(\mathcal{P})\cap\mathcal{F}_2(\mathcal{P})\cap\mathcal{F}_3(\mathcal{P})\cap\mathcal{F}_4(\mathcal{P})\Big\}$. The optimization problem (3.5) can be infeasible since $\mathcal{F}$ can be empty. Nonetheless, when a solution exists, we will call its solution $\tilde{P}$ the *SW-maximizing partition*. Equivalently to our definition of stable partition, a partition of an SPFG is stable if it is in $\mathcal{F}$.

There are important real-world contexts, such as ride sharing among individuals [7] and risk sharing among wind producers [8], where a central planner coordinates the activities of the participating agents (players). In such cases, most works in the literature guide this coordination through the maximization of the SW. Moreover, a coalition structure maximizing SW allow us to reduce inefficiencies. We are aware that, in an asymmetric setting, using SW as a selection criterion among stable partitions can cause a lack of fairness. For instance, if one player regularly achieves greater payoff allocations in partitions that are not maximizing the SW, it means that the central planner would force this player to under-perform. However, in a symmetric setting, this issue is mitigated by the fact that if the chosen partition is the SP, the GC, or a partition with coalitions of same cardinality exclusively, coalitions will already receive equal payoffs. Moreover, if the chosen partition is not a partition where all coalitions are equal in size, coalitions can use extracoalitional payments, if they are not prohibited, to compensate between themselves until the desired level of fairness is reached.

Example 3.2.1 (Finding stability in a PFG using the $\mathcal{F}_1$ to $\mathcal{F}_4$ constraints)**.** In this example, we aim to show the application of our stability constrains, while demonstrating that the set $\mathcal{F}$ can lead to (stable) partitions with no benefit to the players with respect to their individual payoffs. Consider the PFG $(\{1,2,3,4,5\}, V)$ where only four partitions lead to coalitions with payoffs different from zero. We have:

- $V(\{1,2\},\{\{1,2\},3,\{4,5\}\}) = 4$
- $V(3,\{\{1,2\},3,\{4,5\}\}) = 2$
- $V(\{4,5\},\{\{1,2\},3,\{4,5\}\}) = 3$
- $V(\{1,2\},\{\{1,2\},3,4,5\}) = 2$
- $V(3,\{\{1,2\},3,4,5\}) = 2$
- $V(4,\{\{1,2\},3,4,5\}) = 4$
- $V(5,\{\{1,2\},3,4,5\}) = 4$
- $V(\{1,2,3\},\{\{1,2,3\},4,5\}) = 7$
- $V(4,\{\{1,2,3\},4,5\}) = 1$
- $V(5,\{\{1,2,3\},4,5\}) = 1$
- $V(\{1,2,3\},\{\{1,2,3\},\{4,5\}\}) = 4$
- $V(\{4,5\},\{\{1,2,3\},\{4,5\}\}) = 5$
- Remaining embedded coalitions have the partition value function equal to 0.

We can see that no partition with a total payoff strictly greater than zero is stable according to our definition (i.e., the $\mathcal{F}_1$ to $\mathcal{F}_4$ constraints):

- Partition $\{\{1,2\},3,\{4,5\}\}$ does not respect $\mathcal{F}_2$, because if $\{4,5\}$ splits into two, players 4 and 5 would make 4 each, while they make only 3 together.
- Partition $\{\{1,2\},3,4,5\}$ does not respect $\mathcal{F}_4$, because if coalition $\{1,2\}$ merges with player 3, they would make 7 together, while they are only making 2 each separately.
- Partition $\{\{1,2,3\},4,5\}$ does not respect $\mathcal{F}_4$, because if player 4 joins player 5, they would make 5 together, while they are only making 1 each separately.
- Partition $\{\{1,2,3\},\{4,5\}\}$ does not respect $\mathcal{F}_2$, because if $\{1,2,3\}$ splits into coalitions $\{1,2\}$ and 3, they would make respectively 4 and 2, while they make only 4 together.

Figure 3.1 summarizes the transitions described above. However, the stable partitions set $\mathcal{F}$ is not empty! For instance, the partition $\mathcal{P} = \{\{1,5\},2,\{3,4\}\}$ belongs to $\mathcal{F}$, because:

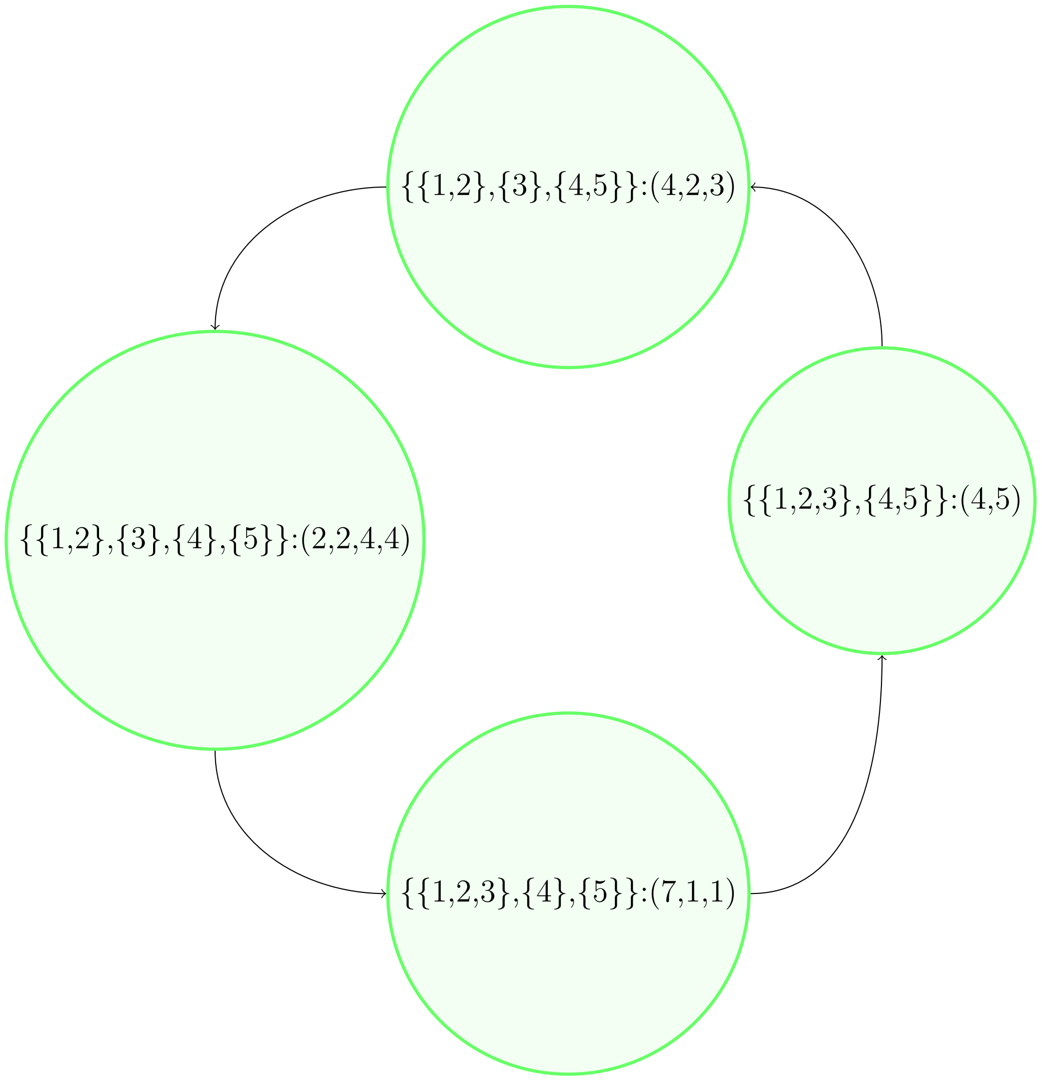


Fig. 3.1. Diagram describing the advantageous partition transitions. For each node, in the left we have the partition, and in the right the vector of payoffs for the associated embedded coalitions.

- All the coalitional payoffs are equal to 0;
- Any merging of two coalitions also leads to null coalitional payoffs, which makes $\mathcal{F}_4(\mathcal{P})$ hold;
- Any splitting of any coalition into two also leads to null coalitional payoffs, which makes $\mathcal{F}_2(\mathcal{P})$ hold;
- SP has null coalitional payoffs, so constraints $\mathcal{F}_1(\mathcal{P})$ and $\mathcal{F}_3(\mathcal{P})$ are guaranteed to hold.

Solving Problem (3.5) would thus return a partition that presents no advantage over any of the four non-zero coalitional payoff partitions. This partition is stable according to our definition, but it is not "efficient" in the sense that it would be possible to strictly improve the utility for some sets of players since there are non-zero partition value embedded coalitions.

If stability is itself the end goal of a study, this non-efficient stable partition would be appropriate. If we instead prefer to primarily focus on enabling the players to achieve some gains, restricting the candidate partition space to a set of embedded coalitions satisfying some minimum payoff value beforehand would be wise. This minimum payoff would need to be formalized and, possibly, defined according to the real-world situation at hand. In our four non-zero coalitional payoff partition example, this would encourage the players to seek another kind of stability – or even to rely more on fairness or side payments – in order to finish the game with profits. In summary, this example demonstrates that the context of the game is very important to define a satisfying solution concept.

3.2.1. PFG with positive and negative externalities

In Chapter 1, we defined PFGs with positive (see Definition 1.3.8) and negative (see Definition 1.3.9) externalities. Such games usually have an underlying payoff structure that makes the finding of a stable partition more systematic and logical.

Theorem 1 (PFG with positive externalities)**.** A PFG with positive externalities has exactly one stable partition which is the grand coalition (GC).

Proof. In a PFG with positive externalities, players' associations are beneficial to all the coalitions. With the exception of the GC, all other partitions have more than one coalition, and thus there is always incentive for them to merge, according to $\mathcal{F}_4$. Hence, the GC, where every player agrees to collaborate with every other player, is the most beneficial partition and is stable. $\square$

Theorem 2 (PFG with negative externalities)**.** A PFG with negative externalities has exactly one stable partition which is the SP.

Proof. In a PFG with negative externalities, players' associations are detrimental to all the coalitions. Except for the SP, all other partitions have at least one coalition with at least two players. Thus, the latter has incentive to deviate by splitting, according to $\mathcal{F}_2$. Hence, the SP, where every player is alone in their partition, is the most beneficial partition and is stable. $\square$

3.3. Payoff sharing

We conceive that the determination of a game outcome is done in two steps: first, the computation of $\tilde{\mathcal{P}}$ solving Problem (3.5), and second, the allocation of payoffs. Symmetry motivates this 2-step design choice, since we know that a subset of players from a coalition will be as tempted to depart from it as any other subset of the same size. Hence, the overall stability of a coalition within its neighbourhood will not be affected by diverging opinions from its members: they will all share the same view on the coalition. A natural action to take in an SPFG payoff sharing problem is to split the payoff of a coalition equally among its members. However, with the intention of targeting mainly stability, we decided to not enforce the equal allocation. Each player payoff will be constrained to be greater or equal to the utility of the player in the deviation neighbourhood.

We detailed in Chapter 2 the core, a solution concept which seeks stability by validating payoff allocations that are both efficient and coalitionally rational. We take inspiration from those two criteria to design our allocation constraints. First, we have efficiency:

$$\sum_{i=1}^{n} \vec{y}_i = \sum_{\mathcal{C} \in \tilde{\mathcal{P}}} W(\mathcal{C}, \tilde{\mathcal{P}}). \tag{3.1}$$

Second, we have a slightly restrained version of coalitional rationality:

$$\sum_{i \in \mathcal{S}} \vec{y}_i \geq W(\mathcal{S}, (\tilde{\mathcal{P}} - \mathcal{C}, \mathcal{C} \setminus \mathcal{S}, \mathcal{S})) \; \forall \mathcal{S} \subseteq \mathcal{C} \in \tilde{\mathcal{P}}. \tag{3.2}$$

In this version, we do not consider all subsets $\mathcal{S} \subseteq \mathcal{N}$, as done in the original definition of the core, because at this stage we are not looking for additional agreements to be made. Thus, the deviating players are made such that they come from the same coalition, i.e. $\mathcal{S} \subseteq \mathcal{C}$.

There is no guarantee on the existence of a payoff allocation vector $\vec{y}$ satisfying the efficiency and coalitional rationality constraints. Thus, for the payoff sharing, we strive to find the *most stable* sharing by starting with the efficiency constraint and gradually adding coalitional rationality constraints. The latter are added in increasing order of the size of the deviating coalition $\mathcal{S}$. Those constraints are added simultaneously for each embedded coalition $(\mathcal{C}, \mathcal{P})$. This is because we assume that small-sized coalition deviations are more likely to happen than large-sized coalition deviations because they require less coordination from the deviating players. If the size of $\mathcal{S}$ can go all the way to $\max_{\mathcal{C} \in \tilde{\mathcal{P}}} |\mathcal{C}|$ without making the set of feasible payoff allocations empty, it means the solutions it contains are robust with respect to the neighbourhood defined by us. Else, if one increment of the maximal size of the deviating coalition $\mathcal{S}$ empties the set of stable payoff allocations, we backtrack to the previous maximal size and content ourselves with one of its elements. This will be a weaker solution, but it will provide us some degree of stability according with the last size of $\mathcal{S}$ for which there was a feasible allocation vector. By following this procedure, we are certain to always obtain a payoff sharing vector. The worst-case scenario would be that we are given a solution that respects the efficiency constraint, but it cannot integrate any coalitional rationality constraint to the model without emptying the set of feasible payoff allocations.

At last, those two criteria of efficiency and coalitional rationality say nothing on coalitions having to keep their utility to themselves. It means the algorithm might return a solution that contains extracoalitional exchanges of utility. If we want to prevent those exchanges from happening, we can add an anti-extracoalitional exchange constraint:

$$\sum_{i \in \mathcal{C}} \vec{y}_i = W(\mathcal{C}, \tilde{\mathcal{P}}) \; \forall \mathcal{C} \in \tilde{\mathcal{P}}. \tag{3.3}$$

The convenience of having an MILP-representation of the partition function W will be demonstrated in the following chapters, when using our definition of stable partition and of *most stable* payoff allocation as defined in this chapter.

Chapter 4

Integer programming representation

Recall from Chapter 1 that solving a PFG means answering to the questions about how players partition themselves and how they share (allocate) payoffs. In Chapter 3, where we start to turn our attention to SPFGs, we propose solutions to these two questions by modeling certain aspects of strategical behavior by the players. Indeed, it is the behavior of each player that renders the answers to these questions more complex, as players are driven by the goal of maximizing their individual gains. Hence, these two problems are naturally related to optimization. This brings us to the heart of this thesis: we are interested in designing a representation of PFGs that is compatible with MILP tools for solving a game, all because they are a very powerful modeling framework and there are solvers capable of efficiently solving MILPs. To this end, we only need to find a concise and compact way to express a PFG using an MILP-formulation, and then let the solvers work for us. As a first step in this research direction, we decided to keep our focus on the representation and solving of SPFGs, as they are expected to be much simpler because of their naturally more compact representation (recall Table 1.1), and their payoff sharing made uncomplicated due to indistinct players. We also anticipated that SPFGs could be a good starting point to showcase the usefulness of applying integer programming on PFGs.

4.1. SPFG representation

In this section, we show that we can represent a family of SPFGs through an MILP-formulation of size bounded by a polynomial in the number of players n. The main ingredient of the formulation is the definition of an $n \times n$ matrix of parameters. Of course, the expressiveness of this representation lessens as n grows, and thus as the difference between PN(n) and n^2 accentuates. Nevertheless, the advantages of this framework are: *(i)* a compact representation, *(ii)* the compatibility with the integer programming paradigm, which allows solving the game with off-the-shelf solvers, and *(iii)* its promising computational results, even when used to approximate SPFGs, as we will see in Chapter 5.

4.1.1. Variables and constraints

The main idea behind our representation is to have three matrices whose product gives the partition function values for a specific partition. The first matrix $\mathbf{X}$ encodes the coalition formation, i.e., how the players partition themselves. The matrix $\mathbf{X}$ is a variable for the MILP-formulation. The second (auxiliary) matrix $\mathbf{M}$ plays the role of the intermediary between $\mathbf{X}$ and the third matrix $\mathbf{A}$ by ensuring that each embedded coalition gets the right payoff from $\mathbf{A}$. The third matrix $\mathbf{A}$ encodes information of the SPFG partition function values and is not a variable, but an input parameter. Next, we go into the details for the description of $\mathbf{X}$ and $\mathbf{M}$, and in Section 4.1.2, we describe $\mathbf{A}$.

The first set of variables is given by the binary square matrix $\mathbf{X} \in \{0{,}1\}^{n\times n}$. Rows represent players and columns represent coalitions indexed from 1 to n. Therefore, $\mathbf{X}_{ij}$ is 1 if player i is in coalition j, and 0 otherwise.

Example 4.1.1 (Matrix $\mathbf{X}$ expressing various partitions for a game with four players)**.** Below are possible representations of the singleton partition (SP):

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \text{ or } \mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Next, we present possible representations of a γ-partition with two players in its largest coalition, here players 2 and 3:

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{ or } \mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The grand coalition (GC) can be expressed as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{ or } \mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We can already see that each row must have exactly one entry equal to 1, i.e., $\mathbf{X}$ provides a partition of the players. Moreover, we remark that in Example 4.1.1, there are multiple possibilities on how to express a partition with a matrix. More precisely, the variations result from the fact that we are trying to represent a set, but sets are non-ordered, so all column permutations are valid. This forces us to add constraints on the order in which the columns present themselves in the matrix to achieve uniqueness. We add two uniqueness-related constraints: *(i)* the coalitions are sorted in descending order of their cardinality, i.e., the columns with the most ones come first, and *(ii)* the coalitions of same size are further sorted by their lowest player indexes. Hence, we will formulate constraints on $\mathbf{X}$ which will enforce the following: 1) every row has exactly one entry 1 so that each player belongs exactly to one coalition, 2) coalitions (rows) are ordered from the largest to the smallest, and 3) if two coalitions have the same cardinality, the player indexes are used as a tiebreaker.

