Technical University of Crete, Greece
School of Electrical and Computer Engineering

Extending the MC-nets representation scheme to cooperative game settings with uncertainty or overlaps

Errikos Streviniotis

Thesis Committee
Supervisor: Associate Professor Georgios Chalkiadakis (ECE)
Associate Professor Michail G. Lagoudakis (ECE)
Associate Professor Vasileios Samoladas (ECE)

Chania, November 2020

Abstract

Cooperative game theory studies how self rational agents interact with each other in order to form coalitions and achieve a common goal by collaborating, while maximising their own profits. Representing the collaborations efficiently is key for supporting coalition formation decisions and achieving tractable computation in cooperative game settings. Moreover, research in cooperative games often assumes that there is no uncertainty in the settings of the game, or that agents cannot participate in many coalitions simultaneously.

In this thesis, we focus on a well-known coalitional representation scheme, the MC-nets representation, and extend it to settings where we remove the aforementioned unrealistic assumptions, namely having complete information and no-overlaps.

We begin by extending the Relational Rules representation, a scheme that itself extends MC-nets to cooperative games with overlapping coalitions, so that it now includes both positive and negative literals. Our proposed representation reduces to the classic MC nets representation for non-overlapping environments.

We then introduce a novel succinct representation scheme for cooperative games under uncertainty, the ε-MC nets. The proposed representation is discussed first in the context of transferable utility games, and exploits estimates over marginal contributions to form compact rules representing collaboration patterns with potentially uncertain value.

In more detail, given a set of MC-nets rules that use prior beliefs over values instead of the actual ones, we provide a polynomial algorithm for reaching the proposed succinct representation. We provide theoretical results regarding the information loss (regarding the perceived value of the agent collaboration patterns) after the compression of the original representation of the set. We show that the loss from compressing the set is bounded by a value directly proportional to ε, which represents an error regarding the believed value of an original rule which we are willing to accept in order to compress the representation.

We then extend our algorithm to exploit equivalence classes of agents. This allows us to obtain an even more compact representation, and to derive new, previously unheld beliefs over the value of unobserved agent collaboration patterns. Moreover, we show that our approach extends naturally to non-transferable utility games.

We conduct a systematic experimental evaluation of our algorithm's variants, studying its behaviour in various realistic settings, and provide results on the compression achieved in each evaluated setting. Our experimental results confirm the effectiveness of our approach.

Abstract in Greek

Η συνεργατική θεωρία παιγνίων μελετά πως ορθολογικοί πράκτορες αλληλεπιδρούν μεταξύ τους προκειμένου να δημιουργήσουν συνασπισμούς ώστε συνεργαζόμενοι να πετύχουν κάποιο κοινό στόχο, ενώ παράλληλα μεγιστοποιούν τις ατομικές τους απολαβές. Η αποδοτική αναπαράσταση των συνεργασιών στα πλαίσια ενός συνασπισμού, είναι σημαντική για την υποστήριξη λήψης αποφάσεων σχετικά με τη δημιουργία των συνασπισμών, και γενικότερα για την επίτευξη υπολογισμών σε συνεργατικά παίγνια. Επιπλέον, στην σχετική με συνεργατικά παίγνια έρευνα, χρησιμοποιούνται συχνά οι παραδοχές ότι δεν υπάρχει αβεβαιότητα στις ρυθμίσεις του παιχνιδιού ή/και ότι οι πράκτορες δεν μπορούν να συμμετέχουν σε πολλούς συνασπισμούς ταυτόχρονα.

Στην παρούσα διπλωματική εργασία, επικεντρωνόμαστε σε ένα από τα πλέον γνωστά σχήματα αναπαράστασης συνεργατικών παιγνίων, την αναπαράσταση MC-nets, και το επεκτείνουμε ώστε να αφαιρέσουμε τις προαναφερθείσες μη-ρεαλιστικές παραδοχές (δηλαδή, αυτή της πλήρους πληροφόρησης και της μη ύπαρξης επικαλύψεων).

Η πρώτη μας συνεισφορά συνίσταται στην επέκταση της αναπαράστασης Relational Rules, ενός πρόσφατα διατυπωμένου σχήματος αναπαράστασης που επέκτεινε εν μέρει τα MC-nets σε συνεργατικά παίγνια με επικαλυπτόμενους συνασπισμούς χρησιμοποιώντας λογικές προτάσεις με θετικά μόνο λεκτικά. Η προτεινόμενη στην παρούσα διπλωματική επέκταση, κατορθώνει να επεκτείνει πλήρως τα MC-nets, χρησιμοποιώντας για την αναπαράσταση συνεργασιών λογικές προτάσεις που μπορεί να περιλαμβάνουν τόσο θετικά όσο και αρνητικά λεκτικά. Η προτεινόμενη αναπαράσταση, αντιστοιχεί επακριβώς στην κλασική MC-nets αναπαράσταση σε περιβάλλοντα με μη επικαλυπτόμενους συνασπισμούς.

Στη συνέχεια, προτείνουμε ένα νέο, περιεκτικό σχήμα αναπαράστασης για συνεργατικά παίγνια με αβεβαιότητα, το οποίο καλούμε ε-MC nets. Η προτεινόμενη αναπαράσταση ορίζεται αρχικά στο πλαίσιο των παιγνίων με μεταβιβάσιμη αξία (transferable utility), και εκμεταλλεύεται εκτιμήσεις περιθωρίων συνεισφορών πρακτόρων για να σχηματίσει συμπαγείς κανόνες αναπαριστώντας μοτίβα συνεργασίας με ενδεχομένως αβέβαιη αξία.

Πιο συγκεκριμένα, δεδομένου ενός σετ από MC-nets κανόνες που χρησιμοποιούν πρότερες εκτιμήσεις αξίας συνεργασιών, παρέχουμε έναν πολυωνυμικό αλγόριθμο που επιτυγχάνει την προτεινόμενη συμπαγή αναπαράσταση. Επιπροσθέτως, παρέχουμε θεωρητικά αποτελέσματα σχετικά με την απώλεια πληροφορίας (όσον αφορά τις εκλαμβανόμενες αξίες των μοτίβων συνεργασίας πρακτόρων) μετά τη συμπίεση της αρχικής αναπαράστασης για το σετ κανόνων, δείχνοντας ότι φράσσεται από μια τιμή ευθέως ανάλογη με το αποδεκτό περιθώριο απόστασης από την αξία ενός αρχικού MC-net κανόνα.

Κατόπιν, επεκτείνουμε τον αλγόριθμό μας ώστε να εκμεταλλεύεται την ύπαρξη κλάσεων

ισοδυναμιών των πρακτόρων. Αυτό μας επιτρέπει να αποκτήσουμε μια ακόμα πιο συμπαγή αναπαράσταση, καθώς και να παράγουμε νέες, προηγουμένως ανύπαρκτες πεποιθήσεις για τις αξίες μη-παρατηρήσιμων μοτίβων συνεργασίας πρακτόρων. Επιπλέον, δεικνύουμε ότι η προσέγγιση μας μπορεί να επεκταθεί και σε συνεργατικά παίγνια με μή μεταβιβάσιμη αξία (non-transferable utility games).

Τέλος, διεξάγουμε μια συστηματική πειραματική αξιολόγηση των παραλλαγών του αλγορίθμου μας, μελετώντας τη συμπεριφορά τους μέσω προσομοιώσεων σε ποικίλα ρεαλιστικά περιβάλλοντα, και παρέχουμε αποτελέσματα σχετικά με την συμπίεση που επιτυγχάνεται σε κάθε περίπτωση. Τα πειραματικά μας αποτελέσματα επιβεβαιώνουν την αποτελεσματικότητα της προσέγγισής μας.

Acknowledgements

First and foremost, I would like to deeply thank my supervisor, Professor Georgios Chalkiadakis, for all his guidance and trust throughout this thesis, and for motivating me to continue my studies on topics related to Artificial Intelligence. I also thank the members of the Intelligent Systems Laboratory, Dimitris and Antonis, for our fruitful discussions, and especially Athina for her help and our excellent collaboration.

Second, I wish to thank the members of the committee, Professor Michael G. Lagoudakis and Professor Vasilis Samoladas, for their helpful comments and time.

Finally, i wish to thank all of my friends and family for their unconditional love and support, not only throughout my studies, but also in every part of my whole life.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Multiagent systems [1] or simply MAS, constitutes a well-studied field of artificial intelligence, and aims to provide solutions to problems that are difficult or even impossible for an individual agent. MAS research captures many real life scenarios as online trading, disaster response, logistics, smart grids etc; often modeling them using game theoretic paradigms. Cooperative game theory [2] in particular, has attracted the interest of many MAS researchers, since it provides a rich framework for the coordination of the actions of self-interested agents in strategic settings. In general, cooperative or coalitional games [2] capture settings where individuals need to form coalitions in order to fulfil some complicated task, which they would not be able to accomplish on their own or they can achieve better outcomes.

In order to capture coalitional games and perform any kind of computation, we need to find an efficient way to represent such games. The naive representation, lists every coalition together with its value, requiring space exponential in the number of agents in the game. As such, it is critical to find more succinct representation schemes for coalitional games. In the past, many researchers focused their interest to find efficient ways to represent coalitional games [3, 4, 5]. Ieong and Shoham [3], introduced the MC-nets representation, a *complete* representation language for characteristc function games. The main idea is to decompose the game into a set of rules that assign marginal contributions to groups of agents.

Now, a common assumption in cooperative game theory is that coalitions have to be disjoint—i.e., an agent participates in exactly one coalition at the time. Nevertheless, such an assumption can be considered extremely restrictive, since in many real-life scenarios an agent can be a member of many coalitions at the same time. Supposing that each coalition is associated with tasks to be performed by its members, many agents may be involved with more than one task. In that case, agents may need to distribute their resources among different coalitions, in an efficient way, in order to fulfil their goals. For example, consider online trading agents representing individuals, and facing the challenge of allocating their owners' capital to a variety of projects (each funded by a coalition of investors) at the same time [2]. Therefore, it is natural to consider that each agent holds an amount of divisible resources (i.e., money, computational power etc.), which can invest into different coalitions simultaneously in order to achieve his goals and to maximise his profits. Such a scenario results in coalition structure with *overlapping coalitions* [6, 7].

In [8], the authors inspired by [3], introduced the Relational Rules (RRs), a representation scheme for overlapping coalitions. Similarly to [3], a game is described by a set of rules, while the utility of a coalition C can be computed, by summing the values of rules that apply to C. In addition with the classic MC nets, the values of rules are not standard, since affected by the portions of agents that participate in C. As such, we provide an extension of RRs, that includes both positive and negative literals, that reduces (like [8]) to MC nets for *non-overlapping* environments.

Another interest line of research, is to find schemes for representing large coalitional games in an efficient way. Specifically, as the number of individuals scales up, the number of different possible coalitions one may participate in rises exponentially. Moreover, in large open multiagent systems, we may have hundreds or even thousands of agents which form coalitions in order to perform complex tasks. In such large settings, it is unrealistic to assume that we can have complete knowledge over every possible collaboration pattern between the agents. As such, it is natural to assume that we are in a partially observed environment, and therefore we have *beliefs* (estimates) *over the value of potential collaboration patterns*. Fully representing such multiagent systems can be extremely inefficient as the number of agents rises, and such taking into consideration that the environment is not fully observable.

In this light, here we provide a novel representation that encodes the prior beliefs of the agents over the value of some observed collaboration patterns in a succinct way. In order to do so, we exploit similarities on agents' behaviour when they work with each other, retaining the uncertainty over our beliefs within acceptable limits, with $\varepsilon \in \mathbb{R}_+$

signifying how far away from our perceived value of a collaboration pattern we are willing to deviate in order to compress an original representation. Our proposed representation, the $\varepsilon-$MC nets, uses the ideas above while building on the classic MC-nets representation, in order to capture collaboration patterns with similar values among similar agents, and encode them into compact rules.

1.1 Contributions

Our contributions in this paper are as follows. First, we generalize the proposed representation scheme of [8], for cooperative games with *overlapping coalitions*. Specifically we extend this representation scheme, which is build on the idea of well-known MC-nets [3], in a way that includes both positive and negative literals in the patterns of the rules. At the same time, our proposed extension, achieves the desirable reduction to classic MC nets representation for *non-overlapping* settings.

Following that, we propose a novel representation for cooperative games under the form of uncertainty we described earlier. We study the complexity of the algorithm, and provide theoretical results regarding the information loss (regarding the perceived value of agent collaboration patterns) after the representation's compression, showing that it is bounded by a value directly proportional to ε. We then extend this algorithm in a way that exploits "equivalence classes" of agents, in order to produce an even more compact representation of the game. This variant of the algorithm can also *produce* new, previously unknown, collaboration patterns among agents. Our approach extends naturally to non-transferable utility games. Finally, we conduct a systematic evaluation of the algorithm, studying its behaviour in various realistic settings, and provide experimental results on the percentage of reduction in each evaluated setting. Our experimental results confirm the effectiveness of our approach in environments with this particular form of uncertainty.

1.2 Outline

In Chapter 2 we present all the necessary theoretical background for this thesis. We present the basic aspects of *cooperative game theory* and the *overlapping coalitions*, present various representation schemes , like MC nets, and discuss their use in various settings. In Chapter 3 we extend the Relational Rules (RRs) [8] representation in a way that considers both positive and negative literals, while preserving the desirable characteristics of the RRs representation (including their reduction to classic MC nets for non-overlapping settings). In Chapter 4 we introduce our novel representation scheme for cooperative games under uncertainty. We design an algorithm that reaches such representations, and we provide theoretical results regarding the information loss that this representation may have. We then extend this algorithm, so that it exploits equivalence classes of agents. In Chapter 5 we evaluate the performance of our proposed algorithms concerning the reduction percentage (i.e., the number of rules comprising the new representation compared to the number of the initial representation) that they achieve. Chapter 6 acts as an epilogue for this thesis, presenting our conclusions along with future directions of work. Finally, in Chapter 7 we extend the classic MC nets representation to NTU environments and more specifically to *Hedonic Games*, while we discuss the *kernel* stability concept.

Chapter 2

Theoretical Background

In this chapter we discuss the theoretical background required for this thesis.

2.1 Cooperative Games

Game theory provides a mathematical framework for the analysis of self-interested computational entities interactions. In cooperative (or coalitional) games, agents cooperate with each other in order to achieve a specific goal and maximise their profits. By this collaboration agents may be able to achieve goals that were impossible if they acted individually. It is also usual that agents can gain more by cooperating with other agents.

As such, in cooperative game theory agents form some groups, which are called *coalitions*. Every coalition consists of a set of agents, that make decisions and act as a whole. More formally, for a non-empty set of agents $N = \{1, \cdots, n\}$, a coalition C is any subset of N ($C \subseteq N$). It is natural to say that in cooperative games, actions are taken by group of agents. If a group consists of all the agents then it is called the *grand coalition*, while if it consists of only one agent it is called a *singleton*.

2.1.1 Characteristic Function Games

Characteristic function game (CFG) is a widely-studied subclass of cooperative games. Such games are populated by a non-empty set N of agents. Moreover any *CFG* has a characteristic function, known as utility function, which assigns a numeric value to every possible coalition.

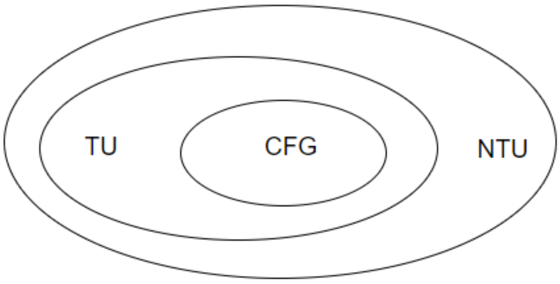

Figure 2.1: Classes of Cooperative Games.

Formally we define a *characteristic function game* as:

Definition 1 ([2]) *A characteristic function game G is given by a pair (N, v), where $N = \{1, ..., n\}$ is a finite, non-empty set of agents and $v : 2^N \rightarrow \mathbb{R}$ is a characteristic function, which maps each coalition $C \subseteq N$ to a real number $v(C)$. The number $v(C)$ is usually referred to as the value of the coalition C.*

Usually in *characteristic function games* we consider that the coalition value $v(C)$ can be divided among the members that participate in C in any way. Games with such property are said to be *transferable utility games (TU games)*. As such, the outcome of a TU game, is a pair $\langle CS, x\rangle$, where CS is a coalition structure, i.e., a partition of N into coalitions, and x is a payoff vector which distributes the value among agents.

