Coalition Formation among Rational Agents in Uncertain and Untrustworthy Environments

Dissertation zur Erlangung des Grades des Doktors der Ingenieurwissenschaften der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

Bastian Blankenburg

30. September 2013

Tag des Kolloquiums:
9. April 2014

Dekan der NTF I:
Prof. Dr. Markus Bläser

Vorsitzender des Prüfungsausschusses:
Prof. Dr. Jörg Siekmann

Berichterstatter:
PD Dr. Matthias Klusch
Prof. Dr. Markus Bläser
Prof. Dr. Sarit Kraus

Akademischer Mitarbeiter:
Dr. Klaus Fischer

Acknowledgements

Initial results on which this work is based on were obtained while working at Deutsches Forschungszentrum für künstliche Intelligenz GmbH (DFKI), Saarbrücken, on the SCALLOPS and CASCOM research projects, which were funded by Bundesministerium für Bildung und Forschung and the European Union IST programme, respectively.

First, I heartily thank Dr. Matthias Klusch for his supervision, guidance and support on professional but also personal levels, as well as fruitful and inspiring discussions.

Likewise, my gratitude goes to Prof. Jörg Siekmann for initially accepting me as doctoral student and letting me start this work as a member of his high quality multiagent research group at DFKI.

I cordially thank Prof. Sarit Kraus and Prof. Markus Bläser for their willingness to review this thesis.

Further, I thank Prof. Nicholas R. Jennings, Dr. Raj Dash, Dr. Sarvapali D. Ramchurn and Dr. Minghua He for the fruitful collaborative work at the Univ. of Southampton.

My thanks also is also extended to (former) members of the multiagent research group at DFKI, for the good working atmosphere and insightful discussions.

Finally, I especially thank my family and friends, in particular my parents, my brother Henning, Mercy Khamala, Michael Schneider, Verena Ayere and Marco Leiter, for being very supportive and patient, as well as for proof-reading.

Zusammenfassung

Auf *kooperativer Spieltheorie basierende Koalitionsbildung in Multiagensystemen* ermöglicht es *rationalen Softwareagenten*, zu kooperieren um ihren Nutzen zu erhöhen. Wenn dies in offenen Umgebungen stattfindet, ergeben sich einige Probleme. In dieser Arbeit geht es speziell um:

- *Unsicherheit*: in offenen Umgebungen haben Agenten oft nur unvollständige Information. Gezeigt wird, wie sich *mit unscharfen Koalitionswerten effizient Koalitionen bilden lassen*, und es werden Simulationsergebnisse präsentiert. Weiter wird eine Methode vorgestellt, mir der sich *stabile und garantiert risikobegrenzende Koalitionen bilden lassen.*

- *Betrügerische und unverlässliche Agenten*: in offenen Umgebungen ist zu erwarten, dass einige Agenten versuchen, ihren Profit auf Kosten anderer unberechtigterweise zu steigern. Andere könnten unzuverlässig sein. Es wird gezeigt, wie ein *Vertrauensmodell mit kryptographischen Techniken* verbunden werden kann, sodass ein *erfolgreicher Betrug erschwert* wird.

- *Wahrung der Privatsphäre*: viele Koalitionsbildungsverfahren verlangen von den Agenten, umfangreiche Informationen auszutauschen, z.B. individuelle Kosten und Bewertungen von möglichen Ergebnissen. Es kann jedoch ein Problem sein, wenn persönliche, finanzielle oder andere heikle Daten verlangt werden. Für solche Situationen wird der (nach unserem besten Wissen) *erste privatsphärensichernde Koalitionsbildungsalgorithmus* vorgestellt.

Summary

Multiagent coalition formation based on *cooperative game theory* is a means to let *rational software agents* cooperate to increase their benefits. When applied in *open environments*, a number of problems arise that are not well coped with by existing approaches. Some specific problems are addressed:

- *Uncertainty*: in open environments, agents often do not have complete knowledge. In this thesis, it is shown how to *form coalitions efficiently by modeling uncertainty as fuzzy numbers*. Simulation results are provided. Additionally, a method for coalition formation is proposed which is shown to *form stable coalitions with guaranteed risk bounds*.

- *Defrauding or unreliable agents*: it might be expected in open environments that some agents try to unsolicitedly increase their own profits at the cost of others by deception, or are generally unreliable. In this thesis, we *combine a trust measure* based approach *with cryptographic techniques* to obtain payment and communication protocols that are *shown to hamper successful deception*.

- *Privacy preservation*: most approaches require agents to reveal a considerable amount of information to each other, such as individual costs and valuations of certain outcomes. This might be unacceptable when personal, financial or other delicate data is concerned. To this end, the (to our best knowledge) *first privacy preserving coalition formation algorithm* is proposed.

Contents

Chapter 1

Introduction

Today's highly networked computational world provides its users with a multitude of available online services which they can employ to satisfy their needs. This includes services for all sorts of trades, such as retail, travel, transportation, shipping, finance or entertainment, to name but a few. Others are purely computing oriented, then also called cloud services, such as services for online storage, database access or computation itself. More and more of those services are also made available via machine-usable APIs, then also called *web services*[1]. Typically, each web service thereby offers some narrowly defined functionality. For example, a travel agency might provide a web service to query available flights, and another one to actually book a flight. A traditional application designed for end-users, such as a flight booking app for smartphones, might then combine a fixed set of web services to implement its functionality.

Because commercial web service providers aim to make some profit, the usage of many web services comes at some costs for their users. This cost might be applicable either for accessing the web service itself, which is typically the case for cloud services such as online storage. Or because of a secondary effect, such as the cost of a plane ticket that is booked via the web service. Common models include pay-per-user or subscriptions.

Alternatively, web service providers and users might employ *rational software agents* to negotiate service use. Agents thereby act on behalf of their owners, trying to fulfill their goals. For web service providers, this typically involve maximizing their profit. User agents,

[1]For example, the website http://www.programmableweb.com lists 9699 services in 68 categories as of August 2013

on the other hand, will try to fulfill their given requests at a low cost. However, the sets of web service providers and users need not be disjunct. A company offering some service might at the same time be requesting other services from other providers.

Furthermore, more complex requests might only be fulfilled by employing a set of offered services in combination. For example, a user might request her agent to book a ticket for a concert in another city, as well as a matching train ticket and hotel. However, her budget is bounded. Therefore, the three bookings have to be made in concert. Now, it might turn out that this combination is not possible within the given budget and the usual fixed prices. However, relevant service provider agents might still agree to offer a matching bundle at a reduced price which is still profitable for them, but within the budget. Actually, there might be a number of possible combinations of service prodiver agents which are able to do this, while there might be more similar requests from other user agents at the same time. So who should cooperate with whom, and under which terms?

Questions of this kind are studied in the field of *cooperative game theory*, which provides *solution concepts* that specify which agents should get into a *binding agreement*, i.e. form a *coalitions* bound by some specific *contract*. Typically the contract specifies which actions are to be performed by which agent in the coalition such that the coalitional profit is achieved. Further, a solution concept determines and how coalition members should distribute the profit among themselves such that it is profitable for each agent, and that no agent has a significant incentive to break away from the coalition. Therefore, a number of coalition formation algorithms and protocols for use in multiagent systems have been introduced in the literature (Chalkiadakis et al., 2011, Klusch and Shehory., 1996, Shehory and Kraus, 1998, Shoham and Leyton-Brown, 2009). Use of such algorithms is not limited to web service scenarios, but can be applied to any situation involving rational agents trying to negotiate profitable cooperation when contracts among the agents can be made. Quite a few problems in this undertaking have been tackled so far with some success, such as circumventing the high computational complexity inherent in cooperative game theory solution concepts. But when applied in *open environments* such as the internet, additional challenges arise that are not well coped with by existing approaches. In this thesis, we tackle three specific of such problems: *uncertainty*, *deceiving and unreliable agents* and *privacy preservation*. We provide coalition formation algorithms for each of these, and show relevant properties mostly theoretical, although a few simulation results are provided as well.

First, we consider uncertainty in coalition formation. In classic cooperative games, perfect information is assumed in order to compute a solution. But this is generally not realistic to assume in real-world scenarios. For example, the exact profit of a potential coalition might in general not be known in advance. To this end, a few approaches have been introduced in the literature to allow for coalition formation under uncertainty. Most of these rely on probabilistic methods, such as Bayesian coalition formation (Chalkiadakis and Boutilier, 2012). In this thesis, we propose an alternative approach which allows to use *fuzzy numbers* in the setting of *possibility theory* to model input values. Possibility theory has been shown to be more flexible and more in line with human expectations than probability theory in some settings Chowdhury et al. (2009), Hüllermeier (2003), Raufaste et al. (2003), especially when knowledge is subjective and probability distributions are not known. Therefore, we extend the coalition formation algorithm BSCA (Klusch, 1997), which is based on the solution concept of the *bilateral Shapley value*, to operate on fuzzy numbers. The proposed algorithm is shown to maintain the low computational and communication complexity of the BSCA. Additionally, simulation results are provided which show generally favorable quality of negotiated solutions, both with respect to overall achieved payoffs as well as stability.

Another kind of uncertainty in coalition formation arises when coalitions might fail completely, thus inducing only costs instead of profits. Thus, even though the potential profit of coalition might be high, and the expected payoff (in probabilistic terms) might be positive, the possible loss in case of failure might be unacceptable to agents. For this kind of situation, we show how to use a *coherent risk measure* to assess and quantify the financial risk of a coalition. One property of coherent risk measures is that the combined risk of two coalitions is less or equal than their added single risks. We exploit this fact to let agents take part in multiple coalitions at the same time, thus lowering their risks. We then extend the cooperative game solution concept of the *kernel* to respect the agents' individual risk bounds when computing a payoff distribution. Based on this, we outline an algorithm that guarantees to adhere to the risk bounds.

A further issue in open environments is that an agent might come across others which try to unjustifiably increase their own payoffs via deception. For the *kernel*, we show that it is prone to such manipulations. Furthermore, agents might fail to make their assigned payments to each other (side-payments) in order to implement a solution. While contracts are assumed to be *binding*, the question of who ultimately enforces the adherence to the contracts is typically not part of cooperative game theory or coalition formation research. Instead, this is assumed to be part of the environment in which agents engage in negotiations, such as the legal framework at the top most level. However, the cost of exercising one's right at that level can be high, and thus situations which necessitate such a step should be avoided if possible. Therefore, we introduce a novel *side payment protocol* which ensures that it is rational for each agent to make its side-payment at the specified time, since we prove that deviating from it would induce a loss for the agent. Finally, we show how agents can combine this with a *trust model* to measure each other's performance when executing the coalitional actions. This way, unreliable agents and agents which deceived with respect to their announced capabilities can be coped with.

Finally, we tackle the issue of privacy preservation in coalition formation in open environments. When agents representing different, unrelated, independent and unacquainted entities such as individual users and service-offering companies engage in online negotiation, it might be unacceptable for an agent that others learn which services are accessed and which profits are achieved. For instance, a company negotiating with potential component suppliers for some product might not want to let its competitors, which might also take part in negotiations, know about its requests. Unfortunately, existing coalition formation approaches generally assume that all agents share such information freely with each other in order to negotiate. To this end, we propose the first privacy preserving coalition formation algorithm (to our best knowdlegde).

Having outlined the basic concerns of this thesis, we provide the specific research questions in the next section.

1.1 Research Questions

In this thesis, we present methods for multiagent coalition formation which mitigate some of the strict requierements for classical game-theoretic approaches while trying to preserve their beneficial solution qualities. The particular research questions we answer are the following:

Fuzzy-valued coalition formation

1. *How can stable coalitions be formed efficiently when the coalition values are fuzzy?*
2. How can a resulting fuzzy solution be used to determine concrete, non-fuzzy side-payments in a stable manner?
3. What impact does the choice of possibilistic ranking operators have on the resulting payoffs?
4. Are the resulting fuzzy payoffs core-stable?

Risk-bounded coalition formation

1. How can resource-bounded agents reduce the risk of suffering losses due to failing coalitions according to an approved measures of risk?
2. How can stability of risk-bounded coalitions be ensured and what is the computational cost of such an approach?

Coalition formation with deceiving agents

How can stable coalitions be formed while preventing rational agents from deceiving during

1. coalition negotiations and
2. side payment executions?

Privacy-preserving coalition formation

1. How can profitable coalitions efficiently be formed while adhering to privacy constraints?
2. How can adherence to these privacy constraints be maintained also during the side payment and execution phases?