To write them down in our MILP, we need an auxiliary binary vector $\vec{q} \in \{0,1\}^{n-1}$, an indicator that flags a 1 at index j if the coalitions from column j and $j+1$ in $\mathbf{X}$ are of the same cardinality, and that is 0 otherwise.

Example 4.1.2 (Vector $\vec{q}$ works as an indicator for coalitions of the same size that are adjacent in $\mathbf{X}$)**.** Here are some example of $\vec{q}$ given $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies \vec{q}^T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \implies \vec{q}^T = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \implies \vec{q}^T = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}.$$

With variables $\mathbf{X}$ and $\vec{q}$ defined, we can present our constraints forcing the representation of a partition through $\mathbf{X}$ to be unique:

$$\sum_{j=1}^{n} \mathbf{X}_{ij} = 1 \qquad \forall i \in \{1, ..., n\} \tag{4.1}$$

$$\sum_{i=1}^{n} \mathbf{X}_{ij} \geq \sum_{i=1}^{n} \mathbf{X}_{i(j+1)} \qquad \forall j \in \{1,..., n-1\} \tag{4.2}$$

$$\sum_{i=1}^{n} \left(\mathbf{X}_{ij} - \mathbf{X}_{i(j+1)}\right) \geq 1 - \vec{q}_j \qquad \forall j \in \{1,..., n-1\} \tag{4.3}$$

$$\begin{bmatrix}2^{n-1} & 2^{n-2} & ... & 2^0\end{bmatrix} \times \mathbf{X}_{*j} \geq \begin{bmatrix}2^{n-1} & 2^{n-2} & ... & 2^0\end{bmatrix} \times \mathbf{X}_{*j+1} - \bar{L}(1-\vec{q_j}) \quad \forall j \in \{1,...,n-1\}, \tag{4.4}$$

$$\mathbf{X} \in \{0,1\}^{n \times n}, \quad \vec{q} \in \{0,1\}^{n-1}, \tag{4.5}$$

where $\bar{L}$ is a very large number. These constraints allow a partition to be uniquely defined by a $\mathbf{X}$. Constraints (4.1) force each player to be in one coalition. Constraints (4.2) sort coalitions based on their size, in decreasing order. Constraints (4.3) ensure that $\vec{q_j} = 1$ when two coalitions of the same cardinality are adjacent in indices j and $j+1$. Constraints (4.4) are "relaxed" when $\vec{q_j} = 0$, but "activated" $\vec{q_j} = 1$, to enforce sorting based on the player indices. Indeed, Constraints (4.3) to (4.4) model the following implications:

$$\vec{q_j} = 0 \implies \sum_{i=1}^{n} \mathbf{X}_{ij} > \sum_{i=1}^{n} \mathbf{X}_{i(j+1)} \ \forall j\{1,...,n-1\} \tag{4.6}$$

$$\vec{q_j} = 1 \implies \sum_{i=1}^{n} \mathbf{X}_{ij} = \sum_{i=1}^{n} \mathbf{X}_{i(j+1)} \ \forall j\{1,...,n-1\}. \tag{4.7}$$

When $q_j = 1$, the respective Constraint (4.4) is "activated", forcing the ordering of coalitions of the same size by player labels since $2^k > \sum_{i=0}^{k-1} 2^i, \forall k \in \mathbb{N}$. Finally, Constraints (4.5) ensure that the variables are binary.

Let us now introduce the binary rectangular $\mathbf{M} \in \{0,1\}^{(n+1)\times n}$. Each row represents the number of players in a coalition (which can be zero), and columns represent the coalitions. Thus, an entry $\mathbf{M}_{ij}$ equals 1 if coalition j has a cardinality of $n-i+1$, and 0 otherwise.

Example 4.1.3 (Matrix $\mathbf{M}$ represents the sizes of the coalitions of a partition.)**.** Consider an SPFG with 10 players and the numerical partition $\mathcal{P}^{num} = \{4,3,1,1,1\}$, i.e., there is a coalition with 4 players, one with 3 players, and the remaining players are by themselves. Then, $\mathbf{M}$ is

$$\begin{array}{c c}
\text{Coalition size} & \\
\begin{array}{c}10\\9\\8\\7\\6\\5\\4\\3\\2\\1\\0\end{array} &
\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&1&1&1&1&1
\end{bmatrix}
\end{array}$$

To represent the described $\mathbf{M}$, we need the following constraints: 1) each column has exactly one entry equal to 1, as a coalition cannot have two cardinalities, and 2) this non-zero entry must be placed in row i such that the cardinality of coalition j as written in $\mathbf{X}_{*}j$ equals $n-i+1$:

$$\sum_{i=1}^{n+1} \mathbf{M}_{ij} = 1 \qquad \forall j \in \{0,...,n\} \tag{4.8}$$

$$\begin{bmatrix}1 & 1 & ... & 1 & 1\end{bmatrix}_{1\times n} \times \mathbf{X} = \begin{bmatrix}n & n-1 & ... & 1 & 0\end{bmatrix}_{1\times n+1} \times \mathbf{M} \tag{4.9}$$

$$\mathbf{M} \in \{0,1\}^{(n+1)\times n}. \tag{4.10}$$

Based on Constraints (4.8)-(4.10), it is clear that $\mathbf{M}$ is a variable entirely determined with respect to $\mathbf{X}$. Consequently, we can see $\mathbf{X}$ as the sole "true" variable of this model with $\vec{q}$ and $\mathbf{M}$ as auxiliary variables. Still, $\mathbf{M}$ is crucial to our representation. Recall that our goal is to describe SPFGs, this is, their partition functions with MILP-representations. In the next section, we will introduce the second matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ that will provide information on the partition function values. If we limit our MILP-representation of the partition function of an SPFG to $\mathbf{X}$ and $\mathbf{A}$, e.g., coalition k values $\sum_{j=1}^{n} X_{jk} \left(\sum_{i=1}^{n} A_{ij}\right)$, then symmetry would not be necessarily guaranteed, i.e., coalitions of same cardinality could have different payoffs (values). Under a symmetric game, this is unacceptable, because the result would be different based on the order in which the players are labeled. This motivates the introduction of our second matrix, $\mathbf{M}$.

4.1.2. Parameters and MILP-representation

Our MILP-representation does not have a wide array of parameters. Two suffice to set it up. The first parameter is the number of players n, while the second is the matrix describing the information on the partition function, $\mathbf{A} \in \mathbb{R}^{n\times n}$. Since we have restricted ourselves to an MILP-representation, next we describe our formulation of the partition function stressing that 1) it is linear in the decision variables, 2) it guarantees symmetry, i.e., the partition function value only depends on the numerical partition and cardinality of the coalition under evaluation, and 3) empty coalitions have null payoff. On the latter point note that our representation $\mathbf{X}$ can have coalitions with zero players (e.g., the GC in Example 4.1.1), since we insist that $\mathbf{X}$ always has fixed dimensions of $n \times n$. To comply with empty coalitions, we need to append a column of zeros to $\mathbf{A}$ which we denote by $[\mathbf{A}|\mathbf{0}]$. With this new form, we ensure that empty coalitions always get a null payoff when we define the payoffs as follows:

$$\vec{1}^{T} \times \mathbf{X} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}. \tag{4.11}$$

In particular, entry ℓ of the vector given by the expression above provides the payoff (partition function value) of coalition ℓ

$$M_{n+1,\ell} \times 0 + \sum_{k=1}^{n} M_{k\ell} \left(\sum_{j=1}^{n} A_{jk} \left(\sum_{i=1}^{n} X_{ij} \right) \right),$$

so if the cardinality of coalition ℓ is 0, then $M_{k\ell} = 0$ for $k = 1, \ldots, n$, and $M_{n+1,\ell} = 1$. In simple words, we first obtain the $(n+1)$-dimensional vector by multiplying a row vector containing, in decreasing order, the cardinalities of the coalitions under the current partition $\mathbf{X}$ with $[\mathbf{A}|\mathbf{0}]$, and second, we obtain the payoff of coalition ℓ by keeping from that vector the entry corresponding to the cardinality of that coalition. In this way, we ensure that coalitions with the same cardinality have the same payoff (partition function value).

Expression (4.11) is non-linear. This formulation of the payoffs for each coalition contains bilinear terms between entries of the binary matrices $\mathbf{X}$ and $\mathbf{M}$. The recent version of the MIP solver we use, Gurobi [33], can cope with bilinear (quadratic) MIP formulations without manual linearization. The manual linearization, using the McCormick envelope, will give an exact linearization in this context because the variables that have been multiplied are binary (see Appendix A for more details about our manual linearization). Nonetheless, we can claim that the formulation described is MILP-representable. Hence, in the remainder of this thesis we define an SPFG as *MILP-representable* as follows:

Definition 4.1.1. *An SPFG* $(\mathcal{N}, W)$ *is MILP-representable if there is a matrix* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *such that for any partition* $\mathbf{X}$ *satisfying Constraints* (4.1)-(4.5) *and its associated numerical representation* $\mathbf{M}$ *satisfying Constraints* (4.8)-(4.10)*, the partition function payoffs for each embedded coalition of* $\mathbf{X}$ *are given by Expression* (4.11)*, i.e.,*

$$(W(\mathbf{X}_{*1}, \mathbf{X}), W(\mathbf{X}_{*2}, \mathbf{X}), \ldots, W(\mathbf{X}_{*n}, \mathbf{X})) = \vec{1}^T \times \mathbf{X} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}.$$

Of course, other SPFGs could be represented by MILP, if other design choices were taken. This definition only concerns our framework.

Example 4.1.4 (Matrix $\mathbf{A}$ with positive/negative externalities for an SPFG with 5 players.)**.** Firstly, an example of what $\mathbf{A}$ looks like for a game with positive externalities:

$$\mathbf{A}_{pos} = \begin{bmatrix} 100 & 80 & 60 & 40 & 20 \\ 0 & 30 & 25 & 20 & 15 \\ 0 & 0 & 15 & 10 & 5 \\ 0 & 0 & 0 & 5 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

And, for a game with negative externalities:

$$\mathbf{A}_{neg} = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 0 & 4 & 5 & 6 & 7 \\ 0 & 0 & 6 & 7 & 8 \\ 0 & 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}.$$

To highlight the positive/negative externalities, let us look at the case where the coalition structure is $(2, 2, 1)$, so

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

We have the partition function payoffs for each embedded coalition of $\mathbf{X}$ given by:

$$\begin{aligned} &\vec{1}^T \times \mathbf{X} \times [\mathbf{A}_{pos}|\mathbf{0}] \times \mathbf{M} \\ &= \vec{1}^T \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 100 & 80 & 60 & 40 & 20 & 0 \\ 0 & 30 & 25 & 20 & 15 & 0 \\ 0 & 0 & 15 & 10 & 5 & 0 \\ 0 & 0 & 0 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 130 & 130 & 75 & 0 & 0 \end{bmatrix}, \end{aligned}$$

in case of positive externalities, and by:

$$\begin{aligned} &\vec{1}^T \times \mathbf{X} \times [\mathbf{A}_{neg}|\mathbf{0}] \times \mathbf{M} \\ &= \vec{1}^T \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 0 \\ 0 & 4 & 5 & 6 & 7 & 0 \\ 0 & 0 & 6 & 7 & 8 & 0 \\ 0 & 0 & 0 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 29 & 29 & 34 & 0 & 0 \end{bmatrix}, \end{aligned}$$

in case of negative externalities. Now, let us suppose that the first coalition of players would like to split as $(2, 1, 1, 1)$. This would change matrices $\mathbf{X}$ and $\mathbf{M}$ into:

$$\mathbf{X}_{split} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{M}_{split} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We can then compute again the partition function payoffs for each embedded coalition of $\mathbf{X}_{split}$ given by:

$$\begin{aligned} &\vec{1}^T \times \mathbf{X}_{split} \times [\mathbf{A}_{pos}|\mathbf{0}] \times \mathbf{M}_{split} \\ &= \begin{bmatrix} 115 & 62 & 62 & 62 & 0 \end{bmatrix}, \end{aligned}$$

in case of positive externalities, and by:

$$\begin{aligned} &\vec{1}^T \times \mathbf{X}_{split} \times [\mathbf{A}_{neg}|\mathbf{0}] \times \mathbf{M}_{split} \\ &= \begin{bmatrix} 31 & 36 & 36 & 36 & 0 \end{bmatrix}, \end{aligned}$$

in case of negative externalities. We can already see that with $\mathbf{A}_{pos}$, players are all disadvantaged by the split. Initially, a coalition of cardinality 2 was receiving 130, but after the split it is only receiving 115, while the utility of a coalition of cardinality 1 dropped from 75 to 62. However, with $\mathbf{A}_{neg}$, we observe the opposite. Initially, a coalition of cardinality 2 was receiving 29, but after the split it is receiving 31, while a coalition of cardinality 1 saw its payoff increase from 34 to 36.

Finally, if instead of a split, we have a merge where player 2 joins the first coalition to create the coalition structure (3,2,1), we obtain:

$$\mathbf{X}_{merge} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{M}_{merge} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

We can then compute again the partition function payoffs for each embedded coalition of $\mathbf{X}_{merge}$ given by:

$$\begin{aligned} &\vec{1}^T \times \mathbf{X}_{split} \times [\mathbf{A}_{pos}|\mathbf{0}] \times \mathbf{M}_{merge} \\ &= \begin{bmatrix} 230 & 160 & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

in case of positive externalities, and by:

$$\begin{aligned}&\vec{1}^T \times \mathbf{X}_{merge} \times [\mathbf{A}_{neg}|\mathbf{0}] \times \mathbf{M}_{merge}\\&= \begin{bmatrix}22 & 27 & 0 & 0 & 0\end{bmatrix},\end{aligned}$$

in case of negative externalities. When the partition has a coarser granularity, we can see that with $\mathbf{A}_{pos}$, players are advantaged compared to $\mathbf{A}$. A coalition of cardinality 2 was receiving 130 before, and is now receiving 160. Plus, the new coalition of cardinality 3 receives 230, which is more than what the two joining parties were making separately initially(130 and 75). Again, with $\mathbf{A}_{neg}$, the result goes the opposite way. A coalition of cardinality 2 was receiving 29 before, and is now receiving only 27, while the payoff of the new coalition of cardinality 3 is 22, which is less than what the two joining parties were making separately initially (29 and 34).

In a nutshell, this example displays simultaneously how we can compute the partition function values and how a game with positive/negative externalities reacts to coalitions merging and splitting.

4.1.3. Solving MILP-representable games

Now that we have our MILP-representation (4.1)-(4.5), (4.8)-(4.10) with payoffs given by Expression (4.11), we can demonstrate its usefulness to solve SPFGs.

In this thesis, as specified in Chapter 3, we are targeting a stability-oriented solution concept. Moreover, remembering Problem (3.5), if there are multiple partitions in the stable partition set, we select the partition maximizing the SW, i.e., the sum of the coalition values (payoffs).

Using the variables and parameters described in Sections 4.1.1 and 4.1.2, given an MILP-representable game, it is straightforward to model the problem of finding the partition maximizing the SW:

$$\begin{aligned}(SW_{max}) \qquad \max \quad & \vec{1}^T \times \mathbf{X} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M} \times \vec{1} \qquad (4.12)\\ s.t. \quad & \text{Constraints } (4.1) - (4.5)\\ & \text{Constraints } (4.8) - (4.10),\end{aligned}$$

where the objective (4.12) is the so-called SW. Once we input the model (SW_{max}) to a solver and solve it to optimality, matrix $\mathbf{X}$ identifies the partition that maximizes the SW. To know the payoff of each coalition, we can keep the first part $\vec{1}^T \times \mathbf{X} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}$ of the objective function. This row vector will indicate the partition function payoff of coalition j at index j.

Recall that Problem (3.5) imposes the restriction of optimizing SW among the stable partitions. Thus, in order to integrate the stability constraints, we devise an algorithm with the following steps:

(1) Create the mathematical program in (4.12) and denote it by (SW_{max}).
(2) Using an off-the-shelf solver, solve (SW_{max}) and save the corresponding optimal partition $\mathbf{X}^*$.
(3) Generate the stability Constraints (3.1)-(3.4) for the partition $\mathbf{X}^*$; see Appendix B. If $\mathbf{X}^*$ verifies those constraints, i.e., it is stable, return $\tilde{\mathbf{X}} = \mathbf{X}^*$.
(4) Add to (SW_{max}) a constraint banning $\mathbf{X}^*$ and go to step (2).

In step (4), we add the following banning constraint when partition $\mathbf{X}^*$ is not stable:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\Big((2\mathbf{X}^*_{ij} - 1)\mathbf{X}_{ij} + 1 - \mathbf{X}^*_{ij}\Big) \leq n^2 - 1. \tag{4.13}$$

This constraint sums the number of entries of the same indices that are equal in $\mathbf{X}$ and $\mathbf{X}^*$. If this number equals n^2, it means the two matrices are identical and we want to prevent that. This is why we restrict the sum to $n^2 - 1$.

Once this first task is performed, i.e., we have a stable partition, our algorithm has a second task to achieve: the allocation of payoffs as described in Section 3.3. In particular, recall from that section that we describe an iterative procedure to find the *most stable* allocation. In this way, given the chosen partition $\tilde{\mathbf{X}}$ (output of the first task), at iteration k, we find a vector $\vec{y}$ satisfying the following constraints:

$$\sum_{i=1}^{n} \vec{y}_i = \vec{1}^T \times \tilde{\mathbf{X}} \times [\mathbf{A}|\mathbf{0}] \times \tilde{\mathbf{M}} \times \vec{1} \tag{4.14a}$$

$$\begin{gathered}\mathbf{X}^{split} : \Big(\exists! j : \mathbf{X}^*_{*j} \notin \mathbf{X}^{split}, \vec{1}^T \times \mathbf{X}^*_{*j} \geq 2\Big) \wedge \\ \Big(\exists! i, \exists! k : \mathbf{X}^{split}_{*i} \notin \mathbf{X}^*, \mathbf{X}^{split}_{*k} \notin \mathbf{X}^*, \mathbf{X}^{split}_{*i} + \mathbf{X}^{split}_{*k} = \mathbf{X}^*_{*j}\Big),\end{gathered} \tag{4.14b}$$

$$\begin{gathered}\sum_{i:\mathbf{X}^{neigh}_{ij}=1}^{n} \vec{y}_i \geq \Big(\vec{1}^T \times \mathbf{X}^{neigh} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^{neigh}\Big)_j \, \forall j : \mathbf{X}^{neigh}_{*j} \notin \tilde{\mathbf{X}}, \\ \forall \mathbf{X}^{neigh} \in \mathbf{X}^{split} : \vec{1}^T \times \mathbf{X}^{neigh}_{*j} \leq k.\end{gathered} \tag{4.14c}$$

Operator $\notin$ between a column and a matrix means there is no such column in the matrix. In summary, the algorithm for this second task is as follows:

(1) Create a feasibility problem (P) with the efficiency Constraint (4.14a).
(2) Make $k = 0$ and add to (P) the coalitional rationality Constraints (4.14c) with respect to k.
(3) While $k < n$ and there is $\vec{y}$ solving (P):
    (a) Save a feasible $\vec{y}$ to (P)

(b) Make $k = k+1$ and add to (P) the coalitional rationality constraints (4.14c) with respect to k.