Definition 2 ([2]) *Given a characteristic function game $G = (N, v)$, a coalition structure over N is a collection of non-empty subsets $CS = \{C^1, ..., C^k\}$ such that*

- $\bigcup_{j=1}^{k} C^j = N$*, and*
- $C^i \cap C^j = \emptyset$*, for any $i, j \in \{1, ..., k\}$ such that $i \neq j$.*

A vector $x = (x_1, ..., x_n) \in \mathbb{R}^n$ is a payoff vector for a coalition structure $CS = \{C^1, ..., C^k\}$ over $N = \{1, ..., n\}$ if

- $x_i \geq 0$ *for all $i \in N$, and*
- $\sum_{i \in C^j} x_i \leq v(C^j)$ *for any $j \in \{1, ..., k\}$.*

An outcome of G is a pair (CS, x), where CS is a coalition structure over G and x is a payoff vector for CS.

2.1.2 Non-Transferable Utility Games

Another interesting setting of cooperative games, are the instances where the utility cannot be transferred among the agents who collaborate, but intuitively it is assigned to each member directly following a coalitional action. Those games are known as *non-transferable utility games (NTU games).*

Formally, in such games, every coalition has a set of *choices* or *consequences* $\Lambda = \{\lambda_1, \lambda_2, ...\}$ and the agents have *preferences* over these choices, captured by preference relations. As such each agent $i \in N$ is associated with a preference relation $\succeq_i$.

Definition 3 ([2]) *A preference relation on Λ is a binary relation $\succeq \subseteq \Lambda \times \Lambda$, which is required to satisfy the following properties:*

- *Completeness: For every $\{\lambda, \lambda'\} \subseteq \Lambda$, we have $\lambda \succeq \lambda'$ or $\lambda' \succeq \lambda$;*
- *Reflexivity: For every $\lambda \in \Lambda$, we have $\lambda \succeq \lambda$; and*
- *Transitivity: For every $\{\lambda_1, \lambda_2, \lambda_3\} \subseteq \Lambda$, if $\lambda_1 \succeq \lambda_2$ and $\lambda_2 \succeq \lambda_3$ then $\lambda_1 \succeq \lambda_3$.*

We say that a choice λ is preferred at least as much as choice λ' if and only if $\lambda \succeq \lambda'$. That is, a non-transferable utility game can be defined as:

Definition 4 ([2]) *A non-transferable utility game (NTU game) is given by a structure $G = (N, \Lambda, \succeq_1, \cdots, \succeq_n)$, where $N = \{1, \cdots, n\}$ is a non-empty set of players, $\Lambda = \{\lambda, \lambda_1, \cdots\}$ is a non-empty set of choices, $v : 2^n \to 2^\Lambda$ is the characteristic function of G, which for every coalition C defines the choices $v(C)$ available to C, and, for each player $i \in N$, $\succeq_i \subseteq \Lambda \times \Lambda$ is a preference relation on Λ.*

2.1.3 Partition Function Games

In many real life scenarios, the choices that a coalition C can make (or, in TU settings, the payoff that C can earn) may depend on the coalition structure formed by all agents in N, that is, the set of all coalitions that form a partition π of the game. Such scenarios are modeled by *partition function games* or games with *externalities.*

A *partition* (or coalition structure) is a set of coalitions $\pi = \{C_1, \cdots, C_m\}$ such that for every $i, j = \{1, \cdots, m\}$ the following conditions must be satisfied:

- $C_i \cap C_j = \emptyset$, when $i \neq j$.

- $\bigcup_{i\in 1,\cdots,m} C_i = N$.

In these games, the value of a coalition depends on the coalition structure that it appears in. As such, in [2], authors define an *embedded coalition* as:

Definition 5 ([2]) *An embedded coalition over N is a pair of the form (C,CS), where CS is a coalition structure over N, and* $C \in CS$*.*

The set of all embedded coalitions over N is denoted with $\mathcal{E}_N$, while the set of the embedded coalitions that contain agent i is denoted with $\mathcal{E}_N(i)$.

Partition function games can be defined for both *transferable* and *non-transferable* utility settings.

Definition 6 ([2]) *A partition function game G is given by a pair (N,u), where* $N = \{1, \cdots, n\}$ *is a finite non-empty set of agents and* $u : \mathcal{E}_N \to \mathbb{R}$ *is a mapping that assigns a real number* $u(C, CS)$ *to each embedded coalition (C,CS).*

Definition 7 ([9]) *A coalitional game in partition function form (PFF) with non-transferable utility (NTU) is defined by a pair* $\langle N, V\rangle$*, where N is the set of players, and V is a mapping such that for every* $\pi \in \Pi$ *and every coalition* $C \subseteq N$*,* $C \in \pi$*,* $V(C, \pi)$ *is a closed convex subset of* $\mathbb{R}^{|C|}$ *that contains the payoff vector that players in S can achieve. Alternatively, if we consider a payoff vector in* $\mathbb{R}^n$ *for every coalition* $C \subseteq N$ *(let for any* $i \notin C$ *the corresponding payoff be 0 or* $-\infty$*), then V can be viewed as a mapping* $V : \mathcal{E}_N \to \mathbb{R}^n$ *that assigns to n-vector of real numbers to each embedded coalition* (C, π)*.*

2.1.4 Hedonic Games

Hedonic games [10] form a subclass of NTU games in which agents have *preferences* over the coalitions in which they can participate. Essentially each agent has *preferences* over her collaboration with the others. As such, the *payoff* of each agent, corresponds to their satisfaction from the collaboration itself, while the outcome of such games is a coalition structure. More formally, we define a hedonic game as:

Definition 8 ([11]) *Let N be a finite set of agents. A coalition is a non-empty subset of N. Let* $N_i = \{S \subseteq N : i \in S\}$ *be the set of all coalitions (subsets of N) that include agent* $i \in N$*. A coalition structure is a partition* π *of agents N into disjoint coalitions. A hedonic coalition formation game is a pair* $(N, \succeq)$*, where* $\succeq$ *is a preference profile that* *specifies for every agent* $i \in N$ *a reflexive, complete, and transitive binary relation* $\succeq_i$ *on* N_i. *We call* $\succeq_i$ *a preference relation.*

In [12] authors provide the following definition for *hedonic games in partition function form.*

Definition 9 **([12])** *A hedonic game (HG) in partition function form (PFF) is defined by a pair* $\langle N, \succeq \rangle$, *where* N *is the set of players, and* $\succeq = \{\succeq^{\pi_1}, ..., \succeq^{\pi_m}\}$ *with* $|\Pi| = m$; *and for all* $\pi_j \in \Pi$ $\succeq^{\pi_j} = \{\succeq_1^{\pi_j}, ..., \succeq_n^{\pi_j}\}$, *and each* $\succeq_i^{\pi_j} \subseteq N_i \times N_i$ *is a complete, reflexive and transitive preference relation describing agent* i*'s preferences over coalitions it can participate in when* π_j *is in place.*

2.2 Overlapping Coalition Formation

Overlapping coalition formation possibly first appeared as a term in [13]. The model assumes that agents have specific goals and capabilities and agents have to form coalitions in order to achieve the goals. Each agent contribute some of his capabilities in each coalition that he participates. Dang et al. [6] presented a work that uses a greedy algorithm for overlapping coalition formation in a multi-sensor network.

In [7], Chalkiadakis et. al. provide a model, where an agent holds some resources that can distribute among different coalitions simultaneously. Their work extended the classic cooperative games to cooperative games with overlapping coalitions—or *overlapping coalition formation games*. In such model, the value of any coalition depends not only on the agents that are its members, but also to the amount of their resources that they contribute to it.

Formally they define an *overlapping coalition formation game* (OCF game) as:

Definition 10 **([7])** *An OCF-game* G *with player set* $N = \{1, \cdots, n\}$ *is given by a function* $v : [0, 1]^n \to \mathbb{R}$, *where* $v(0^n) = 0$.

"This function v is defined on *partial coalitions*, i.e., vectors of the form $r = (r_1, \cdots, r_n)$, where r_i is the fraction of agent i's resources contributed to this coalition; function v maps any such coalition r to a corresponding payoff" [2].

Chalkiadakis et. al. [7] also studied the *stability* of their model. Specifically, they define three stability concepts for such games—the *conservative core*, the *refined core* and the *optimistic core*, which generalize the classic rationality solution concept of the core [14] to overlapping settings. Finally in [15], authors proposed the *arbitrated core*, a notion that encodes the three types of core of [7].

2.3 Representation schemes

The naive solution in order to represent a coalitional game is to enumerate the payoffs to each set of agents. As such, the required space rises exponentially to the number of agents, making such an approach impractical for many real-world scenarios. Due to this problem, many researchers focused their efforts on finding efficient ways to represent coalitional games. In what follows, we briefly describe the most celebrated of those representations.

2.3.1 Induced Subgraph games

This representation was introduced by Deng and Papadimitriou [4] and uses a weighted undirected graph in order to represent a game. Formally, a game is described by a weighted undirected graph $G = (N, E)$, where each node represent an agent $i \in N$. Each edge between two nodes i and j is denoted as $w_{i,j}$. The value of coalition C, denoted as $u(C)$ can be computed as:

$$u(C) = \sum_{\{i,j\} \in C \cap E} w_{i,j}$$

The representation includes self-loops, since the value of any *singleton* can be non-zero.

Example 1 *Given an induced subgraph game (Figure 2.2) with $N = \{1, 2, 3, 4\}$ the value of coalitions $C_1 = \{1, 3, 4\}$, $C_2 = \{1, 2, 3\}$ and $C_3 = \{1, 2, 3, 4\}$ can be computed as:*

- $C_1 = 5 + 7 = 12.$
- $C_2 = 5 + 1 + 2 = 8.$
- $C_3 = 1 + 2 + 5 + 7 = 15.$

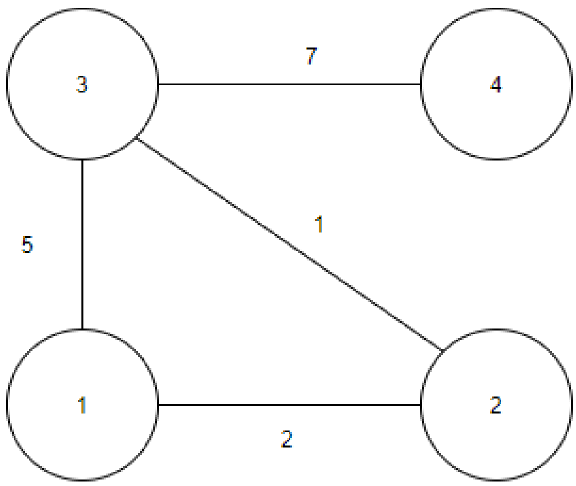

Figure 2.2: Induced subgraph representation of Example 1.

However the Induced Subgraph representation is not a complete representation scheme, since in the general case the value of any coalition is not determined exclusively by pairs of agents and cannot be represented by such a graph.

2.3.2 Marginal Contribution Nets

Marginal Contribution Nets, or simply MC nets, were introduced by Ieong and Shoham in [3] and constitute a complete representation scheme for coalitional games with transferable utility. It is widely used because of their simplicity and their ability to be fully expressive. The basic idea of MC nets is to represent a coalitional game by a set of rules. Every rule has the following form:

$$Pattern \rightarrow Value$$

where *Pattern* is a Boolean expression that consists only conjuctions of agents and *Value* is a real number.

A rule $r : p_1 \wedge p_2 \wedge \cdots \wedge p_x \wedge \neg n_1 \wedge \neg n_2 \wedge \cdots \wedge \neg n_y \rightarrow val_r$ is said to apply on a coalition $C \subseteq N$, denoted by $r \models C$, if and only if each positive literal p_i exists in C, i.e, $p_i \in C$ for $i = 1, \ldots, x$, and no negative literal n_j exists in C, i.e, $n_j \notin C$ for $j = 1, \ldots, y$. For example, a rule $r : 1 \wedge 2 \wedge \neg 3 \rightarrow val_r$ would apply to coalitions $\{1, 2\}$ and $\{1, 2, 4\}$, but not to the coalitions $\{1, 2, 3\}$ and $\{1, 4\}$.

Given a coalition C, we can compute C's utility by summing up the values of all the rules that apply to C:

$$v(C) \equiv \sum_{r \models C} val_r$$

Any characteristic function game can be represented by a set of such rules [3].

2.3.2.1 Embedded MC-nets

Michalak et al. in [16] proposed an extension of the above representation scheme, the *Embedded MC-nets*. The embedded MC-nets are used to represent coalitional games with externalities—i.e., in partition function form. In that work the rules refer to *embedded coalition into partitions*; that is, the values of the pattern, and therefore the utility of a coalition, depends not only on the members in the coalitions, but also on the overall partitioning of the agents into coalitions.

The form of an embedded MC-net rule is: $P_0|P_1 \dots P_k \rightarrow Value$, where P_i, for $i = 0, \dots, k$, is a pattern and $Value$ is a real value. An embedded MC-net rule $r : P_0|P_1 \dots P_k \rightarrow Value$ is applied on an embedded coalition (C, π) if and only if $P_0 \models C$ and for every P_i, for $i = 1, \dots k$, there is at least one coalition $C' \in \pi \setminus C$ such that $P_i \models C'$. The utility of an embedded coalition is computed by summing all rules that apply on i: $v(C, \pi) = \sum_{r \models (C,\pi)} r$.

2.3.2.2 Weighted MC-nets

Another extension, the *Weighted MC-nets*, was proposed in [17] by Michalak et al., and also conceives a representation of coalitional games with externalities. In the weighted MC-nets, the rules take the following form: $(P_1^1; v_1^1) \dots (P_{r_1}^1; v_{r_1}^1)| \dots |(P_1^s; v_1^s) \dots (P_{r_s}^s; v_{r_s}^s)$, where P_i^j is a pattern containing at least one positive literal, and v_i^j is its corresponding value. A partition π applies on a weighted MC-net rule r if π can be partitioned into s disjoint non-empty sets of coalitions such that $\pi_1 \cup \dots \cup \pi_s = \pi$; and for every pattern P_k^l, with $l = 1, \dots, s$ and $k = 1, \dots, r_l$, there is at least one coalition in π_l that satisfies pattern P_k^l.

2.3.2.3 Relational Rules

Finally, Mamakos and Chalkiadakis [8] proposed *Relational Rules (RRs)*, an MC-nets extension for representing overlapping coalitions. In that work, the rules are of the form:

$$A \rightarrow \frac{\sum_{i \in A} \pi_{i,C}}{|A|} \cdot value \tag{2.1}$$

where A is a subset of agents (corresponding to positive literals), C is a coalition such that $A \subseteq C$, and $\pi_{i,C}$ is the portion of the resource that agent i has invested in C—i.e., $\pi_{i,C} = \frac{resource_{i,C}}{resource_i}$, where $resource_i$ is the total resource quantity (continues or discrete) that i holds and $resource_{i,C}$ is the amount she has invested in C. A relational rule r applies to a coalition C if and only if $A \subseteq C$; while the value of a coalition is computed by summing the rules that apply to coalition C. Finally the Relational Rules representation reduce to classic MC nets [3] for non-overlapping settings since it holds that $\frac{\sum_{i \in A} \pi_{i,C}}{|A|} = 1$. However, this reduction is possible only when the set or rules is assumed to not contain negative literals.

Example 2 *Let $N = \{1, 2, 3\}$, $resource_1 = 6$, $resource_2 = 10$, $resource_3 = 3$, and let the Relational Rules of the game be:*

$$(r_1) : \{2\} \rightarrow \pi_{2,C} \cdot 2$$
$$(r_2) : \{1, 2\} \rightarrow \tfrac{\pi_{1,C} + \pi_{2,C}}{2} \cdot 5$$
$$(r_3) : \{1, 3\} \rightarrow \tfrac{\pi_{1,C} + \pi_{3,C}}{2} \cdot 3$$

Then, assume coalition $C_1 = \{1, 2\}$ forms, with $resource_{1,C_1} = 6$ ($\pi_{1,C_1} = \frac{resource_{1,C_1}}{resource_1} = \frac{6}{6} = 1$) and $resource_{2,C_1} = 2$ ($\pi_{2,C_1} = \frac{2}{10} = 0.2$). Applying rule (r_1) to C_1 will result a value $\pi_{2,C} \cdot 2 = 1 \cdot 2 = 2$ and applying rule (r_2) to C_1 will result a value of $\frac{\pi_{1,C} + \pi_{2,C}}{2} \cdot 5 = \frac{1+0.2}{2} \cdot 5 = 3$. Rule (r_3) does not apply in C_1, since agent $3 \notin C_1$. As such, the value of coalition C_1 is equal to $2 + 3 = 5$.