1.2 Thesis structure

The thesis is structured into three main parts: the first part consists of chapters 2 and 3. Chapter 2 starts with a clarification of what type of multiagent systems we are concerned with in this thesis. Then, we introduce the basic notions of cooperative game theory and provide background on how it has so far been applied to multiagent systems. We conclude the chapter by outlining the problems to which solutions are proposed in this thesis. In chapter 3, we discuss related work with respect to tackling these problems.

In the second part, chapters 4 - chapters 7, we introduce and discuss a number of coalition formation algorithms, the main contributions of this thesis.Chapters 4 and 5 tackle coalition formation under uncertainty using fuzzy values and risk bounds, respectively. Chapter 6 is concerned with the potential untrustworthiness of other agents, while chapter 7 introduces privacy-preserving coalition formation. As these four chapters employ different formalisms in addition to the theory of cooperative games and are concerned with quite separate problems, we introduce these formalisms within the preliminaries at the beginning of each chapter.

Finally, we conclude in chapter 8 by giving answers to the research questions and providing an outlook on worthwhile future work.

Chapter 2

Background

In this chapter we introduce the idea of *coalition formation* as a means to enable *autonomous rational agents* to cooperate to their own benefit in *open multiagent systems*.

For this purpose, we first clarify the notions of *autonomous rational agents* and *open multiagent systems* in detail in section 2.1. In section 2.2, we introduce *cooperative game theory* as an appropriate theory to model the cooperative settings that we are concerned with. We then give an overview of how this theory has already been applied to multiagent coalition formation in the literature in section 2.3. Some of the more basic challenges of that undertaking, such as high computational complexity of cooperative game theory solutions, and how they been tackled in existing approaches, are also outlined in that section.

Finally, we introduce the specific problems of coalition formation which we tackle in this thesis: in section 2.4, we introduce the problem of coalition formation under uncertainty. To our best knowledge, almost all of the existing work in this area employs probabilistic approaches, which we briefly introduce first. Next, we look at employing fuzzy set theory as means to model uncertainty in the context of multiagent coalition formation. Such approaches have received little attention, albeit fuzzy methods have been shown in recent literature to have certain practical advantages over probabilistic ones in certain settings.

As another challenge, we introduce the problems of deception, fraud, and privacy preservation in coalition formation in section 2.4.

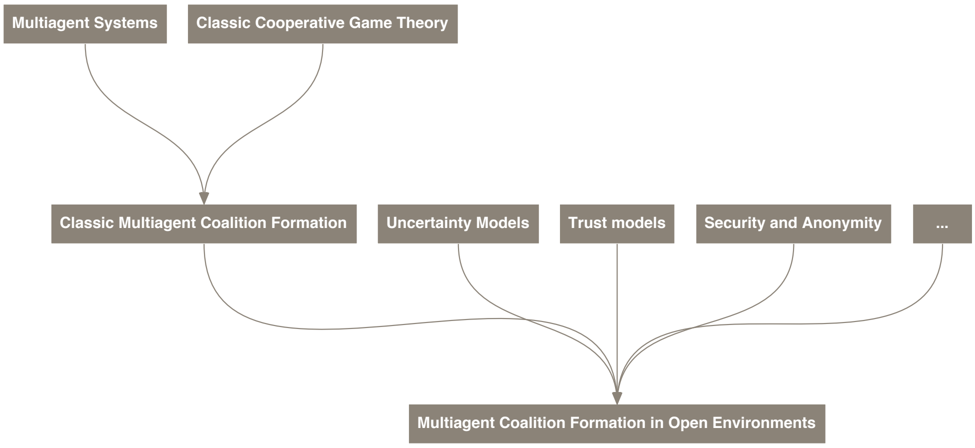

Figure 2.1: Overview of Multiagent Coalition Formation and contributing research fields.

2.1 Autonomous Rational Agents and Open Multiagent Systems

The research presented in this work is concerned with enabling *autonomous rational agents* to form of coalitions in *open multiagent systems* in order to jointly satisfy their goals when they cannot do so on their own, or only sub-optimally. But what exactly are those agents and multiagent systems? Unfortunately, general and definitive answers to these questions are still missing in the relevant literature, as Shoham and Leyton-Brown (2009) note in their recent book on multiagent systems:

"Somewhat strangely for a book that purports to be rigorous, we will not give a precise definition of a multiagent system. The reason is that many competing, mutually inconsistent answers have been offered in the past. Indeed, even the seemingly simpler question — What is a (single) agent? — has resisted a definitive answer. For our purposes, the following loose definition will suffice: Multiagent systems are those systems that include multiple autonomous entities with either diverging information or diverging interests, or both."

In addition to this, we also assume that there is no central authority which monitors, controls, schedules or otherwise regulates the agents. That is, the agents' are theoretically only bound by applicable juridical laws. This is what we mean by *open* multiagent system.

While the above broad definition in fact includes all sorts of system, including the "real world", in this work we consider only fully automated (i.e. software or robotic) multiagent systems. That is, artificial, non-human agents. Nevertheless, in the remainder, we often say just "system" instead of "open multiagent system", and just "agent" instead of "autonomous rational agent".

We assume the agents themselves to be autonomous and goal-oriented in the sense of the definition from Franklin and Graesser (1996):

"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

However, while it might be imaginable that such an autonomous agent is completely independent and truly has "its own agenda", we assume that usually an agent represents another entity, i.e. an "owner". This owner might e.g. be a human user, a company or another (software) agent, and sets the goal(s) for its agent(s) to achieve. An agent then autonomously acts on its owner's behalf, i.e. it is free to make its own decisions on how to take action in the system to achieve its goals.

We additionally assume that the owners of agents in the system are generally independent, and that they might not poss any prior knowledge about each other. We indeed assume that this will be a typical case in larger systems with many paricipants.

If agents might not know each other in advance, they will need a way to discover each other. We however let it open whether the system provides some centralized functionality for this (such as a directory service), or the agents implement some distributed peer-to-peer protocol to achieve this.

We further assume that the agents' owners typically also provide for the agents' resources. This includes e.g. computational power, memory, and financial resources. We assume these resources to be bounded, and also that agents generally will have more or less specific time constraints for achievement of their goals (i.e. from strict deadlines to soft requirements such as "find me a solution soon enough").

Also, agents might not be able to achieve complex goals on their own. They must therefore take advantage of some form of cooperation and coordination with other to be able to satisfy such goals. However, if we further assume that the agents' owners are selfish, they wouldn't want their agents to just use their finite resources for altruistic achievement of other agents' (or owners') goals. Agents are thus required to cooperate only if it is beneficial for themselves. Such benefit might e.g. stem from mutual help to achieve each cooperating agent's goals, or, if the help is one-sided, via direct monetary payment to the helping agent. Agents which try to maximize their own (or their owner's) benefit are called *rational agents*.

Please note that to enable the payments between agents, it might be helpful if the system has a built-in payment mechanism, but this might also be achieved via external means.

To summarize, the software agents considered in this thesis are

- goal-oriented (where overall goals are typically given by agents' owners),
- autonomous (makes its own decisions and acts independently to fulfill its owner's goals),
- generally ignorant of each other (agents might a priori not know other agents, their owners, their goals and/or their resources).
- communicative (agents can send messages to each other, including contracts and payments),
- resourceful and -bounded (each agent is running on some real, i.e. finite, computer system and has financial resources typically provided by its owner),
- rational (i.e. try to maximize their own, or their owner's, benefit).

And the considered multiagent systems are

- open (agents from independent owners might participate, and there is no central controlling entity),
- providing communication infrastructure and
- providing some discovery mechanism for the agents.

To give a more concrete idea of what kind of environments we look at, figure 2.2 shows an example system including agents which are owned by and acting on the behalf of companies and individual users. They specify certain goals for their agents, and the agents would then try to find other agents who are able to satisfy those goals.

Now, assume that the satisfaction of a given goal requires the execution of some specific operations, also called *tasks*, of some other agents. Then, depending on how the goal and available operations are specified, different techniques that have been developed in artificial intelligence research might be used to determine which tasks to execute.

For instance, if available operations are made available on the Internet as *web services*, *semantic web service descriptions* can be used to express service offers (available operations) and requests (modeling the goals). Examples of suitable description languages include *OWL-S* or *WSMO* and *annotated WSDL* (see OWL-Services-Coalition, 2004, Patil et al., 2004, Roman et al., 2005, respectively). This approach has the advantage that there already exist technologies for discovery, matchmaking, composition planning and automated execution for semantic web services. For instance, Klusch (2008) provides an overview of solutions for each of these tasks.

More recently, Bartalos and Bieliková (2011) compared different composition planning approaches for semantic web services. But they also note that despite a number of problems thereof have already been solved, there has not been a wide adoption of semantic web services in practice. But there exist also alternative approaches which circumvent the requirement of semantic descriptions by employing other artificial intelligence techniques such as genetic programming (e.g. Xiao et al., 2012).

However, in this thesis we are not particularly concerned about the concrete technologies which the agents employ to determine the actions that will satisfy some goal. We only assume that such technologies exist and can be used by the agents. On the other hand, the service model provides explicit notions of requests, offers, service composition etc. This is sometimes helpful, and we specifically exploit this in chapters 5 and 7.

Nevertheless, it should be kept in mind that we mostly use the terms “task” and “(web) service” in an abstract way and mostly interchangeably. So even if some of the proposed algorithms are for-

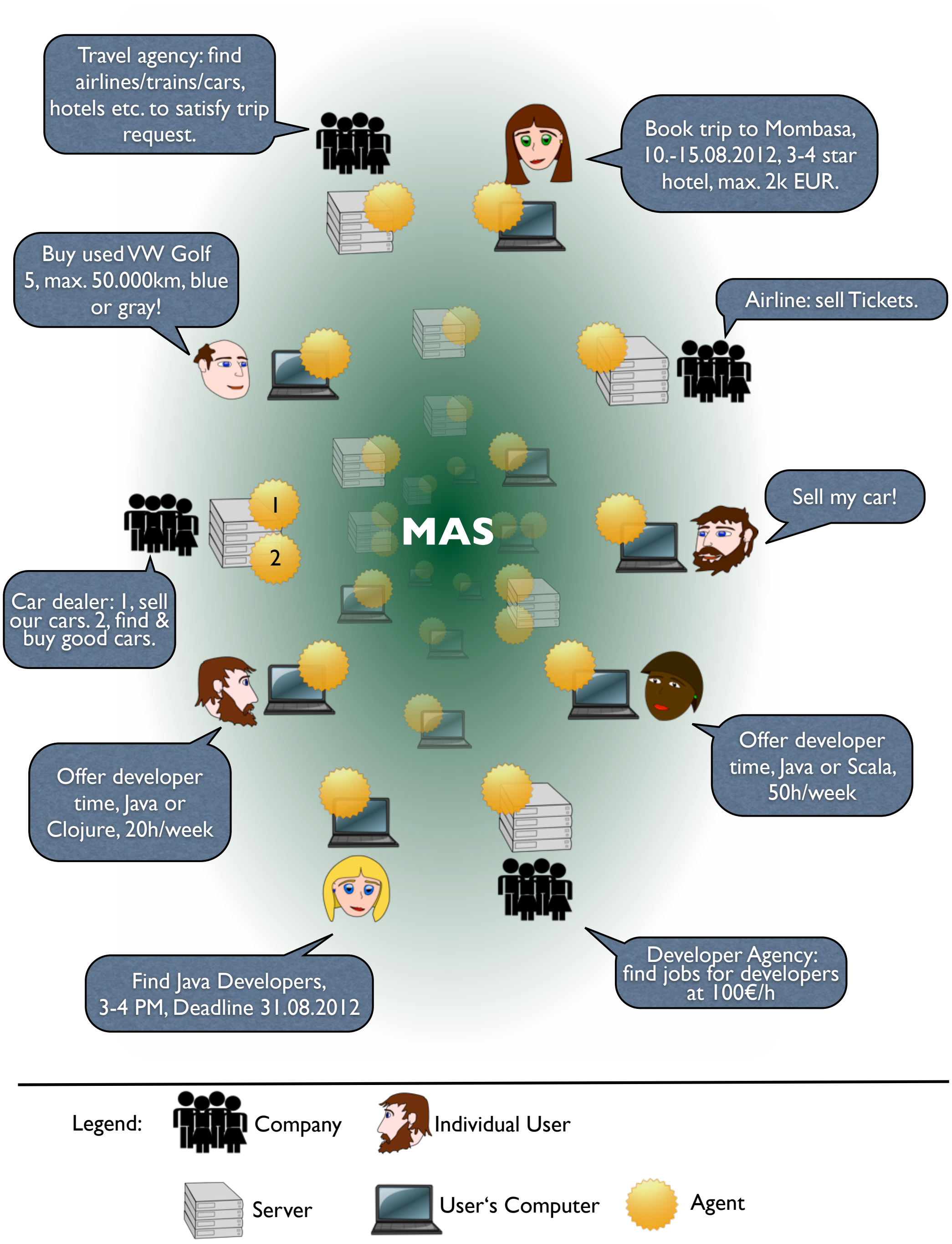

Figure 2.2: Example multiagent system with different types of agents owned by companies and human end users. Goals for the agents are stated in the balloons.

mulated in terms of services, they should still be generally adaptable to scenarios where task executions lead to goal satisfactions.