(4) Return $\vec{\tilde{y}} = \vec{y}$.

We refer the reader to the thesis's repository [65] for more details on the implementation.

Example 4.1.5 (Numerical example)**.** Let us define an SPFG $(4, W)$ with:

$$\begin{aligned}
&W = \Big\{\\
&(4, \{4\}) \mapsto 16,\\
&(3, \{3, 1\}) \mapsto 9,\\
&(1, \{3, 1\}) \mapsto 3,\\
&(2, \{2, 2\}) \mapsto 4,\\
&(2, \{2, 1, 1\}) \mapsto 6,\\
&(1, \{2, 1, 1\}) \mapsto 1,\\
&(1, \{1, 1, 1, 1\}) \mapsto 2\\
&\Big\}.
\end{aligned}$$

This matrix $\mathbf{A}$ is an exact MILP-representation of the game:

$$\mathbf{A} = \begin{bmatrix} 4 & 3 & 4 & -1 \\ 0 & 0 & -2 & 6 \\ 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

With our two parameters defined, $n = 4$ and $\mathbf{A}$, we are now ready to optimize. There are five partitions in this case, and they get the following SW values:

$$\begin{aligned}
&SW = \Big\{\\
&(\{4\}, 16),\\
&(\{3, 1\}, 12),\\
&(\{2, 2\}, 8),\\
&(\{2, 1, 1\}, 8),\\
&(\{1, 1, 1, 1\}, 8)\\
&\Big\}.
\end{aligned}$$

The chosen partition is thus the GC, and we get the following variable values at the optimal state:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Hence, we know that the first coalition is the only one to receive a non-zero payoff, and that this payoff will be 16. The next step to perform is to introduce the stability constraints $\mathcal{F}$ described in section 3.2 to check if the chosen partition belongs to the stable partition set.

- $\mathcal{F}_1$: Since the chosen partition has a greater total payoff than the one of the SP ($16 \geq 8$), this constraint holds.
- $\mathcal{F}_2$: The neighbourhood of the chosen partition, which is the GC, is {{3,1},{2,2}}. The comparison with the first neighbour favors the GC ($W(3, \{3{,}1\}) + W(1,\{3{,}1\}) = 12 \leq 16$), and the comparison with the second neighbour does too ($W(2, \{2{,}2\}) + W(2,\{2{,}2\}) = 8 \leq 16$). This constraint holds.
- $\mathcal{F}_3$: Because the chosen partition is the GC, $\mathcal{F}_3$ is equivalent to $\mathcal{F}_1$. This constraint holds.
- $\mathcal{F}_4$: There are no neighbours in this case, so this constraint holds by default.

We find that yes, the chosen partition belongs to the stable partition set. The last step to perform is to find a payoff allocation vector that respects efficiency and coalitional rationality. We enforce the following constraints on $\vec{y}$:

$$\begin{aligned} \sum_{i=1}^{4} \vec{y}_i &= 16 \\ \vec{y}_i + \vec{y}_j + \vec{y}_k &\geq 9 \ \ \forall i,j,k \in \{1,2,3,4\} : i \neq j \neq k \neq i \\ \vec{y}_i + \vec{y}_j &\geq 4 \ \ \forall i,j \in \{1,2,3,4\} : i \neq j \\ \vec{y}_i &\geq 3 \ \ \forall i \in \{1,2,3,4\}. \end{aligned}$$

The set of feasible payoff allocations is not empty; it will contain $\{\vec{y} : \sum_{i=1}^{4} \vec{y}_i = 16, \vec{y}_i \geq 3 \forall i \in \{1,2,3,4\}\}$. In other words, to achieve stability, the players could receive each anything equal or above 3, as long as the sum of their payoffs is 16.

4.2. MILP-Representable Family of SPFG

First, we observe that the MILP-based representation motivated in Section 4.1 cannot be used to represent *every* SPFG. This can be shown by a straightforward counting argument. Every SPFG with n players is defined by $\mathrm{CN}(n-1)$ real numbers, i.e., a payoff associated with each embedded numerical coalition, while the matrix $\mathbf{A}$ is defined by n^2 real numbers. It is known that $\mathrm{PN}(n) \leq \mathrm{CN}(n-1) = \sum_{i=0}^{n-1} \mathrm{PN}(i)\ \forall n \geq 1$, because there are as many or more embedded numerical coalitions than there are partitions for a given n. We saw in Equation (1.4) a lower bound for $\mathrm{PN}(n)$ that follows an exponential growth such that $n^2 \ll \mathrm{PN}_{lb}(n) \leq \mathrm{PN}(n) \leq \mathrm{CN}(n-1)$, and hence it is guaranteed that there are SPFGs not representable in the form defined in Section 4.1. In this section, we characterize the MILP-representable SPFGs by giving an explicit construction when they are representable, and an approximation when they are not.

4.2.1. The matrix A

First we prove result that, without loss of generality, $\mathbf{A}$ can always be chosen as an upper-triangular matrix.

Proposition 4.2.1. *Let P_1 and P_2 be two SPFGs defined by matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ respectively. The SPFGs P_1 and P_2 are equal if*

$$[\mathbf{A}_1]_{ij} \quad = \quad [\mathbf{A}_2]_{ij} \qquad \forall\, i \leq j;\ i,j \in \{1,\ldots,n\}.$$

Proof. The proposition statement says that if all the elements above (and including) the principal diagonal of two matrices are the same, then they necessarily model the same SPFG. Let us show this.

When computing the payoffs of the embedded coalitions (Equation (4.11)), the partition with the least empty coalitions to be multiplied with column j of $\mathbf{A}$ is the γ-partition with $n-j+1$ players in its largest coalition. Thus, the last non-zero component of the γ-partition will be at index j for column j of $\mathbf{A}$. This implies that all the entries of $\mathbf{A}$ that are below the principal diagonal will be multiplied by zero. For the computations involving the upper-triangular part of $\mathbf{A}$, if we have

$$[\mathbf{A}_1]_{ij} \quad = \quad [\mathbf{A}_2]_{ij} \qquad \forall\, i \leq j;\ i,j \in \{1,\ldots,n\},$$

then for partitions $\mathbf{X}$, the embedded coalitions payoffs satisfy

$$\vec{1}^T \times \mathbf{X} \times [\mathbf{A_1}|\mathbf{0}] \times \mathbf{M} = \vec{1}^T \times \mathbf{X} \times [\mathbf{A_2}|\mathbf{0}] \times \mathbf{M},$$

i.e., every embedded numerical coalition has equal partition function values for P_1 and P_2. □

While we have said that the entries in the upper triangular part of the matrix $\mathbf{A}$ determine the SPFG that is being modeled, the theorem below shows that: *(i)* The set of all such MILP-representable SPFGs form a vector space. *(ii)* More importantly, the vector space is given as the range of a matrix, $\bar{\mathbf{Q}}^n$, which depends only on the number of players in the game. *(iii)* Further, the theorem also shows that solving a system of linear equations of the form $\bar{\mathbf{Q}}^n\vec{x} = \vec{b}$ identifies the matrix $\mathbf{A}$ that has to be used to model the given SPFG.

Theorem 4.2.2 (MILP-representation theorem)**.** *Given the number of players n, there exist a matrix $\bar{\mathbf{Q}}^n \in \mathbb{R}^{\mathrm{CN}(n-1) \times \frac{n(n+1)}{2}}$ such that the following holds: For each SPFG $(\mathcal{N}, W)$, there exists $\vec{\bar{b}} \in \mathbb{R}^{\mathrm{CN}(n-1)}$, such that the SPFG is MILP-representable if and only if $\bar{\mathbf{Q}}^n \vec{x} = \vec{\bar{b}}$ has a solution. Further, any solution to the system of linear equations can be directly interpreted as the entries of the matrix $\mathbf{A}$ after appropriate rearrangement.*

Proof. We construct matrix $\bar{\mathbf{Q}}^n$ by encoding every embedded numerical coalition $(c, \mathcal{P}^{num}) \in \mathcal{E}^{num}(\mathcal{N})$ in the $\mathrm{CN}(n-1)$ rows of $\bar{\mathbf{Q}}^n$ such that:

- Each row i of $\bar{\mathbf{Q}}^n$ will display $\mathcal{P}^{num}$ of $(c, \mathcal{P}^{num})$ following a decreasing order in the cardinality of the coalitions, i.e., $\forall d,e \in \mathcal{P}^{num}$ such that $\bar{\mathbf{Q}}^n_{ij} = d$ and $\bar{\mathbf{Q}}^n_{ik} = e$ we have $d < e \implies j < k$. We need $|\mathcal{P}^{num}|$ entries to do so. Unused entries are kept to 0.
- The non-zero entries of each row i of $\bar{\mathbf{Q}}^n$ are contiguous, i.e., $\bar{\mathbf{Q}}^n_{ij} = d \neq 0$ and $\bar{\mathbf{Q}}^n_{i\,j+1} = 0 \implies \forall k > j+1, \bar{\mathbf{Q}}^n_{ik} = 0$, while $\bar{\mathbf{Q}}^n_{ij} = d \neq 0$ and $\bar{\mathbf{Q}}^n_{i\,j-1} = 0 \implies \forall k < j-1, \bar{\mathbf{Q}}^n_{ik} = 0$.
- Each row i of $\bar{\mathbf{Q}}^n$ will display $\mathcal{P}^{num}$ of $(c, \mathcal{P}^{num})$ with the first non-zero entry starting at index $1 + (n-c) * (n-c+1)/2$, i.e., $\bar{\mathbf{Q}}^n_{ij} = d \neq 0$ and $\forall k < j, \bar{\mathbf{Q}}^n_{ik} = 0 \implies j = 1 + (n-c) * (n-c+1)/2$.
- If the row i of $\bar{\mathbf{Q}}^n$ encodes $(c, \mathcal{P}^{num})$, $\vec{\bar{b}}_i = W(c, \mathcal{P}^{num})$.

This construction allows us to say that if the SPFG is MILP-representable, $\bar{\mathbf{Q}}^n \vec{x} = \vec{\bar{b}}$ has a solution, and reciprocally. Furthermore, if we permute the rows of $\bar{\mathbf{Q}}^n$ (and of $\vec{\bar{b}}$ in the exact same fashion) such that $\bar{\mathbf{Q}}^n$ achieves the row echelon form, then

$$\mathbf{A}_{ij} = \vec{x}_{\frac{i(i-1)}{2}+j} \quad \forall i \leq j; i,j \in \{1,2,\ldots,n\}.$$

$\square$

Corollary 4.2.3. *The following are true.*

(1) The set of SPFGs with n players is a vector space of dimension of $\mathrm{CN}(n)$.

(2) The set of MILP-representable SPFGs with n players is a vector space of dimension $\frac{n(n+1)}{2}$.

(3) The set of MILP-representable SPFGs is smaller than the vector space we can represent with the $\frac{n(n+1)}{2}$ entries of matrix $\mathbf{A}$ that we use.

Second of all, the columns of $\mathbf{A}$ are treated independently by our model to get the payoff of any embedded coalition. Thus, what matters is the relationship between the elements of a column of $\mathbf{A}$. This ascertainment makes the analysis of $\mathbf{A}$ very convenient. For instance, to determine $\mathbf{A}$ given an SPFG $(\mathcal{N}, W)$, we just need to define for each column $n-i+1$ of $\mathbf{A}$ a system of equations $\mathbf{Q}\vec{v} = \vec{b}$ where:

- $\mathbf{Q} \in \mathbb{Z}_+^{\mathrm{PN}(n-i) \times (n-i+1)}$ is a matrix whose rows are all numerical partitions of n players with at least one coalition of cardinality i, ordered from the largest coalition to the smallest, and zero-padded to get exactly $n-i+1$ elements in each row.

- $\vec{b} \in \mathbb{R}^{\mathrm{PN}(n-i)}$ is a payoff vector for a coalition of cardinality i such that if partition $\mathbf{Q}_k$ forms, the payoff will be $\vec{b}_k$. Note that this vector is computed through the partition function W.
- $\vec{v} \in \mathbb{R}^{n-i+1}$ is column $\mathbf{A}_{*(n-i+1)}$ truncated to its last useful entry before the zero padding.

In this way, by interpreting each column of $\mathbf{A}$ as a vector of variables with its own system of linear equations, we can determine whether an SPFG is MILP-representable. The key to this problem is the rank of matrix $\mathbf{Q}$, as stated by Theorem 3.

Theorem 3 (Solution to a linear equation system [63])**.** Given a matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$ and a vector $\vec{b} \in \mathbb{R}^m$, let $\left[\mathbf{Q}|\vec{b}\right]$ be the augmented matrix of system $\mathbf{Q}\vec{v} = \vec{b}$. We have that

(1) if $\mathrm{rank}(\left[\mathbf{Q}|\vec{b}\right]) > \mathrm{rank}(\mathbf{Q})$, then the system admits no solution;

(2) if $\mathrm{rank}(\left[\mathbf{Q}|\vec{b}\right]) = \mathrm{rank}(\mathbf{Q}) = r$ and $r = n$, then the system admits a unique solution;

(3) if $\mathrm{rank}(\left[\mathbf{Q}|\vec{b}\right]) = \mathrm{rank}(\mathbf{Q}) = r$ and $r < n$, then the system admits an infinite number of solutions.

Now having concluded that the MILP-representability of an SPFG relies on the defined n matrices $\mathbf{Q}$ (one for each column of $\mathbf{A}$), we can underline that each of them can be immediately studied, as they only depend on the parameter n.

Next, we will show that all SPFGs with $n \leq 5$ are MILP-representable. As expected, due to our compact representation, there are not enough linear dependencies between rows of $\mathbf{Q}$ to guarantee exact MILP-representations for all SPFGs with $n > 5$. This will reveal that the first four columns of $\mathbf{A}$ (when they exist) never experience representation limitations, as there are more variables than equations constraining them. Finally, we will prove that the rank of $\mathbf{Q}$ for the systems of equations related to the reminding columns of $\mathbf{A}$ is either $n-i$ or $n-i+1$, depending on the combination of parameters n and i. This will enable us to characterize the families of SPFGs which are MILP-representable.

4.2.2. Characterization of MILP-representable SPFGs

Note that Theorem 3 allow us to determine the cases where an SPFG is MILP-representable through the study of the rank of the matrices $\mathbf{Q}$. We begin this study for SPFGs with five or less players due to its simplicity which enable us to become familiar with the notation and the construction of matrices $\mathbf{Q}$ for different n and $n-i+1$ (column index of $\mathbf{A}$) setups. This is shown in Table 4.1. From it, we conclude that any SPFG with five players or less is MILP-representable. Also, we can observe that, no matter the value of n, for column indexes of $\mathbf{A}$ going from 1 to 4, $\mathrm{rank}(\mathbf{Q}) = \mathrm{rank}(\left[\mathbf{Q}|\vec{b}\right]) =$ number of rows in $\mathbf{Q}$. This is always the case because the rows in $\mathbf{Q}$, once removed the zero padding, all have different lengths, and thus cannot be linearly dependent. The following lemma summarizes these findings:

n	column index of $\mathbf{A}$	$\left[\mathbf{Q}\vert\vec{b}\right]$	rank($\left[\mathbf{Q}\vert\vec{b}\right]$)	rank($\mathbf{Q}$)	Verdict
1	1	$\begin{bmatrix}1 & b_1\end{bmatrix}$	1	1	Unique solution
2	1	$\begin{bmatrix}2 & b_1\end{bmatrix}$	1	1	Unique solution
2	2	$\begin{bmatrix}1 & 1 & b_1\end{bmatrix}$	1	1	Infinity of solutions
3	1	$\begin{bmatrix}3 & b_1\end{bmatrix}$	1	1	Unique solution
3	2	$\begin{bmatrix}2 & 1 & b_1\end{bmatrix}$	1	1	Infinity of solutions
3	3	$\begin{bmatrix}2 & 1 & 0 & b_1\\ 1 & 1 & 1 & b_2\end{bmatrix}$	2	2	Infinity of solutions
4	1	$\begin{bmatrix}4 & b_1\end{bmatrix}$	1	1	Unique solution
4	2	$\begin{bmatrix}3 & 1 & b_1\end{bmatrix}$	1	1	Infinity of solutions
4	3	$\begin{bmatrix}2 & 2 & 0 & b_1\\ 2 & 1 & 1 & b_2\end{bmatrix}$	2	2	Infinity of solutions
4	4	$\begin{bmatrix}3 & 1 & 0 & 0 & b_1\\ 2 & 1 & 1 & 0 & b_2\\ 1 & 1 & 1 & 1 & b_3\end{bmatrix}$	3	3	Infinity of solutions
5	1	$\begin{bmatrix}5 & b_1\end{bmatrix}$	1	1	Unique solution
5	2	$\begin{bmatrix}4 & 1 & b_1\end{bmatrix}$	1	1	Infinity of solutions
5	3	$\begin{bmatrix}3 & 2 & 0 & b_1\\ 3 & 1 & 1 & b_2\end{bmatrix}$	2	2	Infinity of solutions
5	4	$\begin{bmatrix}3 & 2 & 0 & 0 & b_1\\ 2 & 2 & 1 & 0 & b_2\\ 2 & 1 & 1 & 1 & b_3\end{bmatrix}$	3	3	Infinity of solutions
5	5	$\begin{bmatrix}4 & 1 & 0 & 0 & 0 & b_1\\ 3 & 1 & 1 & 0 & 0 & b_2\\ 2 & 2 & 1 & 0 & 0 & b_3\\ 2 & 1 & 1 & 1 & 0 & b_4\\ 1 & 1 & 1 & 1 & 1 & b_5\end{bmatrix}$	5	5	Unique solution

Table 4.1. Our MILP-representation can express any SPFG composed of five or less players, as there is always at least one solution of elements to combine for each column of $\mathbf{A}$ to give an embedded coalition its correct partition function value. Note that each b_i in the table corresponds to a partition function value as defined in Section 4.2, e.g., b_2 in row $n = 4$ and column index of $\mathbf{A}$ equal to 4 is $W(1,\{2,1,1\})$.