Chapter 3

MC nets in overlapping environments

In this chapter we present an extension of the MC-nets representation for *overlapping* settings, holding on the RRs concept.

3.1 RRs reduction to classic MC nets

As described in section 2.3.2.3, the authors in [8] introduced the Relational Rules, a representation scheme for cooperative games with *overlapping coalitions*. Relational Rules, like MC nets, use a set of rules in order to represent a coalitional game. The form of these rules permit the reduction of Relational Rules representation to a classical MC nets representation for non-overlapping settings, since it holds that $\frac{\sum_{i \in A} \pi_{i,C}}{|A|} = 1$. However, this desirable property holds without the existence of negative literals, i.e, only positive literals.

We now extend Relational Rules in order to achieve the desired reduction in non-overlapping settings even when negative literals are allowed. Considering a rule r and $A \subseteq N$ be the set of agents of r, we define the subset A^+, that consist of all the positive literals of A, and a subset A^- that consist of all the negative literals. Obviously it holds that $A^+ \cap A^- = \emptyset$ and $A^+ \cup A^- = A$.

As such, a Relational Rule r, that contains both positive and negative literals, can be defined as:

$$r : A = A^{+} \cup A^{-} \rightarrow \frac{\sum_{i \in A^{+}} \pi_{i,C} + |A^{-}| \cdot overlap()}{|A^{+}| + |A^{-}|} \cdot value = value_{ovl} \tag{3.1}$$

where A^{+} is the set of positive literals, A^{-} is the set of negative literals, $\pi_{i,C}$ is the portion of its resource that some $i \in A^{+}$ has invested in coalition C—i.e., $\pi_{i,C} = \frac{resource_{i,C}}{resource_{i}}$, where $resource_{i}$ is the total resource quantity (continues or discrete) that i holds and $resource_{i,C}$ is the amount she has invested in C. The *overlap* function is defined as:

$$overlap() = \begin{cases} 0, & if \quad \frac{\sum_{i \in A^{+}} \pi_{i,C}}{|A^{+}|} \neq 0 \quad and \quad A^{+} \neq \emptyset. \\ 1, & else. \end{cases} \tag{3.2}$$

The presence of $\frac{\sum_{i \in A^{+}} \pi_{i,C}}{|A^{+}|}$ in the conditions, captures the "pathological scenario", in which every agent i that participates in some coalition C, has zero contributions in C. In such a case, from Equation 3.2 we get that $overlap() = 1$ (since $\frac{\sum_{i \in A^{+}} \pi_{i,C}}{|A^{+}|} = 0$) and from Equation 3.1 we get that:

$$\frac{0 + |A^{-}| \cdot 1}{|A^{+}| + |A^{-}|} \cdot value = \frac{|A^{-}|}{|A^{+}| + |A^{-}|} \cdot value$$

That is, if all positive literals of a rule r have zero contributions for a coalition C (i.e., $\pi_{i,C} = 0, \forall i \in A^{+}$), then r applies to C but its value depends on the ratio $\frac{|A^{-}|}{|A^{+}|+|A^{-}|}$ only. Intuitively, we have a "discount factor" for the value of rule r that depends on the number of positive literals (with zero contributions). In such case, if a rule consists of many negative literals and the number of positive literals is small, then the "discount factor" is small as well. As the number of positive literals (with zero contributions) raises, the "discount factor" raises too.

If a rule r consists of only negative literals (i.e, $|A^{+}| = \emptyset$), then from Equation 3.2 we get that $overlap() = 1$ (since $|A^{+}| = \emptyset$) and from Equation 3.1 we get that:

$$\frac{0 + |A^{-}| \cdot 1}{0 + |A^{-}|} \cdot value = value$$

That is, the value of a rule that does not consist of any positive literals, does not decrease since we are only interested on the non-existence of the negative literals.

Finally, if a rule r consist of only positive literals (i.e, $|A^{-}| = \emptyset$), then this extension transforms to Equation 2.1 (proposed in [8]).

We say that, a rule r applies to coalition C, denoted as $r \models C$, if and only if:

- $\forall x \in A^+$, $x \in C$; and
- $\forall y \in A^-$, $y \notin C$.

In that case, $value_{ovl}$ is added to the coalition value of C.

Example 3 *Assume that* $N = \{1,2,3,4\}$, $resource_1 = 10$, $resource_2 = 8$, $resource_3 = 8$, $resource_4 = 2$, *and the Relational Rules of the game are:*

$$(r_1) : \{1,2,\neg 4\} \rightarrow \tfrac{\pi_{1,C}+\pi_{2,C}+1\cdot overlap()}{3} \cdot 10$$
$$(r_2) : \{1,3\} \rightarrow \tfrac{\pi_{1,C}+\pi_{3,C}}{2} \cdot 5$$
$$(r_3) : \{4\} \rightarrow \pi_{4,C} \cdot 2$$

Let coalition $C = \{1,2,3\}$ *form, with* $resource_{1,C} = 5$, $resource_{2,C} = 8$, $resource_3 = 2$. *Applying rule* (r_1) *to* C *will result a value of* $\frac{\pi_{1,C}+\pi_{2,C}+1\cdot overlap()}{3} \cdot 10 = \frac{0.5+1+1\cdot 0}{3} \cdot 10 = 5$. *Similarly, applying rule* (r_2) *to* C *will result a value of* $\frac{\pi_{1,C}+\pi_{3,C}}{2} \cdot 5 = \frac{0.5+0.25}{2} \cdot 5 = 1.88$. *Rule* (r_3) *does not apply in* C, *since agent* $4 \notin C$. *As such, the value of coalition* C *is equal to* $5 + 1.88 = 6.88$.

Lemma 1 ([8]) *In non-overlapping games, Relational Rules reduce to MC-nets rules without negative literals.*

Proof. Let N be a non-empty set of agents in a non-overlapping game. For each agent $i \in N$ holds that $\pi_{i,C} = 1$, since there are no overlaps and every agent participates in exactly one coalition C. As such, for every rule r (that consist only positive literals), using Equation 2.1, it holds that:

$$A \rightarrow \frac{\sum_{i\in A}\pi_{i,C}}{|A|} \cdot value = \frac{\sum_{i\in A} 1}{|A|} \cdot value = \frac{|A|}{|A|} \cdot value = value$$

That is, RRs reduce to classic MC-nets rules in non-overlapping games. □

Theorem 1 *The Relational Rules representation reduce to classic MC nets representation for non-overlapping settings.*

Proof. Firstly, we consider that there are only positive literals, i.e., $A^- = \emptyset$. That is, Equation 3.1 is equal to:

$$A = A^+ \rightarrow \frac{\sum_{i \in A} \pi_{i,C}}{|A|} \cdot value = value_{ovl} \tag{3.3}$$

Equation 3.3 is the same that introduced in [8]. As such, from Lemma 1 we know that Relational Rules reduces to classic MC nets .

On the other hand, considering that there are only negative literals, i.e., $A^+ = \emptyset$, Equation 3.1 transforms to:

$$\begin{aligned} A =& A^- \rightarrow \frac{|A^-| \cdot overlap()}{|A^-|} \cdot value \Rightarrow \\ & A^- \rightarrow \frac{|A^-| \cdot 1}{|A^-|} \cdot value \Rightarrow \\ & A^- \rightarrow value = value_{ovl} \end{aligned} \tag{3.4}$$

As such, Relational Rules reduces to classic MC nets in that case too.

Finally, we consider that there are both positive and negative literals, i.e., $A^+ \neq \emptyset$ and $A^- \neq \emptyset$. For non-overlapping coalitions it holds that:

$$\begin{aligned} A =& A^+ \cup A^- \rightarrow \frac{\sum_{i \in A^+} \pi_{i,C} + |A^-| \cdot overlap()}{|A^+| + |A^-|} \cdot value \Rightarrow \\ & A^+ \cup A^- \rightarrow \frac{\sum_{i \in A^+} 1 + |A^-| \cdot 1}{|A^+| + |A^-|} \cdot value \Rightarrow \\ & A^+ \cup A^- \rightarrow \frac{|A^+| + |A^-|}{|A^+| + |A^-|} \cdot value \Rightarrow \\ & A^+ \cup A^- \rightarrow value = value_{ovl} \end{aligned} \tag{3.5}$$

□

Chapter 4

ε-MC nets

In this chapter we describe the $\varepsilon-$MC nets representation scheme. First we discuss environments for $\varepsilon-$MC nets, i.e., large open multiagent systems with uncertainty. Then we present the form of $\varepsilon-$MC nets rules along with an algorithm that constructs the $\varepsilon-$MC nets representation. We propose a variant of the algorithm that exploits prior knowledge over the agents in order to exploit their perceived similarities and achieve an even more compact representation. To do so, we also define a notion of *agent equivalence* in the process. This variant of the algorithm can also produce new, previously unknown, collaboration patterns among agents. Finally, our approach extends naturally to non-transferable utility games.

4.1 Multiagent Environments with Uncertainty

In multiagent environments where we have a large number of agents, it is unrealistic to assume perfect and complete information. That is, the number of different coalitions rises exponentially, and a prevalent assumption that we are aware of the utility of each one of these coalitions cannot stand.

The issue of uncertainty has gained a lot of attention in the game theory community, and there is a host of research papers tackling the problem. For instance, [18] proposes a class of cooperative games where agents are uncertain about their partners' *type*, and express beliefs over the type of other agents. The authors in [19] study a series of strategies and protocols for coalition formation under uncertainty; while [20] provides a definition of *Transferable Utility Games with Uncertainty (TUU)*. According to this definition a

TUU G is a tuple $\langle N, \mathcal{S}, v, u\rangle$, where N is a set of agents; $\mathcal{S}$ is a finite set of states; v_s is a characteristic function for state $s \in \mathcal{S}$; and u^i assigns to every profile of payoffs a utility level.

In our work we consider environments with uncertainty over the utility of coalitions. Let N be a non-empty set of agents, and v be an underlying characteristic function of some TU game. Function v is hidden to the whole system, however we have in our disposal a function that comprises beliefs over coalitions' utility.

Definition 11 (Believed Characteristic Function) *Given a TU game $G = \langle N, v\rangle$ where v is an unknown characteristic function, a function $\tilde{v} : A \rightarrow \mathbb{R}$, where $A \subseteq 2^N$, constitutes a* believed characteristic function *estimating the underlying characteristic function v.*

That is, $\tilde{v}(C)$ corresponds to an estimate of the utility of coalition C (e.g., inferred by past observations), rather than to the actual $v(C)$.

4.1.1 ε-MC nets Rules

Given a set of agents N, assume we have estimates on collaborations between pairs of agents, i.e., we have rules of the form $i \wedge j \rightarrow val$, where i and j refers to an agent $a_i \in N$ and $a_j \in N$ respectively, either as a positive ($i \equiv a_i$) or a negative literal ($i \equiv \neg a_i$); and val is our estimate about collaboration pattern $i \wedge j$. Intuitively, pairwise collaborations can be considered as the basis for estimating the utility of collaboration with many agents. This concept suggests that larger collaboration patterns follow an additive behaviour. We will refer to such rules as "MC-net-like" rules.

Definition 12 (MC-net-like Rules) *An MC-net-like rule r is of the form Pattern $\rightarrow$ val, where Pattern is restricted to pairs of agents, where an agent is represented by either a positive or a negative literal; and val is an estimate about the utility that portrays the collaboration between the agents indicated by the pattern, where val is provided by a believed characteristic function $\tilde{v}$.*

Having a set of MC-net-like rules at hand, we can build a $\tilde{v} : A \rightarrow \mathbb{R}$ such that $A = \{C \subseteq N : \exists r_i \ s.t.\ r_i \models C\}$, and $\tilde{v}(C) = \sum_{r_i \models C} val_i$, where $\tilde{v}$ is the believed characteristic function, and val_i is the value of the i^{th} MC-net-like rule that applies to C. In the $\varepsilon-$MC net representation we propose a compact set of rules originating from a set of MC-net-like rules, by merging patterns and regulating the value correspondingly. Now we are ready to define the $\varepsilon-$MC net rules.

Definition 13 (*$\varepsilon-$MC net Rule*) An $\varepsilon-$MC net rule is of the form $i \wedge \mathrm{CG} \rightarrow val$, where $i \in N$ is called *the reference agent*, $\mathrm{CG} \subseteq N$ is a subset of agents called *its collaborators group*, and $val \in \mathbb{R}$ expresses the estimate[1] we have on the value of collaboration pattern between agent i and any agent $j \in \mathrm{CG}$.

Intuitively, the reference agent corresponds to the "common agent" for all agents in the collaborating group, such that by establishing a collaboration between i and any agent j in CG, has an expected value val provided by the *believed characteristic function; and $\varepsilon \in \mathbb{R}_+$ signifies how far from the $\tilde{v}$ value, according to the MC-net-like rules at hand, we are willing to depart in order to compress an original MC-net-like representation.* In other words, ε represents the margin of information loss we are willing to accept in order to compress the representation. Naturally, the larger the ε, the wider these margins are, and therefore the more compact the representation will be.

In the process of compressing the initial MC-net-like rules to a final set of $\varepsilon-$MC nets rules, we distinguish two types of merging: *(a)* the full-merge, and *(b)* the half-merge. The *full merge* describes the merge of two MC-net-like rules that produces a new $\varepsilon-$MC net rule. A full merge can occur if there is a *mutual* agent between the rules, and if the values of the two rules differ by at most ε, where ε is the margin of information loss that we are willing to accept. Formally, two MC-net-like rules $r : i \wedge j \rightarrow val_{i,j}$ and $r' : k \wedge l \rightarrow val_{k,l}$ can be full-merged iff:

(I) $i \equiv k$ or $i \equiv l$ or $j \equiv k$ or $j \equiv l$. Remember that i,j,k and l are positive or negative literals, thus if it stands for example that $i \equiv k$ it means that they are both referring to the same agent, and they are both positive or both negative; and

(II) $|val_{i,j} - val_{k,l}| \leq \varepsilon$.

The resulting $\varepsilon-$MC net rule is $r_{merged} : a_{mutual} \wedge \mathrm{CG} \rightarrow \frac{val_{i,j}+val_{k,l}}{2}$, where a_{mutual} is the mutual agent, and CG contains the non-mutual agents. The second type of merges, the *half-merge*, describes a merge of an MC-net-like rule r to an $\varepsilon-$MC net rule r'. A half-merge rule can occur if the reference agent of the $\varepsilon-$MC net exists in the pattern

[1]Nothing in our model precludes val from being the actual value of $i \wedge CG$.

of the MC-net-like rule; if the values of the two rules differ by at most ε, similarly to full-merge; and if the value of the soon-to-be-produced $\varepsilon-$MC net rule is bounded by the maximum and the minimum value of the rules merged to r' so far. Formally, let an $\varepsilon-$MC net rule $r : i \wedge \mathrm{CG} \rightarrow val_{\mathrm{CG}}$, with val_{CG} the value of rule r. Also let V_{CG} be the set containing all the values of the rules merged so far to produce r. Consider now an MC-net-like rule $r' : j \wedge k \rightarrow val_{j,k}$. Then, r' can be merged with r if and only if:

(III) $i \equiv j$ or $i \equiv k$, where two literals are identical if they refer to the same agent, and they are both positive or both negative; and

(IV) $val_{\mathrm{CG}} - \varepsilon \leq val_{j,k} \leq val_{\mathrm{CG}} + \varepsilon$; and

(V) $\max_{v \in V_{\mathrm{CG}}} v - \varepsilon \leq \frac{val_{\mathrm{CG}} \cdot |V_{\mathrm{CG}}| + val_{r'}}{|V_{\mathrm{CG}}|+1} \leq \min_{v \in V_{\mathrm{CG}}} v + \varepsilon$.

The newly merged rule is denoted as:

$$r_{merged} : i \wedge \mathrm{CG}' \rightarrow \frac{val_{\mathrm{CG}} \cdot |V_{\mathrm{CG}}| + val_{j,k}}{|V_{\mathrm{CG}}| + 1},$$

where $\mathrm{CG}' \equiv \mathrm{CG} \cup \{a_{non-mutual}\}$, $a_{non-mutual}$ corresponds to agent j if $i \equiv k$ or to agent k if $i \equiv j$.