We now provide the theoretical background of cooperative game theory and how it can be employed as a basis for multiagent coalition formation in general in the following sections.

2.2 Cooperative Game Theory

Strategic decision situations involving independent and rational participants are the subject of study in the microeconomic field of *game theory*, first introduced in von Neumann and Morgenstern (1944). Such situations are called *games*, and the participating agents are also called *players* in the game theory context. In the following, we use the terms "player" and "agent" interchangeably. Game theory has numerous theoretical and practical applications in the fields of economics and politics (see e.g. Aumann, 2010, Aumannn, 1997, 1999, Aumannn and Hart, 1992, Rapoport, 2001).

There are two main strands of games theory, *non-cooperative* and *cooperative*. In both variants, agents are assumed to be rational and selfish. But cooperative game theory allows them to engage in binding *contracts* by negotiation, while non-cooperative game theory considers settings where this, or in fact any agreement among the agents, is not possible or not allowed (agents that nevertheless manage to cooperate in non-cooperative settings are also said to *collude*). Therefore, if cooperation is desired in a multiagent system, cooperative game theory provides an applicable theoretical basis to study the cooperation opportunities. For general introductions to game theory we also point to, for instance, Osborne (2003), Peleg and Sudhölter (2007) or Owen (1995).

In cooperative game theory, agents which reach an agreement and thus establish a contract among each other are said to from a *coalition*. Being rational and selfish, they will only do so if they can somehow profit from being in the coalition. Assume that an agent has at its disposal a *utility function* u which maps the set of possible outcomes X of the game to real values:

$$u : X \mapsto \mathbb{R}$$

Thus, a utility function establishes a complete ordering of an agent's preferences over the outcomes: being rational, an agent will (strictly) prefer an outcome x to another one y if $u(x) > u(y)$(von Neumann and Morgenstern, 1944). For example, if x implies that the agent is a member of certain coalition and y implies that it stays alone, then it is preferable for the agent to join this coalition rather than not joining any coalition. It is then said to be *individually rational* for the agent to join the coalition. However, the agent might still have the possibility to join another coalition, represented by the outcome z, such that $u(z) > u(x) > u(y)$. Therefore, theoretic game theory encompasses *solution concepts*, also called *stability concepts*, which determine outcomes such that no agent is motivated to break from its coalition in the respective outcome.

For this purpose, cooperative game theory distinguishes between to types of games:

Games without transferable utility , where the utility $u(x)$ of an outcome x is determined exclusively by the agent's coalition membership.

Games with transferable utility , where $u(x)$, in addition to coalition membership, also captures some utility transfers among the members of a coalition.

Utility transfers are in particular possible in monetarian settings where coalitions obtain a certain certain amount of money which is then to be distributed among its members. This is the type of game which we are concerned with in this thesis, and we proceed by providing the relevant formal concepts.

Cooperative Games

Definition 2.2.1 Coalition
Given a set of agents $\mathcal{A}$, a *coalition* C is any subset of $\mathcal{A}$:

$$C \subseteq \mathcal{A}$$

In the following, when denoting coalitions of specific agents $a_i, a_k, \ldots$, $i, k, \ldots \in \mathbb{N}$, we also write $\{i, k, \ldots\}$, instead of $\{a_i, a_k, \ldots\}$. Further, the coalition of all agents $C = \mathcal{A}$ is called the *grand coalition*, while subsets $C* \subseteq C$ are called *subcoalitions* of C. △

Definition 2.2.2 Cooperative game
A *cooperative game* or *coalition game* in characteristic function form

is a pair $(\mathcal{A}, v)$ with the set of agents $\mathcal{A}$ and the *characteristic function* v mapping coalitions to real numbers:

$$v : 2^{\mathcal{A}} \mapsto \mathbb{R}$$

$v(C)$ is called the *(coalition) value* of the coalition $C \subseteq A$. $v(\{a\})$, $a \in \mathcal{A}$, is also called a's *self value*. It is assumed to be the *maximum payoff* that C can achieve *independently* of which other coalitions are formed. The value of the empty coalition is defined as zero: $v(\emptyset) := 0$. △

Thus, the value of a coalition C can be viewed as a measure of the payoff achievable by C by cooperating behaviour of its members. In the following, we also say just "game" instead of cooperative or coalition game. Certain classes of games are induced by specific interesting properties of the characteristic function, such as its *additivity*:

Definition 2.2.3 Additivity
The *additiviy* of a coalition game is determined by the relation of the sum $v(C_1) + v(C_2)$ of the values of two disjunct coalitions $C_1, C_2 \subset \mathcal{A}$, $C_1 \cap C_2 = \emptyset$ to the value of the union coalition $v(C_i \cup C_k)$. The game $(\mathcal{A}, v)$ is called

1. *superadditive*, iff[1]

$$\forall C_1, C_2 \subseteq \mathcal{A}, C_1 \cap C_2 = \emptyset : \; v(C_1 \cup C_2) \geq v(C_1) + v(C_2)$$

2. *locally superadditive for* $C_1, C_2 \subseteq \mathcal{A}, C_1 \cap C_2 = \emptyset$, iff

$$v(C_1 \cup C_2) \geq v(C_1) + v(C_2)$$

3. *subadditive*, iff

$$\forall C_1, C_2 \subseteq \mathcal{A}, C_1 \cap C_2 = \emptyset : \; v(C_1 \cup C_2) \leq v(C_1) + v(C_2)$$

4. *locally subadditive for* $C_1, C_2 \subseteq \mathcal{A}, C_1 \cap C_2 = \emptyset$, iff

$$v(C_1 \cup C_2) \leq v(C_1) + v(C_2)$$

△

[1] we write "iff" for "if and only if" throughout this thesis.

Remark 2.2.4. *Because super- and subadditivity are defined in terms of non-strict inequality, a game may be both super- and subadditive at the same time iff*

$$\forall C_1, C_2 \subset \mathcal{A}, C_1 \cap C_2 = \emptyset : \; v(C_1 \cup C_2) = v(C_1) + v(C_2)$$

The additivity of a game plays an important role in determining which coalitions are profitable and which solution concepts are applicable. Some classic solution concepts assume that the grand coalition is always formed, which is only the best option if the game is superadditive. This includes the core and the Shapley value (these are defined in an upcoming section).

There has been some dispute on the question on whether non-superadditive games are a viable model for realistic situations. For example, Wooders (2008) argues that (where "worth" amounts to the coalition value and "group of players" to coalition): "the worth of a group of players is independent of the total player set in which it is embedded and an option open to a group is to achieve the total worths realizable by a partition of the total player set into smaller groups". That is, if cooperation of agents in coalition C_1 with agents in coalition C_2 is somehow detrimental to the joint utility of agents in $C = C_1 \cup C_2$, they simply might abstain from such cooperation even if C is formed formally. However, it has also been argued that the environment might impose restrictions on contracts, such as penalizing coalitions of certain sizes. Examples include anti-trust or anti-monopoly laws (Chalkiadakis et al., 2011, p. 15) or the increased cost of computation and communication involved with coordinating a bigger coalition (Shehory and Kraus, 1999).

Another interesting property of cooperative games is *convexity*, which is yet stronger than superadditivity:

Definition 2.2.5 Convexity of a game
A game $(\mathcal{A}, v)$ is called *convex* if it satisfies

$$\forall C_1, C_2 \subseteq \mathcal{A} : \; v(C_1 \cup C_2) + v(C_1 \cap C_2) \geq v(C_1) + v(C_2)$$

△

Remark 2.2.6. *A convex game is always also superadditive because the latter only considers the unions of coalitions C_1 and C_2 for which $C_1 \cap C_2 = \emptyset$. Because $v(\emptyset) = 0$, for such coalitions the constraint for convexity coinsides with that for superadditivity.*

Definition 2.2.7 Symmetry
In a game $(\mathcal{A}, v)$, two agents $a_1, a_2 \in \mathcal{A}$ are called *symmetric* iff

$$\forall C \subset \mathcal{A}, a_1, a_2 \notin C \ : \ v(C \cup \{1\}) = v(C \cup \{2\})$$

The game $(\mathcal{A}, v)$ itself is called symmetric if all agents $a \in \mathcal{A}$ are pairwise symmetric. △

Finally, we also define the *addition of games*.

Definition 2.2.8 Addition
A game $(\mathcal{A}, v)$ is called an addition of two games $(\mathcal{A}, v_1)$ and $(\mathcal{A}, v_2)$ iff

$$\forall C \subseteq \mathcal{A} : v(C) = v_1(C) + v_2(C)$$

v is then also denoted as $v = v_1 + v_2$. △

Configurations and Solutions

Having introduced cooperative games and their properties, we now consider their outcomes and which of the possible outcomes are *solutions* of the game. In a transferable utility game, an outcome is specified by a *configuration* which assignes agents to coalitions and payoffs to agents.

Definition 2.2.9 Configuration
A configuration $(\mathcal{C}, u)$ for a game (A, v) specifies a *payoff distribution*

$$u : A \mapsto \mathbb{R}$$

for a *coalition structure* $\mathcal{C}$, a partition of A:

$$\mathcal{C} = \{C_1, \dots, C_n\} \subseteq 2^{\mathcal{A}}$$

$u(a)$ denotes the *payoff* for agent a. For the joint payoff of the agents in a coalition $C \subseteq \mathcal{A}$, we also write

$$u(C) = \sum_{a \in C} u(a)$$

Further, for an agent set $\mathcal{A} = \{a_1, \dots, a_n\}, n \in \mathbb{N}$, u is also used in vector notation with $u \in \mathbb{R}^n$ and $u_i = u(a_i), 1 \leq i \leq n$. △

To determine which configurations might be considered solutions, we introduce the following properties.

Definition 2.2.10 Configuration properties
For a configuration $\mathcal{C}$, the payoff distribution u is called

- *feasible* iff $\forall\ C \in \mathcal{C}\ :\ u(C) \leq v(C)$
- *efficient* iff $\forall\ C \in \mathcal{C}\ :\ u(C) = v(C)$,
- *individually rational* iff $\forall a \in \mathcal{A}\ :\ u(a) \geq v(a)$,
- an *imputation* iff it is both efficient and individually rational.
- *(locally) individually rational for* $a \in \mathcal{A}$ iff $u(a) \geq v(a)$
- *pareto optimal* iff there exists no other configuration $(\mathcal{C}^*, u^*)$ such that u^* is feasible and

$$\forall a \in \mathcal{A} : u^*(a) \geq u(a) \text{ and } \nexists a \in \mathcal{A} :\ u^*(a) > u(a)$$

△

Intuitively, efficiency of a configuration means that the value of every coalition is distributed completely among its members and so nothing is lost or gained to or from "outside". Individual rationality implies that each agent is better off staying in its coalition than breaking off to form its single-agent coalition. And pareto optimality implies that no agent can obtain a higher payoff in a different configuration without decreasing that of some agent. Therefore, to obtain a solution to the game, one generally has to additionally employ some *stability concept*, of which the core might be the most intuitive one:

Definition 2.2.11 Core
The *core* of a game $(\mathcal{A}, v)$ is defined as the set of configurations whose payoff distribution u satisfies

$$\forall C \subseteq \mathcal{A}\ :\ u(C) \geq v(C)$$

A configuration in the core is also said to be *core-stable*. △

Thus, in the core, alternative coalition $C \notin \mathcal{C}$ can form (thereby breaking up existing coalitions in $\mathcal{C}$) to obtain a higher payoff. The payoff distributions of configurations in the core have been shown to be pareto optimal. However, for certain games, multiple configurations might be core-stable, or sometimes none at all(see e.g. Owen, 1995). This is demonstrated by the following two examples.