Lemma 4.2.4. *Any SPFG with $n \leq 5$ is MILP-representable. Moreover, for any SPFG, we can always find a matrix $\mathbf{A}$ such that the payoffs of coalitions of size n, $n-1$, $n-2$ and $n-3$ are exact.*

But then, how good are the payoffs if we look at games with $n > 5$ and column indexes of $\mathbf{A}$ with $n - i + 1 > 4$? To answer this question, we need to find a more generalized approach to study those games. Two observations will allow us to do so. The first observation is as follows. Recall from Section 4.2.1 that in the matrix $\mathbf{Q}$, the first element in each row corresponds to the size of the largest coalition in a partition. Also, recall that the partition function value of a coalition of cardinality i is represented through the column $n-i+1$ of matrix $\mathbf{A}$. Now, once $i \geq \frac{n}{2}$, i will always appear as the first element in the rows of matrix $\mathbf{Q}$, since it will be the largest coalition. This allows us to handle this case separately from all other. In the analysis below, case 1 corresponds to when $i < \frac{n}{2}$ and case 2 corresponds to when $i \geq \frac{n}{2}$. In both these cases, we restrict to $n > 5$, because from 4.2.4, we already know that the SPFG can be exactly represented when $n \leq 5$. Secondly, γ-partitions have all a different number of non-zero elements, meaning that once they are written in vector form, the resulting vectors are linearly independent. Thus, they constitute a good starting point to find the rank of $\mathbf{Q}$.

Lemma 4.2.5. *For any SPFG with $n > 5$, there is an MILP-representation of the partition function value of an embedded coalition of cardinality $i \geq \frac{n}{2}$, if $n-i+1 > 4$ and $[\mathbf{Q}|\vec{b}]$ has rank $n-i$.*

Case 1: $n > 5$**,** $i \geq n/2$ **and the column index of matrix A is** $n-i+1 > 4$. We extract the γ-partitions from $\mathbf{Q}$ with respect to $n-i$ players to define its sub-matrix $\mathbf{Q}'$:

$$\mathbf{Q}' = \begin{bmatrix} i & 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\ i & 2 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 0 \\ i & 3 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & 0 \\ i & 4 & 1 & 1 & 1 & \dots & 1 & 0 & 0 & 0 \\ \vdots & & & & & & & & & \\ i & n-i-2 & 1 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ i & n-i-1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ i & n-i & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix}_{\in \mathbb{Z}_+^{(n-i)\times(n-i+1)}}$$

Note that if we exclude the first column, then each row is a γ-partition for the remaining $n-i$ players. We now prove that the rows of $\mathbf{Q}'$ are linearly independent. To this end we subtract from each row the next, starting from the top. The last row is then divided by $n-i$, and added to the previous rows.

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \\ i & 2 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 0 \\ i & 3 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & 0 \\ i & 4 & 1 & 1 & 1 & \dots & 1 & 0 & 0 & 0 \\ \vdots & & & & & & & & & \\ i & n-i-2 & 1 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ i & n-i-1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ i & n-i & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{Q'}_1 \leftarrow \mathbf{Q'}_1 - \mathbf{Q'}_2$$

$$\begin{bmatrix}0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1\\ 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0\\ i & 3 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & 0\\ i & 4 & 1 & 1 & 1 & \dots & 1 & 0 & 0 & 0\\ \vdots & & & & & & & & & \\ i & n-i-2 & 1 & 1 & 0 & \dots & 0 & 0 & 0 & 0\\ i & n-i-1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0\\ i & n-i & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0\end{bmatrix} \quad \begin{matrix} \\ \mathbf{Q'}_2 \;\leftarrow\; \mathbf{Q'}_2 - \mathbf{Q'}_3 \\ \\ \\ \\ \\ \\ \\ \end{matrix}$$

$$\vdots$$

$$\begin{bmatrix}0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1\\ 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0\\ 0 & -1 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0\\ \vdots & & & & & & & & & \\ 0 & -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0\\ i/(n-i) & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0\end{bmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \mathbf{Q'}_{n-i} \;\leftarrow\; \mathbf{Q'}_{n-i}/(n-i)\end{matrix}$$

$$\begin{bmatrix}i/(n-i) & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1\\ i/(n-i) & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0\\ i/(n-i) & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0\\ i/(n-i) & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0\\ \vdots & & & & & & & & & \\ i/(n-i) & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0\\ i/(n-i) & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0\\ i/(n-i) & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0\end{bmatrix} \quad \begin{matrix} \mathbf{Q'}_1 & \leftarrow & \mathbf{Q'}_1 + \mathbf{Q'}_{n-i}\\ \mathbf{Q'}_2 & \leftarrow & \mathbf{Q'}_2 + \mathbf{Q'}_{n-i}\\ \mathbf{Q'}_3 & \leftarrow & \mathbf{Q'}_3 + \mathbf{Q'}_{n-i}\\ \mathbf{Q'}_4 & \leftarrow & \mathbf{Q'}_4 + \mathbf{Q'}_{n-i}\\ & & \\ \mathbf{Q'}_{n-i-2} & \leftarrow & \mathbf{Q'}_{n-i-2} + \mathbf{Q'}_{n-i}\\ \mathbf{Q'}_{n-i-1} & \leftarrow & \mathbf{Q'}_{n-i-1} + \mathbf{Q'}_{n-i}\\ & & \end{matrix}$$

It is now clear that the rows of $\mathbf{Q'}$ are linearly independent. Let us rename $\mathbf{Q'}$ the final matrix obtained from the described linear operations in $\mathbf{Q'}$. Thus, matrix $\mathbf{Q}$ has at least $n-i$ independent rows (or columns), because we can clearly see the (vertically mirrored) identity matrix pattern starting from the second column. Since the maximum rank that $\mathbf{Q}$ can have is $n-i+1$, the next question would now be: is there a row of $\mathbf{Q}$ that is linearly independent from the rows in $\mathbf{Q'}$? We can verify that, thanks to a generic row that we append to the end of $\mathbf{Q'}$, and which (again) we rename as $\mathbf{Q'}$:

$$\begin{bmatrix}
i/(n-i) & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \\
i/(n-i) & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\
i/(n-i) & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 \\
i/(n-i) & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\
\vdots & & & & & & & & & \\
i/(n-i) & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\
i/(n-i) & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
i/(n-i) & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
i & \mathbf{Q}_{k2} & \mathbf{Q}_{k3} & \mathbf{Q}_{k4} & \mathbf{Q}_{k5} & \dots & \mathbf{Q}_{k(n-i-2)} & \mathbf{Q}_{k(n-i-1)} & \mathbf{Q}_{k(n-i)} & \mathbf{Q}_{k(n-i+1)}
\end{bmatrix}$$

We seek to determine whether the last row is independent by reducing it to its first element using linear operations. Effectively, to ensure that columns with index 2 or larger will have a single non-zero element, we do the following:

$$\mathbf{Q'}_{n-i+1} \leftarrow \mathbf{Q'}_{n-i+1} - \mathbf{Q}_{k(n-i+1)}\mathbf{Q'}_1 - \mathbf{Q}_{k(n-i)}\mathbf{Q'}_2 - \mathbf{Q}_{k(n-i-1)}\mathbf{Q'}_3 - \dots - \mathbf{Q}_{k2}\mathbf{Q'}_{(n-i)}.$$

This operation will have the expected effect, but it will also reduce the first element to zero:

$$\begin{aligned}
& i - \frac{i}{n-i}\mathbf{Q}_{k(n-i+1)} - \frac{i}{n-i}\mathbf{Q}_{k(n-i)} - \frac{i}{n-i}\mathbf{Q}_{k(n-i-1)} - \dots - \frac{i}{n-i}\mathbf{Q}_{k2} \\
&= i - \frac{i}{n-i}\sum_{\ell=2}^{n-i+1}\mathbf{Q}_{k\ell} \\
&= i - \frac{i}{n-i}(n-i) \\
&= i - i \\
&= 0.
\end{aligned}$$

Thus, under case 1, we conclude that matrix $\mathbf{Q}$ possesses exactly $n-i$ independent rows or columns. If the SPFG has, under case 1, augmented matrices $[\mathbf{Q}|\vec{b}]$ of rank $n-i$, it means the corresponding columns of $\mathbf{A}$ will be able to represent their part of the SPFG with fidelity.

Lemma 4.2.6. *For any SPFG with $n > 5$, there is an MILP-representation of the partition function value of an embedded coalition of cardinality $i < \frac{n}{2}$, if $n-i+1 > 4$ and $[\mathbf{Q}|\vec{b}]$ has rank $n-i+1$.*

Case 2: $n > 5$, $i < \frac{n}{2}$ and the column index of matrix A is $n-i+1 > 4$. Again, using the γ-partitions extracted from $\vec{Q}$ with respect to $n-i$ players, we define the matrix $\mathbf{Q}'$:

$$\mathbf{Q}' = \begin{bmatrix} i & 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 \\ i & 2 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 0 \\ i & 3 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & 0 \\ \vdots & & & & & & & & & \\ i & i-1 & 1 & 1 & 1 & \dots & 0 & 0 & 0 & 0 \\ i & i & 1 & 1 & 1 & \dots & 0 & 0 & 0 & 0 \\ i+1 & i & 1 & 1 & 1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & & \\ n-i-2 & i & 1 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ n-i-1 & i & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ n-i & i & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix}_{\in \mathbb{Z}_+^{(n-i)\times(n-i+1)}}.$$

Using the same steps as in the previous case, i.e., subtracting from each line the next and dividing the last line by i before adding it to leave a single non-zero element in the second column, we get a matrix where we can clearly see linear independence between the vectors given by its rows.

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 \\ \vdots & & & & & & & & & \\ 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & & \\ -1 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ (n-i)/i & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix} \text{ After subtractions and division;}$$

$$\begin{bmatrix} (n-i)/i & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \\ (n-i)/i & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\ (n-i)/i & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 \\ \vdots & & & & & & & & & \\ (n-i)/i & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & & \\ -1 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ (n-i)/i & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix} \text{After additions.}$$

We redefine the matrix $\mathbf{Q'}$ as the one above. We remark that the first $i-1$ rows and row $n-i$ of $\mathbf{Q'}$ have their first element equal to $(n-i)/i$, while rows i to $n-i-1$ have their first element equal to 1. Thus, adding a generic $\mathbf{Q}_k$ row to matrix $\mathbf{Q'}$ will leave us performing the following operation to verify if the rank of $\mathbf{Q}$ can reach its maximum $n-i+1$:

$$\mathbf{Q'}_{n-i+1} \leftarrow \mathbf{Q'}_{n-i+1} - \mathbf{Q}_{k(n-i+1)}\mathbf{Q'}_1 - \mathbf{Q}_{k(n-i)}\mathbf{Q'}_2 - \mathbf{Q}_{k(n-i-1)}\mathbf{Q'}_3 - \cdots - \mathbf{Q}_{k2}\mathbf{Q'}_{(n-i)}. \quad (4.1)$$

This time however, we are uncertain of the value of $\mathbf{Q}_{k1}$ since $i < \frac{n}{2}$, and thus of the associated modified value resulting from operation (4.1):

$$\mathbf{Q}_{k1} - \frac{n-i}{i}\mathbf{Q}_{k(n-i+1)} - \frac{n-i}{i}\mathbf{Q}_{k(n-i)} - \cdots - \frac{n-i}{i}\mathbf{Q}_{k(n-2i+3)} - \mathbf{Q}_{k(n-2i+2)} - \cdots - \mathbf{Q}_{k(n-i-1)} - \mathbf{Q}_{k2}$$
$$= \mathbf{Q}_{k1} - \frac{n-i}{i}\mathbf{Q}_{k2} - \sum_{l=3}^{n-2i+2}\mathbf{Q}_{kl} - \frac{n-i}{i}\sum_{l=n-2i+3}^{n-i+1}\mathbf{Q}_{kl}. \quad (4.2)$$

Consequently, since we subtract from $\mathbf{Q}_{k1}$ multiples of other first elements, we identify two disjoint subcases:

- Subtractions of a multiple of $(n-i)/i$, coming from rows 1 to $i-1$ and $n-i$. This multiple will be equal to the sum of players in coalitions 2 and $n-2i+3$ to $n-i+1$.
- Subtractions of a multiple of -1, coming from rows i to $n-i$. This multiple will be equal to the sum of players in coalitions 3 to $n-2i+2$.

To illustrate better this distinction, let us introduce simpler notation for the appended row. We use $\vec{x} = \vec{x}_1 = \mathbf{Q}_{k1}$ for the vector composed of the first entry, $\vec{y}$ is the vector composed of entries 2 and entries $n-2i+3$ to $n-i+1$ (i.e., the entries associated with multiples of $(n-i)/i$) and $\vec{z}$ is the vector entries from 3 to $n-2i+2$ (i.e., the entries associated with multiples of -1). Under the matrix form, we get:

$$\begin{bmatrix}
(n-i)/i & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\
(n-i)/i & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 \\
(n-i)/i & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\
\vdots & & & & & & & & & \\
(n-i)/i & 0 & 0 & 0 & \dots & 0 & 1 & \dots & 0 & 0 \\
-1 & 0 & 0 & 0 & \dots & 1 & 0 & \dots & 0 & 0 \\
-1 & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\
\vdots & & & & & & & & & \\
-1 & 0 & 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 \\
-1 & 0 & 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\
(n-i)/i & 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\
\vec{x}_1 & \vec{y}_2 & \vec{z}_3 & \vec{z}_4 & \dots & \vec{z}_{n-2i+2} & \vec{y}_{n-2i+3} & \dots & \vec{y}_{n-i} & \vec{y}_{n-i+1}
\end{bmatrix}_{\in \mathbb{Z}_+^{(n-i+1)\times(n-i+1)}}.$$

We proceed with some very simple statements. We know that the sum of the appended row is n, i.e., $n = \vec{1}^T \cdot \vec{x} + \vec{1}^T \cdot \vec{y} + \vec{1}^T \cdot \vec{z}$. Moreover, at least one entry of this numerical partition equals i. Nor $\vec{x}$, $\vec{y}_2$ or $\vec{z}_3$ equals to zero and $\vec{x}_1 \geq \vec{y}_2 \geq \vec{z}_3 \geq \vec{z}_4 \geq \dots \geq \vec{y}_{n-i+1} \geq 0$. Also, we make the hypothesis that there is another independent row we can find in $\mathbf{Q}$. Thus, going back to the calculation of the modified first entry in (4.2), if we assume that it is different from zero, we have:

$$\begin{aligned}
& & 0 \neq \vec{1}^T\vec{x} - \frac{n-i}{i}\vec{1}^T\vec{y} - \vec{1}^T\vec{z} \\
&\Leftrightarrow & \frac{n-i}{i}\vec{1}^T\vec{y} \neq \vec{1}^T\vec{x} - \vec{1}^T\vec{z} \\
&\Leftrightarrow & \frac{n-i}{i} \neq \frac{\vec{1}^T\vec{x} - \vec{1}^T\vec{z}}{\vec{1}^T\vec{y}} \\
&\implies & \vec{1}^T\vec{x} - \vec{1}^T\vec{z} \neq n - i \bigwedge \vec{1}^T\vec{y} \neq i.
\end{aligned}$$

A critical remark here is that we need at least one coalition of cardinality i for the numerical partition $\mathbf{Q}_k$. Therefore, as there are exactly $i-1$ coalitions under the $\vec{y}$ label besides $\vec{y}_2$, we will always have that all components of $\vec{y}$ are zero except $\vec{y}_2$, since there are not enough players left to populate the rightmost coalitions of the partition. This means that if $\mathbf{Q}_{k1}$ is i, we should restrict $\vec{y}_2 \leq i-1$ and all remaining $\vec{z}_l \leq i-1$, while still obtaining $n = \vec{1}^T\vec{x} + \vec{1}^T\vec{y} + \vec{1}^T\vec{z}$. Since under this case we have $n > 5$, for what values of i would it be possible to find a partition of $\mathbf{Q}$ not already in $\mathbf{Q}'$ and conforming to the described criteria?

Under this case, we work in the integer range $1 \leq i \leq (n-1)/2$. Also, we know that whatever the outcome, the maximal number of non-empty coalitions contributing to $\vec{z}$ is $n-2i$, which from the last sentence oscillates between 1 and $n-2$, knowing that $n \geq 6$.

Given that we set $\vec{x}_1 = i$ and $\vec{y} = (i-1, 0, \ldots, 0)$, we are left with $\vec{1}^T \vec{z} = n - 2i + 1$. Do we always have $(n-2i+1)/(i-1) \leq n - 2i$, or in other words, is there enough room in the coalitions of $\vec{z}$ for the remaining players?

$$\begin{array}{rcll} & \frac{n-2i+1}{i-1} & \leq & n-2i \\ \Leftrightarrow & \frac{n-2i+1}{n-2i} & \leq & i-1 \\ \Longrightarrow & 2 & \leq & i-1 \quad \text{since } \lceil \frac{n-2i+1}{n-2i} \rceil = 2 \text{ and } i-1 \in \mathbb{Z} \\ \Longrightarrow & \text{Yes, if} & i \geq 3. & \end{array}$$

We just found that it is possible to find an additional partition in $\mathbf{Q}$ that is linearly independent from the initial rows in $\mathbf{Q}'$, for all $n \geq 6$ and $3 \leq i \leq (n-1)/2$. We simply have to set $\vec{x}_1 = i$, $\vec{y}_2 = i - 1$ and the first $\lfloor \frac{n-2i+1}{i-1} \rfloor$ coalitions in $\vec{z}$ to $i-1$. The remaining players from that division can invest the next unoccupied entry in $\vec{z}$. The rank of a $\mathbf{Q}$ under these conditions is therefore $n - i + 1$.