Algorithm 1 performs a series of such merges, compressing the initial set of rules R to a succinct representation captured in a generated set R' of $\varepsilon-$MC nets rules. Its complexity is quadratic to the number of the initial MC-net-like rules. Going through Algorithm 1, we see that the outer loop in line 3 needs exactly m iterations, where m is the size of the initial set of MC-net-like rules. The inner loop in lines 11-23 needs at most m iterations; while the condition in lines 12 is trivial—specifically in the case of full-merge we need 4 comparisons for detecting the reference agent, and 2 comparisons for the value condition; while in the case of half-merge we need 2 comparisons for detecting the reference agent, and 4 comparisons for the two value conditions. As such the complexity of the algorithm is $\mathcal{O}(6 \cdot m^2)$, where m is the total number of the initial MC-net-like rules (i.e., the size of the R set). According to the algorithm, a final $\varepsilon-$MC net rule is derived from κ merges, that is *(i)* a single full-merge, and *(ii)* $\kappa - 1$ half-merges, with $\kappa = 1, 2, \ldots$. Example 4 illustrates Algorithm 1's functionality.

Algorithm 1 Merging MC-net-like Rules

$R \leftarrow$ initial set of MC-net-like rules of size m;
$R' \leftarrow \emptyset$;
for $r \in R$ **do**
    $\mathrm{CG} \leftarrow \emptyset$;
    $\kappa \leftarrow 0$;
    $V_{\mathrm{CG}} \leftarrow val_r$;
    $\min \leftarrow \min_{v \in V_{\mathrm{CG}}} \{v\}$;
    $\max \leftarrow \max_{v \in V_{\mathrm{CG}}} \{v\}$;
    $\mathrm{avg} \leftarrow val_r$;
    remove r from R;
    for $r' \in R$ **do**
        if $\textsc{CanFullMerge}(r, r')$ **OR** $\textsc{CanHalfMerge}(r, r')$ **then**
            add $val_{r'}$ in V_{CG};
            add non mutual agents in CG;
            $\min \leftarrow \min_{v \in V_{\mathrm{CG}}} \{v\}$;
            $\max \leftarrow \max_{v \in V_{\mathrm{CG}}} \{v\}$;
            $\mathrm{avg} \leftarrow \frac{\mathrm{avg} \cdot |V_{\mathrm{CG}}| + val_{r'}}{|V_{\mathrm{CG}}| + 1}$;
            $\kappa \leftarrow \kappa + 1$;
            $r_{new} \leftarrow a_{mutual} \wedge \mathrm{CG} \rightarrow \mathrm{avg}$; {# intermediate rule}
            $r \leftarrow r_{new}$;
            remove r' from R;
        end if
    end for
    add r in R';
end for
return R';

Algorithm 2 $\textsc{CanFullMerge}(r, r')$

if r and r' *satisfy* Cond. (I) **AND** (II) **then**
    return True;
end if
return False;

Example 4 *Consider a set of 5 agents $N = \{1,2,3,4,5\}$, and the following initial MC-net-like rules in R:*

$$(r_1) : 1 \wedge 2 \to 5$$

$$(r_2) : 3 \wedge 4 \to 6$$

$$(r_3) : 1 \wedge 4 \to 7$$

$$(r_4) : 3 \wedge 2 \to 16$$

$$(r_5) : 4 \wedge 5 \to 7$$

Moreover, we let ε (the accepted information loss) be equal to 1. Employing Algorithm 1 on the above set of rules we get the following result. We begin with rule r_1, and we set the variables $\max = 5, \min = 5, \text{avg} = 5$. The rules that share mutual agents with r_1 are r_3 and r_4; however, none of the two can be full-merged with r_1 since the condition (II) does not hold for any of the two rules. Rule r_1 remains intact, we move to rule r_2, and set the variables to $\max = 6, \min = 6, \text{avg} = 6$. Rule r_2 shares a mutual agent with rules r_3, r_4 and r_5. Rules r_2 and r_3 can be full-merged since $6 - 1 \leq 7 \leq 6 + 1$ (condition (II)); so we add 1 and 3 in CG, we set $V_{\text{CG}} = \{6, 7\}$, we update the variables $\max = \max_{v \in V_{\text{CG}}}\{v\} = 7, \min = \min_{v \in V_{\text{CG}}}\{v\} = 6, \text{avg} = 6.5$, we generate a new rule $r_6 : 4 \wedge \{1, 3\}$, and replace r_2 with r_6 as well as remove r_3 from R. Now instead of looking for rules with mutual agents with rule r_2, we use rule r_6, and specifically we are looking for rules containing the reference agent 4. The pattern of rule r_4 does not contain the reference agent of r_6, while rule r_5 does. Rule r_5 can be half-merged to rule r_6 since $6.5 - 1 \leq 7 \leq 6.5 + 1$ (condition (IV)) and $7 - 1 \leq \frac{6.5 \cdot 2 + 7}{2+1} \leq 6 + 1$ (condition (V)). Now we add agent 5 in CG, we set $V_{\text{CG}} = \{6, 7, 7\}$, we update the variables $\max = \max_{v \in V_{\text{CG}}}\{v\} = 7, \min = \min_{v \in V_{\text{CG}}}\{v\} = 6, \text{avg} = 6.667$, we generate a new rule $r_7 : 4 \wedge \{1, 3, 5\}$, and replace r_6 with r_7 as well as remove r_5 from R.

At the end of this process, we have a new, more compact representation with the following rules:

$$(r_1) : 1 \wedge 2 \to 5$$

$$(r_4) : 3 \wedge 2 \to 16$$

$$(r_7) : 4 \wedge \{1, 3, 5\} \to 6.667$$

Algorithm 3 CANHALFMERGE(r, r')

if r and r' *satisfy* Cond. (III) **AND** (IV) **AND** (V) **then**
  return True;
end if
return False;

Intuitively, the ε-MC nets representation *detects similar collaboration patterns.* For instance, in a setting where agents acquire skills, a rule $i \wedge \mathrm{CG} \rightarrow val$ could be interpreted as: *if we combine the skills of agent i with the skills of any agent in* CG *we observe a similar change in the utility of the coalition; that is, agents in* CG *have similar skill-sets that impact the outcome similarly.*

We have to mention that the (final) set of rules R' that our algorithm will produce, depends on the sequence of merges that will take place, i.e., for the same initial set of rules, R, if we change the order of the initial rules, then our algorithm will produce a different representation R'.[1]

4.1.1.1 Bounds on the values of $\varepsilon-$MC nets rules

Next in Theorem 4, we show that our estimate of the value of any collaboration pair given by an $\varepsilon-$MC net rule, lies in a distance of at most $\kappa \cdot \varepsilon$ from our initial estimate on the values of the rules merged in order to reach the $\varepsilon-$MC net rule. As such, we provide a bound on the maximum information loss incurred by using our "compressed" representation instead of an MC-net-like one.

Lemma 2 *For any half-merge between an MC-net-like rule r_x and $\varepsilon-$MC net rule r_y producing a new $\varepsilon-$MC net rule r_z, it holds that: $|val_z - val_y| \leq \varepsilon$.*

Proof. Let $r_x : i \wedge j \rightarrow val_x$ be an MC-net-like rule, and $r_y : i \wedge \mathrm{CG} \rightarrow val_y$ be an $\varepsilon-$MC net rule with V_{CG} containing all the values of the rules that have been merged in r_y, so far. We assume that r_x can be half-merged with r_y (i.e., the conditions (III), (IV), and (V) are satisfied), producing a new $\varepsilon-$MC net rule $r_z : i \wedge (\mathrm{CG} \cup \{j\}) \rightarrow val_z$. From half-merge condition (IV) we have that:

$$val_y - \varepsilon \leq val_x \leq val_y + \varepsilon \tag{4.1}$$

[1] This opens the possibility for a variant of our algorithm that employs randomized sequences of the initial set of rules

Moreover due to the way we construct the new $\varepsilon-$MC net rule r_z, it holds that

$$val_z = \frac{val_y \cdot |V_{\mathrm{CG}}| + val_x}{|V_{\mathrm{CG}}| + 1} \Rightarrow val_x = (|V_{\mathrm{CG}}| + 1) \cdot val_z - |V_{\mathrm{CG}}| \cdot val_y \tag{4.2}$$

Combining Equations (4.1) and (4.2) we have that:

$$\begin{aligned}
val_y - \varepsilon \leq (|V_{\mathrm{CG}}| + 1) \cdot val_z - |V_{\mathrm{CG}}| \cdot val_y \leq val_y + \varepsilon \Rightarrow \\
|V_{\mathrm{CG}}| \cdot val_y + val_y - \varepsilon \leq (|V_{\mathrm{CG}}| + 1) \cdot val_z \leq |V_{\mathrm{CG}}| \cdot val_y + val_y + \varepsilon \Rightarrow \\
(|V_{\mathrm{CG}}| + 1) \cdot val_y - \varepsilon \leq (|V_{\mathrm{CG}}| + 1) \cdot val_z \leq (|V_{\mathrm{CG}}| + 1) \cdot val_y + \varepsilon \Rightarrow \\
-\frac{\varepsilon}{|V_{\mathrm{CG}}| + 1} \leq val_z - val_y \leq \frac{\varepsilon}{|V_{\mathrm{CG}}| + 1} \Rightarrow \\
|val_z - val_y| \leq \frac{\varepsilon}{|V_{\mathrm{CG}}| + 1} \leq \varepsilon \Rightarrow \\
|val_z - val_y| \leq \varepsilon
\end{aligned}$$

□

Lemma 3 *For any half-merge between an MC-net-like rule r_x and $\varepsilon-$MC net rule r_y producing a new $\varepsilon-$MC net rule r_z, it holds that: $|val_z - val_x| \leq \varepsilon$.*

Proof. Let $r_x : i \wedge j \rightarrow val_x$ be an MC-net-like rule, and $r_y : i \wedge \mathrm{CG} \rightarrow val_y$ be an $\varepsilon-$MC net rule with $|V_{\mathrm{CG}}|$ containing all the values of the rules that have been merged in r_y, so far. We assume that r_x can be half-merged with r_y (i.e., the conditions (III), (IV), and (V) are satisfied), producing a new $\varepsilon-$MC net rule $r_z : i \wedge (\mathrm{CG} \cup \{j\}) \rightarrow val_z$. Due to the way we construct the new $\varepsilon-$MC net rule r_z, it holds:

$$\begin{aligned}
val_z &= \frac{|V_{\mathrm{CG}}| \cdot val_y + val_x}{|V_{\mathrm{CG}}| + 1} && \Rightarrow \\
val_z - val_x &= \frac{|V_{\mathrm{CG}}| \cdot val_y + val_x}{|V_{\mathrm{CG}}| + 1} - val_x && \Rightarrow \\
val_z - val_x &= \frac{|V_{\mathrm{CG}}| \cdot val_y + val_x - (|V_{\mathrm{CG}}| + 1) \cdot val_x}{|V_{\mathrm{CG}}| + 1} && \Rightarrow \\
val_z - val_x &= \frac{|V_{\mathrm{CG}}| \cdot val_y - |V_{\mathrm{CG}}| \cdot val_x}{|V_{\mathrm{CG}}| + 1} && \Rightarrow \\
val_z - val_x &= \frac{|V_{\mathrm{CG}}|}{|V_{\mathrm{CG}}| + 1} \cdot (val_y - val_x) && (4.4)
\end{aligned}$$

From half-merge condition (IV) we have that:

$$val_y - \varepsilon \leq val_x \leq val_y + \varepsilon \qquad \Rightarrow$$

$$
\begin{aligned}
&-\varepsilon \leq val_y - val_x \leq \varepsilon && \Rightarrow \\
&-\frac{|V_{\mathrm{CG}}|}{|V_{\mathrm{CG}}|+1} \cdot \varepsilon \leq \frac{|V_{\mathrm{CG}}|}{|V_{\mathrm{CG}}|+1} \cdot (val_y - val_x) \leq \frac{|V_{\mathrm{CG}}|}{|V_{\mathrm{CG}}|+1} \cdot \varepsilon && \Rightarrow \\
&-\varepsilon \leq \frac{|V_{\mathrm{CG}}|}{|V_{\mathrm{CG}}|+1} \cdot (val_y - val_x) \leq \varepsilon
\end{aligned} \tag{4.5}
$$

Combining Equations (4.4) and (4.5) we have that:
$-\varepsilon \leq val_z - val_x \leq \varepsilon \Rightarrow |val_z - val_x| \leq \varepsilon.$ □

Theorem 2 *For every $\varepsilon-$MC net rule r_{merged} derived by one full-merge and $\kappa-1$ half-merges, and for the two MC-net-like rules r_x and r_y (with estimates val_x and val_y) that participate in the full-merge, it holds that: $|val_{merged}-val_x| \leq \kappa \cdot \varepsilon$ and $|val_{merged}-val_y| \leq \kappa \cdot \varepsilon$.*

Proof. Let an $\varepsilon-$MC net rule $r_{\mathrm{merged}} : i \wedge \mathrm{CG} \rightarrow val_{\mathrm{merged}}$ produced by κ merges (1 full-merge and $\kappa-1$ half-merges); and let $r_x : i \wedge j \rightarrow val_x$ and $r_y : i \wedge k \rightarrow val_y$ be the two initial MC-net-like rules participating in the full-merge (first merge) in progress of reaching final $r_{\mathrm{merged}} : i \wedge \mathrm{CG} \rightarrow val_{\mathrm{merged}}$.

Base Case: Rules r_x and r_y can be full-merged and produce an $\varepsilon-$MC net $r_1 : i \wedge \{j,k\} \rightarrow val_1$, with $\{j,k\} \subset \mathrm{CG}$, and $val_1 = \frac{val_x+val_y}{2}$. As such at step 1, from condition (II) we have that:

$$
\begin{aligned}
-\varepsilon \leq val_y - val_x \leq \varepsilon \Rightarrow \frac{2 \cdot val_x - \varepsilon}{2} \leq \frac{val_x + val_y}{2} \leq \frac{2 \cdot val_x + \varepsilon}{2} \\
-\frac{\varepsilon}{2} \leq val_1 - val_x \leq \frac{\varepsilon}{2} \Rightarrow |val_1 - val_x| \leq \frac{\varepsilon}{2} \leq \varepsilon
\end{aligned} \tag{4.6}
$$

Induction step: After $\kappa-1$ merges (1 full-merge and $\kappa-2$ half-merges), we have produced an $\varepsilon-$MC net rule $r_{\kappa-1} : i \wedge \mathrm{CG}_{\kappa-1} \rightarrow val_{\kappa-1}$, where $\mathrm{CG}_{\kappa-1} = \{j,k,l,\dots\} \subset \mathrm{CG}$ contains κ agents. Our hypothesis is that it holds:

$$
|val_{\kappa-1} - val_x| \leq (\kappa-1) \cdot \varepsilon \tag{4.7}
$$

In step κ we have a half-merge of $r_z : i \wedge m \rightarrow val_z$ to $r_{\kappa-1}$, and produce a rule $r_\kappa : i \wedge (\mathrm{CG}_{\kappa-1} \cup \{m\}) \rightarrow val_\kappa$, where $\mathrm{CG}_{\kappa-1} \cup \{m\} \equiv \mathrm{CG}_\kappa$, and $val_{\mathrm{merged}} \equiv val_\kappa = \frac{val_{\kappa-1} \cdot (\kappa-1) + val_z}{\kappa}$. From Lemma 2 it holds that:

$$
|val_\kappa - val_{\kappa-1}| \leq \varepsilon \tag{4.8}
$$

Summing (4.7) and (4.8) we have:

$$
\begin{aligned}
&-(\kappa-1)\cdot\varepsilon-\varepsilon \le val_\kappa - val_{\kappa-1} + val_{\kappa-1} - val_x \le (\kappa-1)\cdot\varepsilon+\varepsilon \Rightarrow \\
&\kappa\cdot\varepsilon \le val_\kappa - val_x \le \kappa\cdot\varepsilon \Rightarrow |val_\kappa - val_x| \le \kappa\cdot\varepsilon
\end{aligned} \tag{4.9}
$$

Similarly we can show that the same holds for val_y of r_y, thus two MC-net-like rules composing the full-merge it holds that after κ merges we have $|val_\kappa - val_x| \le \kappa\cdot\varepsilon$ and $|val_\kappa - val_y| \le \kappa\cdot\varepsilon$. □

Theorem 3 *For any MC-net-like rule r_z that is half-merged to the intermediate*[1] *rule $\varepsilon-$MC net at step λ when producing a final $\varepsilon-$MC net rule r_{merged} after κ merges, it holds $|val_{merged} - val_z| \le (\kappa-\lambda+1)\cdot\varepsilon$.*

Proof. Let an $\varepsilon-$MC net rule $r_{\text{merged}} : i \wedge \text{CG} \to val_{\text{merged}}$ produced by κ merges (1 full-merge and $\kappa-1$ half-merges); and r_z is an MC-net-like rule that is half-merged at step λ (with $2 \le \lambda \le \kappa$) with intermediate rule r_λ with value val_λ, in progress of reaching final $r_{\text{merged}} : i \wedge \text{CG} \to val_{\text{merged}}$.
Base Case From Lemma 3 we know that $|val_\lambda - val_z| \le \varepsilon$.
Induction step Our hypothesis is that at step $\kappa-1$ (after 1 full-merge and $\kappa-1$ half-merges) it holds $|val_{\kappa-1} - val_z| \le (\kappa-1-\lambda+1)\cdot\varepsilon$. At step κ Equation (4.8) stands, so by summation we have:

$$
\begin{aligned}
&-(\kappa-\lambda)\cdot\varepsilon-\varepsilon \le val_\kappa - val_{\kappa-1} + val_{\kappa-1} - val_z \le (\kappa-\lambda)\cdot\varepsilon+\varepsilon \Rightarrow \\
&-(\kappa-\lambda+1)\cdot\varepsilon \le val_\kappa - val_z \le (\kappa-\lambda+1)\cdot\varepsilon \Rightarrow
\end{aligned}
$$

Thus we have that $|val_\kappa - val_z| \le (\kappa-\lambda+1)$, where $val_\kappa \equiv val_{\text{merged}}$. □

Theorem 4 *For any MC-net-like rule $r : i \wedge j \to val$ that is merged (either full- or half-merged) in the process of reaching an $\varepsilon-$MC net rule r_{merged} after κ merges, it holds $|val_{merged} - val| \le \kappa\cdot\varepsilon$.*

Proof. From Theorem 2 we have that for the two rules r_x and r_y participating in the full-merge it holds $|val_{\text{merged}} - val_x| \le \kappa\cdot\varepsilon$ and $|val_{\text{merged}} - val_y| \le \kappa\cdot\varepsilon$. From Theorem 3 we have that for any rule r_z being half-merges at any step λ (with $2 \le \lambda \le \kappa$) it holds that $|val_{\text{merged}} - val_z| \le (\kappa-\lambda+1)\cdot\varepsilon \overset{2\le\lambda\le\kappa}{\Rightarrow} |val_{\text{merged}} - val_z| \le \kappa\cdot\varepsilon$. □

[1] See line 19 of Algorithm 1.