Example 2.2.12
Consider the symmetric, superadditive and convex game $(\{1, 2\}, v)$ with $v(\{1\}) = v(\{2\}) = 1$ and $v(\{1, 2\}) = 3$. Then every configuration with coalition structure $\{\{1, 2\}\}$ and efficient payoff distribution u,

with $u(a_1) \geq 1$ and $u(a_2) \geq 1$, is core-stable. This example further demonstrates that the core does not necessarily assign equal payoffs to symmetric agents. For instance, $(\{\{1,2\}\}, u)$ with $u(a_1) = 2)$ and $u(a_2) = 1$ is core-stable, although a_1 and a_2 are symmetric. △

Example 2.2.13

Consider this symmetric and superadditive but non-convex game:

coalition(s) C	v(C)
$\{a_1\}$, $\{a_2\}$, $\{a_3\}$	0
$\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$	10
$\{a_1, a_2, a_3\}$	12

This game has an empty core: if a two-agent coalition C is formed, then at least one of its members a must receive a payoff $u(a) < 10$. But then a can threaten to form the two-agent coalition C^* with the third agent a^* instead, because $v(C) = 10 > u(a)+0 = u(a)+u(a^*)$. If the grand coalition if formed, any pair of agents a, a^* receiving together less than 10 can threat to form the coalition $\{a, a^*\}$. However, since $v(\{a_1, a_2, a_3\}) = 12 < 15$, this cannot be prevented for all three pairs of agents at the same time. △

These examples show that the core might not be the best approach to obtain stable solutions in a multiagent setting. A popular alternative is the kernel (Davis and Maschler, 1965, Kahan and Rapoport, 1984):

Definition 2.2.14 Kernel

Let $(\mathcal{C}, u)$ be a configuration for the cooperative game $(\mathcal{A}, v)$.

1. The *excess* $e(C^*, u)$ of a coalition $C^* \notin \mathcal{C}$ is defined as

$$e(C^*, u) := v(C^*) - \sum_{a \in C^*} u(a)$$

2. The *surplus* sur_{ik} of agent a_i with respect to agent a_k with $a_i, a_k \in C \in \mathcal{C}, a_i \neq a_k$, is defined as

$$sur_{ik} := max\{e(C^*, u) \mid C^* \notin \mathcal{C}, a_i \in C^*, a_k \notin C^*\}$$

3. $(\mathcal{C}, u)$ is in an *equilibrium of surpluses*, if $\forall a_i, a_k \in C \in \mathcal{C}$:

$$\begin{aligned}
&(sur_{ik} = sur_{ki})\\
&\vee(sur_{ik} > s_{ki} \wedge u(a_k) = v(\{a_k\}))\\
&\vee(sur_{ki} > s_{ik} \wedge u(a_i) = v(\{a_i\}))
\end{aligned}$$

Then the *kernel* of the game $(\mathcal{A}, v)$ is the set of all efficient and individually rational configurations which are in an equilibrium of surpluses. △

Example 2.2.15
Consider again the game from example 2.2.13. Since the agents are symmetrical, we should expect that all agents in the same coalition should get the same payoff:

coalition structure	$u(a_1)$	$u(a_2)$	$u(a_3)$
$\{\{a_1\}, \{a_2\}, \{a_3\}\}$	0	0	0
$\{\{a_1, a_2\}, \{a_3\}\}$	5	5	0
$\{\{a_1, a_3\}, \{a_2\}\}$	5	0	5
$\{\{a_1\}, \{a_2, a_3\}\}$	0	5	5
$\{\{a_1, a_2, a_3\}\}$	4	4	4

Because the Kernel treats symmetric agents equally, these are indeed the Kernel-stable configurations of this game. △

Another popular solution concept is the Shapley value (Shapley, 1953). As opposed to set based stability concepts, the Shapley value in its original form does not define a set of stable configurations, but directly assigns stable payoffs to the agents. It is implicitly assumed that the grand coalition is formed and thus the Shapley value might reasonably applied only to superadditive games.

Definition 2.2.16 Shapley value
The *Shapley value* $\sigma(a, v)$ for an agent $a \in \mathcal{A}$ in the game $(\mathcal{A}, v)$ is defined as

$$\sigma(a, v) = \sum_{C \subseteq \mathcal{A}} \frac{(|\mathcal{A}| - |C|)!(|C| - 1)!}{|\mathcal{A}|!}(v(C) - v(C \setminus \{a\}))$$

$v(C) - c(C \setminus \{a\})$ is called *marginal contribution* of a to $v(C)$. △

Since the number of possible coalitions for a set of agents $\mathcal{A}$is $2^{|\mathcal{A}|} - 1$, the computation of the Shapley value for one agent requires exponential time wrt. the number of agents. However, when it is assumed that coalitions are only bilaterally merged, one might instead use the bilateral Shapley value. It considers only a sub game of two coalitions which are to be merged but allows to design coalition formation algorithms with low polynomial runtime (Ketchpel, 1995, Klusch, 1997).

Definition 2.2.17 Bilatral coalition
A union $C_1 \cup C_2$ of two disjunkt coalitions $C_1, C_2 \subset \mathcal{A} \setminus \emptyset$ is called a *bilateral coalition.* C_1 and C_2 are called *subcoalitions* of $C_1 \cup C_2$. △

Definition 2.2.18 Bilateral Shapley value
The *bilateral Shapley value* $\sigma_b(C_1 \cup C_2,\, C_i,\, v)$,$C_i, i \in \{1, 2\}$ in the bilateral coalition $C_1 \cup C_2$ is defined as the Shapley value of C_i in the game $(\{C_1, C_2\}, v)$:

$$\sigma_b(C_i, v) = \frac{1}{2} \cdot v(C_i) + \frac{1}{2} \cdot (v(C_1 \cup C_2) - v(C_k)),\ k \in \{1, 2\}, k \neq i$$

△

If coalitions are formed by repeatly merging two coalitions together, one might keep track of the binary tree structure of subcoalition that is thereby built up. Therefore, we also define recursively bilateral variants of coalitions, coalition structures and the bilateral Shapley value in the following definitions.

Definition 2.2.19 Recursively bilateral coalition
A bilateral coalition C is called a *recursively bilateral coalition* iff it is the root node of a binary tree denoted T_C for which holds that

1. every non-leaf node is a bilateral coalition whose subcoalitions are its only children, and
2. every leaf node is a single-agent coalition.

△

Definition 2.2.20 Recursively bilateral coalition structure
A coalition structure $\mathcal{C}$ for $(\mathcal{A}, v)$ is called a *recursively bilateral coalition structure* if

$$\forall C \in \mathcal{C} :\ C \text{ is (recursively) bilateral or } C = \{a\},\ a \in \mathcal{A}$$

△

Definition 2.2.21 Recursively bilateral Shapley value
Given a recursively bilateral coalition C for a game $(\mathcal{A}, v)$ a payoff distribution u is called *recursively bilateral Shapley value stable* iff it is efficient and the payoff for every non-root node C_i^* in T_C is the bilateral Shapley value applied to its parent C^*, the subcoalition C_i^* itself and a modified coalition value function v_{C^*}, where $v_{C^*}(C^*)$ is a modified v which maps the parent C^* to its recursively bilateral Shapley value payoff instead of its coalition value.

Formally, u is therefore recursively bilateral Shapley value stable iff it holds that

$$u(C_i^*) = \sigma_b(C^*,\, C_i^*,\, v_{C^*}),\ i \in 1,2, \text{ with}$$
$$\forall C^{**} \subseteq \mathcal{A}:\ v_{C^*}(C^{**}) = \begin{cases} \sigma_b(C^p,\, C_k^p,\, v_{C^p}) & \text{if } C^{**} = C^*, C^p \in T_C, \\ & C^* = C_k^p, k \in 1,2 \\ v(C^{**}) & \text{otherwise} \end{cases}$$

△

In other words, when merging two recursively bilateral coalitions into one its value will be distributed down the corresponding coalition tree to its members by means of recursively replacing the coalition value of the respective parent coalition with its payoff, that is the bilateral Shapley value (for an illustration, we refer to examples 7.1.3 and 7.1.4 in chapter 7).

Local worth and side payments

As already laid out in section 2.1, we generally assume that agents in a coalition

1. request certain goals to be satisfied (possibly modeled as service requests), where the fulfillment of a goal is of a certain value for the requesting agent; and

2. perform assigned tasks (possibly modeled as offered web services), where a task execution has a certain cost for the executing agent.

We therefore introduce the following notations:

Definition 2.2.22 Goal valuations and task execution costs
We call

- G_a the *set of goals* of agent a and $w_a : G_a \mapsto \mathbb{R}^+$ a's *(goal satisfaction) valuation function*, and

- $\mathcal{T}_a$ the *set of tasks* which a is capable of performing and $c_a : \mathcal{T}_a \mapsto \mathbb{R}^+$ a's *(task execution) cost function.*

For simplicity, we also call $w_a(g)$, $g \in G_a$, a's *valuation of (the satisfaction of) g* and $c_a(\tau)$, $\tau \in \mathcal{T}_a$, a's *cost of (the execution of)* τ. △

Definition 2.2.23 Task allocation
Given a coalition C, we define the *task allocation*

$$\alpha_C \subseteq \bigcup_{a \in C} \mathcal{T}_a$$

as the set of tasks that is to be executed by C's members (and so is part of C's coalitional contract). For agent $a \in C$, we also write

$$\mathcal{T}_a(C) := \mathcal{T}_a \cap \alpha_C$$

△

Definition 2.2.24 Satisfied goals
$G_a(C) \subseteq G_a$ denotes the set of goals of agent $a \in \mathcal{A}$ which are satisfied in coalition C via the execution of tasks in α_C. △

Corollary 2.2.25. *Remember from definition 2.2.2 that the coalition values are defined as the* maximum payoff *that a coalition "can independently achieve". Therefore, if agents have perfect information and unbounded resources, the task allocation within a coalition C must be optimal with respect to valuations $w_a(g)$ and costs $c_a(C)$ for all $a \in C$.*

Remark 2.2.26. *The constraints of perfect information and unbounded (or nearly so) resources in corollary 2.2.25 often do not hold in practical applications. For example, an optimal task allocation in a large coalition might exist in the form of a complicated interplay among all its members. But this might not be found in feasible time by any of its members. Therefore, the task allocation that a coalition comes up with and is included in its contract might in practice often be better understood loosely as "the best one which its members could find in time".*

From the above definitions it follows that every agent a in a coalition C has a local income and local cost induced by only its own task executions and goal satisfactions, but without regard for side payments. Klusch (1997) defined this as the *local worth* of an agent:

Definition 2.2.27 Local worth
The *local worth* $lw_a(C)$ of a in C is defined as

$$lw_a(C) := \sum_{g \in G_a(C)} w_a(g) - \sum_{\tau \in \mathcal{T}_a(C)} c_a(\tau)$$

For an agent a_i, we also write $lw_i(C)$ instead of $lw_{a_i}(C)$. △

Corollary 2.2.28. *From the definition of the local worth, and assuming that there exist no externalities (i.e. coalitions do not get or make payments from/to entities outside the coalition), it follows directly that the coalition value of a coalition C is the sum of its local worths:*

$$v(C) = \sum_{a \in C} lw_a(C)$$

Note that to actually *implement* the payoffs assigned by a given solution, the agents have to make side payments to each other. From the definition of the local worth, it further follows:

Corollary 2.2.29. *Let $(\mathcal{C}, u)$ be a solution for a game $(\mathcal{A}, v)$. Then the* net side payment *that an agents has to receive to obtain its assigned payoff $u(a)$, denoted $sp_u(a, C)$, is the difference of its assigned payoff and its local worth of its assigned coalition:*

$$sp_u(a, C) = u(a) - lw_a(C)$$

Note that if $sp_u(a, C)$ is positive, then a is to receive the amount from other agents in C. If it is negative, then a has to pay the amount to other agents in C.