Now, what if $i = 1$ or $i = 2$? Well, it is certain that we cannot have $\vec{x}_1 = i$ or $\vec{y}_2 = i$, so let us try $\vec{x}_1 \geq \vec{y}_2 > \vec{z}_3 = i$. If we decide to set $\vec{x}_1 = \vec{y}_2 = i + 1$, the maximal number of non-empty coalitions contributing to $\vec{z}$, $n - 2i - 1$, has now to support $n - 3i - 2$ players. Do we always have $(n - 3i - 2)/i \leq n - 2i - 1$?

- For $i = 1$, $n-5 \leq n-3$ is always true. Plus, the configuration with $\vec{x}_1 = \vec{y}_2 = i+1 = 2$ and $\vec{z}_3 = i = 1$ demands at least 5 players, which causes no trouble under our current case with $n \geq 6$.
- For $i = 2$, $n/2 - 4 \leq n - 5$ is true when $n \geq 2$, which is always the case for us here. However, the configuration $\vec{x}_1 = \vec{y}_2 = i + 1 = 3$ and $\vec{z}_3 = i = 2$ demands at least 8 players, leaving a shadow on what happens when $n = 6$ or $n = 7$ and $i = 2$. We can elucidate these two cases by simply analyzing their respective $\mathbf{Q}$, as we do in Table 4.2. We find that $n = 7$ and $i = 2$ is the only exception to the rule, with $\text{rank}(\mathbf{Q}) = n - i$.

Thus, we conclude that matrix $\mathbf{Q}$, under case 2, has exactly $n - i + 1$ independent rows or columns, unless $n = 7$ and $i = 2$ (in which case the rank is $n - i$). If the SPFG has, under case 2, augmented matrices $[\mathbf{Q}|\vec{b}]$ of rank $n - i + 1$ ($n - i$ for the exception), it means that the corresponding columns of $\mathbf{A}$ provide an accurate MILP-representation of the associated embedded coalitions for the SPFG.

With all of the lemmata stated, we can affirm the following:

Theorem 4. Consider an SPFG $(\mathcal{N}, W)$ and the following matrix and vector for each $i = 1, \ldots, n$:

n	column index	i	$\mathbf{Q}$	rank($\mathbf{Q}$)	Verdict
6	5	2	$\begin{bmatrix}4&2&0&0&0\\3&2&1&0&0\\2&2&2&0&0\\2&2&1&1&0\\2&1&1&1&1\end{bmatrix}$	5	Rank $= n-i+1$.
7	6	2	$\begin{bmatrix}5&2&0&0&0&0\\4&2&1&0&0&0\\3&2&2&0&0&0\\3&2&1&1&0&0\\2&2&2&1&0&0\\2&2&1&1&1&0\\2&1&1&1&1&1\end{bmatrix}$	5	Rank $< n-i+1$.

Table 4.2. For the second to last column of $\mathbf{A}$, when $n \geq 6$, we almost always have rank($\mathbf{Q}$) $= n-i+1$, which is the largest rank we can aim for. There is one exception when $n = 7$ and $i = 2$ with rank($\mathbf{Q}$) $= n-i$.

- A matrix $\mathbf{Q} \in \mathbb{Z}_+^{\mathrm{PN}(n-i)\times(n-i+1)}$ whose rows are all numerical partitions of n players composed of at least one coalition of cardinality i, ordered from the largest coalition to the smallest, zero-padded to get exactly $n-i+1$ elements in every row;
- A payoff vector $\vec{b} \in \mathbb{R}^{n-i+1}$ for a coalition of cardinality i such that if partition $\mathbf{Q}_k$ forms, the payoff is $\vec{b}_k = W(i,\mathbf{Q}_k)$.

The SPFG $(\mathcal{N}, W)$ is MILP-representable under the following cases:

(1) $n \leq 5$;

(2) $n = 6$ or $n \geq 8$ and:
  - $\mathrm{rank}([\mathbf{Q}|\vec{b}]) = n-i \ \ \forall i \in \mathbb{Z}_+, n/2 \leq i \leq n-4$;
  - $\mathrm{rank}([\mathbf{Q}|\vec{b}]) = n-i+1 \ \ \forall i \in \mathbb{Z}_+, 1 \leq i \leq (n-1)/2$;

(3) $n = 7$ and:
  - $\mathrm{rank}([\mathbf{Q}|\vec{b}]) = 5 \ \ \forall i \in \{2,3\}$;
  - $\mathrm{rank}([\mathbf{Q}|\vec{b}]) = 7 \ \ \forall i \in \{1\}$.

Corollary 4.2.7. *Suppose an SPFG $(\mathcal{N}, W)$ is known to be MILP-representable. Then, the matrix $\mathbf{A}$ can be computed in polynomial time on the number of players, and at most polynomial calls to an oracle providing the partition function value $W(\mathcal{C}, \mathcal{P})$.*

Proof. In the proof of Theorem 4, we identified the rank of $\mathbf{Q}$ for each $i = 1, \ldots, n$ through an explicit determination of a basis for the vector space generated by the linear map $\mathbf{Q}\vec{v}$. Therefore, we can restrict $\mathbf{Q}$ to the row vectors (numerical coalitions) associated with the determined basis. Since the cardinality of the set of vectors forming the basis is linear on the number of players, determining

$$\mathbf{A}_{*(n-i+1)} = \mathbf{Q}^{-1}\vec{b}$$

with $\mathbf{Q}$ restricted to the described basis can be done in polynomial time on the number of players [9]. Since $\mathbf{A}$ has n columns, then its computation takes polynomial time on the number of players. □

4.2.3. MILP-approximations of SPFG

For SPFGs that are not MILP-representable, we propose to use the Moore-Penrose inverse matrix [64], also known as the pseudo-inverse, in order to generate a least-squares solution to our linear systems of equations $\mathbf{Q}\vec{v} = \vec{b}$. Concretely, given an SPFG $(\mathcal{N},W)$, for each $i = 1, \dots, n$, column $n - i + 1$ of $\mathbf{A}$ is the solution of the following optimization problem:

$$\mathbf{A}_{*(n-i+1)} \in \arg\min_{\vec{v}} \quad ||\mathbf{Q}\vec{v} - \vec{b}||_2 \tag{4.3}$$

where $\mathbf{Q}$ and $\vec{b}$ are as defined in Section 4.2.1. Hence, $\mathbf{A}_{*(n-i+1)} = \mathbf{Q}^{+}\vec{b}$, where $\mathbf{Q}^{+}$ is the pseudo-inverse of $\mathbf{Q}$.

Definition 4.2.8. *Given an SPFG $(\mathcal{N},W)$, we say that $(\mathcal{N},\mathbf{A})$ is its approximated MILP-representation if each column $n - i + 1$ of $\mathbf{A}$ for $i = 1, \dots, n$ is a solution of Problem* (4.3).

Accordingly with Definition 4.1.1, when the optimal value of the objective in Problem (4.3) is zero for each $i = 1, \dots, n$, then we say that its approximated game $(\mathcal{N},\mathbf{A})$ is an MILP-representation of the SPFG $(\mathcal{N},W)$.

The use of approximated MILP-representations is very attractive because it can promote the use of our MILP-representation to (approximately) solve SPFGs. Moreover, it is not difficult at all to implement when the rest of the framework to solve SPFGs with MILP-representations is already in place. In our implementation, we used Numpy [35] as there is a specific function retrieving the pseudo-inverse given $\mathbf{Q}$ and $\vec{b}$. In Chapter 5, we will discuss the performance of this method through the display of an application. However, before this, we discuss approximation-factor bounds when solving SPFGs through their approximated MILP-representations.

4.3. Bounding approximations

Consider an SPFG $(\mathcal{N},W)$ and its approximated MILP-representation $(\mathcal{N},\mathbf{A})$. In what follows, let us denote by W' the partition function induced by $\mathbf{A}$, i.e., the payoff calculated as in Expression (4.11). We define two approximation measurements:

- ϵ_1 is the largest absolute difference between a value returned by W and the value returned by W' for the same embedded numerical coalition, i.e., ϵ_1 is the L^{∞}-norm between W and W'.
- ϵ_2 is the largest gain a coalition can make, in the context of $(\mathcal{N},W)$, by deviating from the partition selected when solving the approximated MILP-representation game,

given that the only deviations allowed are those described in the stability set $\mathcal{F}$ (defined in section 3.2).

Formally, we have:

$$\epsilon_1 = \max\Big\{|W(\mathcal{C},\mathcal{P}) - W'(\mathcal{C},\mathcal{P})| \ : (\mathcal{C},\mathcal{P}) \in \mathcal{E}(\mathcal{N})\Big\} \tag{4.1a}$$

$$\tilde{\mathcal{P}} = \max_{\mathcal{P}\in\mathcal{F}} \sum_{\mathcal{C}\in\mathcal{P}} W'(\mathcal{C},\mathcal{P}) \tag{4.1b}$$

$$\mathcal{F} := \Big\{\mathcal{P} \in \Pi(\mathcal{N}) : \mathcal{F}_1(\mathcal{P}) \cap \mathcal{F}_2(\mathcal{P}) \cap \mathcal{F}_3(\mathcal{P}) \cap \mathcal{F}_4(\mathcal{P})\Big\} \tag{4.1c}$$

$$\mathcal{F}_1(\mathcal{P}) : \sum_{\mathcal{C}\in\mathcal{P}} W'(\mathcal{C},\mathcal{P}) \geq \sum_{i\in\mathcal{N}} W'(\{i\}, SP) \tag{4.1d}$$

$$\begin{aligned}\mathcal{F}_2(\mathcal{P}) : W'(\mathcal{C},\mathcal{P}) \geq W'(\mathcal{S},(\mathcal{P}-\mathcal{C},\mathcal{S},\mathcal{T})) + W'(\mathcal{T},(\mathcal{P}-\mathcal{C},\mathcal{S},\mathcal{T}))\\ \forall\mathcal{C}\in\mathcal{P}, \forall\mathcal{Q}\in\Pi(\mathcal{C}) : |\mathcal{Q}| = 2, \{\mathcal{S},\mathcal{T}\} = \mathcal{Q}.\end{aligned} \tag{4.1e}$$

$$\mathcal{F}_3(\mathcal{P}) : W'(\mathcal{C},\mathcal{P}) \geq \sum_{i\in\mathcal{C}} W'(\{i\}, SP) \ \forall\mathcal{C}\in\mathcal{P} \tag{4.1f}$$

$$\mathcal{F}_4(\mathcal{P}) : W'(\mathcal{C},\mathcal{P}) + W'(\mathcal{S},\mathcal{P}) \geq W'(\mathcal{C}\cup\mathcal{S},(\mathcal{P}-\mathcal{C}-\mathcal{S},\mathcal{C}\cup\mathcal{S})) \ \forall\mathcal{C},\mathcal{S}\in\mathcal{P} : \mathcal{C}\neq\mathcal{S} \tag{4.1g}$$

$$\epsilon_2 = \max\{0, \delta_{\mathcal{F}_1}, \delta_{\mathcal{F}_2}, \delta_{\mathcal{F}_3}, \delta_{\mathcal{F}_4}\} \tag{4.1h}$$

$$\delta_{\mathcal{F}_1} = \sum_{i\in\mathcal{N}} W(\{i\}, SP) - \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} W(\mathcal{C},\tilde{\mathcal{P}}) \tag{4.1i}$$

$$\begin{aligned}\delta_{\mathcal{F}_2} = \max\Big\{W(\mathcal{S},(\tilde{\mathcal{P}}-\mathcal{C},\mathcal{S},\mathcal{T})) + W(\mathcal{T},(\tilde{\mathcal{P}}-\mathcal{C},\mathcal{S},\mathcal{T})) - W(\mathcal{C},\tilde{\mathcal{P}}) :\\ \mathcal{C}\in\tilde{\mathcal{P}}, \mathcal{Q}\in\Pi(\mathcal{C}), |\mathcal{Q}| = 2, \{\mathcal{S},\mathcal{T}\} = \mathcal{Q}\Big\}\end{aligned} \tag{4.1j}$$

$$\delta_{\mathcal{F}_3} = \max\Big\{\sum_{i\in\mathcal{C}} W(\{i\}, SP) - W(\mathcal{C},\tilde{\mathcal{P}}) \ : \mathcal{C}\in\tilde{\mathcal{P}}\Big\} \tag{4.1k}$$

$$\begin{aligned}\delta_{\mathcal{F}_4} = \max\Big\{W(\mathcal{C}\cup\mathcal{S},(\tilde{\mathcal{P}}-\mathcal{C}-\mathcal{S},\mathcal{C}\cup\mathcal{S})) - W(\mathcal{C},\tilde{\mathcal{P}}) - W(\mathcal{S},\tilde{\mathcal{P}}) :\\ \mathcal{C},\mathcal{S}\in\tilde{\mathcal{P}}, \mathcal{C}\neq\mathcal{S}\Big\}.\end{aligned} \tag{4.1l}$$

Is the SPFG ($\mathcal{N}$,W') different from the original game? If yes, does the computation of a stable partition provide a useful solution to the original SPFG ($\mathcal{N}$,W)? Assessing whether ϵ_1 and ϵ_2 are equal to zero or strictly positive can help us to answer these questions. The first measurement, $\epsilon_1 \geq 0$, indicates that W' is different from W if and only if it is greater than 0. Otherwise, the SPFG ($\mathcal{N}$,W) is MILP-representable and solving the SPFG ($\mathcal{N}$,W') provides a stable (SW-maximizing) partition to the SPFG ($\mathcal{N}$,W), i.e., $\epsilon_2 = 0$. The second measurement, $\epsilon_2 \geq 0$, has similar behaviour, excepted for the reciprocity: W' can be different from W without it affecting the stability verdict apposed by solving its approximated representation. Hence, we obtain the following logical implications connecting ϵ_1 and ϵ_2:

$$\epsilon_1 = 0 \Leftrightarrow W' = W \tag{4.2a}$$

$$\epsilon_1 = 0 \Rightarrow \epsilon_2 = 0 \tag{4.2b}$$

$$W' = W \Rightarrow \epsilon_2 = 0 \tag{4.2c}$$

$$\epsilon_1 > 0 \Leftrightarrow W' \neq W \tag{4.2d}$$

$$\epsilon_2 > 0 \Rightarrow \epsilon_1 > 0 \tag{4.2e}$$

$$\epsilon_2 > 0 \Rightarrow W' \neq W. \tag{4.2f}$$

In practice, we are mainly concerned with whether $\epsilon_2 = 0$ or not: the usefulness of our approximated MILP-representation relies on its capacity to determine a stable partition of the original game (accordingly with the definition of stability settled by $\mathcal{F}$). When $\epsilon_2 > 0$, our representation introduces approximation errors that compromise stability from the original SPFG viewpoint. Additionally, the value of ϵ_2 signifies how far from stable the chosen partition is. Thus, in what follows, our goal is to show that ϵ_2 is upper-bounded, effectively demonstrating how much, in the worst-case scenario, the approximation error affects the stability of the solution. Before we continue let us note that we will not analyze whether the determined partition $\tilde{\mathcal{P}}$ also maximizes the SW of the original game among its stable partitions. We set as priority being able to use our MILP-representation for finding a stable partition.

Theorem 5. If $\tilde{P}$ is a stable partition for $(\mathcal{N},\mathbf{A})$, then $\epsilon_2 \leq 2n\epsilon_1$.

Proof. First, we define ϵ_2', which is the equivalent of ϵ_2 but using W'.

$$\epsilon_2' = \max\{0, \delta'_{\mathcal{F}_1}, \delta'_{\mathcal{F}_2}, \delta'_{\mathcal{F}_3}, \delta'_{\mathcal{F}_4}\} \tag{4.3}$$

$$\delta'_{\mathcal{F}_1} = \sum_{i\in\mathcal{N}} W'(\{i\}, SP) - \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} W'(\mathcal{C}, \tilde{\mathcal{P}}) \tag{4.4}$$

$$\begin{aligned}\delta'_{\mathcal{F}_2} = \max\Big\{&W'(\mathcal{S}, (\tilde{\mathcal{P}} - \mathcal{C}, \mathcal{S}, \mathcal{T})) + W'(\mathcal{T}, (\tilde{\mathcal{P}} - \mathcal{C}, \mathcal{S}, \mathcal{T})) - W'(\mathcal{C}, \tilde{\mathcal{P}}) :\\ &\mathcal{C} \in \tilde{\mathcal{P}}, \mathcal{Q} \in \Pi(\mathcal{C}), |\mathcal{Q}| = 2, \{\mathcal{S},\mathcal{T}\} = \mathcal{Q}\Big\}\end{aligned} \tag{4.5}$$

$$\delta'_{\mathcal{F}_3} = \max\Big\{\sum_{i\in\mathcal{C}} W'(\{i\}, SP) - W'(\mathcal{C}, \tilde{\mathcal{P}}) \ : \mathcal{C} \in \tilde{\mathcal{P}}\Big\} \tag{4.6}$$

$$\begin{aligned}\delta'_{\mathcal{F}_4} = \max\Big\{&W'(\mathcal{C} \cup \mathcal{S}, (\tilde{\mathcal{P}} - \mathcal{C} - \mathcal{S}, \mathcal{C} \cup \mathcal{S})) - W'(\mathcal{C}, \tilde{\mathcal{P}}) - W'(\mathcal{S}, \tilde{\mathcal{P}}) :\\ &\mathcal{C},\mathcal{S} \in \tilde{\mathcal{P}}, \mathcal{C} \neq \mathcal{S}\Big\}.\end{aligned} \tag{4.7}$$

Since $\tilde{\mathcal{P}}$ satisfies all stability criteria in (4.1d)-(4.1g), we have $\epsilon_2' = 0$ and thus:

$$\delta'_{\mathcal{F}_1} = \sum_{i\in\mathcal{N}} W'(\{i\}, SP) - \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} W'(\mathcal{C}, \tilde{\mathcal{P}}) \leq 0 \tag{4.8}$$

$$\begin{aligned}\delta'_{\mathcal{F}_2} = \max\Big\{&W'(\mathcal{S}, (\tilde{\mathcal{P}} - \mathcal{C}, \mathcal{S}, \mathcal{T})) + W'(\mathcal{T}, (\tilde{\mathcal{P}} - \mathcal{C}, \mathcal{S}, \mathcal{T})) - W'(\mathcal{C}, \tilde{\mathcal{P}}) :\\ &\mathcal{C} \in \tilde{\mathcal{P}}, \mathcal{Q} \in \Pi(\mathcal{C}), |\mathcal{Q}| = 2, \{\mathcal{S},\mathcal{T}\} = \mathcal{Q}\Big\} \leq 0\end{aligned} \tag{4.9}$$

$$\delta'_{\mathcal{F}_3} = \max\Big\{\sum_{i\in\mathcal{C}} W'(\{i\}, SP) - W'(\mathcal{C}, \tilde{\mathcal{P}}) \ : \mathcal{C} \in \tilde{\mathcal{P}}\Big\} \leq 0 \tag{4.10}$$

$$\delta'_{\mathcal{F}_4} = \max\Big\{W'(\mathcal{C}\cup\mathcal{S},(\tilde{\mathcal{P}}-\mathcal{C}-\mathcal{S},\mathcal{C}\cup\mathcal{S})) - W'(\mathcal{C},\tilde{\mathcal{P}}) - W'(\mathcal{S},\tilde{\mathcal{P}}) : \mathcal{C},\mathcal{S}\in\tilde{\mathcal{P}},\mathcal{C}\neq\mathcal{S}\Big\} \leq 0. \tag{4.11}$$

Second, we study each stability criterion separately. By subtracting $\delta'_{\mathcal{F}_i}$ from $\delta_{\mathcal{F}_i}$ and then applying the triangular inequality $|a+b| \leq |a|+|b|$ using the corresponding embedded coalitions bounded by (4.1a), we get upper-bounds for ϵ_2 as a function of ϵ_1.