4.1.1.2 Bounding the number of merges per ε-MC net rule

Note that we can easily adapt Algorithm 1 so that the resulting succinct representation consists of a set of ε-MC nets rules, where for each such rule r it holds that the value of r, i.e. val_r, is at most $\kappa_{max} \cdot \varepsilon$ away from our initial MC-net-like estimates. That is, we can define an upper bound κ_{max} on the number of merges that we allow per ε-MC net rule. As such, we introduce Algorithm 4 that permits κ_{max} number of merges for every ε-MC net rule. Compared to Algorithm 1, in Algorithm 4 we need to check the additional condition (line 13) whether the total number of merges (for a specific ε-MC net rule) is greater than κ_{max}. The complexity of Algorithm 4 remains $\mathcal{O}(m^2)$, where m is the size of the R set of rules, since checking the new condition is trivial.

4.1.1.3 Rules with larger collaborative patterns

Our work so far considers MC-net-like rules containing only collaborative pairs. However, with a slight change on Algorithm 1 we can consider initial rules with patterns containing more than two literals. In this variant the $\varepsilon-$MC nets rules will be of the form:

$$i \wedge \mathrm{CG} \rightarrow val$$

where CG now is a set of sub-patterns, e.g., $\mathrm{CG} = \big\{\{j \wedge k\} \vee \{l \wedge o \wedge p\} \vee \{q\}\big\}$, which is produced by merging rules $r_1 : i \wedge j \wedge k \rightarrow val_1$, $r_2 : i \wedge l \wedge o \wedge p \rightarrow val_2$, and $r_3 : i \wedge q \rightarrow val_3$.

In this case, again, we distinguish the same two types of merging (full-merge and half-merge). However, the conditions that need to stand true must be slightly changed. Specifically, two rules $r_1 : Pattern_1 \rightarrow val_1$ and $r_2 : Pattern_2 \rightarrow val_2$, where $Pattern_1$ and $Pattern_2$ are a conjunction of g agents ($2 \leq g \leq |N|$), can be full-merged iff:

(VI) $i \equiv j$, where i is any agent in $Pattern_1$ and j represents the same agent, in $Pattern_2$. Remember that i and j denote positive or negative literals, thus if it holds for example that $i \equiv j$, it means that they are both referring to the same agent, and they are both positive or both negative; and

(VII) $|val_1 - val_2| \leq \varepsilon$.

Algorithm 4 Bounded number of merges per ε-MC net rule

```
R ← initial set of MC-net-like rules of size m;
R' ← ∅;
for r ∈ R do
    CG ← ∅;
    κ ← 0;
    V_CG ← val_r;
    min ← min_{v∈V_CG} {v};
    max ← max_{v∈V_CG} {v};
    avg ← val_r;
    κ_max ← maximum number of merges;
    remove r from R;
    for r' ∈ R do
        if κ_max < κ then
            break;
        end if
        if CanFullMerge(r, r') OR CanHalfMerge(r, r') then
            add val_r' in V_CG;
            add non mutual agents in CG;
            min ← min_{v∈V_CG} {v};
            max ← max_{v∈V_CG} {v};
            avg ← (avg · |V_CG| + val_r') / (|V_CG| + 1);
            κ ← κ + 1;
            r_new ← a_mutual ∧ CG → avg;
            r ← r_new;
            remove r' from R;
        end if
    end for
    add r in R';
end for
return R';
```

The resulting $\varepsilon-$MC net rule is $r_{merged} : a_{mutual} \wedge \mathrm{CG} \to \frac{val_1+val_2}{2}$, where a_{mutual} is the mutual agent, and $\mathrm{CG} = \{Pattern'_1, Pattern'_2\}$, where $Pattern'_1 = Pattern_1 \backslash \{i\}$ and $Pattern'_2 = Pattern_2 \backslash \{j\}$.

Similarly, given an ε-MC net rule $r : i \wedge \mathrm{CG} \to val_{\mathrm{CG}}$, where CG is a set of sub-patterns and V_{CG} is the set containing all the values of the rules merged so far to produce r; and a rule $r_3 : Pattern_3 \to val_3$, we say that r_3 can be merged with r iff:

(VIII) $i \equiv j$, where $j \in Pattern_3$. The two literals are identical if they refer to the same agent, and they are both positive or both negative; and

(IX) $val_{\mathrm{CG}} - \varepsilon \leq val_3 \leq val_{\mathrm{CG}} + \varepsilon$; and

(X) $\max_{v \in V_{\mathrm{CG}}} v - \varepsilon \leq \frac{val_{\mathrm{CG}} \cdot |V_{\mathrm{CG}}| + val_3}{|V_{\mathrm{CG}}|+1} \leq \min_{v \in V_{\mathrm{CG}}} v + \varepsilon$.

The newly merged rule is denoted as:

$$r_{merged} : i \wedge \mathrm{CG}' \to \frac{val_{\mathrm{CG}} \cdot |V_{\mathrm{CG}}| + val_3}{|V_{\mathrm{CG}}| + 1},$$

where $\mathrm{CG}' \equiv \mathrm{CG} \cup \{Pattern'_3\}$ and $Pattern'_3 = Pattern_3 \backslash \{j\}$.

We can easily show that the Theorems and Lemmas presented in Section 4.1.1.1, hold as well in this scenario, using the notion of *auxiliary agents*. Essentially, we will substitute each additional collaboration pattern that is presented in the rule for an auxiliary agent.

Proposition 1 *Lemmas 2, 3 and Theorems 2, 3, 4 hold for larger collaborative patterns as well.*

Proof. Let $N = \{i, j, k, m, n, q, o, w\}$ be a set of agents and we have the following rules:

$$(\mathrm{r}_1) : \; i \wedge j \wedge k \to val_1$$

$$(\mathrm{r}_2) : \; i \wedge m \wedge n \wedge q \to val_2$$

$$(\mathrm{r}_3) : \; i \wedge w \wedge o \to val_3$$

which constitute a game G. Considering that r_1 and r_2 can full-merged—i.e., the conditions (VI) and (VII) are satisfied—we get the new ε-MC net rule:

$$r_4 : i \wedge \{\{j \wedge k\}, \{m \wedge n \wedge q\}\} \rightarrow val_4$$

where $val_4 = \frac{val_1+val_2}{2}$. Then, considering that r_3 and r_4 can half-merged—i.e., the conditions (VIII), (IX) and (X) are satisfied—we get:

$$r_5 : i \wedge \{\{j \wedge k\}, \{m \wedge n \wedge q\}, \{w \wedge o\}\} \rightarrow val_5$$

where $val_5 = \frac{val_1+val_2+val_3}{3}$.

Let x be an *auxiliary* agent, that represents the pattern $p_1 = j \wedge k$. Similarly, the *auxiliary* agents y and z represents the patterns $p_2 = m \wedge n \wedge q$ and $p_3 = w \wedge o$ respectively. Note that here we abuse the term of an agent since our new agents x, y and z are in fact patterns of agents. Then, we can build a new game G' with $N' = \{i, x, y, z\}$ and the rules:

$$(r'_1) \,:\, i \wedge x \rightarrow val_1$$

$$(r'_2) \,:\, i \wedge y \rightarrow val_2$$

$$(r'_3) \,:\, i \wedge z \rightarrow val_3$$

In such game, the Theorems and Lemmas (from Section 4.1.1.1) hold. Considering that r'_1 and r'_2 can full-merged—i.e., the conditions (I) and (II) are satisfied—we get the new ε-MC net rule:

$$r'_4 : i \wedge \{x, y\} \rightarrow val_4$$

where $val_4 = \frac{val_1+val_2}{2}$. As such, from Lemma 2 we have that $|val_4 - val_1| \leq \varepsilon$ and $|val_4 - val_2| \leq \varepsilon$. Then, considering that r'_3 and r'_4 can half-merged—i.e., the conditions (III), (IV) and (V) are satisfied—we get:

$$r'_5 : i \wedge \{x, y, z\} \rightarrow val_5$$

where $val_5 = \frac{val_1+val_2+val_3}{3}$. As such, from Lemma 3 we have that $|val_5 - val_3| \leq \varepsilon$. With Lemma 2 and 3 at hand, Theorems 2 and 3 hold in game G', and therefore Theorem 4 give us the following result:

- $|val_5 - val_1| \leq \kappa \cdot \varepsilon$, where $\kappa = 2$.

- $|val_5 - val_2| \leq \kappa \cdot \varepsilon$, where $\kappa = 2$.
- $|val_5 - val_3| \leq \kappa \cdot \varepsilon$, where $\kappa = 2$.

Since r'_1 is equivalent to r_1, r'_2 is equivalent to r_2 and r'_3 is equivalent to r_3, the Lemmas and Theorems that hold for r'_1, r'_2 and r'_3 (game G') also hold for r_1, r_2 and r_3 (game G). Note that, conditions (I) and (II) for r'_1 and r'_2 correspond to conditions (VI) and (VII) for r_1 and r_2, respectively. Similarly, conditions (III), (IV) and (V) for r'_3 and r'_4 correspond to conditions (VIII), (IX) and (X) for r_3 and r_4, respectively. As such, for each game G with larger collaborative patterns, all of the Lemmas and Theorems of Section 4.1.1.1 hold as well. □

As such, we propose Algorithm 5, that performs a series of such merges. Obviously, Algorithm 1 consists a sub-case of Algorithm 5, since it operates with only pairwise rules (i.e., $g = 2$). Asymptotically Algorithm 5 has a computational complexity of $\mathcal{O}(m^2)$, where m is the size of R set of rules.

Nevertheless, in the rest of this thesis we consider initial rules with collaborative pairs.

Algorithm 5 Merging rules with larger collaborative patterns

$R \leftarrow$ initial set of MC-net-like rules of size m;
$R' \leftarrow \emptyset$;
for $r \in R$ **do**
  CG $\leftarrow \emptyset$;
  $\kappa \leftarrow 0$;
  $V_{\text{CG}} \leftarrow val_r$;
  min $\leftarrow \min_{v \in V_{\text{CG}}}\{v\}$;
  max $\leftarrow \max_{v \in V_{\text{CG}}}\{v\}$;
  avg $\leftarrow val_r$;
  remove r from R;
  for $r' \in R$ **do**
    if CanFullMergePetterns(r, r') **OR** CanHalfMergePatterns(r, r') **then**
      add $val_{r'}$ in V_{CG};
      add the sub-patterns without the mutual agent in CG;
      min $\leftarrow \min_{v \in V_{\text{CG}}}\{v\}$;
      max $\leftarrow \max_{v \in V_{\text{CG}}}\{v\}$;
      avg $\leftarrow \frac{\text{avg} \cdot |V_{\text{CG}}| + val_{r'}}{|V_{\text{CG}}|+1}$;
      $\kappa \leftarrow \kappa + 1$;
      $r_{new} \leftarrow a_{mutual} \wedge \text{CG} \rightarrow \text{avg}$;
      $r \leftarrow r_{new}$;
      remove r' from R;
    end if
  end for
  add r in R';
end for
return R';

Algorithm 6 CANFULLMERGEPATTERNS(r, r')

```
if r and r' satisfy Cond. (VI) AND (VII) then
    return True;
end if
return False;
```

Algorithm 7 CANHALFMERGEPATTERNS(r, r')

```
if r and r' satisfy Cond. (VIII) AND (IX) AND (X) then
    return True;
end if
return False;
```

4.1.2 Equivalent Agents

In this section we discuss a variant of the ε-MC nets representation that exploits not only *mutual* agents, but also *equivalence classes* of agents: agents belonging in the same class may have similar behaviour, preferences or properties—for example, in a search & rescue mission all firefighters comprise one equivalence class, while all nurses another. Considering equivalences among agents we manage to: *(a) compress even more* the representation compared to the initial version; *(b) extract underling patterns* that were not observed in the initial set of MC-net-like rules.

Definition 14 (Equivalent Agent) *Given a set of agents N, and a similarity metric $s : N \times N \to [0, 1]$, two agents i and j are equivalent if and only if $s(i, j) \geq$ threshold.*

The threshold denotes the minimum similarity degree for two agents in order for them to be equivalent, and depends on the problem at hand. Note also that any similarity metric s of choice can be used given a real-world scenario of interest. In Section 5.1.3 below we demonstrate how to employ to this purpose specific correlation metrics that are used in many real life scenarios.

Now, given Definition 14 above, we obtain a new version of ε-MC nets representation, where the rules take the form:

$$\Omega_{\text{equiv}} \wedge \text{CG} \to val \tag{4.11}$$

where Ω_{equiv} is a set of equivalent agents and it replaces the *reference agent* of the initial representation. In words, a rule $\Omega_{\text{equiv}} \wedge \text{CG} \rightarrow val$ is interpreted as: *Our estimate of the collaboration between any agent $i \in \Omega_{equiv}$ with any agent $j \in$ CG is equal to val.*

In order to take advantage of equivalence classes, we need to modify Algorithm 1 slightly. Specifically we need to alter line 12 so it checks if the rules r and r' have agents in the same *equivalence class*. We consider that for every agent $i \in N$, there is a list $Equiv_i$, which contains all the agents that are equivalent with agent i. It also natural to assume that if $j \in Equiv_i$ then it holds that $i \in Equiv_j$. As such, considering two MC-net-like rules $r : i \wedge j \rightarrow val_{i,j}$ and $r' : k \wedge l \rightarrow val_{k,l}$, for the full-merge, we have to replace condition (I) with the following condition:

(XI) $i \in Equiv_k$ or $i \in Equiv_l$ or $j \in Equiv_k$ or $j \in Equiv_l$. Remember that i,j,k and l are positive or negative literals, thus if it stands for example that $i \in Equiv_k$ it means that they are both positive or both negative;

Similarly, considering an $\varepsilon-$MC net rule $r : \Omega_{\text{equiv}} \wedge \text{CG} \rightarrow val_{\text{CG}}$ and an MC-net-like rule $r' : j \wedge k \rightarrow val_{j,k}$, for the half-merge, we have to replace the condition (III) with the following condition:

(XII) For every $z \in \Omega_{\text{equiv}}$, $j \in Equiv_z$ or $k \in Equiv_z$, where two literals are identical if they are both positive or both negative;

As such, we introduce Algorithm 8 that exploits equivalence classes of agents.