For a subcoalition $C^* \subseteq C$ we also write $sp_u(C^*, C)$ to denote the sum of the net payoffs of the agents in the subcoalition:

$$sp_u(C^*, C) := \sum_{a \in C^*} sp_u(a, C)$$

It follows that if and only if u is efficient, all side payments $sp_u(C, C)$ within a coalition C, also denoted $sp_u(C)$, sum to zero, because

$$\begin{aligned} sp_u(C) &= \sum_{a \in C^*} (u(a) - lw_a(C)) \\ &= \sum_{a \in C^*} u(a) - \sum_{a \in C^*} lw_a(C) \\ &= \sum_{a \in C^*} u(a) - v(C) \\ &(= 0 \text{ iff } u \text{ is efficient}) \end{aligned}$$

2.3 Coalition Formation in Multiagent Systems

In this section, we outline how rational software agents might generally utilize game theoretic coalition formation protocols to determine who should cooperate with whom. That is, how cooperative game theory can and has been used as a basis for devising algorithms and protocols for rational agents in multiagent systems. The basis in game theory therein generally ensures desired properties of the solution such as individual rationality, pareto-optimality and stability according to the chosen stability concept. The basic idea of multiagent coalition formation is to enable agents to arrive at stable configurations efficiently in terms of runtime, space and communication complexity. [2]

From Game Theory to Coalition Formation

We note that cooperative game theory is a descriptive model, i.e. it gives tools to analyze strategic negotiation situations from a global perspective. The definitions of the classic solution concepts are therefore given in global and centralized terms because the game is provided in characteristic form with the set of agents and all the coalition values already known. In an open multiagent system, however, no agent can be assumed to possess all the required information initially.

Thus, starting with the first algorithms for coalition formation in multiagent systems that have been proposed in the literature, protocols for necessary information exchange were incorporated within the algorithms. Examples include Contreras et al. (1998), Klusch and Shehory (1996), Shehory and Kraus (1998) and Shehory and Kraus (1999).

Kraus (1997) points out three issues that should be handled by any coalition formation scheme:

1. devising an interaction protocol,
2. developing coalition formation algorithm and
3. to address the computational constraints (e.g. communication and computational costs).

[2]We note that there are other works in the literature on coalition formation which do not relate to cooperative game theory concepts such as individual rationality or stability (e.g. Arib and Aknine, 2012, Barton and Allan, 2008, Xiao-fei et al., 2012). While those might be applicable in their respective domains, we do not further consider such approaches in this thesis.

Naturally for multiagent algorithms, another focus is to distribute the computation of a solution. Apart from the obvious advantage of parallelizing the computation, this allows each agent or coalition to compute the parts of the solution by which it is directly affected. It thus potentially reduces the need to trust others on the correctness of computations.

Note that Shehory and Kraus (1998) address a multiagent task allocation setting and focus on minimizing the cost of coalitions to execute their tasks. It does not account for agents' self-interest, and indeed the utilities of single agents are not included in their model. That is, only a coalition structure but no configuration is computed. The other three works mentioned above however account for self-interested agents and yield stable configurations. These are arrived at via successive negotiation rounds, in each of which some coalitions might form. Although the proposed algorithms differ considerably in their details, an outline can be given as follows (see figure 2.3 for an illustration):

1. Each agent communicates initial data to other agents and sets up the initial configuration (usually consisting only of single-agent coalitions).

2. Possibly communicate necessary data for the next coalition formation round.

3. Each coalition in the present configuration computes new stable (possibly partial) configurations which result from merging coalitions[3]. It retains only the configurations which are profitable to itself.

4. Coalitions negotiate (according to a protocol specified in the algorithm) with the goal to obtain the configuration which is

   (a) stable according to the employed stability concept and

   (b) most profitable for themselves.

   If any new coalition was formed, the algorithms will then proceed with the coalition formation next round. Otherwise, if no new coalition was formed, the coalition formation terminates and the coalitions start executing their actions.

[3]Shehory and Kraus (1999) also allows to add a single members from another coalition, thereby breaking the other coalition up.

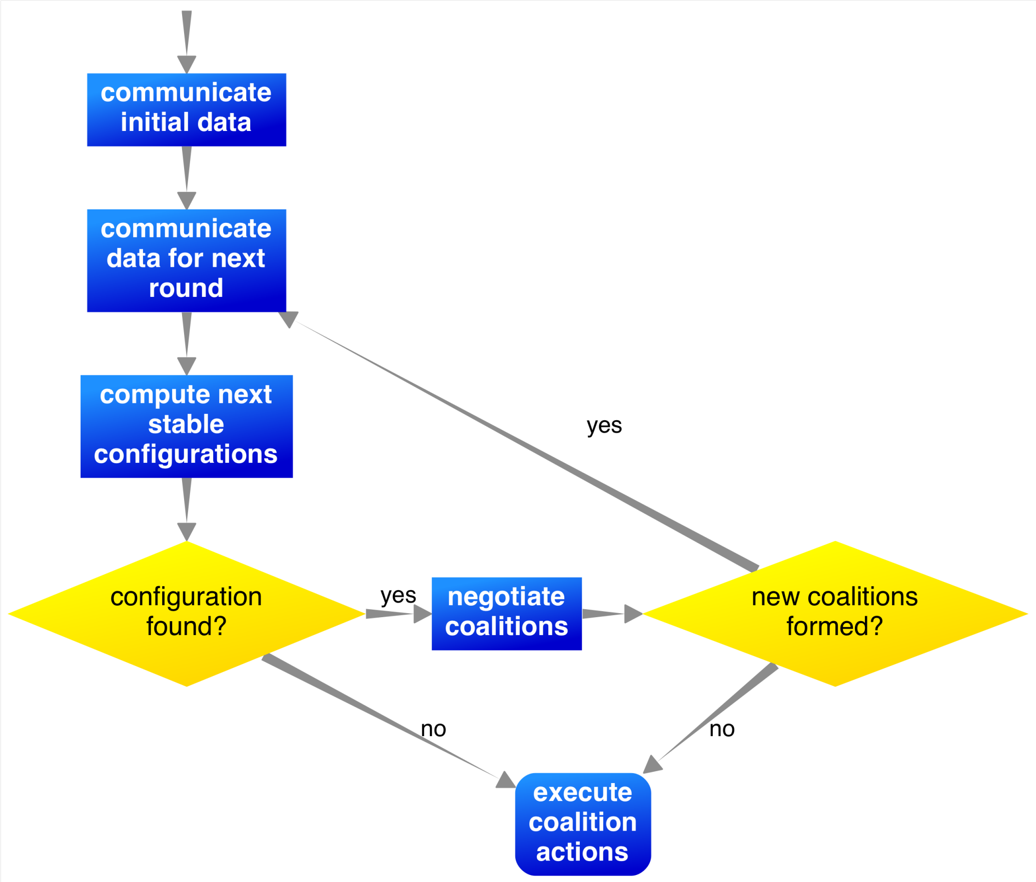

Figure 2.3: Approximate design of round-based coalition formation algorithms.

From this outline, it is clear that algorithms of this structure

- are inherently distributed among the agents;
- also specify protocols for the communication steps;
- are only suitable for stability concepts which allow for coalitions structures other than the one containing only the grand coalition can be used (including e.g. the Kernel or modified versions of the Shapley value);
- are also anytime algorithms since a stable configuration is obtained after each negotiation step.

In chapters 4 and 7, we introduce two coalition formation algorithms following this negotiation round-based scheme. The former enables coalition formation under uncertainty, and the latter allows for privacy-preserving coalition formation.

In contrast to this approach to building up stable configurations by negotiation rounds, Sandholm (1999) proposed to employ a more direct approach to obtain a stable configuration. It consists of just two steps (see figure 2.4):

1. compute *social welfare maximizing* coalition structure, which maximizes the sum of the values of the contained coalitions, and then

2. compute the payoffs to obtain a stable configuration according to the chosen stability concept.

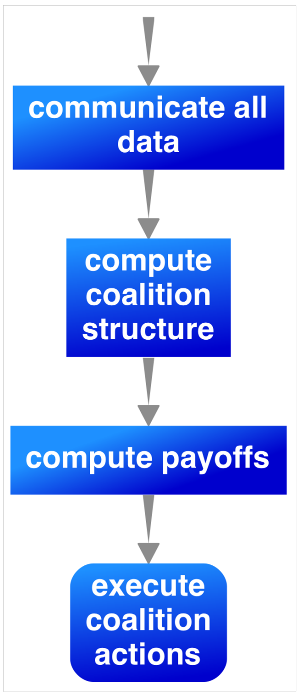

Figure 2.4: General design of “two-step” coalition formation algorithms.

But a social welfare-optimizing coalition structure might not always be the best one for every individual agent, even if the game is superadditive. In particular, this happens if the game is non-convex. We show this in the following example.

Example 2.3.1
Consider again the superadditive but non-convex game from example 2.2.13:

coalition(s) C	v(C)
$\{a_1\}$, $\{a_2\}$, $\{a_3\}$	0
$\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$	10
$\{a_1, a_2, a_3\}$	12

Note that this game is not as far-fetched as it might seem at first glance. For example, each of the 2-agent coalitions might be able to provide a service for a given request at the same cost. However, each of the agents can perform its necessary action only once, for example due to resource constraints. Thus the service can still be provided only once in the grand coalition All three agents however might together provide another requested service when they work together in the grand coalition, leading to an additional value of 2.

The social-welfare maximizing coalition structure in this game is the one containing the grand coalition. The two-step coalition formation approach thus forms the grand coalition in a top-down manner. But a rational autonomous agent who does not benefit from it would not agree to form its assigned coalition in a negotiation process. Instead, each agent would prefer to form one of the two-agent coalitions. In a round-based coalition formation approach, two agents would thus form a coalition in the first negotiation round, possibly employing some additional preference function other than the maximal payoff. In the next round, this coalition would not profit from merging with the third agent and thus no new coalition would form, finishing the negotiation. △

Airiau and Sen (2010) explore still more problematic situations for the two-step approach which might happen if the game is subadditive. They argue that stability might be hampered in such cases, and propose to re-establish stability in such situations by allowing inter-coalitional side-payments. However, this means that agents which are not bound by coalitional contracts have to make side-payments to each other. This is contrary to the basic model of cooperative games where agents are in a contractual agreement if and only if they are in the same coalition.

Nevertheless, we also employ the two-step coalition formation approach in our proposed trusted coalition formation algorithm in chapter 6.

Having outlined the basic structures of coalition formation algorithms which we employ in this work, we now discuss their computational complexity.

Complexity of Coalition Formation

The issue of computational and communication complexity is central for all multiagent coalition formation approaches. We already argued that in a multiagent system, agents are unlikely to already have the complete information about coalition values when they come together to negotiate coalitions. But there is another problem with the assumption that the input to a coalition formation algorithm is provided literally in characteristic function form, i.e. specifying each individual $v(C)$ for all possible coalitions $C \subseteq \mathcal{A}$ in the game $(\mathcal{A}, v)$. Then, the input to an algorithm to compute a solution is already exponential in the size of the agent set $n = |\mathcal{A}|$, as there are 2^n coalitions. Therefore, Deng and Papadimitriou (1994) note that many solution concepts can be computed "efficiently", since computational complexity classes are a function of the problem input length.

As a consequence, it is often assumed that coalition values are not given as explicit input but are either specified in some compact representation, or can be computed by each agent efficiently by some procedure which may or may not be further specified. In the latter case, the coalition value function is an oracle, i.e. a black box function.

For compact representation schemes, the space requirement of the input varies depending on the specific representation and the concrete game that is being negotiated. For coalition value functions as procedures, the computational complexity of a coalition formation algorithm depends on the number of agents with the complexity of the procedure factored-in.

We discuss these two approaches in the following two subsections a bit more detailed. However, please keep in mind that while our algorithms are designed to be of polynomial complexity, tackling the computational complexity of coalition formation in general is not in the scope of this thesis.

Compact representation schemes

A compact representation scheme aims to model the coalition value function in polynomial space with respect to the number of agents (see e.g. Chalkiadakis et al., 2011, , chapter 3 for a recent overview). The idea is to exploit structure in the coalition value function, which is argued to be often existing in real-world scenarios.

For example, Deng and Papadimitriou (1994) introduced *graph games*, which are weighted graphs where the agents are the nodes and the edges between them have some weight. Thus, the input length is bounded by the number of agents plus the max. number of edges, i.e. $n + n^2$ (edges of weight 0 might obviously be omitted). The coalition value $v(C)$ in a graph game is determined as the sum of all edges within C, i.e.

$$v(C) = \sum_{a_1, a_2 \in C} weight(a_1, a_2)$$

. For such games, they show that the Shapley value can be efficiently computed in $O(n^2)$. However, they also show that the problems of checking non-emptiness of the core and core-membership of a given payoff distribution are NP-complete. Further, they conjecture that checking membership in the kernel of a payoff distribution is NP-hard. On the other hand, if all edges have non-negative weights, all of these problems can be solved in polynomial time.