Claim 4.3.1. If $\delta_{\mathcal{F}_1} = \epsilon_2$, then $\epsilon_2 \leq 2n\epsilon_1$.

We suppose that $\delta_{\mathcal{F}_1} = \epsilon_2$, i.e., the largest stability criterion violation occurs for $\mathcal{F}_1(\tilde{P})$.

(1) Subtraction of $\delta_{\mathcal{F}_1}$ by $\delta'_{\mathcal{F}_1}$.

$$\delta'_{\mathcal{F}_1} = \sum_{i\in\mathcal{N}} W'(\{i\},SP) - \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} W'(\mathcal{C},\tilde{\mathcal{P}}) \leq 0 \tag{4.12}$$

$$\delta_{\mathcal{F}_1} = \sum_{i\in\mathcal{N}} W(\{i\},SP) - \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} W(\mathcal{C},\tilde{\mathcal{P}}) = \epsilon_2 \tag{4.13}$$

$$\delta_{\mathcal{F}_1} - \delta'_{\mathcal{F}_1} = \sum_{i\in\mathcal{N}} \left(W(\{i\},SP) - W'(\{i\},SP)\right) + \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} \Big(W'(\mathcal{C},\tilde{\mathcal{P}}) - W(\mathcal{C},\tilde{\mathcal{P}})\Big) \geq \epsilon_2 \tag{4.14}$$

(2) Triangular inequality.

$$\begin{aligned}\epsilon_2 &\leq |\sum_{i\in\mathcal{N}} \left(W(\{i\},SP) - W'(\{i\},SP)\right) + \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} \Big(W'(\mathcal{C},\tilde{\mathcal{P}}) - W(\mathcal{C},\tilde{\mathcal{P}})\Big)| \\ &\leq \sum_{i\in\mathcal{N}} |W(\{i\},SP) - W'(\{i\},SP)| + \sum_{\mathcal{C}\in\tilde{\mathcal{P}}} |W'(\mathcal{C},\tilde{\mathcal{P}}) - W(\mathcal{C},\tilde{\mathcal{P}})|\end{aligned} \tag{4.15}$$

(3) Bounding by ϵ_1 using the worst-case scenario.

$$|W(\{i\},SP) - W'(\{i\},SP)| \leq \epsilon_1 \ \forall i\in\mathcal{N} \tag{4.16}$$

$$|W(\mathcal{C},\tilde{\mathcal{P}}) - W'(\mathcal{C},\tilde{\mathcal{P}})| \leq \epsilon_1 \ \forall \mathcal{C}\in\tilde{\mathcal{P}} \tag{4.17}$$

$$|\mathcal{P}| \leq |\mathcal{N}| = n \ \forall \mathcal{P}\in\Pi(\mathcal{N}) \tag{4.18}$$

$$\implies \epsilon_2 \leq n\epsilon_1 + n\epsilon_1 = 2n\epsilon_1 \tag{4.19}$$

Claim 4.3.2. If $\delta_{\mathcal{F}_2} = \epsilon_2$, then $\epsilon_2 \leq 3\epsilon_1$.

We suppose that $\delta_{\mathcal{F}_2} = \epsilon_2$, i.e., the largest stability criterion violation happens for $\mathcal{F}_2(\tilde{P})$. It means $\exists \mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^* : \delta_{\mathcal{F}_2} = W(\mathcal{S}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) + W(\mathcal{T}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) - W(\mathcal{C}^*,\tilde{\mathcal{P}}) = \epsilon_2$.

(1) Subtraction of $\delta_{\mathcal{F}_2}$ by $\delta'_{\mathcal{F}_2}$.

$$\delta'_{\mathcal{F}_2} = W'(\mathcal{S}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) + W'(\mathcal{T}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) - W'(\mathcal{C}^*,\tilde{\mathcal{P}}) \leq 0 \tag{4.20}$$

$$\delta_{\mathcal{F}_2} = W(\mathcal{S}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) + W(\mathcal{T}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) - W(\mathcal{C}^*,\tilde{\mathcal{P}}) = \epsilon_2 \tag{4.21}$$

$$\begin{aligned}\delta_{\mathcal{F}_2} - \delta'_{\mathcal{F}_2} = {} & W(\mathcal{S}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) - W'(\mathcal{S}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) \\ & + W(\mathcal{T}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) - W'(\mathcal{T}^*,(\tilde{\mathcal{P}}-\mathcal{C}^*,\mathcal{S}^*,\mathcal{T}^*)) \\ & + W'(\mathcal{C}^*,\tilde{\mathcal{P}}) - W(\mathcal{C}^*,\tilde{\mathcal{P}}) \qquad \geq \epsilon_2\end{aligned} \tag{4.22}$$

(2) Triangular inequality.

$$\begin{aligned}\epsilon_2 \leq \quad & |W(\mathcal{S}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) - W'(\mathcal{S}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) \\ & +W(\mathcal{T}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) - W'(\mathcal{T}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) \\ & +W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})| \\ \leq \quad & |W(\mathcal{S}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) - W'(\mathcal{S}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*))| \\ & +|W(\mathcal{T}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) - W'(\mathcal{T}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*))| \\ & +|W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})|\end{aligned} \tag{4.23}$$

(3) Bounding by ϵ_1 using the worst-case scenario.

$$|W(\mathcal{S}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) - W'(\mathcal{S}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*))| \leq \epsilon_1 \tag{4.24}$$

$$|W(\mathcal{T}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*)) - W'(\mathcal{T}^*, (\tilde{\mathcal{P}} - \mathcal{C}^*, \mathcal{S}^*, \mathcal{T}^*))| \leq \epsilon_1 \tag{4.25}$$

$$|W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})| \leq \epsilon_1 \tag{4.26}$$

$$\implies \epsilon_2 \leq 3\epsilon_1. \tag{4.27}$$

Claim 4.3.3. If $\delta_{\mathcal{F}_3} = \epsilon_2$, then $\epsilon_2 \leq (n+1)\epsilon_1$.

By hypothesis $\delta_{\mathcal{F}_3} = \epsilon_2$, i.e., the largest stability criterion violation occurs in $\mathcal{F}_3(\tilde{P})$. It means $\exists \mathcal{C}^* : \sum_{i \in \mathcal{C}^*} W(\{i\}, SP) - W(\mathcal{C}^*, \tilde{\mathcal{P}}) = \epsilon_2$.

(1) Subtraction of $\delta_{\mathcal{F}_3}$ by $\delta'_{\mathcal{F}_3}$.

$$\delta'_{\mathcal{F}_3} = \sum_{i \in \mathcal{C}^*} W'(\{i\}, SP) - W'(\mathcal{C}^*, \tilde{\mathcal{P}}) \leq 0 \tag{4.28}$$

$$\delta_{\mathcal{F}_2} = \sum_{i \in \mathcal{C}^*} W(\{i\}, SP) - W(\mathcal{C}^*, \tilde{\mathcal{P}}) = \epsilon_2 \tag{4.29}$$

$$\delta_{\mathcal{F}_2} - \delta'_{\mathcal{F}_2} = \sum_{i \in \mathcal{C}^*} W(\{i\}, SP) - W'(\{i\}, SP) + W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}}) \geq \epsilon_2 \tag{4.30}$$

(2) Triangular inequality.

$$\begin{aligned}\epsilon_2 &\leq |\sum_{i \in \mathcal{C}^*} W(\{i\}, SP) - W'(\{i\}, SP) + W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})| \\ &\leq \sum_{i \in \mathcal{C}^*} |W(\{i\}, SP) - W'(\{i\}, SP)| + |W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})|\end{aligned} \tag{4.31}$$

(3) Bounding by ϵ_1 using the worst-case scenario.

$$|W(\{i\}, SP) - W'(\{i\}, SP)| \leq \epsilon_1 \; \forall i \in \mathcal{C}^* \tag{4.32}$$

$$|W(\mathcal{C}^*, \tilde{\mathcal{P}}) - W'(\mathcal{C}^*, \tilde{\mathcal{P}})| \leq \epsilon_1 \tag{4.33}$$

$$|\mathcal{C}^*| \leq |\mathcal{N}| = n \tag{4.34}$$

$$\implies \epsilon_2 \leq n\epsilon_1 + \epsilon_1 = (n+1)\epsilon_1 \tag{4.35}$$

Claim 4.3.4. If $\delta_{\mathcal{F}_4} = \epsilon_2$, then $\epsilon_2 \leq 3\epsilon_1$.

Finally, let us suppose that $\delta_{\mathcal{F}_4} = \epsilon_2$, i.e., the largest stability criterion violation happens for $\mathcal{F}_4(\tilde{\mathcal{P}})$. It means $\exists \mathcal{C}^*, \mathcal{S}^* : \delta_{\mathcal{F}_4} = W(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W(\mathcal{S}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}}) = \epsilon_2$.

(1) Subtraction of $\delta_{\mathcal{F}_4}$ by $\delta'_{\mathcal{F}_4}$.

$$\delta'_{\mathcal{F}_4} = W'(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W'(\mathcal{S}^*, \tilde{\mathcal{P}}) - W'(\mathcal{C}^*, \tilde{\mathcal{P}}) \leq 0 \tag{4.36}$$
$$\delta_{\mathcal{F}_4} = W(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W(\mathcal{S}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}}) = \epsilon_2 \tag{4.37}$$

$$\begin{aligned}\delta_{\mathcal{F}_4} - \delta'_{\mathcal{F}_4} = \\ & W(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W'(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) \\ & +W'(\mathcal{S}^*, \tilde{\mathcal{P}}) - W(\mathcal{S}^*, \tilde{\mathcal{P}}) \\ & +W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}}) \\ & \geq \epsilon_2\end{aligned} \tag{4.38}$$

(2) Triangular inequality.

$$\begin{aligned}\epsilon_2 \leq \; & |W(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W'(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) \\ & +W'(\mathcal{S}^*, \tilde{\mathcal{P}}) - W(\mathcal{S}^*, \tilde{\mathcal{P}}) \\ & +W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})| \\ \leq \; & |W(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W'(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*))| \\ & +|W'(\mathcal{S}^*, \tilde{\mathcal{P}}) - W(\mathcal{S}^*, \tilde{\mathcal{P}})| \\ & +|W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})|\end{aligned} \tag{4.39}$$

(3) Bounding by ϵ_1 using the worst-case scenario.

$$|W(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*)) - W'(\mathcal{S}^* \cup \mathcal{C}^*, (\tilde{\mathcal{P}} - \mathcal{C}^* - \mathcal{S}^*, \mathcal{S}^* \cup \mathcal{C}^*))| \leq \epsilon_1 \tag{4.40}$$
$$|W'(\mathcal{S}^*, \tilde{\mathcal{P}}) - W(\mathcal{S}^*, \tilde{\mathcal{P}})| \leq \epsilon_1 \tag{4.41}$$
$$|W'(\mathcal{C}^*, \tilde{\mathcal{P}}) - W(\mathcal{C}^*, \tilde{\mathcal{P}})| \leq \epsilon_1 \tag{4.42}$$
$$\implies \epsilon_2 \leq 3\epsilon_1. \tag{4.43}$$

In conclusion to these four cases, in the worst-case scenario, we have $\epsilon_2 \leq \max\{2n\epsilon_1, 3\epsilon_1, (n+1)\epsilon_1\} = 2n\epsilon_1$; recall that for $n \leq 5$, our approximated representation is exact (ϵ_1=0) and thus, $\epsilon_2 = 0$. □

Chapter 5

Application: competitive markets

In this chapter, we seek to demonstrate the use of our MILP-representation to tackle SPFGs, namely to determine a stable partition for them. For that purpose, we consider the SPFG presented by Nagarajan and Sošić [54], since *(i)* the authors study the problem of determining a stable partition and *(ii)* their setup captures classical features of games (competition for demand, maximization of profits, pricing, etc). In their game, retailers producing interchangeable products compete by setting the prices. The game is played in two-stages, the first consisting of the formation of coalitions (coalition structure), i.e., a partition of the set $\mathcal{N}$ of players (retailers), and the second involving the setting of a price by each coalition.

In Section 5.1, we explain the backward reasoning of Nagarajan and Sošić [54], which allow us to focus simply on the first stage of the game, i.e., determining a stable partition. In particular, this backward process where we compute the equilibria prices of the second stage, provide us with the partition function of their game.

In their article, Nagarajan and Sošić are interested in identifying stable coalition structures when players are farsighted. Thus, their definition of what constitutes a stable partition is different from that used in this thesis and presented in Section 3.2. Indeed, our definition of stability only considers a neighborhood of deviations for a given partition but not sequences of deviations, which can allow to model farsightedness. Moreover, they study the effect of multiple parameters, such as substitutability of the individual demands, variability of the demand process and inventory cost parameters, on the outcome (coalition structure) of the game. In Section 5.2, we apply our approximated MILP-representation to study the coalition structure formation for instances of this game combining different parameters of it: varying degrees of substitutability, number of players and market size. However, we maintain our definition of stability and do not consider players to be farsighted. This will be sufficient to showcase the value of the contribution in this thesis and to have a basis of comparison between the two stability approaches. For the application in hand, we observe from our computational experiments that when a partition is stable as per our definition, then it is also stable as per the definition adopted in Nagarajan and Sošić [54], described in the next section. We will see this empirically in Section 5.2 and also justify why it is true.

5.1. Game setup

Let us describe concretely the SPFG by Nagarajan and Sošić [54], namely, its partition function value[1]. In this game, we have n (symmetric) players who represent retailers. The players compete for demand since they produce interchangeable goods. Each player i aims to set a price $\vec{p_i}$ for their product in order to maximize their profit; the costs are normalized so that they are equal to zero and we only have to consider profits (price times the demand). In the first stage of the game, players form coalitions, which results in a partition of the set of players $\mathcal{N}$. In the second stage, prices are set, with players in the same coalition coordinating their prices to be equal. The demand faced by player i belonging to coalition $\mathcal{C}$, given that the partition formed in the first stage is $\mathcal{P} \ni \mathcal{C}$, is expressed by:

$$D_{i\in\mathcal{C},\mathcal{C}\in\mathcal{P}}(\mathcal{P}) = A - (1+\alpha)\vec{p_i} + \frac{\alpha}{n}|\mathcal{C}|\vec{p_i} + \frac{\alpha}{n}\sum_{j\notin\mathcal{C}}\vec{p_j}, \tag{5.1}$$

where $\alpha \in [0,\infty[$ is the substitutability level between products of any two retailers i and j, A is the market size and $\vec{p}$ is the price vector. Therefore, we can write the profit for player i,

$$\begin{aligned}\text{Profit}_i(D(\mathcal{P}),\vec{p}) &= D_{i\in\mathcal{C},\mathcal{C}\in\mathcal{P}}(\mathcal{P}) \times \vec{p_i} \\ &= A\vec{p_i} - (1+\alpha)\vec{p_i}^2 + \frac{\alpha}{n}|\mathcal{C}|\vec{p_i}^2 + \frac{\alpha}{n}\sum_{j\notin\mathcal{C}}\vec{p_j}\vec{p_i}. \end{aligned} \tag{5.2}$$

Next, we apply a backward induction process to determine the value of the partition function for this game for any embedded coalition. This means that we start by determining the equilibrium price $\vec{p}$ of the second-stage given a partition $\mathcal{P}$ (outcome of the first-stage). By equilibrium price, we mean the prices $\vec{p}$ such that no player has incentive to unilaterally deviate from the selected price. Through the profit formula (5.2), we can find the equilibrium price for each player by finding the maximum profit. In this way, the set of prices $\vec{p}$ simultaneously maximizes each player individual profit, and thus, there will be no incentive to deviate.