This modification changes the complexity of the algorithm to $\mathcal{O}(n \cdot m^2)$, where n is the number of agents in the system, and m is the size of the initial MC-net-like rules set. That is, in each iteration, given two rules $r : i \wedge j \rightarrow val_r$ and $r' : k \wedge l \rightarrow val_{r'}$, we need to check if we can perform either full-merge (line 13). As such, we access the lists $Equiv_i$, $Equiv_j$, $Equiv_k$, and $Equiv_l$ (every list has a size of $size_{list}$, where $1 \leq size_{list} \leq n$). Similarly, we work for the half-merges. Due to these modifications, asymptotically the complexity of Algorithm 8 is $\mathcal{O}(n \cdot m^2)$.

An important thing to mention here, is that the employment of Algorithm 8 that considers equivalence classes, may result in *ambiguous* $\varepsilon-$MC nets rules. That is, depending on the way agents' equivalence is determined, we may end up producing overlapping rules, i.e., multiple $\varepsilon-$MC nets rules may apply to the very same collaborative pair. To overcome this ambiguity we consider that the *post-merge estimate* for a collaborative pair $i \wedge j$ equals the average value of the rules that apply to this pattern.

Algorithm 8 Equivalent classes of agents

$R \leftarrow$ initial set of MC-net-like rules of size m;
$R' \leftarrow \emptyset$;
for $r \in R$ **do**
  CG $\leftarrow \emptyset$;
  $\Omega_{\text{equiv}} \leftarrow \emptyset$;
  $\kappa \leftarrow 0$;
  $V_{\text{CG}} \leftarrow val_r$;
  min $\leftarrow \min_{v \in V_{\text{CG}}}\{v\}$;
  max $\leftarrow \max_{v \in V_{\text{CG}}}\{v\}$;
  avg $\leftarrow val_r$;
  remove r from R;
  for $r' \in R$ **do**
    if CanFullMergeEquiv(r, r') **OR** CanHalfMergeEquiv(r, r') **then**
      add $val_{r'}$ in V_{CG};
      add equivalent agents in Ω_{equiv};
      add the rest of the agents in CG;
      min $\leftarrow \min_{v \in V_{\text{CG}}}\{v\}$;
      max $\leftarrow \max_{v \in V_{\text{CG}}}\{v\}$;
      avg $\leftarrow \frac{\text{avg} \cdot |V_{\text{CG}}| + val_{r'}}{|V_{\text{CG}}|+1}$;
      $\kappa \leftarrow \kappa + 1$;
      $r_{new} \leftarrow \Omega_{\text{equiv}} \wedge \text{CG} \rightarrow \text{avg}$;
      $r \leftarrow r_{new}$;
      remove r' from R;
    end if
  end for
  add r in R';
end for
return R';

Algorithm 9 CanFullMergeEquiv(r, r')

```
if r and r' satisfy Cond. (XI) AND (II) then
    return True;
end if
return False;
```

Algorithm 10 CanHalfMergeEquiv(r, r')

```
if r and r' satisfy Cond. (XII) AND (IV) AND (V) then
    return True;
end if
return False;
```

We demonstrate this via an example:

Example 5 *Consider a set of* 4 *agents* $N = \{1, 2, 3, 4\}$*, and the following initial MC-net-like rules:*

$$(r_1) : 1 \wedge 3 \rightarrow 5$$

$$(r_2) : 2 \wedge 4 \rightarrow 6$$

$$(r_3) : 1 \wedge 4 \rightarrow 8$$

Moreover, we let $\varepsilon = 1$*, while we also have the information that agents* 1 *and* 2 *are equivalent. Employing Algorithm 8, with equivalent agents, we get the following result. We begin with rule* r_1 *which can be merged with* r_2 *since agent 1 is equivalent to agent 2, and* $5-1 \leq 6 \leq 5+1$*. Thus, we generate a new* $\varepsilon-MC$ *net rule* $r_4 : \{1, 2\} \wedge \{3, 4\} \rightarrow \frac{5+6}{2}$*, and we replace* r_1 *with* r_4*. Rule* r_4 *has a mutual agent with* r_3*, however it does not stand* $5.5 - 1 \leq 8 \leq 5.5 + 1$*, thus rule* r_3 *is not merged with* r_4*, and remains intact. As such the final set of rules is:*

$$(r_3) : 1 \wedge 4 \rightarrow 8$$

$$(r_4) : \{1, 2\} \wedge \{3, 4\} \rightarrow 5.5$$

As we can see the collaboration pattern $1 \wedge 4$ *satisfies both of the rules* r_3 *and* r_4*. So the value of this collaboration pattern is equal to* $\frac{5.5+8}{2} = 6.75$*. Moreover notice that in the final* $\varepsilon-MC$ *nets representation there is information about the collaboration pattern* $2 \wedge 3 \rightarrow 5.5$*, which was previously unknown.*

4.1.3 Extension to NTU Games

The ε-MC nets representation can also be used in non-transferable utility (NTU) games [2], where each member of a coalition C now derives an individual utility from participating in C. In such settings, every agent i could have a personal ε_i, that would represent the margins of information loss that agent i is willing to accept in order to produce a compact representation. Therefore each agent i can execute its own instance of any algorithm that has been described in the previous sections, using its personal ε_i and yield a representation regarding its personal estimates. In this extension, given an environment with n agents, we would end up with n compact $\varepsilon-$MC nets representations, where the i^{th} representation corresponds to agent's i estimates. However, in our experimental evaluation (in Section 5 below) we consider only transferable utility environments.

Chapter 5

Experimental Evaluation

In this section we evaluate the performance of our algorithms via simulations. All experiments ran on an i5@2.2GHz processor PC with 8GB of RAM, and the framework was coded in Python 3.8.

5.1 $\varepsilon-$MC net with Mutual Agents

First we present a series of experiments performed to evaluate Algorithm 1 with mutual agents, using synthetic data.

5.1.1 Dataset

We generated synthetic data with varying number of agents n and varying number of rules m. Specifically, n takes the values 100, 200 and 300; while m varies depending on n and takes the values $\frac{n}{2}, n$, and $2 \cdot n$. In each dataset, every rule consist of a pair of agents randomly selected out of $\binom{n}{2}$ possible unordered pairs; and the rule's value is drawn from uniform distribution $\mathcal{U}(1, 200)$. Finally, for each combination $\langle n, m \rangle$ we generated 5 different datasets. As such, in total we use 45 different datasets.

5.1.2 Experimental Analysis

We ran our algorithm for each setting using different values of ε, i.e., the margin of information loss. The evaluation metric we used is the *percentage of reduction*, i.e., we

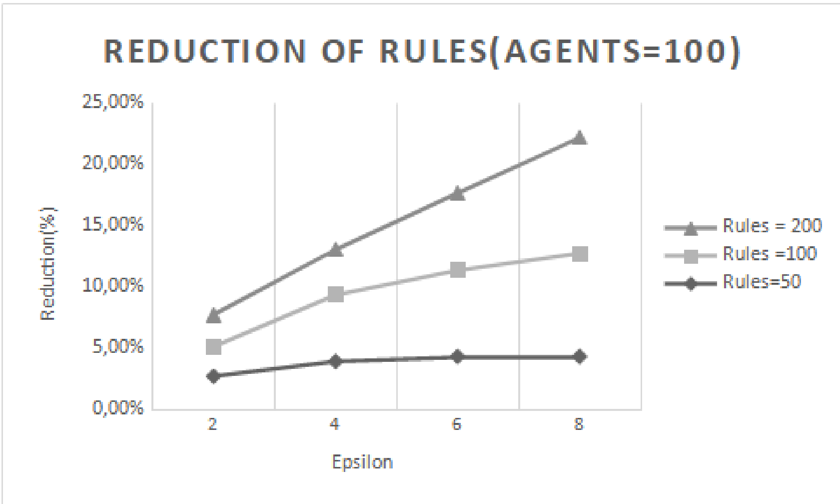

Figure 5.1: Reduction of rules using Algorithm 1, when $N = 100$.

measured the compactness achieved in the $\varepsilon-$MC nets representation by computing the number of rules comprising the new representation compared to the number of the initial MC-net-like rules. Formally, the reduction percentage (RP) is:

$$RP = \left(1 - \frac{\#\varepsilon\text{-MC nets rules} + \#\text{un-merged rules}}{\#\text{initial MC-net-like rules}}\right) \cdot 100\%$$

In Figures 5.1, 5.2 and 5.3 we illustrate graphically the results of our implementation of Algorithm 1. The results represent the average reduction percentage achieved in each setting $\langle n, m \rangle$, where the average is over 5 datasets within the same setting. As we can observe, as the number of rules increases, with ε kept fixed across different settings, our algorithm achieves greater reduction percentage. Such a result is expected, since when we have more rules it is more likely to find MC-net-like rules that satisfy the conditions for merging, and thus the algorithm produces more compact representations. Also, for the same number of MC-net-like rules, as ε increases, we observe that the achieved reduction increases as well. This result is due to the fact that for greater values of ε, the conditions for merging are more relaxed, and thus easier to be met.

In Figures 5.4, 5.5 and 5.6 we compare the achieved reduction percentage between different settings with fixed ratio $\frac{n}{m}$. Here we notice that the RPs exhibit similar behaviour, regardless of the setting at hand.

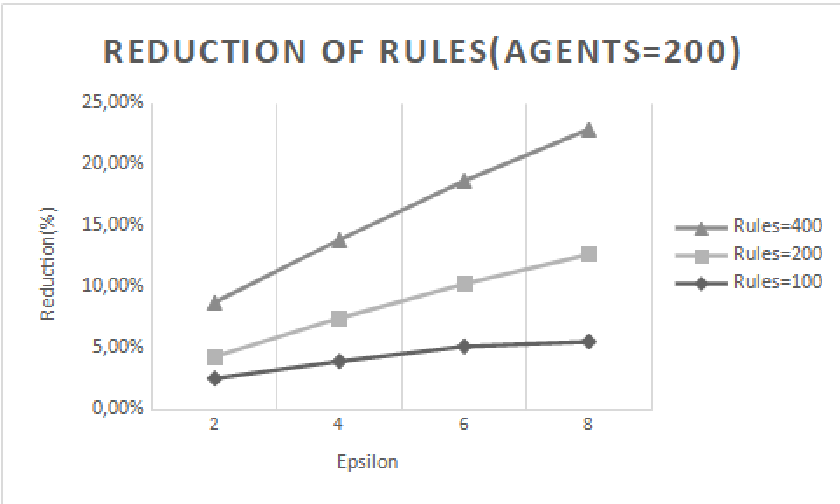

Figure 5.2: Reduction of rules using Algorithm 1, when $N = 200$.

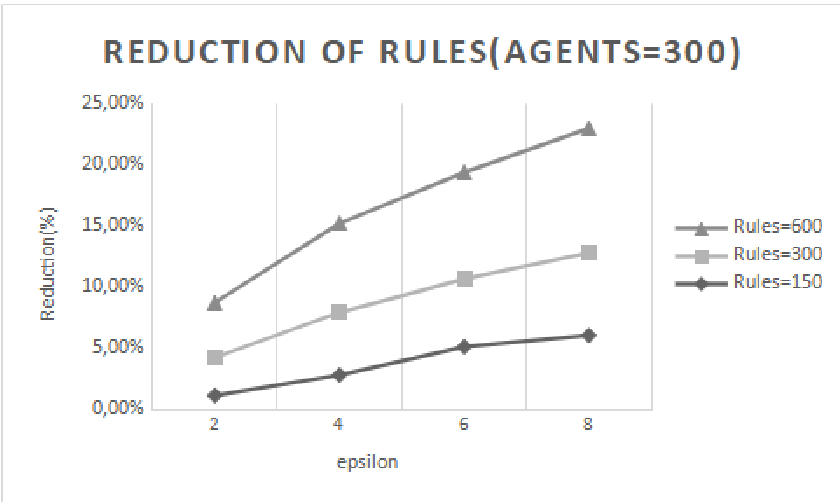

Figure 5.3: Reduction of rules using Algorithm 1, when $N = 300$.

5.1.3 RP: Mutual vs Equivalent Agents

In this line of evaluation we compared the performance of Algorithm 1 against its variant, Algorithm 8, that considers equivalent agents, in terms of percentage of reduction. Here we generated 75 *synthetic datasets* following the process described in Sec. 5.1.1; now n takes the values $50, 100, 200, 300$ and 400, while $m = \frac{n}{2}, n$ and $2 \cdot n$; and again we generated different datasets for each $\langle n, m \rangle$ combination. In order to determine equivalence

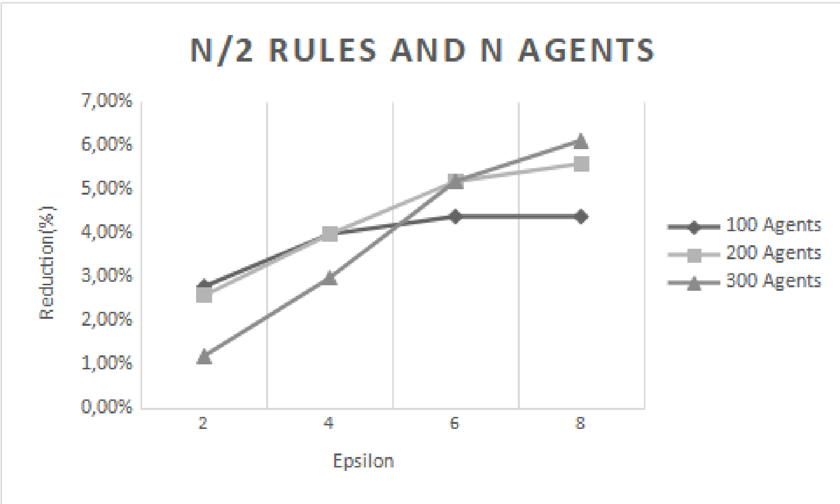

Figure 5.4: Reduction Percentage with fixed ratio $\frac{n}{m} = 2$.

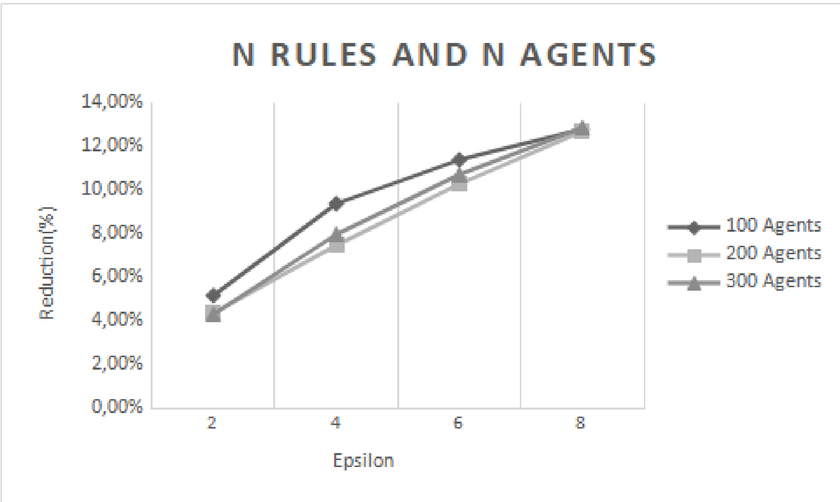

Figure 5.5: Reduction Percentage with fixed ratio $\frac{n}{m} = 1$.

among agents we adopted the following scenario: agents participate in a *ridesharing setting* as drivers or commuters. First, to determine the payoffs of agents participating in the ridesharing scenario, we ran the PK Algorithm (Algorithm 5) of work [21],[1] which computes *kernel-stable* [2, 22] payments for such scenarios. Specifically, for each dataset we run the PK algorithm for a number of partitions depending on the number of agents in the dataset. Each such partition consisted of a randomly sampled coalition C contain-

[1]We retrieved the PK implementation code from https://github.com/filippobistaffa/PK

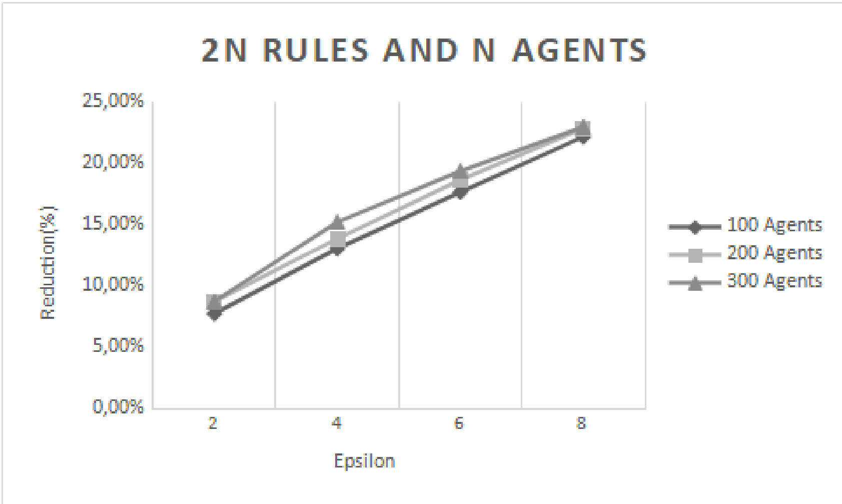

Figure 5.6: Reduction Percentage with fixed ratio $\frac{n}{m} = 0.5$.

ing one driver,[1] at least 1 commuter, and at most 4 (commuters), while all other agents where considered as singletons—i.e., $\pi = \{C\} \cup \{\{i\} \in N \setminus C\}$. The total number of partitions used per dataset family is shown in Table 5.1.