However, graph games do not allow for representation of arbitrary coalition games, since coalition value contributions of groups of more than two agents cannot be modeled. In our example SWS agent scenario, this translates to not being able to express value contributions of composed services which require more than two agents for execution. For this purpose, Deng and Papadimitriou (1994) also consider hypergraphs, where each edge connects $k >= 2$ nodes. But the representation power of a hypergraph modeling a coalition value function is still limited:

Example 2.3.2
Consider again the game from example 2.2.13:

coalition(s) C	v(C)
$\{a_1\}$, $\{a_2\}$, $\{a_3\}$	0
$\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$	10
$\{a_1, a_2, a_3\}$	12

This game cannot be modelled as a graph game, as its set of edges E would need to contain $(a_1, a_2) : 10$, $(a_1, a_3) : 10$ and $(a_2, a_3) : 10$. But then the value of the grand coalition is $\sum_{e \in E} weight(e) = 30 \neq 12$. △

More recently, Ieong and Shoham (2005) introduced the more general *marginal contribution nets* (MC-nets). This model allows to specify only marginal contributions of agents to the coalition values, that is, the difference of the value of a coalition C without a given agent a and the value of $C \cup a$. Rules specified as

$$pattern \rightarrow value$$

pairs, where a pattern is a conjunction of terms representing the requirement to include or exclude a certain agent. The (non-negative) value specifies the marginal contribution to the coalition value if the coalition satisfies the pattern. For example, the rule $a_1 \wedge a_2 \wedge -a_3 \rightarrow 10$ specifies that this rule contributes a value of 10 to all coalitions C containing a_1 and a_2 but not a_3. It is shown that the Shapley value can be computed in polynomial time in the size of the input. However, the problems of core non-emptiness and core-membership are shown to be $coNP$-hard and $coNP$-complete.

While it is also shown that MC-nets are fully expressive, i.e. that every cooperative game can be modelled in this way, an exponential number of rules might be required(Elkind et al., 2008). This is essentially the case if many coalitions' values are not greater or equal than the sum of the values of their respective subcoalitions (subsets). The example from above again provides an idea of this behaviour:

Example 2.3.3
The game of example 2.3.2 can be modeled as an MC-net by these rules:

$$a_1 \wedge a_2 \wedge -a_3 \rightarrow 10$$
$$a_1 \wedge a_3 \wedge -a_2 \rightarrow 10$$
$$a_2 \wedge a_3 \wedge -a_1 \rightarrow 10$$
$$a_1 \wedge a_2 \wedge a_3 \rightarrow 12$$

Thus, if we allow the literal specification of v to also omit 0-coalition values, the MC-net representation is not smaller. △

Elkind et al. (2008) also introduced an extension to MC-nets, *read-once MC-nets*, which allows for more flexible pattern specifications. Therein, *read-once boolean formulas* are allowed with arbitrary binary Boolean connectives such as $\vee$, $wedge$ and $\oplus$ (exclusive or), while negation is allowed only on the atoms. In a read-once formula, each atom (i.e. agent) is allowed to appear maximally once. They provide an algorithm to compute the Shapley value for a game specified with such rules and prove that is has polynomial runtime. The algorithm and prove are given for patterns with the connectives $\vee$, $wedge$ and $\oplus$, but it is argued that all other binary Boolean connectives might be treated analogously. In our example, this allows us to save one rule:

Example 2.3.4
The game of example 2.3.2 can be modeled as a read-once MC-net by these rules:

$$a_1 \wedge (a_2 \oplus a_3) \rightarrow 10$$
$$-a_1 \wedge (a_2 \wedge a_3) \rightarrow 10$$
$$a_1 \wedge a_2 \wedge a_3 \rightarrow 12$$

△

Still more representation schemes have been introduced in the literature, such as skill-based representations. In Ohta et al. (2006), agents are modelled to possess certain skills, and there exists a function mapping subsets of skills to values. It allows for a succinct coalition value function representation if each coalition requires few skills (Chalkiadakis et al., 2011). The representation introduced in Bachrach and Rosenschein (2008) additionally employs sets of tasks, where each task requires certain skills to be fulfilled. They show that certain restricted games can be represented compactly with this representation. Computing the Shapley value turns out to be NP-hard for two of the considered restricted game classes, and remains unknown for the others.

Finally, Shrot et al. (2010) further reduce the compactness of game representation to agents' *strategic and representational types*. Therein, two agents have the same strategic type if they are symmetric in cooperative game theory terms. They show that several problems, such as determining core membership or computing the Shapley value, can be computed in polynomial time with such a model if the number of agent types is fixed. However, they also show that in general, it is NP-hard to compute the strategic types of the agents, such as in coalition resource games (see Wooldridge and Dunne, 2006).

Summarizing, it can be said that using one of these alternative representation schemes, games might have either a

- *succinct representation* – i.e. they can be specified in polynomial space wrt. to the number of agents – but with computational polynomial complexity of classical stability concepts only for restricted classes of games (indeed, too restrictive already for our service-agent example), or a
- *potentially exponential representation* wrt. the number of agents, thus rendering the complexity of algorithms wrt. to the overall input size irrelevant.

Coalition value function as procedure

If no structure or other helpful properties like convexity of the coalition value function is known, but it is assumed that coalition values can be computed in polynomial time, a coalition formation algorithm may regard it as a black box. Alternatively, the procedure to compute coalition values may be specified as part of the algorithm itself. This approach is followed by e.g. Contreras et al. (1998), Klusch and Shehory (1996), Shehory and Kraus (1998, 1999) and Klusch (1997). The latter employs the local worth model from definition 2.2.27.

If coalition values are computed using some such procedure, the complexity of the algorithm is given in terms of the number of agents, with the complexity of the coalition value function factored in. Because all the classical solution concepts require to evaluate an exponential number of coalitions, one has to employ some kind of modification or approximization to obtain polynomial runtime As we mentioned above, the round-based building up of coalitions lends itself to a reduction of the runtime complexity by bounding the coalition size. The algorithms introduced in the above cited works all follow this approach.

On the other hand, the two-step coalition formation schemes outlined above follow a different route. They typically allow coalitions of all sizes to form, but approximate the social welfare maximizing coalition structure. Sandholm (1999) have shown that to establish a finite bound from the optimum if the game is not known in advance to be super- or subadditive, one has to examine exactly $2^N - 1$ coalition structures [4]. This is because in a general and unconstrained game, coalition values are arbitrary and independent of each other,

[4]If the game is super- or subadditive, finding the social-welfare optimizing coalition structure is trivial: they are the structures containing only the grand and only single-agent coalitions, respectively

so every possible coalition and its value have to be evaluated least once. And as we saw in 2.2, there are $2^N - 1$ coalitions in a game of N agents. Michalak et al. (2009), Rahwan (2007), Rahwan et al. (2009) and other have refined this approach to find good solutions faster in practice, sometimes compromising on the worst-case runtime. Rahwan et al. (2012) further extends this line of algorithms to take externalities (external effects on the performance or utility of coalitions which are not modeled by the cooperative game itself) into account. However, these algorithms do not take the payoff distribution into account. This is to be done "after the fact", i.e. after the coalitions have already been formed. As mentioned above, the classical solution concepts require to examine an exponential number of coalitions to find a stable solution. An interesting question is thus whether the approximation of the above cited two-step approaches to coalition structure generation can be efficiently re-used for also computing a stable payoff distribution, but this is not taken into account at all in those works.

Furthermore, remember that the two-step coalition formation approach may prove problematic for subadditive or non-convex games. The above mentioned approach by Airiau and Sen (2010) to remedy this problem is based on inter-coalitional side-payments. These determined by a stability concept $Kernel^+$ which is extended from the Kernel such that all agents have to be in equilibrium, not just those in the same coalition. To arrive at such an equilibrium, an agent has to evaluate 2^{n-1} coalitions, similar to the classic Kernel. Thus, it remains to be seen whether the complexity advantages which can be gained by fast algorithms for generating a social-welfare optimizing coalition structure can be preserved also over the payoff distribution phase.

Finally, Arib and Aknine (2011) propose to use an explicit model of planned actions, where each set of actions has a certain desirability for the agents. But their model uses non-transferable utilities, so no side-payments are allowed and the coalition to which an agent belongs and which plans are to be executed by that coalition directly determines the agent's utility. They argue that by using a preference order on actions, the search space for desirable coalitions can be reduced by each agent. While they provide some experimental results to support this, they neither give information on how many plans the agents had in those experiments, nor give a theoretical analysis of complexity.

Summary

In summary, we can say that tackling the complexity of cooperative game theory based coalition formation is a very difficult problem. Effectively, algorithms with feasible runtime thus either yield solutions which are arbitrarily different from optimal and stable solutions, or require considerable assumptions on the types of games they can solve (in particular the coalition value function) and are thus applicable only in very specific settings.

Before we discuss further open problems in coalition formation, in particular the ones for which we provide solution approaches in this thesis, we now briefly consider some other aspects and models of coalition formation in multiagent systems.

2.4 Open Problems of Coalition Formation

So far we introduced the general approaches to use cooperative game theory concepts as a basis to model multiagent coalition formation algorithms and protocols. We saw that a basic challenge is computational complexity, which all such approaches which are supposed to be of practical value must tackle. In coalition formation settings in open multiagent systems, however, more problems arise. In this section, we summarize the specific challenges that we tackle in this thesis and how. They are then set into the context of related literature in the following chapter 3. We also acknowledge here that these are not the only open problems in this field. For example, some more are listed in Chalkiadakis et al. (2011).

Coalition Formation under Uncertainty

As we pointed out in section 2.3, coalition formation algorithms and protocols which are based on traditional stability concepts imply that the the information required to compute coalition values needs to be known exactly in advance. But in many real-world environments, such perfect information cannot be assumed. An agent might not certainly know the (exact) execution cost of a task that itself is to perform, or the utility of satisfying a goal of its own:

On one hand, an agent might not be able to assess the costs that will arise due to its required actions in a given coalition precisely before actually performing these actions. For example, in the application domain of semantic web service agents, the resource consumption of a service's execution — such as the amount of memory needed — might not be exactly known before executing the instance. Or it might depend on the input, which will not be known if it is a part of the outputs of previous executions of other services in a composition plan.

On the other hand, the value resulting for an agent of successful execution of a coalition's actions might be only vaguely known by the requesting agent. For semantic web service agents, that is the valuation of the satisfaction of a service request. For instance, an agent could request a complex service in order to resell it to one of its own customers, but might not have negotiated a fixed price with its customer yet.

For this kind of uncertainty, we propose a *fuzzy number*-based coalition formation algorithm in chapter 4, BSCA-F. Fuzzy numbers are a special case of fuzzy set theory as introduced by Zadeh (1965). A fuzzy set extends the notion of traditional crisp sets such that membership is not a Boolean property (i.e. an element either is a member of a given set, or it is not), but a *degree of membership*. Applying this notion to the set of real numbers yields the set of *fuzzy numbers*. The idea then is to extend solution concepts of cooperative game theory to operate on games with fuzzy coalition values, which we also call *fuzzy cooperative games* or *fuzzy-valued games*. Thus, the use of fuzzy numbers allows agents to only vaguely specify costs and rewards. To achieve low computational and communication complexities, we extend the efficient coalition formation algorithm BSCA (Klusch, 1997), which is based on the solution concept of the recursively bilateral Shapley value (see 2.2.21). While considering only intra-coalitional relationships to determine the payoffs, and thus yielding only subgame-stability, it is very quick to compute.

However, another kind of uncertainty for a game might result from coalitions failing completely, and thereby not obtaining any profit. For example, this might happen if the agents are resource-bounded and have to complete the coalitional action in a given time. If they fail to achieve this, the respective request is not satisfied, producing no value. However, some services might have already been executed at some cost. Therefore, the coalition experiences a net loss in this case.

In situations such as these, an agent might not only be interested in maximizing the payoff it might expect, but especially how much it might loose in the worst cases. For this kind of situation, we show how to use a *coherent risk measure* to assess and quantify the financial risk of a coalition(Artzner et al., 1999). Risk measures are often applied in finance and insurance applications to evaluate the risk of portfolios. One property of coherent risk measures is that the combined risk of two portfolios is less or equal than their added single risks. Therefore, portfolio diversification can lead to reduced overall risk (i.e., not "putting all one's eggs in one basket"). In chapter 5, we apply this analogously to coalitions by allowing the agents to be part of multiple coalitions at the same time. But this means that each agent is part of its coalitions only to some degree. The degree determines how much of its resources an agent will "invest" in the coalition, thereby influencing the probability that the coalition will finish its task in time. Coalitions of this sort are also called *fuzzy coalitions* (not to be confused with fuzzy-valued games as explained above). Fuzzy coalitions have been first introduced by Aubin (1979) and Butnariu (1980), including extensions of cooperative game theoretic solution concepts. However, they assume a linear relationship between an agent's membership degree in a coalition and its payoff. Unfortunately, as it turns out, this does not fit our model, as the membership have a non-linear relationship with a coalition's probability of success. We therefore extend the cooperative game solution concept of the *kernel* to fit with our model and to respect the agents' individual risk bounds when computing stable payoff distributions. Based on this, we outline an algorithm for resource-bounded service provider agents that guarantees to adhere to the risk bounds, the RCF.