We are able to deduce any equilibrium price $\vec{p_k}$ given the size of the coalition $\mathcal{T} \ni k$, the price of another player $\vec{p_i}$ and the size of the coalition $\mathcal{C} \ni i$.

$$\begin{aligned}0 = \frac{\partial \text{Profit}_i}{\partial \vec{p_i}} &= A - 2(1+\alpha)\vec{p_i} + 2\frac{\alpha}{n}|\mathcal{C}|\vec{p_i} + \frac{\alpha}{n}\sum_{j\notin\mathcal{C}}\vec{p_j} \\ &= A - 2(1+\alpha)\vec{p_i} + \frac{\alpha}{n}|\mathcal{C}|\vec{p_i} + \frac{\alpha}{n}\sum_{\mathcal{S}\in\mathcal{P}}|\mathcal{S}|\vec{p}_{j\in\mathcal{S}}\end{aligned}$$

[1]We consider the deterministic SPFG version in [54].

$$
\begin{aligned}
\implies \frac{\alpha}{n}\sum_{\mathcal{S}\in\mathcal{P}}|\mathcal{S}|\vec{p}_{j\in\mathcal{S}} &= 2(1+\alpha)\vec{p}_i - \frac{\alpha}{n}|\mathcal{C}|\vec{p}_i - A \\
&= 2(1+\alpha)\vec{p}_k - \frac{\alpha}{n}|\mathcal{T}|\vec{p}_k - A \\
\implies \frac{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{C}|}{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{T}|}\vec{p}_i &= \vec{p}_k.
\end{aligned}
$$

Now, we can replace the price of any player as a function of the price of player i, and get the equilibrium price $\vec{p}_i$ for any player $i\in\mathcal{C}$ in any partition $\mathcal{P}\ni\mathcal{C}$.

$$
\begin{aligned}
0 = \frac{\partial \text{Profit}_i}{\partial \vec{p}_i} &= A - 2(1+\alpha)\vec{p}_i + 2\frac{\alpha}{n}|\mathcal{C}|\vec{p}_i + \frac{\alpha}{n}\sum_{j\notin\mathcal{C}}\vec{p}_j \\
&= A - 2(1+\alpha)\vec{p}_i + \frac{\alpha}{n}|\mathcal{C}|\vec{p}_i + \frac{\alpha}{n}\sum_{\mathcal{T}\in\mathcal{P}}|\mathcal{T}|\frac{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{C}|}{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{T}|}\vec{p}_i \\
\implies \vec{p}_i &= \frac{A}{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{C}| - (2(1+\alpha)-\frac{\alpha}{n}|\mathcal{C}|)\frac{\alpha}{n}\sum_{\mathcal{T}\in\mathcal{P}}\frac{|\mathcal{T}|}{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{T}|}} \\
&= \frac{A}{(2(1+\alpha)-\frac{\alpha}{n}|\mathcal{C}|)(1-\frac{\alpha}{n}\sum_{\mathcal{T}\in\mathcal{P}}\frac{|\mathcal{T}|}{2(1+\alpha)-\frac{\alpha}{n}|\mathcal{T}|})} \qquad (5.3)
\end{aligned}
$$

Using Equation (5.3), we can calculate the profit for any embedded coalition at equilibrium, and thus create the partition function W for this SPFG:

$$W(\mathcal{C},\mathcal{P}) = \sum_{i\in\mathcal{C}} \text{Profit}_i(D(\mathcal{P}),\vec{p}), \qquad (5.4)$$

where $\vec{p}$ is computed as in Equation (5.3).

The presented backward reasoning enabled us to define the SPFG $(\mathcal{N},W)$ by Nagarajan and Sošić [54]. Now, we can obtain our approximated MILP-representation $(\mathcal{N},\mathbf{A})$, and run the algorithm described in Section 4.1.3 to solve the approximated game. This algorithm will return the stable partition maximizing the SW of the approximated game if a stable partition exists; otherwise, it returns the empty set. It will also return two approximation measurements for the MILP-representation, ϵ_1 and ϵ_2, that we defined in section 4.3. The first approximation measurement, $\epsilon_1 \geq 0$, describes how different is the MILP-representation from the original SPFG. If $\epsilon_1 = 0$, the representation perfectly captures the game. If $\epsilon_1 > 0$, there are embedded coalitions that will receive different payoffs under the MILP-representation and under the original SPFG, and ϵ_1 is the greatest value among those differences. Up until $n \leq 5$, ϵ_1 is 0 (Theorem 4) since our MILP-representation is exact for small symmetrical games. The second approximation measurement, $\epsilon_2 \geq 0$, describes how far from stable is the algorithm-chosen partition in the original SPFG. If $\epsilon_2 = 0$, the chosen partition is stable in both the approximated representation and the original game. If $\epsilon_2 > 0$, an embedded coalition, according to the original SPFG, could gain a maximum of ϵ_2 when deviating from the chosen partition. In the next section, we analyze our approximated representations for the described game.

5.2. Computational analysis

We tested our algorithm from Section 4.1.3 for instances of the SPFG by Nagarajan and Sošić [54] with $A \in \{20,100\}$, $\alpha \in \{0, 1, 2, \ldots, 9\}$ and $n \in \{3, 4, 5, \ldots, 14\}$. When $\alpha = 0$, it means the players are indifferent to cooperation since the prices of the other players do not affect their individual demand. Hence, all partitions are stable. For any other combination of parameters, as long as $\alpha > 0$, we observe that our algorithm chooses the GC. This is concordant with Theorem 1 of Nagarajan and Sošić [54], as it states that when $\alpha > 0$, the total profit generated by the GC exceeds the total profit generated in any other coalition structure. Consequently, we deduce that this is the first partition whose stability is analyzed by our stability constraints in $\mathcal{F}$. If this partition is stable according to the stability criteria in $\mathcal{F}$, then we expect to have the GC be the selected partition. Also because of their Theorem 1, we know that our stability constraints $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ hold, as they involve exclusively neighbourhoods including all the players, and thus are always comparing the total profit of a coalition structure that is not the GC to the total profit of the GC. There are no other neighbour to compare GC to in $\mathcal{F}_4$, so this constraint holds by default. Hence, we expect our algorithm to return the GC.

When analyzing our computational results, we remarked that when $\epsilon_1 > 0$, for $\alpha > 0$ and $n > 5$, ϵ_2 stays equal to 0. In other words, even though the MILP-representation is a bit off (i.e., it is indeed an approximation to the original game), it never induces an error on the stability verdict and produces a valid stable partition to the original game. Since this is a general trend in our results, we provide a sample of them in Table 5.1 for the case with 14 players. We chose $n = 14$ because we expect our approximated MILP-representation to deteriorate with the number of players (increase in ϵ_1) as well as the stability error ϵ_2 (recall from Theorem 5 that n and ϵ_1 play an important role on the upper-bounding of ϵ_2). In Table 5.1, except for the cases with $\alpha = 0$, it is possible to observe that the GC is a stable partition to the approximated game $(\mathcal{N},\mathbf{A})$, column "Chosen partition". Moreover, we always have $\epsilon_2 = 0$, which implies that the GC is also stable to the original game. Finally, we observe that augmenting A and α leads to an increase in ϵ_1. With respect to the increase in A, the observed increase in ϵ_1 is due to the fact that A, the size of the market, increases the profits of the players and thus, can potentially create larger divergences between W and the approximated partition function values. Concerning the increase of α, we note that it appears in the equilibrium prices (Equation (5.3)) always together with the coalitions sizes. Thus, a larger value of α introduces more variability in the equilibrium prices and, consequently, in the profit of each player which is used to compute W (Equation (5.4)).

Hence, we hypothesize that different embedded coalitions will have significantly different partition function values, resulting in worst approximated MILP-representations.

n	A	α	Chosen partition	ϵ_1	ϵ_2
14	20	0	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13], [14]]	0.00	0.00
14	20	1	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	8.22	0.00
14	20	2	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	21.56	0.00
14	20	3	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	34.02	0.00
14	20	4	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	44.31	0.00
14	20	5	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	52.41	0.00
14	20	6	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	58.62	0.00
14	20	7	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	63.29	0.00
14	20	8	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	69.07	0.00
14	20	9	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	74.24	0.00
14	100	0	[[1, 4, 5, 6, 7, 9, 10, 12], [8, 11, 13], [2], [3], [14]]	0.00	0.00
14	100	1	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	205.57	0.00
14	100	2	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	538.91	0.00
14	100	3	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	850.44	0.00
14	100	4	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	1107.84	0.00
14	100	5	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	1310.34	0.00
14	100	6	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	1465.54	0.00
14	100	7	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	1582.24	0.00
14	100	8	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	1726.74	0.00
14	100	9	[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]	1856.04	0.00

Table 5.1. Snippet of the results for $n = 14$.

In Nagarajan and Sošić [54], the farsighted approach by the authors finds that when $\alpha \to \infty$, the GC is the unique stable outcome, because the products are perfectly substitutable and full cooperation is beneficial for the players. It also finds that when $\alpha = 0$, every partition is stable, because the players are indifferent to cooperation. In our computational experiments, the associated results for $\alpha = 0$ and large values of α lead to the same findings. The comparison with farsighted stability diverges however for small values of α. Their method will allow more partitions to be stable as stated in their Theorem 2. This can be explained by the fact that our algorithm is built to return only one stable outcome, the SW-maximizing partition that is also stable. This is why even with α as small as 0.01, our algorithm returns the GC.

In conclusion, our stability constraints in $\mathcal{F}$, combined to the SW-maximizing component seem to capture the same reasoning as the farsightedness as described by Nagarajan and Sošić [54], since both methods find stability in the GC. Even if our algorithm only replicates a part of that reasoning by focusing on finding the SW-maximizing partition that is also stable according to the constraints in $\mathcal{F}$, we could say that it share some farsighted traits, because it does not contain any stability constraint that is especially protective of unilateral gains (that could have been, for example, Nash Equilibrium stability requirement), since both parties are included for comparison in $\mathcal{F}_2$ (respectively $\mathcal{F}_4$) when there is a split (respectively a fusion). Again, this highlights the need to define accurate stability criteria based on the context of the game.

Conclusion

In Chapter 1, we explain the PFG-related notions that form the basis on which the thesis is built. In order to solve such games, we state in particular the need to tackle two questions: *(i)* “How do players partition themselves?” and *(ii)* “How do players allocate the payoffs attributed to every coalition?”. Thus, the literature review of Chapter 2 targets works addressing one (or both) of those questions. Although PFGs are more general than CFGs, we find that the cooperative game theory literature tilts towards the latter, because they have many applications, but also because of their relative simplicity. We remark that PFGs also do not lack applicability such as inventory pooling [38] or audit sharing [24], to name only a few. Thus, we believe in the well-founded good of having scientists interested in PFGs and pushing their study forward. On the subject of payoff allocations, many concepts developed for CFGs have been extended to PFGs, e.g., the strong-core [15] is an extension of the core [28]. On the subject of stable coalition formation, a wide array of techniques have been used, e.g., the Nash Equilibrium [56], the largest consistent set [17], bargaining methods [53], Markov processes [49]and custom stability constraints [4].

In Chapter 3, we start by proposing a new set of stability constraints in the context of coalition structure formation for PFGs as well as in the context of payoff allocation. Notably, we employ the stability constraints presented in Basso et al. [4] for the formation of a partition, with the addition of one custom stability constraint of our own, because we find they are a good compromise between stability and non-emptiness of the stable partitions set. We also design a core-like approach to find a stable payoff allocation vector that is guaranteed to always return a payoff allocation.

Chapter 4 contains the bulk of the thesis. We present the first compact MILP-representation for SPFGs. We also introduce a framework exploiting the MILP-representation in order to find the SW-maximizing stable partition and a payoff allocation vector to solve the represented SPFG. Given that our MILP-representation has polynomial size on the number of players, it does not reach full expressiveness for all SPFGs. Thus, we characterize the family of SPFGs that our model can represent accurately in Theorem 4. To palliate the issue of not being able to represent exactly all SPFGs, we describe a step allowing to find an approximated MILP-representation for any SPFG. The approximation is based on the least-squares method determining the Moore-Penrose inverse matrix [64]. The chapter ends with the study of the MILP-approximation bounds of our model. Concretely, we give insights on the relation between the partition function approximation error and the violation error of the stability constraints coming from solving the game with the approximated MILP-representation. An observation to make about the representation introduced in this chapter, relative to Lemma 4.2.4, is that it can always express fully the values of any coalition of size $n-3$ to n. This hints that our model is very advantageous for situations where we expect partitions of coarser granularity to form.

Finally, Chapter 5 applies the whole framework to a game described in Nagarajan and Sošić [54], where firms selling (substitutable) products can form coalitions in the hope of improving their individual payoffs. We observe that the solution provided to our approximated MILP-representation is useful to solve the original game, yielding results similar to those of [54]. Indeed, the stability solution concept picked in Chapter 3 captures some far-sightedness, because it includes both parties in the deviations, i.e., not just the coalition deviating, but also the players that are left behind or joined by it. This highlights the importance to pick or design a solution concept appropriated to the context of the game that is studied. Overall, our compact and well-defined MILP-representation, embedded in a flexible framework, manages to find satisfying answers to our two guiding questions.

Future work should address many aspects revolving around the MILP-representation and its framework. A straightforward work direction is the testing of our framework with other solution concepts for payoff allocation and for stable coalition formation as well as other game applications. This could give an even better sense of the performance and value of MILP-representations. Namely, the solving of games coming from real-world contexts through our approximated MILP-representation would allow us to further analyze its performance and compare it with existent approaches. Another direction is related with addressing the existence of multiple ways to encode an SPFG through an MILP-representation. A few constraints could be devised and added to the determination of matrix $\mathbf{A}$ in order to have a unique (approximated) representation given an SPFG. Finally, generalizing the MILP-representation to non-symmetric PFGs would undoubtedly be an important step, as many more applications could find interest in our work.

Références bibliographiques

[1] Anupindi, R., Bassok, Y., and Zemel, E. (2001). A general framework for the study of decentralized distribution systems. *Manufacturing & Service Operations Management*, 3(4):349–368.

[2] Aumann, R. J. (1961). The core of a cooperative game without side payments. *Trans. Amer. Math. Soc*, 98:539–552.

[3] Bartholdi, J. J. and Kemahlioğlu-Ziya, E. (2005). Using shapley value to allocate savings in a supply chain. In Geunes, J. and Pardalos, P. M., editors, *Supply Chain Optimization*, pages 169–208. Springer US, Boston, MA.

[4] Basso, F., Basso, L. J., Rönnqvist, M., and Weintraub, A. (2021). Coalition formation in collaborative production and transportation with competing firms. *European Journal of Operational Research*, 289(2):569–581.

[5] Benedek, M., Fliege, J., and Nguyen, T.-D. (2021). Finding and verifying the nucleolus of cooperative games. *Mathematical Programming*, 190(1):135–170.

[6] Bistaffa, F., Farinelli, A., Cerquides, J., Rodríguez-Aguilar, J., and Ramchurn, S. D. (2014). Anytime coalition structure generation on synergy graphs. AAMAS '14, page 13–20, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Bistaffa, F., Farinelli, A., Chalkiadakis, G., and Ramchurn, S. D. (2017). A cooperative game-theoretic approach to the social ridesharing problem. *Artificial Intelligence*, 246:86–117.

[8] Bitar, E. Y., Baeyens, E., Khargonekar, P. P., Poolla, K., and Varaiya, P. (2012). Optimal sharing of quantity risk for a coalition of wind power producers facing nodal prices. In *2012 American Control Conference (ACC)*, pages 4438–4445. IEEE.

[9] Björck, Å. (2009). *Least squares problems*, pages 1856–1866. Springer US, Boston, MA.

[10] Bloch, F. and van den Nouweland, A. (2014). Expectation formation rules and the core of partition function games. *Games and Economic Behavior*, 88(C):339–353.

[11] Caprara, A. and Letchford, A. N. (2010). New techniques for cost sharing in combinatorial optimization games. *Mathematical Programming*, 124:93–118.

[12] Carraro, C. and Marchiori, C. (2002). Stable coalitions. Technical Report 3258, CEPR Press Discussion Papers.

[13] Carvalho, M., Lodi, A., and Pedroso, J. (2022). Computing equilibria for integer programming games. *European Journal of Operational Research*, 303(3):1057–1070.

[14] Chalkiadakis, G., Wooldridge, M. J., and Elkind, E. (2012). Coalition structure formation. In *Computational aspects of Cooperative Game Theory*, page 87–105. Morgan & Claypool Publishers.

[15] Chander, P. (2014). A core concept for partition function games. Retrieved from `http://www.parkashchander.com/pdf/Strongcore15.pdf`.

[16] Chander, P. and Tulkens, H. (1997). The core of an economy with multilateral environmental externalities. *International Journal of Game Theory*, 26:379–401.

[17] Chwe, M. S.-Y. (1994). Farsighted coalitional stability. *Journal of Economic Theory*, 63:299–325.

[18] Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer Programming*. Springer Cham.

[19] Conitzer, V. and Sandholm, T. (2004). Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, page 219–225. AAAI Press.

[20] Conitzer, V. and Sandholm, T. (2006). Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6):607–619.

[21] Contreras, J. P., Bosch, P., Varas, M., and Basso, F. (2020). A new genetic algorithm encoding for coalition structure generation problems. *Mathematical Problems in Engineering*, 2020.

[22] CPLEX, IBM ILOG (2009). V12. 1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157.

[23] Deng, X. and Fang, Q. (2008). Algorithmic cooperative game theory. In Chinchuluun, A., Pardalos, P. M., Migdalas, A., and Pitsoulis, L., editors, *Pareto Optimality, Game Theory And Equilibria*, pages 159–185. Springer New York, New York, NY.

[24] Fang, X. and Cho, S.-H. (2020). Cooperative approaches to managing social responsibility in a market with externalities. *Manufacturing & Service Operations Management*, 22(6):1215–1233.

[25] Feng, Q., Li, C., Lu, M., and Shanthikumar, J. G. (2022a). Implementing environmental and social responsibility programs in supply networks through multiunit bilateral negotiation. *Management Science*, 68(4):2579–2599.

[26] Feng, Q., Li, Y., and Shanthikumar, J. G. (2022b). Negotiations in competing supply chains: the Kalai-Smorodinsky bargaining solution. *Management Science*, 68(8):5868–5890.

[27] Gabriel, S. A., Siddiqui, S. A., Conejo, A. J., and Ruiz, C. (2013). Solving discretely-constrained nash–cournot games with an application to power markets. *Networks and Spatial Economics*, 13(3):307–326.