Number of Agents	50	100	200	300	400
Number of Partitions	1150	3700	12000	25000	40000

Table 5.1: Number of partitions used in PK Algorithm per dataset family in Section 5.1.3.

As soon as we have the payoffs, for every pair of agents i, j we build two ranking lists M_i and M_j as follows:
For the k^{th} sampled partitions π (with $C \in \pi$):

1. if $\{i, j\} \subseteq C$:
   - add i's payoff according π in the k^{th} position of M_i;
   - add j's payoff according π in the k^{th} position of M_j.
2. else if $i \in C$ and $\exists\ \pi'$ such that
   $j \in C'$ and $C \setminus \{i\} \equiv C' \setminus \{j\}$ with $C' \in \pi'$:

[1] In all datasets we let 20% of the agents' population be drivers, and 80% of the population be commuters.

- add i's payoff according π in the k^{th} position of M_i;
- add j's payoff according π' in the k^{th} position of M_j.

We then use the lists above to determine the equivalence between any two agents i and j, via combining two widely used correlation metrics, *(a)* the Kendall's Tau ranking distance [23], and *(b)* the Pearson Correlation Coefficient [24].

According to Kendall's Tau, the ranking distance between two lists M_i and M_j depends on the number of misalignments of any pair of elements between the two lists. That is, let $M_i[k]$ and $M_i[l]$ denote the value in k^{th} and l^{th} position in list M_i, respectively (similarly for $M_j[k]$ and $M_j[l]$); there is a misalignment between M_i and M_j if $M_i[k] \geq M_i[l]$ and $M_j[k] < M_j[l]$ or $M_i[k] < M_i[l]$ and $M_j[k] \geq M_j[l]$. Thus, the Kendall's Tau distance between M_i and M_j is defined as the summation of misalignements between any unordered pair of positions k, l, normalized by the total number of unordered pairs of positions.

Now, the Pearson Correlation Coefficient measures the linear correlation between two random variables X and Y; and is computed as $r_{X,Y} = \frac{\sum_i (X_i - \bar{X}) \cdot (Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})} \cdot \sqrt{\sum_i (Y_i - \bar{Y})}}$. Considering lists M_i and M_j as random variables, the PCC between agents i and j is computed as

$$r_{M_i,M_j} = \frac{\sum_k (M_i[k] - \text{avg}\{M_i\}) \cdot (M_j[k] - \text{avg}\{M_j\})}{\sqrt{\sum_k (M_i[k] - \text{avg}\{M_i\})} \cdot \sqrt{\sum_k (M_j[k] - \text{avg}\{M_j\})}}$$

where the summation is over the positions in lists M_i and M_j.[1]

With the Kendall's Tau distance and the Pearson correlation coefficient at hand, two agents i and j are considered equivalent if it holds $K(M_i, M_j) \geq \textit{threshold}$ **and** $r_{M_i,M_j} \geq \textit{threshold}$. For these series of experimental analysis, we set $\textit{threshold} = 0.97$.

Table 5.2 shows the results (average over 5 datasets with the same combination $\langle n, m \rangle$) for every setting, when we employ the algorithm using equivalence classes of agents (denoted as "Equivalent") against when employing the original version, using solely mutual agents (denoted as "Mutual"). Here, we see that for every examined setting, the algorithm that takes advantage of equivalences consistently achieves *manyfold greater reduction* than the algorithm with the mutual agents. Such a result is expected due to the use

[1]Note that the Pearson Correlation Coefficient (PCC) measures correlation, as such its values lie in $[-1, 1]$ where $PCC \rightarrow 1$ indicates positive correlation, $PCC \rightarrow 0$ indicates no correlation, and $PCC \rightarrow -1$ indicates negative correlation.

of equivalences, since in this version we exploit information which is not considered in the version with only mutual agents.

Now considering the ratio $\rho = \frac{\text{Equivalent RP}}{\text{Mutual RP}}$ we can observe that the maximum value of ρ is equal to $\rho = \frac{5.6}{1.6} = 3.5$ for the family with 50 agents, $\rho = \frac{9.6}{1.2} = 8$ for the family with 100 agents, $\rho = \frac{18.8}{1.2} = 15.67$ for the family with 200 agents, $\rho = \frac{23.46}{1.98} = 11.84$ for the family of 300 agents, and finally $\rho = \frac{27.3}{1.6} = 17.06$ for the family with 400 agents. Note that this peak is achieved in the setting with $\frac{n}{2}$ initial MC-net-like rules for each family; this result is due to the fact that with fewer rules, it is hard to find rules to satisfy the merging constraints. Also the maximum reduction percentage for every family is achieved in the setting with $2 \cdot n$ rules. Once again, we can observe that as ε rises, the reduction percentage rises in both algorithms' variations as well, which is due to the relaxed constraints for merging.

Finally our experiments confirm that the extra information on equivalences among agents allows us not only to produce more succinct representations (as discussed above), but also to learn new collaboration patterns. We show this through the NCP ratio:

$$NCP = \frac{\text{New collaboration patterns}}{\text{Total number of collaboration patterns}} \cdot 100\%,$$

where *Total number of collaboration patterns* corresponds to the number of initial MC-net-like rules plus the new collaborative pairs of agents that our algorithm produced, exploiting equivalences among agents. Note that in case there are ambiguities, i.e., the same collaboration pattern is expressed by more than one rule, we consider this rule only once.

Table 5.3 shows the NCPs for every setting, when we employ the algorithm using equivalence classes of agents—the NCPs displayed are averages over the 5 different datasets for each combination $\langle n, m \rangle$. As we can observe, the NCP is rising, for a given n, as m rises. This is natural, since for larger m our algorithm is able to perform more merges. As a result, new collaboration patterns are produced. Furthermore, for the same reason, we observe a similar behaviour as the ε rises.

			$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$
$n = 50$	$m = 25$	Mutual	1.6%	1.6%	2.4%
		Equivalent	4%	5.6%	7.2%
	$m = 50$	Mutual	2%	3.6%	6%
		Equivalent	6.8%	10.4%	15.8%
	$m = 100$	Mutual	5.8%	8%	10.2%
		Equivalent	12.8%	19%	23.6%
$n = 100$	$m = 50$	Mutual	1.2%	3.6%	3.6%
		Equivalent	9.6%	14.8%	18.8%
	$m = 100$	Mutual	3%	5.4%	7.2%
		Equivalent	17.6%	25.8%	30.8%
	$m = 200$	Mutual	6.5%	9.4%	12.1%
		Equivalent	27.5%	37.9%	43.3%
$n = 200$	$m = 100$	Mutual	1.2%	2.4%	4.4%
		Equivalent	18.8%	25.2%	31.2%
	$m = 200$	Mutual	2.9%	5%	6.7%
		Equivalent	26.3%	35.4%	41.2%
	$m = 400$	Mutual	5.5%	8.75%	11.15%
		Equivalent	38.3%	47.75%	53.55%
$n = 300$	$m = 150$	Mutual	1.98%	2.8%	4.13%
		Equivalent	23.46%	31.45%	38.4%
	$m = 300$	Mutual	3.4%	4.6%	6.46%
		Equivalent	34.46%	44.76%	50.4%
	$m = 600$	Mutual	5.62%	8.89%	11.32%
		Equivalent	46.25%	56.13%	61.2%
$n = 400$	$m = 200$	Mutual	1.6%	3.2%	5%
		Equivalent	27.3%	36.5%	43.5%
	$m = 400$	Mutual	2.9%	4.95%	6.2%
		Equivalent	39.85%	50.4%	56.4%
	$m = 800$	Mutual	5.57%	8.92%	11.72%
		Equivalent	50.97%	60.82%	66.32%

Table 5.2: Reduction percentage per setting of Section 5.1.3.

		$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$
$n = 50$	$m = 25$	6.7%	14.3%	18.3%
	$m = 50$	7.7%	16.3%	20%
	$m = 100$	16.9%	22.4%	27.1%
$n = 100$	$m = 50$	14.3%	20%	29.9%
	$m = 100$	23%	33.4%	39%
	$m = 200$	32.7%	42.6%	47.2%
$n = 200$	$m = 100$	23.5%	35%	39%
	$m = 200$	37.2%	45.8%	52.8%
	$m = 400$	46.4%	56.6%	63.2%
$n = 300$	$m = 150$	34%	45.3%	51.2%
	$m = 300$	44.4%	55.2%	60%
	$m = 600$	57.2%	68%	73.1%
$n = 400$	$m = 200$	37.1%	49.5%	56%
	$m = 400$	52.2%	62.8%	67.7%
	$m = 800$	63.5%	73.3%	78.2%

Table 5.3: Percentage of new collaboration patterns.

Chapter 6

Conclusion and Future work

In this thesis, we extended the classic MC nets representation scheme to cooperative games with settings that contain either a specific type of uncertainty or overlaps among the coalitions that agents can form.

First, we extended the Relational Rules [8], so that they describe games with overlapping coalitions that contain both positive and negative literals. Our proposed representation, reduces to classic MC nets for non-overlapping settings.

Our second main contribution in this thesis, was introducing a novel succinct representation for cooperative games under uncertainty. This representation extends the work of [3] to allow for rules that include sets of agents, instead of just individuals; and can be employed in open multiagent settings, possibly under uncertainty regarding the value of collaboration patterns. We formally defined the concept of the $\varepsilon-$MC nets rules; the types of merging that can occur between rules; and we proposed a polynomial algorithm for constructing an $\varepsilon-$MC nets representation. Moreover, we determined a theoretical bound for the maximum information loss that our "compression process" may incur. Then, motivated by the future work envisaged in [3], we considered equivalence classes of agents, and put forward a variant of our algorithm which takes these into account. In this variant it is possible to generate values for collaboration patterns that were initially unknown to us.

We have to mention that the focus of this work is compressing the representation of large coalitional games; however, these are, from a practical point of view, almost by definition uncertain. Value/collaboration-related uncertainty is incorporated in the utility function, but we do not explicitly handle it otherwise; e.g. we do not actively attempt to reduce it, though partly this can be "passively" achieved as a result of our scheme, as explained in the paper (beginning of section 4.1.2 and example 5; and section 5.1.3, last paragraph) and below. Moreover, we believe that the presence of uncertainty in large settings, makes arguments in favor of compressing available information more compelling, despite information loss; one would have been more reluctant to suffer information loss if they knew information is accurate. Hence our approach is well-suited to uncertain environments. In any case, the MC-net-like rules may be derived either from the believed characteristic function or the actual one, if this is at hand (which is unlikely). Apart from that, according to our assumption that we cannot fully observe the game, notice that the exploitation of such equivalence classes can reveal the values of previously unknown collaboration patterns.

Next, we conducted a systematic evaluation of our approach. First, we studied the reduction percentage achieved by Algorithm 1 in various settings considering the combination of different number of agents, number of rules, and information loss margin (ε). We compared the performance in terms of reduction percentage (RP) of the algorithm considering solely mutual agents (Algorithm 1), against the one considering equivalent agents (Algorithm 8) as well. Our results verified the effectiveness of our approach. Specifically, they showed that the algorithm with mutual agents achieved an RP up to 12%, while the variant with equivalent agents achieved an RP up to 66%.

Finally, though we show it is polynomial, the complexity of the algorithm may be a problem in *very large* settings where agents need to make decisions and form coalitions "on-line". However, in practice, even then compressing the representation using our algorithm can prove to be extremely useful: e.g., Algorithm 1 can be employed "off-line" only once for all agents originally in the setting, and then one can use the resulting compact representation for merges of its content rules with rules involving newcomer agents (in a batch-processing style). Moreover, note that in our larger experimental settings, involving 400 agents and 800 rules, compression was achieved in less than a minute on a simple laptop (i5@2.2GHz).

Regarding future work, we intend to extend Algorithm 8 to perform a back-tracking technique. That is, merges that were rejected at some point, may become feasible (i.e., they can be performed) due to the entry of equivalent agents in the set Ω_{equiv} of an ε-MC net rule—i.e., a MC-net-like rule r may cannot half-merge to an ε-MC net rule r' since there are not equivalent agents in Ω_{equiv} of r', but after some merges in r', the Ω_{equiv} of r' may contains some new equivalent agents and r can now merge to r'.

Another interesting line of research is to set acceptable bounds on the maximum uncertainty over a collaborative pair (depending on the number of merges), and devise methods for optimally selecting the rules to be merged in order to maximise the reduction percentage. Moreover, further experimentation in different real-world domains is in order. Finally, since this work extends readily into NTU-games (as discussed in Section 4.1.3), we aim to conduct a systematic evaluation in such settings.

References

[1] Wooldridge, M.: An Introduction to MultiAgent Systems. 2nd edn. Wiley Publishing (2009) 1

[2] Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational Aspects of Cooperative Game Theory (Synthesis Lectures on Artificial Inetlligence and Machine Learning). 1st edn. Morgan & Claypool Publishers (2011) 1, 2, 6, 7, 8, 9, 39, 44, 62, 65

[3] Ieong, S., Shoham, Y.: Marginal contribution nets: a compact representation scheme for coalitional games. In: EC '05. (2005) 1, 2, 3, 11, 12, 13, 51, 59

[4] Deng, X., Papadimitriou, C.H.: On the complexity of cooperative solution concepts. Math. Oper. Res. **19**(2) (May 1994) 257–266 1, 10

[5] Conitzer, V., Sandholm, T.: Complexity of determining nonemptiness of the core. In: Proceedings of the 4th ACM Conference on Electronic Commerce. EC '03, New York, NY, USA, Association for Computing Machinery (2003) 230–231 1

[6] Dang, V.D., Dash, R.K., Rogers, A., Jennings, N.R.: Overlapping coalition formation for efficient data fusion in multi-sensor networks. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1. AAAI'06, AAAI Press (2006) 635–640 2, 9

[7] Chalkiadakis, G., Elkind, E., Markakis, E., Polukarov, M., Jennings, N.: Cooperative games with overlapping coalitions. J. Artif. Intell. Res. (JAIR) **39** (09 2010) 179–216 2, 9, 10

[8] Mamakos, M., Chalkiadakis, G.: Overlapping coalition formation via probabilistic topic modeling. In André, E., Koenig, S., Dastani, M., Sukthankar, G., eds.: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018, International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM (2018) 2010–2012 2, 3, 13, 15, 16, 17, 18, 51

[9] Saad, W., Han, Z., Zheng, R., Hjørungnes, A., Başar, T., Poor, H.V.: Coalitional games in partition form for joint spectrum sensing and. IEEE Journal of Selected Topics in Signal Processing - IEEE J SEL TOP SIGNAL PROCESS **6** (02 2012) 8

[10] Banerjee, S., Konishi, H., Sönmez, T.: Core in a simple coalition formation game. Social Choice and Welfare **18**(1) (2001) 135–153 8

[11] Aziz, H., Savani, R., Moulin, H. In: Hedonic Games. Cambridge University Press (2016) 356–376 8

[12] Georgara, A., Troullinos, D., Chalkiadakis, G.: Extracting hidden preferences over partitions in hedonic cooperative games. In Douligeris, C., Karagiannis, D., Apostolou, D., eds.: Knowledge Science, Engineering and Management, Cham, Springer International Publishing (2019) 829–841 9, 64

[13] Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence **101**(1) (1998) 165 – 200 9

[14] Gillies, D.B. In: 3. Solutions to General Non-Zero-Sum Games. Princeton University Press, Princeton (31 Dec. 1959) 47 – 86 10

[15] Zick, Y., Elkind, E.: Arbitrators in overlapping coalition formation games. Volume 1. (01 2011) 55–62 10

[16] Michalak, T., Marciniak, D., Szamotulski, M., Rahwan, T., Wooldridge, M., McBurney, P., Jennings, N.R.: 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, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2010) 125–132 12

[17] Michalak, T., Rahwan, T., Marciniak, D., Szamotulski, M., Jennings, N.: Computational aspects of extending the shapley value to coalitional games with externalities. (01 2010) 12

[18] Chalkiadakis, G., Elkind, E., Jennings, N.R.: Simple coalitional games with beliefs. In: Proceedings of the 21st International Jont Conference on Artifical Intelligence. IJCAI'09, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2009) 85–90 19

[19] Kraus, S., Shehory, O., Taase, G.: The advantages of compromising in coalition formation with incomplete information. In: 3rd International conference of Autonomous Agent and Multiagent Systems (AAMAS'04). (02 2004) 588– 595 19

[20] Habis, H., Herings, P.: Transferable utility games with uncertainty. Maastricht : METEOR, Maastricht Research School of Economics of Technology and Organization, Research Memoranda **146** (01 2010) 19

[21] Bistaffa, F., Farinelli, A., Chalkiadakis, G., Ramchurn, S.: A cooperative game-theoretic approach to the social ridesharing problem. Artificial Intelligence **246** (02 2017) 86–117 44

[22] Stearns, R.E.: Convergent transfer schemes for n-person games. Transactions of the American Mathematical Society **134**(3) (1968) 449–459 44

[23] Kendall, M.: Rank correlation methods. Griffin, London (1948) 46

[24] Freedman, D., Pisani, R., Purves, R.: Statistics (international student edition). Pisani, R. Purves, 4th edn. WW Norton & Company, New York (2007) 46

[25] Elkind, E., Wooldridge, M.: Hedonic coalition nets. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1. AAMAS '09, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2009) 417–424 60, 61

[26] Saad, W., Han, Z., Zheng, R., Hjørungnes, A., Başar, T., Poor, H.V.: Coalitional games in partition form for joint spectrum sensing and. IEEE Journal of Selected Topics in Signal Processing - IEEE J SEL TOP SIGNAL PROCESS **6** (02 2012) 63

[27] Davis, M., Maschler, M.: The kernel of a cooperative game. Naval Research Logistics Quarterly **12**(3) (1965) 223–259 65

Chapter 7

Appendices

7.1 Appendix A

Here we discuss work in progress that extends the classic MC nets representation to NTU environments and more specifically to *Hedonic Games*.