Truth-telling in Coalition Formation

Another issue with common approaches to multiagent coalition formation is the possibility of the participation of deceiving, defrauding or simply under-performing agents. For example, agents might communicate manipulated data to other agents or coalitions in order to obtain an unjustified higher payoff.

This is possible in many coalition formation methods because they rely on an initial or incremental exchange of data among the agents such that each agent is able to assess the coalitions it might join and its resulting payoff according to the chosen stability concept. Therefore, an agent might deliberately communicate false data in order to unjustifiably increase its own payoff, possibly at the cost of others. For example, an agent could overstate its capabilities, letting the others believe that it will produce more value for the coalition than it actually can. The agent might behave in such a way intentionally, in which case we say that the agent *defrauds* or *deceives*; or it might do it unintentionally, in which case we call the agent *unreliable*.

Another issue is the actual execution of side payments within coalitions. In corollary 2.2.29 we saw how to determine an agent's net side payment $sp_u(a, C)$ in its coalition C in a solution to a coalition game in the form of a configuration $(\mathcal{C}, u)$. However, nothing is said so far to which exact other agent it should pay which amount. Or conversely from which agents it should expect to receive a certain amount, and how it can sure that it is indeed paid by these others. After all, the computation and negotiation of a stable solution is not quite that useful if it is then not actually implemented by the agents.

Finally, agents might underperform when executing tasks, i.e. the task executions turn out to not produce the promised values.

Therefore, in chapter 6, we investigate how kernel stable coalitions might be manipulated. We then devise an appropriate communication protocol using cryptographic techniques to prevent this. Then, we propose a payment scheme and show that it is rational for each agent to adhere to it. At last, agents measure each other's performance via a generic trust model.

Privacy Preservation in Coalition Formation

Also, remember that traditional coalition formation approaches usually devise protocols such that each agent informs each other agent about its requests and offers. This is often necessary to determine coalition values and stability. However, it is problematic if the agents have privacy constraints, such that it is not acceptable for them to provide certain other agents with certain information.

Example applications that would benefit or even require a privacy preserving coalition formation protocol include health care web service agents, which form coalitions e.g. to automatically handle insurance issues, transportation, hospital and medical personal assignments. But an agent responsible for transportation should not need to know which patients are assigned to which doctors.

But this problem has not received much attention in the literature, and we provide the first privacy preserving coalition formation algorithm in chapter 7.

Chapter 3

Related Work

In this chapter, we discuss related work for each of the open challenges in multiagent coalition formation which we tackle in this thesis. Although there are some overlaps, the different approaches tackle quite separate problems. Therefore, we consider related work on uncertainty, truth-telling and privacy preserving coalition formation each in its on section.

3.1 Coalition Formation under Uncertainty

To remedy the situation of uncertainty in coalition formation and relax this assumption of perfect information on the coalition values, some approaches have been presented in the literature. Most popular seem to be probabilistic coalition formation approaches, of which we discuss some related ones. Before we do that, however, we first look at some results on the general applicability of modeling uncertainty with fuzzy numbers as opposed to probabilities in the following subsection. There we also relate to the (very few) other existing approaches to fuzzy-valued coalition formation. To conclude this section, we relate to heuristic approaches to coalition formation under uncertainty.

Fuzzy-valued Coalition Formation

Like probability distributions on numbers, fuzzy numbers are functions of numbers mapping to the unit interval. There are two common interpretations of the resulting degrees, both of which are fundamentally different to probabilities (see e.g. Delgado et al., 1994, Dubois and Prade, 1983, Zadeh, 1978):

1. Vague: the number is interpreted as inherently vague. That is, an exact value for the given entity represented by such a fuzzy number does indeed not exist (in the regarded universe) A vague interpretation thus presumes that the entities of interest are *inherently* imprecise. For example, the word “red” as used in day-to-day human language does not specify a specific colour with a precise wave-length, but a whole range of colours which are more or less red. Fuzzy membership degrees in this sense are thus also called *degrees of truth*.

2. Possibilistic: the number represents an entity that has some exact value, but it is not known. It is only known to which degree it is ‘possible’ or ‘necessary’ that a certain value is equal to that exact value. While this seems similar to the semantics of probabilities, it is more flexible (due to non-unitary and non-additivity) by allowing for a less exact and strict modeling of a problem instance. In particular, it allows for ignorance of membership degrees in cases which are not of interest, without introducing error.

The lack of a well-defined semantics of these interpretations may seem disadvantageous at first. However, they have certain properties that make them worthwhile to consider in conceptually and computationally complex settings such as coalition formation:

- There exist well-defined set and arithmetic operations as well as preference relations on fuzzy numbers.

- Complex operations on fuzzy numbers can be efficiently approximated if some loss of information is acceptable.

- There exist empirical evaluations indicating that possibilistic reasoning is generally more compatible with human conscious reasoning than probabilistic reasoning is, as it was shown e.g. by Raufaste et al. (2003). They were able to confirm the claim by Zadeh (1978) that humans really conjure subjective possibilities rather than subjective probabilities. It means that a possibilistic system should be potentially able to better capture the essence of human-specified inputs than a probabilistic system. This could be especially useful for designing appropriate end user interfaces for agents, and might help produce results which are more in line with human expectations, although this would need to be further investigated.

- Possibility theory has been shown theoretically and experimentally to be more flexible than and performing as least as well as a standard probabilistic approach in instance-based learning (Hüllermeier, 2003). Although that domain is considerably different to that of coalition formation, it shows that possibilistic approaches are worth looking into in domains that call for more flexibility and error-tolerance.

In the context of cooperative game theory, fuzzy-valued games were first employed by Mareš (2001). In particular, this work introduced fuzzified versions of core concepts like additivity, as well as the stability concepts of the fuzzy Core and the fuzzy Shapley value.

However, when it comes to concrete coalition formation algorithms for multiagent systems, not much is available in the literature. In fact, Blankenburg et al. (2003) proposed the only other coalition formation algorithm for fuzzy-valued games that we are aware of. Therein, a fuzzy extension of the kernel and an algorithm KCA-F to form fuzzy-kernel stable coalitions were introduced. It was shown that it is possible to obtain polynomial computational complexity of this algorithm by putting a constant bound on the maximum allowed coalition size. Then the complexity was shown to be in $O(|\mathcal{A}|^7)$ if it is assumed that arithmetic operations on fuzzy numbers can be done in constant time. For practical applications, this is considerably greater than the complexity of $O(|\mathcal{A}|^4)$ of the BSCA-F (see section 4.4). Also, the issue of defuzzification of a fuzzy payoff configuration was not considered for the KCA-F, but we demonstrate how it may be done in section 4.5. But given that in the KCA-F the same possibilistic interpretation of fuzziness is used as in the BSCA-F, the same methods should be applicable.

Probabilistic Coalition Formation

One approach to tackle uncertain coalition values is to model them probabilistically. Suijs et al. (1999) introduced stochastic cooperative games where the payoffs are pairs consisting of a deterministic and an uncertain part. The former determines the side-payments while the latter models a random variable over the uncertain outcomes of the associated coalitional actions. Agents might have individual preferences over the random payoffs and might generally be risk-averse, risk-neutral or risk-loving. Note that it is implied that the probability distributions are common prior knowledge. That is, the agents know the probability distributions of the (financial) outcomes of all possible coalitional actions in advance. The solution concepts of the core and the nucleolus are then extended to such games. As in the classic case, the core might be empty, and it turns out to be difficult to determine non-emptiness of the core in general. The nucleolus is defined for stochastic games in which each agent's preference function satisfies a weak form of continuity. This can be obtained if the set of outcomes of coalitional actions is finite and the preference functions have some arguably natural properties. Both the core and the nucleolus for stochastic games are only applicable for the formation of the grand coalition, and thus for superadditive stochastic games. But it is also shown that, equivalent to deterministic games, if a payoff distribution of a stochastic game is in the nucleolus and the core is non-empty, it is also in the core. While the approach provides a workable theory for the analysis of stochastic games, no algorithms or schemes are provided to actually determine a solution, let alone one that is applicable in open multiagent systems. This concerns in particular a specialization of their theory to allocate financial risk among insurance providers and clients which was detailed by Suijs et al. (1998). For a more detailed and complete account of the theory and the insurance scenario we also point to Suijs (1998).

This risk allocation setting might seem similar to our risk-bound coalition formation algorithm RCF. However, while in Suijs et al. (1998) the risk is the main subject of the game, i.e. the risk is modelled via the coalition values, in the RCF it is a constraint on a service allocation game. Also, since the RCF employs the kernel as a solution concept, solutions to the game always exist.

In the literature, some approaches have been introduced to overcome the assumption that the agents know in advance the probability distributions of the outcomes of coalitional actions. Chalkiadakis and Boutilier (2004), Chalkiadakis et al. (2007) introduce a *Bayesian core* for repeated coalition formation (BCF) where the agents are uncertain about each others *types*. That is, the agents do not know in advance how well other agents will perform their assigned tasks in the coalitional actions, making costs and rewards and thus the coalition values uncertain. Additionally, this model implies that the agents' expectations of each other's types are subjective and thus can be different. This is in contrast to the classic solution concepts which imply perfect information of each agent about the objective values. However, each agent is assumed to know its own type with certainty. Coalition values then are a function of the types of the coalition members and each agent might have a different expectation of a coalition's value. Consequently, if the agents' types of a coalition where known with certainty, so would be the coalition value.

Note that apart from using Bayesian probabilities as opposed to fuzzy numbers, this is different to the uncertainty modeled in the BSCA-F, where uncertainty is a quality of the coalition values themselves. That is, even though in the BSCA-F it is assumed that the agents know each others types' perfectly, the coalition values are still uncertain.

The agents in BCF then form coalitions and execute coalitional actions repeatedly while using reinforcement learning to improve their beliefs about the agents' types. The learning method is modeled as a partially observable Markov decision process (POMDP). Thus, while the agents might have some prior knowledge, they improve their beliefs individually and independently over time based on their experiences. To actually form coalitions, agents can propose certain restricted changes to the coalition structure. If the agents are allowed some experimentation (that is, temporarily accepting a disadvantageous configuration in hopes of obtaining a more profitable one in later stages), it is proved that this process converges to a Bayesian core-stable configuration. Chalkiadakis et al. (2007) further showed that there is non-cooperative justification of the Bayesian core. In particular, they *"prove that if the BC of a coalitional game (and of each subgame) is non-empty, then there exists an equilibrium of the corresponding bargaining game that produces a BC element; and conversely, if there exists a coalitional bargaining equilibrium (with certain properties), then it induces a BC configuration."* The Bayesian core, like the classic core, may be empty, and is in general computationally intractable. The latter also holds for POMDPs on whose solution the coalition formation model is based. Chalkiadakis et al. (2012) thus provide an evaluation of several simulated coalition formation sequences which use different approximation methods to compute the POMDP solutions in feasible time. Although no comparison to a global optimum is made (which is intractable to compute), the different methods show quite widely varying results in different settings with respect to overall payoffs. In a five-agent setting, all simulated methods do not reliably converge to a Bayesian core stable configuration, and in a ten-agent setting, the Bayesian core is empty.

In contrast, in the BSCA-F, by using the recursively bilateral Shapley value, we have taken a different route in that we chose to abandon the use of a high-complexity and often empty solution concept from the start, resulting in very low computational complexity without any further approximation. Still, the simulation results that we present in section 4.6 show that the BSCA-F very often results in fuzzy payoffs that are at least possibly in the core. On the other hand, higher degrees of necessity of core membership turn out to be achievable only when certain ranking operators are employed in the BSCA-F (a ranking operator for fuzzy numbers assigns gives a degree to which one of the numbers is to be considered greater than the other, see section 4.1). Thus, whether the BSCA-F or the BCF method is more promising when the trade-off between runtime performance and optimality or stability of the solution is regarded could only be really assessed by a direct comparison. However, it is still to be kept in mind that two algorithms handle different kinds of uncertainty.