[28] Gillies, D. (1959). Solutions to general nonzero sum games. *Annals of Mathematical Studies*, 40:47–85.

[29] Gopalakrishnan, S. and Sankaranarayanan, S. (2022). Cooperative security against interdependent risks. arXiv:2201.04308.

[30] Granot, D. and Sošić, G. (2003). A three-stage model for a decentralized distribution system of retailers. *Operations Research*, 51(5):771–784.

[31] Granot, D. and Sošić, G. (2005). Formation of alliances in internet-based supply exchanges. *Management Science*, 51(1):92–105.

[32] Guichard, D. (2022). An introduction to combinatorics and graph theory.

[33] Gurobi Optimization, LLC (2022). Gurobi Optimizer Reference Manual.

[34] Hardy, G. H. and Ramanujan, S. (1918). Asymptotic formulae in combinatory analysis. *Proceedings of the London Mathematical Society, Second Series*, 17:75–115. Reprinted in Collected papers of Srinivasa Ramanujan, Amer. Math. Soc. (2000), pp. 276–309.

[35] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

[36] Hart, S. and Kurz, M. (1983). Endogenous formation of coalitions. *Econometrica*, 51(4):1047—64.

[37] Ieong, S. and Shoham, Y. (2005). Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, EC '05, page 193–202, New York, NY, USA. Association for Computing Machinery.

[38] Kemahlıoğlu-Ziya, E. and Bartholdi, J. J. (2011). Centralizing inventory in supply chains by using shapley value to allocate the profits. *Manufacturing & Service Operations Management*, 13(2):146–162.

[39] Klusch, M. and Gerber, A. (2002). Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, 17(3):42–47.

[40] Kóczy, L. Á. (2007). A recursive core for partition function form games. *Theory and Decision*, 63(1):41–51.

[41] Kóczy, L. Á. (2009). Sequential coalition formation and the core in the presence of externalities. *Games and Economic Behavior*, 66(1):559–565.

[42] Kóczy, L. A. (2018). *Partition Function Form Games, Coalitional Games with Externalities*. Springer Cham.

[43] Kóczy, L. Á. (2022). Core-stability over networks with widespread externalities. *Annals of Operations Research*, 318:1001–1027.

[44] Kong, X., Tong, X., and Wang, Y. (2021). Min-k-cut coalition structure generation on trust-utility relationship graph. *Wireless Communications and Mobile Computing*, 2021:1–11.

[45] Konishi, H. and Ray, D. (2003). Coalition formation as a dynamic process. *Journal of Economic Theory*, 110:1–41.

[46] Lasisi, R. O. (2017). Overlapping coalition formation in multi-sensor networks. In *FLAIRS Conference*, pages 194–197.

[47] Leng, M. and Parlar, M. (2009). Allocation of cost savings in a three-level supply chain with demand information sharing: A cooperative-game approach. *Operations Research*, 57(1):200–213.

[48] Li, W. (2016). Approximation of the partition number after hardy and ramanujan: An application of data fitting method in combinatorics. arXiv:1612.05526.

[49] Liao, S. S., Zhang, J.-D., Lau, R., and Wu, T. (2014). Coalition formation based on marginal contributions and the markov process. *Decision Support Systems*, 57:355–363.

[50] McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems. *Mathematical Programming*, 10(1):147–175.

[51] Michalak, T., Marciniak, D., Szamotulski, M., Rahwan, T., Wooldridge, M., McBurney, P., and Jennings, N. R. (2010). A logic-based representation for coalitional games with externalities. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, page 125–132, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[52] Mu, L., Hu, B., Reddy, A. A., and Gavirneni, S. (2022). Negotiating government-to-government food importing contracts: A nash bargaining framework. *Manufacturing & Service Operations Management*, 24(3):1681–1697.

[53] Nagarajan, M. and Bassok, Y. (2008). A bargaining framework in supply chains: The assembly problem. *Management Science*, 54:1482—96.

[54] Nagarajan, M. and Sošić, G. (2007). Stable farsighted coalitions in competitive markets. *Management Science*, 53(1):29–45.

[55] Nagarajan, M. and Sošić, G. (2008). Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *European Journal of Operational Research*, 187(3):719–745.

[56] Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.

[57] Norde, H. and Pham Do, K.-H. (2007). The shapley value for partition function form games. *International Game Theory Review (IGTR)*, 09:353–360.

[58] Nudelman, E., Wortman, J., Shoham, Y., and Leyton-Brown, K. (2004). Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. volume 2, pages 880–887.

[59] OEIS Foundation Inc (2022a). $a(n) = \sum_{k=0..n} p(k)$ where $p(k)$ = number of partitions of k (a000041), entry a000070 in the on-line encyclopedia of integer sequences. (visited on 2022-10-06).

[60] OEIS Foundation Inc (2022b). a(n) is the number of partitions of n (the partition numbers), entry a000041 in the on-line encyclopedia of integer sequences. (visited on 2022-09-13).

[61] OEIS Foundation Inc (2022c). Bell or exponential numbers: number of ways to partition a set of n labeled elements, entry a000110 in the on-line encyclopedia of integer sequences. (visited on 2022-09-13).

[62] Olariu, E. F., Frasinaru, C., and Policiuc, A. A. (2020). A branch and bound algorithm for coalition structure generation over graphs. *CoRR*, arXiv:2004.13425.

[63] Ouellet, G. (2002). Systèmes d'équations linéaires. In *Algèbre linéaire : vecteurs et géométrie*, page 95–145. Le griffon d'argile, 2 edition.

[64] Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.

[65] Pepin, J. (2022). MIP-Representation-for-SPFG. `https://github.com/chair-dsgt/MIP-Representation-for-SPFG`.

[66] Pia, A. D., Ferris, M. C., and Michini, C. (2017). Totally unimodular congestion games. In *SODA*.

[67] Pintassilgo, P. and Lindroos, M. (2008). Coalition formation in straddling stock fisheries: A partition function approach. *International Game Theory Review (IGTR)*, 10:303–317.

[68] Präntare, F. and Heintz, F. (2020). An anytime algorithm for optimal simultaneous coalition structure generation and assignment. *Autonomous Agents and Multi-Agent Systems*, 34(1):29.

[69] Rahwan, T., Ramchurn, S., Viet Dung, D., Giovannucci, A., and Jennings, N. (2007). Anytime optimal coalition structure generation. volume 2, pages 1184–1190.

[70] Ray, D. and Vohra, R. (1997). Equilibrium binding agreements. *Journal of Economic Theory*, 73(1):30–78.

[71] Rodríguez-Pereira, J., Balcik, B., Rancourt, M.-E., and Laporte, G. (2021). A cost-sharing mechanism for multi-country partnerships in disaster preparedness. *Production and Operations Management*, 30(12):4541–4565.

[72] Sandholm, T., Gilpin, A., and Conitzer, V. (2005). Mixed-integer programming methods for finding nash equilibria. In *AAAI*, pages 495–501, Pittsburgh, Pennsylvania.

[73] Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal of Applied Mathematics*, 17:1163–1170.

[74] Shapley, L. S. (1951). *Notes on the N-Person Game — II: The Value of an N-Person Game*. RAND Corporation, Santa Monica, CA.

[75] Shenoy, P. P. (1979). On coalition formation: A game-theoretical approach. *Int. J. Game Theory*, 8(3):133–164.

[76] Skibski, O., Michalak, T., Sakurai, Y., Wooldridge, M., and Yokoo, M. (2015). A graphical representation for games in partition function form. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

[77] Stojkovic, M., Soumis, F., and Desrosiers, J. (1998). The operational airline crew scheduling problem. *Transportation Science*, 32:232–245.

[78] Thrall, R. M. and Lucas, W. F. (1963). N-person games in partition function form. *Naval Research Logistics Quarterly*, 10(1):281–298.

[79] Tripathi, R. R. and Amit, R. (2016). Equivalence nucleolus for coalitional games with externalities. *Operations Research Letters*, 44(2):219–224.

[80] Ueda, S., Iwasaki, A., Conitzer, V., Ohta, N., Sakurai, Y., and Yokoo, M. (2018). Coalition structure generation in cooperative games with compact representations. *Autonomous Agents and Multi-Agent Systems*, 32(4):503–533.

[81] Voice, T., Polukarov, M., and Jennings, N. R. (2012). Coalition structure generation over graphs. *Journal of Artificial Intelligence Research*, 45:165–196.

[82] von Neumann, J. and Morgenstern, O. (1944). Theory of games and economic behavior. *Princeton University Press.*

[83] Westerink-Duijzer, L. E., Schlicher, L. P. J., and Musegaas, M. (2020). Core allocations for cooperation problems in vaccination. *Production and Operations Management*, 29(7):1720–1737.

[84] Wolsey, L. A. (2020). *Integer Programming.* Wiley.

[85] Yi, S.-S. (1997). Stable coalition structures with externalities. *Games and Economic Behavior*, 20:201–237.

[86] Zha, A., Nomoto, K., Ueda, S., Koshimura, M., Sakurai, Y., and Yokoo, M. (2017). Coalition structure generation for partition function games utilizing a concise graphical representation. In An, B., Bazzan, A., Leite, J., Villata, S., and van der Torre, L., editors, *PRIMA 2017: Principles and Practice of Multi-Agent Systems*, pages 143–159, Cham. Springer International Publishing.

[87] Zhao, J. (1992). The hybrid solutions of an n-person game. *Games and Economic Behavior*, 4:145–160.

Appendix A

McCormick envelope implementation

Gurobi copes well with the formulation presented in Chapter 4. Nevertheless, we implemented the McCormick envelope [50] to linearize the bilinear terms and we tested whether the solver could perform better when inputting the obtained linear model. Preliminary results showed that it slightly slowed down Gurobi which possibly performs this linearization itself. Thus, we decided to leave this implementation aside, but describe how we can make it in the present appendix.

This model takes after the one introduced in Chapter 4, but makes it linear by using the McCormick envelope. Given that we have our parameters n (the number of players) and $\mathbf{A} \in \mathbb{R}^{n \times n}$ (square, upper-triangular and continuous matrix describing the SPFG), we have the following variables:

- $\mathbf{X} \in \{0,1\}^{n \times n}$ is a binary square matrix where the rows represent the players and the columns represent the coalitions. There are ones where a given player is in a given coalition.
- $\vec{q} \in \{0,1\}^{n-1}$ is a binary vector where if there is a one at index j, it means the coalitions in $\mathbf{X}$ at indices j and $j+1$ are of the same cardinality.
- $\mathbf{M} \in \{0,1\}^{(n+1) \times n}$ is a binary rectangular matrix where the rows represent the number of players and the columns represent the coalitions. There will be a one at $\mathbf{M}_{ij}$ if coalition j has $n-i+1$ players, and there will be a zero otherwise.
- $\mathbf{W} \in \{0,1\}^{n \times n \times (n+1) \times n}$ is a four-dimension matrix whose elements represent all different products that are generated by multiplying an element of $\mathbf{X}$ with an element of $\mathbf{M}$. Namely, we write $\mathbf{W}_{irkj} = \mathbf{X}_{ir}\mathbf{M}_{kj}$.

We can write the objective using the definition of $\mathbf{W}$ in order to linearize it.

$$\max_{\mathbf{W}} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{k=1}^{n+1} \sum_{r=1}^{n} \mathbf{W}_{irkj} [\mathbf{A}|\mathbf{0}]_{rk}$$

The augmented matrix $[\mathbf{A}|\mathbf{0}]$ correspond to matrix $\mathbf{A}$ with a column of zeros appended to it.

Constraints. This first block of constraints regards **X**. It makes sure every row only has one "1" (as each player can only belong to one coalition), and orders the coalition from the most numerous to the least. The players' indices are used as a tiebreaker.

$$\sum_{j=1}^{n} \mathbf{X}_{ij} = 1 \qquad \forall i \in \{1, ..., n\} \tag{A.1}$$

$$\sum_{i=1}^{n} \mathbf{X}_{ij} \geq \sum_{i=1}^{n} \mathbf{X}_{i(j+1)} \qquad \forall j \in \{1,..., n-1\} \tag{A.2}$$

$$\sum_{i=1}^{n} \left(\mathbf{X}_{ij} - \mathbf{X}_{i(j+1)} \right) \geq 1 - \vec{q}_j \qquad \forall j \in \{1,..., n-1\} \tag{A.3}$$

$$\begin{bmatrix} 2^{n-1} & 2^{n-2} & ... & 2^0 \end{bmatrix} \times \mathbf{X}_{*j} \geq \begin{bmatrix} 2^{n-1} & 2^{n-2} & ... & 2^0 \end{bmatrix} \times \mathbf{X}_{*j+1} - \bar{L}(1 - \vec{q}_j) \quad \forall j \in \{1,..., n-1\}, \tag{A.4}$$

$$\sum_{i=1}^{n+1} \mathbf{M}_{ij} = 1 \qquad \forall j \in \{0,..., n\} \tag{A.5}$$

$$\begin{bmatrix} 1 & 1 & ... & 1 & 1 \end{bmatrix}_{1 \times n} \times \mathbf{X} = \begin{bmatrix} n & n-1 & ... & 1 & 0 \end{bmatrix}_{1 \times n+1} \times \mathbf{M} \tag{A.6}$$

$$\mathbf{X} \in \{0,1\}^{n \times n}, \quad \vec{q} \in \{0,1\}^{n-1}, \quad \mathbf{M} \in \{0,1\}^{(n+1) \times n}. \tag{A.7}$$

We also add the constraints applying to **W**'s elements. They happen to be binary because **X** and **M** also are binary.

$$\mathbf{W}_{irkj} \geq \mathbf{X}_{ir} + \mathbf{M}_{kj} - 1 \qquad \forall i,r,j \in \{0,...,n\} \text{ and } k \in \{0,...,n+1\} \tag{A.8}$$

$$\mathbf{W}_{irkj} \geq 0 \qquad \forall i,r,j \in \{0,...,n\} \text{ and } k \in \{0,...,n+1\} \tag{A.9}$$

$$\mathbf{W}_{irkj} \leq \mathbf{X}_{ir} \qquad \forall i,r,j \in \{0,...,n\} \text{ and } k \in \{0,...,n+1\} \tag{A.10}$$

$$\mathbf{W}_{irkj} \leq \mathbf{M}_{kj} \qquad \forall i,r,j \in \{0,...,n\} \text{ and } k \in \{0,...,n+1\} \tag{A.11}$$

$$\mathbf{X}_{ir} \in \{0,1\} \qquad \forall i,r \in \{0,...,n\} \tag{A.12}$$

$$\mathbf{M}_{kj} \in \{0,1\} \qquad \forall j \in \{0,...,n\} \text{ and } k \in \{0,...,n+1\} \tag{A.13}$$

$$0 \leq \mathbf{W}_{irkj} \leq 1 \qquad \forall i,r,j \in \{0,...,n\} \text{ and } k \in \{0,...,n+1\} \tag{A.14}$$

This model is now ready to be used just like the one described in Chapter 4.

Appendix B

Stability constraints for an MILP-representation

In step (3) of the procedure determining a stable partition (Section 4.1.3), the stability constraints in $\mathcal{F}$ for the partition $\mathbf{X}^*$, written with our MILP-representation notation are as follows:

$$\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^* \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^* \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}_{n\times 1} \geq \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^{SP} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^{SP} \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}_{n\times 1}, \tag{B.1a}$$

$$\left(\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^* \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^*\right)_i \geq \sum_{j:\mathbf{X}^*_{ji}=1} \left(\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^{SP} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^{SP}\right)_j \quad \forall i = 1,2,\dots,n, \tag{B.1b}$$

$$\sum_{j:\mathbf{X}^*_{*j}\notin\mathbf{X}^{neigh}} \left(\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^* \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^*\right)_j \geq \sum_{j:\mathbf{X}^{neigh}_{*j}\notin\mathbf{X}^*} \left(\begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^{neigh} \times [\mathbf{A}|\mathbf{0}] \times \mathbf{M}^{neigh}\right)_j$$
$$\forall \mathbf{X}^{neigh} : \left(\mathbf{X}^{neigh} \in \mathbf{X}^{split}\right) \vee \left(\mathbf{X}^{neigh} \in \mathbf{X}^{merge}\right), \tag{B.1c}$$

$$\mathbf{X}^{split} : \left(\exists! j : \mathbf{X}^*_{*j} \notin \mathbf{X}^{split}, \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^*_{*j} \geq 2\right) \wedge$$
$$\left(\exists! i, \exists! k : \mathbf{X}^{split}_{*i} \notin \mathbf{X}^*, \mathbf{X}^{split}_{*k} \notin \mathbf{X}^*, \mathbf{X}^{split}_{*i} + \mathbf{X}^{split}_{*k} = \mathbf{X}^*_{*j}\right), \tag{B.1d}$$

$$\mathbf{X}^{merge} : \left(\exists! j : \mathbf{X}^{merge}_{*j} \notin \mathbf{X}^{*}, \begin{bmatrix}1 & 1 & \dots & 1\end{bmatrix}_{1\times n} \times \mathbf{X}^{merge}_{*j} \geq 2\right) \wedge \left(\exists! i, \exists! k : \mathbf{X}^{*}_{*i} \notin \mathbf{X}^{merge}, \mathbf{X}^{*}_{*k} \notin \mathbf{X}^{merge}, \mathbf{X}^{*}_{*i} + \mathbf{X}^{*}_{*k} = \mathbf{X}^{merge}_{*j}\right), \tag{B.1e}$$

where operator $\notin$ between a column and a matrix means there is no such column in the matrix. Constraint (B.1a) is equivalent to $\mathcal{F}_1$ and ensures the candidate partition $\mathbf{X}^{*}$ has a larger total payoff than the SP. Constraints (B.1b) are equivalent to $\mathcal{F}_3$ and ensure that each coalition in the candidate partition has a larger payoff than its equivalent in the SP. Constraints (B.1c) is equivalent to $\mathcal{F}_2$ when we use $\mathbf{X}^{neigh} \in \mathbf{X}^{split}$ (B.1d) and to $\mathcal{F}_4$ when we use $\mathbf{X}^{neigh} \in \mathbf{X}^{merge}$ (B.1e). They ensure that given a neighbourhood partition, no coalition will have incentive to deviate from the candidate partition.