Inspired by MC nets representation [3], we introduce a representation scheme for non-transferable utility games, called *Rule by Agent*. The main idea of such representation, is to define the *preferences* of agents over all possible coalitions, using sets of rules.

7.1.1 Rule by Agent representation

Let N be a finite, non-empty set of players of size $|N| = n$. Rule by agent representation uses sets of rules for every agent $i \in N$. The set of rules of agent i is denoted as R^i. Such rules have the following syntactic form:

$$r : Pattern \rightarrow val_i$$

where *Pattern* is a conjunction of m agents, with $m \leq |N|$, and $val_i \in \mathbb{R}$. A rule is said to apply on a coalition $C \subseteq N$, denoted by $r \models C$, if *Pattern* evaluates to true for coalition C.

Given any coalition C, we can compute the utility of C for agent i, denoted as $u_i(C)$, by summing the values of the rules, $r \in R^i$, that apply to C:

$$u_i(C) = \sum_{r \models C, r \in R^i} val_i$$

As such, for every agent i, it holds that $C_1 \succeq_i C_2$ if and only if $u_i(C_1) \geq u_i(C_2)$.

Example 6 *Consider a game of 3 agents $N = \{1, 2, 3\}$, which is described by the following "Rule by Agent" representation:*

$$R^1 = \{r_1 : 1 \wedge 2 \rightarrow 3, r_2 : 1 \wedge 3 \rightarrow 2, r_3 : 1 \wedge 2 \wedge 3 \rightarrow 5\}$$

$$R^2 = \{r_4 : 1 \wedge 2 \rightarrow 2, r_5 : 1 \wedge 2 \wedge 3 \rightarrow 4\}$$

$$R^3 = \{r_6 : 1 \wedge 3 \rightarrow 3, r_7 : 1 \wedge 2 \wedge 3 \rightarrow 6\}$$

We want to compute the utility that agent 1 and 3 will receive from coalition $C_1 = \{1, 3\}$. For agent 1 we sum up the rules from R^1 that apply to C_1, while for agent 3 we are interested for the rules of R^3. Thus, we get that $u_1(C_1) = 2$ and $u_3(C_1) = 3$. Following the same procedure, for coalition $C_2 = \{1, 2, 3\}$ we get that $u_1(C_2) = 3 + 2 + 5 = 10$ and $u_3(C_2) = 3 + 6 = 9$. As such it holds that $C_2 \succeq_1 C_1$ and $C_2 \succeq_3 C_1$.

In the course of this work, we realize that the Rule by agent representation is largely equivalent with the work of Hedonic coalition nets [25].

7.1.2 Rule by Values representation

As an alternative to Rule by Agent representation, we introduce the Rule by Values representation. In this representation scheme we use a set of rules R that pertains to every agent $i \in N$. Formally a rule of such representation has the following form:

$$r : Pattern \rightarrow \overrightarrow{Val}$$

where *Pattern* is a conjunction of m agents, with $m \leq |N|$, and $\overrightarrow{Val}$ is a vector of size $|N|$, that contains the utilities for every agent i. A common assumption regarding the utility of any agent i, is to be considered that if i does not appear in a *Pattern* of rule r, he receives a utility of zero (from rule r). Although that kind of hypothesis does not hold in every case (i.e., Section 7.1.3.1). As an example, consider that $|N| = 3$ and a rule $r : 1 \wedge 3 \rightarrow \{2, 0, 5\}$, we get that agent 1 will receive a value equal to 2 when r applies to a coalition. Similarly agent 2 will receive a value equal to 0 and agent 3 will receive a value equal to 5.

Finally, in order to compute the utility of coalition C for agent i, we sum up the $Val[i]$ of every rule that apply to C:

$$u_i(C) = \sum_{r \models C} Val[i]$$

Again, for any agent i, it holds that $C_1 \succeq_i C_2$ if and only if $u_i(C_1) \geq u_i(C_2)$.

Example 7 *For the same settings as Example 6 the "Rule by Values" representation is as follows:*

$$\begin{gathered} r_1 : 1 \wedge 2 \rightarrow \{3, 2, 0\} \\ r_2 : 1 \wedge 3 \rightarrow \{2, 0, 3\} \\ r_3 : 1 \wedge 2 \wedge 3 \rightarrow \{5, 4, 6\} \end{gathered}$$

Regarding the utilities of agents there is no difference with Example 6.

7.1.3 Additively Separable Hedonic Games (ASHGs)

Additively Separable Hedonic Games (ASHGs) [25], is a class of hedonic games where agents preferences over coalitions are lifted from preferences over agents. That is, each agent $i \in N$ assigns a value $b_i^j \in \mathbb{R}$ to every other agent $j \in N$, meaning that i sets a value b_i^j from the fact that j participates in C. Agent's i preferences for a coalition C corresponds to the summation of the individual values b_i^j, $\forall j \in C$. Formally, it holds that:

$$u_i(C) = \sum_{j \in C} b_i^j$$

Generally ASHGs is more natural to be modelled as NTU games. Thus, we will use the "Rule by Agent" representation in order to represent such games. For each agent $i \in N$ we hold a set of rules R^i, which consists pairwise rules between agent i and every $j \in N$. The value, val, of a rule $r : i \wedge j \rightarrow val$, is equal to b_i^j,[1]. Finally, the value of coalition C for agent i, $u_i(C)$, is defined as:

$$u_i(C) = \sum_{r \in C, r \in R^i} val_i$$

[1]The agents assign a zero value to themselves, i.e., $b_i^i = 0$.

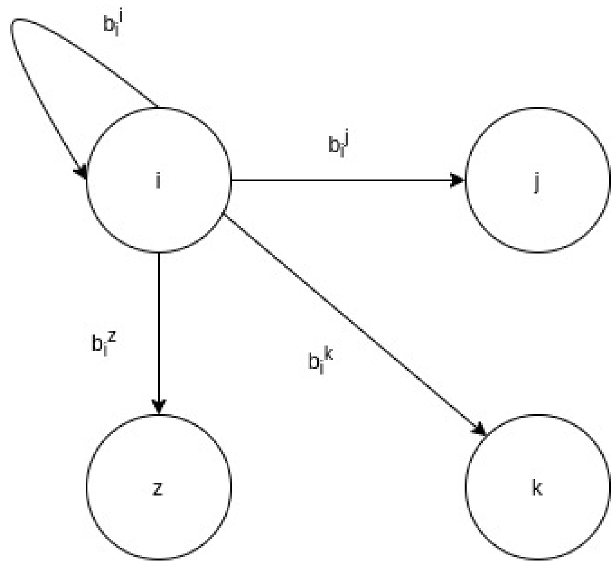

Figure 7.1: Graphical representation of Rule by agent form.

This representation can easily be extended in a graphical way if we restrict the rules to be about pairs of agents. Supposing that we are interested in computing the value that agent i holds for a coalition C. We generate a weighted directed graph, where the node represented agent i has only outgoing edges, while the nodes represented the rest of the agents in C have only incoming edges.

Every edge (i, j) has a weight b_i^j, given by the rule $r : i \wedge j \rightarrow b_i^j$, where $r \in R^i$. An example of such representation shown in Figure 7.1. Again to compute i's utility for a coalition C, we sum the weights of every edge that connects agent i with every agent $j \in C$. In [2] an ASHGs can be defined as:

Definition 15 *A hedonic game $G = (N, \succeq_1, \cdots, \succeq_n)$ is said to be additively separable if there exists an $|N| \times |N|$ matrix of reals M (the value matrix) such that:*

$$C_1 \succeq_i C_2 \text{ iff } \sum_{j \in C_1} M[i, j] \geq \sum_{k \in C_2} M[i, k]$$

where $M[i, j]$ represents the value of agent i to agent j.

Theorem 5 *The "Rule by Agent" constitutes a complete representation scheme for ASHGs.*

Proof. Consider an $|N| \times |N|$ matrix of real values M. The $M[i, j]$ represents the value that agent i holds for agent j. For every i and j we generate a rule $r : i \wedge j \rightarrow M[i, j]$.

Consequently we add rule r to the set of rules R^i. At the end of that procedure, we get $|N|$ sets of rules and each one of these sets consist $|N|$ rules. As such, we need $|N| \times |N|$ rules in order to fully describe the game. Clearly, that is exactly the same amount of space that we need with the M matrix. □

Theorem 6 *The "Rule by Values" constitutes a complete representation scheme for ASHGs.*

Proof. Similarly to the proof of Theorem 5 it can be shown that Rule by Values representation needs exactly $\frac{|N| \times |N|}{2}$ rules in order to fully describe the game. □

Generally, the "Rule by Values" consist a more compact representation scheme than the "Rule by Agent".

Finally, an ASHG is called *symmetric* if $M[i,j] = M[j,i]$, $\forall i,j = 1, \cdots n$. Using the "Rule by Agent" representation an ASHG is called *symmetric* if $value_{ij} = value_{ji}$, $\forall i,j = 1, \cdots n$, where $value_{ij}$ is the value of rule $r : i \land j \to value_{ij}(r \in R^i)$ and $value_{ji}$ is the value of rule $r' : j \land i \to value_{ji}(r' \in R^j)$.

7.1.3.1 Partition Function Form (PFF)

A *partition* (or coalition structure) is a set of coalitions $\pi = \{C_1, \cdots, C_m\}$ such that for every $i, j = \{1, \cdots, m\}$ the following conditions must be satisfied:

- $C_i \cap C_j = \emptyset$, when $i \neq j$.
- $\bigcup_{i \in 1, \cdots, m} C_i = N$.

In these games, the value of a coalition depends on the coalition structure that it appears in.

Partition function games can be expressed either in transferable (TU) or non-transferable (NTU) utility games. As mentioned before, hedonic games is a subclass of NTU games. In [26], a coalitional game in PFF with non-transferable utility defined as:

Definition 16 *A coalitional game in partition function form (PFF) with non-transferable utility (NTU) is defined by a pair $\langle N, V \rangle$, where N is the set of agents, and V is a mapping such that for every $\pi \in \Pi$ and every coalition $C \subseteq N$, $C \in \pi$, $V(C, \pi)$ is a closed convex subset of $\mathbb{R}^{|C|}$ that contains the payoff vector that players in C can achieve.*

In [12], authors provide a definition for *hedonic games in partition function form.*

Definition 17 *A hedonic game (HG) in partition function form (PFF) is defined by a pair $\langle N, \succeq \rangle$, where N is the set of players, and $\succeq = \{\succeq^{\pi_1}, \cdots, \succeq^{\pi_m}\}$ with $|\Pi| = m$; and for all $\pi_j \in \Pi$ $\succeq^{\pi_j} = \{\succeq_1^{\pi_j}, \cdots, \succeq_n^{\pi_j}\}$, and each $\succeq_i^{\pi_j} \subseteq N_i \times N_i$ is a complete, reflexive and transitive preference relation describing agent i's preferences over coalitions it can participate in when π_j is in place.*

Generalizing ASHGs to PFF, each agent $i \in N$ assigns a value $b_i^j(\pi)$ to any agent j when partition π is formed, and the utility of an embedded coalition (C, π) is defined as:

$$u_i(C, \pi) = \sum_{j \in C} b_i^j(\pi)$$

Finally, it holds that $(C_1, \pi_1) \succeq_i (C_2, \pi_2)$ if and only if $u_i(C_1, \pi_1) \geq u_i(C_2, \pi_2)$.

7.1.4 Rule by agent representation for PFF

The “Rule by agent” representation can be fully expressive in hedonic games as described before. We extend “Rule by agent” representation into PFF by modifying the form of rules. Specifically each agent $i \in N$ holds a set of rules R^i, containing pairwise rules of the following form:

$$r : j \wedge k \rightarrow val_i$$

where j, $k \in N$ and $val_i \in \mathbb{R}$. Intuitively each agent evaluates the collaboration of every two agents j, $k \in N$ with a value val_i and the number of rules that each set R^i contains is $|N|^3$, where N is the set of agents.

However, in *partition function games* the value of a coalition depends on the coalition structure that it appears in. Thus, it is more natural to model such games with rules that considers every time the formed coalition structure π, i.e., if a collaboration pattern between two agents is not part of the coalition that contains agent i, then the utility of i (for that collaboration pattern) may be reduced. As such the “Rule by agent” representation for agent i in PFF can be defined by a set of rules R^i of the following form:

$$r : j \wedge k \rightarrow f(C_i, \pi) \cdot val_i$$

where j, $k \in N$, $val_i \in \mathbb{R}$, π is the formed partition, C_i is the coalition that contains i and $f : \mathcal{E}_N \rightarrow [0, 1]$.

7.2 Appendix B

Here we discuss the main idea of *Solution concepts* of an outcome and we provide a description of the *kernel* that we used in Section 5 of this thesis.

7.2.1 Solution concepts

Solution concepts in cooperative game theory, define a specific way to allocate the payoffs among each coalition. As described in Definition 1, the characteristic function, or utility function, maps each coalition C to a real number. However it does not specify the payoff allocation among the members of C. As such, we can evaluate an outcome according to two sets of criteria: (1)*fairness*, i.e., how well each agent's payoff reflects his contribution; and (2)*stability*, i.e., what are the incentives for the agents to stay in the coalition structure [2]. Some well-known solution concepts are: *Shapley value*, *core*, *kernel*, *nucleolus* etc.

7.2.2 Kernel

This is the solution concept that we used for this thesis in Section 5. The *kernel* [27] consists of all outcomes that no player can demand a fraction of another player's payoff. Formally, for any player i we define his *surplus* over the player j with respect to a payoff vector $\mathbf{x}$ as the quantity:

$$S_{i,j}(x) = max\{v(C) - x(C) \quad | \quad C \subseteq N, i \in C, j \notin C\}$$

This is the amount that player i can earn without the cooperation of player j, by asking a set $C\backslash\{i\}$ to join him in a deviation, and paying each player in $C\backslash\{i\}$ what it used to be paid under $\mathbf{x}$. Now, if $S_{i,j}(x) > S_{j,i}(x)$, player i should be able to demand a fraction of player j's payoff—unless player j already receives the smallest payments that satisfies the individual rationality condition, i.e, $v(\{j\})$ [2].