Ieong and Shoham (2008) propose a further refined theoretical model for Bayesian coalition games in multiagent systems. To account for uncertainty, this model explicitly includes a set of *possible worlds* and for each agent a partition of those worlds whose members pool worlds that are indistinguishable from the perspective of the respective agent. Coalitional *contracts* are defined as mappings from the set of possible worlds to payoff vectors, which accommodates the fact that no certain payoffs can be determined before all agents know which world turns out to be true. Based on this and assuming that the grand coalition is always formed (i.e. assuming superadditivity), three notions of Bayesian core-stability for such contracts are introduced:

1. The *ex-ante core* models stable contracts *before* anything is known about which world is true.

2. The *ex-interim core* models stable contracts when each agent knows to which of its partitions of indistinguishable worlds the true world belongs.

3. The *ex-post core* models stable contracts when each agent knows which is the true world.

It is further argued that this model is a generalization of Suijs et al. (1999) if each coalition has only one action, and that the ex-interim core is somewhat similar to the Bayesian core proposed by Chalkiadakis and Boutilier (2004), Chalkiadakis et al. (2007) if the formation of the grand coalition is assumed. However, as is the case with Suijs et al. (1999), no actual coalition formation algorithm is given. Considering the descriptive, all-encompassing nature of their model, doing so in a computationally feasible manner seems to be highly non-trivial.

Heuristic Approaches

Another direction is followed in Kraus et al. (2003), a heuristic coalition formation (HCF) approach in the request for proposal domain. Therein, requester agents issue requests for proposal (RFP), for which a coalition of service provider agents might make bids to jointly try and satisfy the request within a given time limit. This setting is thus comparable to the one considered in our risk-bounded coalition formation algorithm RCF. They evaluate the achieved average payoffs of different heuristics in different settings, in particular involving different degrees of imperfect information. However, no conclusions can be drawn with regards to the risk that agents might experience losses, which is the main concern of the RCF. Also, no relation of the resulting payoffs to stable payoffs in the sense of cooperative game theory is provided.

In the same request for proposal domain, Jones and Barber (2009) advance the heuristic approach to let the agents be able to also react to dynamic changes in the requests, as well as use dual heuristics for task and team (coalition) selection. Then the interdependencies of these heuristics are studied via simulations. However, again only average payoffs are analyzed, but neither are risk nor cooperative game theoretic stability.

Fuzzy Coalitions

Although we employ fuzzy coalitions in our risk-bounded coalition formation approach, the standard models of fuzzy coalitions do not fit our approach(Aubin, 1979, Butnariu, 1980). This is because they assume a linear relationship between coalition membership and coalition values, which we show in chapter 5 does not fit the model employed there.

Nevertheless, we note that Nishizaki and Sakawa (2001) proposed a number of algorithms to compute solutions according to their concepts. They did however not propose a protocol that enables a coalition negotiation among computational autonomous agents.

Shehory and Kraus (1999) considered the formation of overlapping but non-fuzzy coalitions. They however focus on maximising the joint payoff of all agents rather than individual payoffs or minimising potential individual losses. In contrast, our approach focuses especially on the latter points. Thus, the motivations and the properties of the obtained solutions are very different.

More recently, Chalkiadakis et al. (2010) investigated cooperative games with overlapping coalitions in multiagent system settings. They provide definitions of the core and other basic game theoretic concepts and study different aspects of stability. However, no coalition formation algorithm is provided.

3.2 Truth-telling in Coalition Formation

In the literature, there exist two general approaches to encourage agents to be truthful in negotiation settings:

1. Measuring each other's trust-worthiness with the help of a *trust model* and resulting *trust measure*. This is then used by agents' to adjust their assessments of what to expect from other agents.

2. Designing the interaction protocols such that truth-telling is rational for agents, i.e. implement *incentive compatibility*.

We now consider each of these approaches in turn.

Trust measures

If the agents engage in coalition formation repeatedly, they might over time learn to map other agents' communication behaviour to their actual performance. This is the approach of *trust models*, as it was outlined by Dasgupta (1998). Ramchurn et al. (2004) provides an overview of different approaches to trust in multiagent systems, summarizing Dasgupta's definition of trust as follows:

"Trust is a belief an agent has that the other party will do what it says it will (being honest and reliable) or reciprocate (being reciprocative for the common good of both), given an opportunity to defect to get higher payoffs.""

They further view trust as being composed of two main components:

1. *individual-level trust*: the mutual trust among individual agents, which an agent might assess by e.g. incorporating learning, reasoning and/or reputation models.

2. *system-level trust*: the trust of agents in the system to enforce trustworthy behaviour of other agents by e.g. implementing appropriate rules of encounter and interaction protocols.

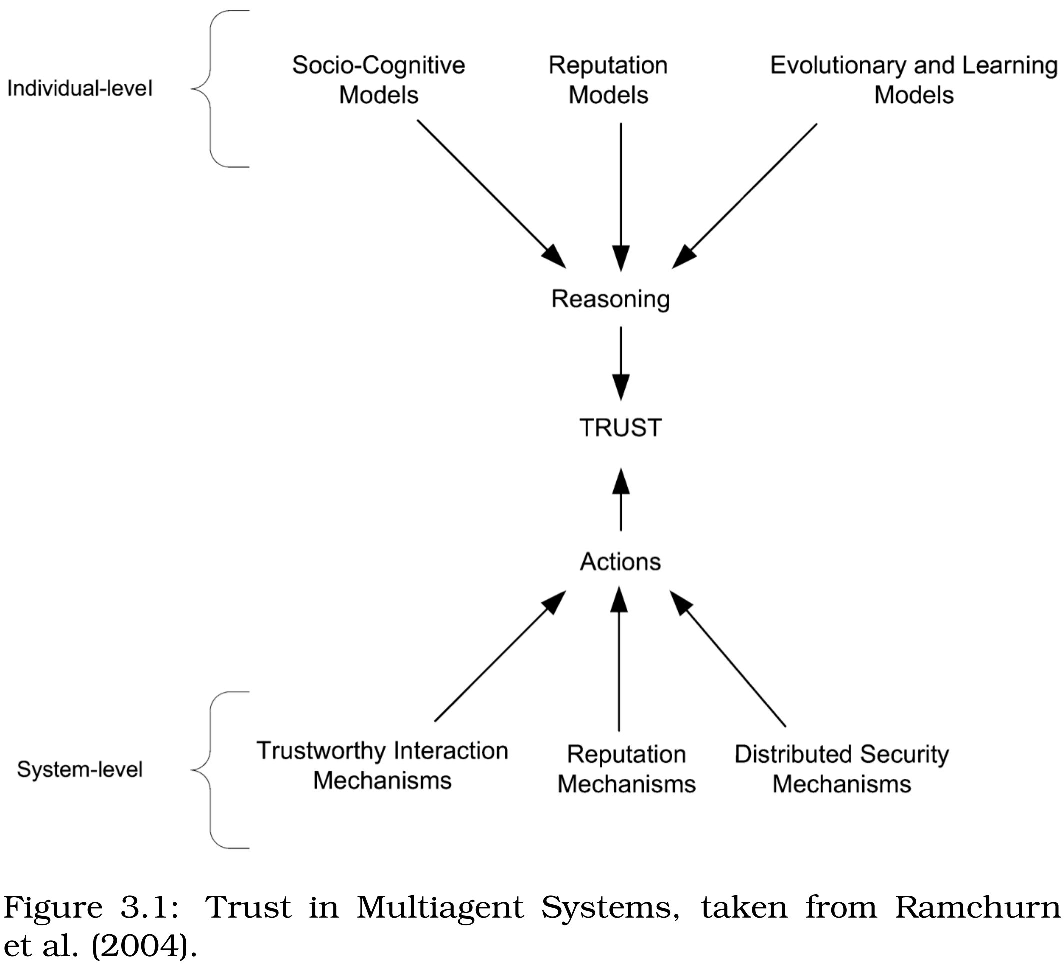

Figure 3.1: Trust in Multiagent Systems, taken from Ramchurn et al. (2004).

Figure 3.1 depicts this view of trust. In the literature on multiagent coalition formation, a number of protocols have been introduced that incorporate individual-level trust models, such as Breban and Vassileva (2002), Griffiths and Luck (2003), Hoelz and Ralha (2012) and Vassileva et al. (2002). These works, however, do not consider stability in the cooperative game-theoretic sense, and thus actually use a different notion of “coalition”. In contrast, in chapter 6 we introduce the coalition formation algorithm TKCF which enables agents to form Kernel-stable coalitions as well as use a trust measure to assess each other’s likelihood of truth-telling or reliability.

In fact, the uncertainty of other agents’ types modelled by the approach of Chalkiadakis and Boutilier (2004), Chalkiadakis et al. (2007) (BCF, see the previous section for a more detailed discussion) seems to be more closely related to the uncertainty about other agents’ trustworthiness in the TKCF. Also there, learning over repeated coalition formation is employed to improve the knowledge about other agents’ behaviours. However, in contrast to the BCF, agents in the TKCF also inform each other about their trust measures of potential coalition partners. Therefore, if the percentage of malicious or underperforming agents in the TKCF is not too high, it might be expected that an agent might be able to assess the trustworthiness of other agents quicker as opposed to the case where it only relies on its own learning as in the BCF. However, remember that the exact choice of trust model in the TKCF is not the focus of this thesis, but the design of the algorithm to hinder deception by agents even before employing the trust model. To this end, the BCF has nothing comparable.

Since coalition formation is usually assumed to be executed just by the participating agents, without direct involvement of a central authority (other than the providers of the basic system infrastructure), the problem of system-level trust might not seem as important for coalition formation. But depending on the exact implementation of coalition formation algorithms and protocols, auxiliary services such as directory services (so that the agents can find each other) might be used. We do however not consider this problem in this work.

Incentive compatibility

The second approach to the problem of deceiving agents is to make the interaction protocol *incentive compatible*. That is, to design the protocol in such a way that truth-telling leads to higher expected utility for the participating agents, and thus make it a preferred strategy of rational agents. This approach was introduced and has been widely studied in the research field of *mechanism design*. As Myerson (2008) puts it,

"A mechanism is a specification of how economic decisions are determined as a function of the information that is known by the individuals in the economy. Mechanism theory shows that incentive constraints should be considered coequally with resource constraints in the formulation of the economic problem."

It should be noted that for a *mechanism*, the *designer* is a central authority who sets the overall goal of the game and typically also implements or oversees the environment where the game takes place. Prominent examples of mechanisms in this sense include auction and stock markets, where the goal might be to achieve an efficient market such that goods are traded at their true market values. Both are examples of regulated markets, where the designers and implementors are auction houses or stock exchanges, respectively. Another example of a mechanism is voting, where the goal is usually to get a result that represents the participants' true preferences. The designer therefore devises the rules such that participating agents are likely to behave in a way that reaches the overall goal, even though they are assumed to be self-interested [1].

Note that, in contrast, coalition formation protocols and algorithms do not assume the existence of a controlling authority. Furthermore, in mechanism design, the agents are also assumed to act non-cooperatively. Indeed, cooperation among agents is usually forbidden in mechanisms, and thus called *collusion*.

Now, in order for the agents to partake in the mechanism such that the overall goal is fulfilled, agent's must be encouraged to act according to their true private preferences. For example, agents should make bids in an auction which reflect their true utility of the

[1] An issue in mechanism design is that the central authority generally has to be fully trusted by participating agents. It would e.g. be a problem for participants if the auction house chooses an auction mode that typically leads to higher prices in order to collect higher fees; or a corrupted voting committee which cheats when counting the votes because of its own interests.

traded good or vote according to their true preferences of the options on vote. In other words, agents should be prevented from bidding or voting strategically due to their expectations of the behaviour of other participants.

A mechanism which fulfills this is said to be *incentive compatible* (Hurwicz, 1972). More precisely, incentive compatibility of a mechanism with rational participants implies that truth-telling leads to a pareto-optimal Nash-equilibrium. That is, assuming that the agents are rational, and each agent assuming that all other agents are rational, the strategy with the highest expected value for each agent is to tell the truth. If truth-telling is even a dominant strategy of the mechanism, i.e. it is always the best strategy independently of other participants' behaviour (and thus the rationality constraint can be relaxed), then the mechanism features *strong incentive compatibility* (which is also called *strategy-proofness*). Unfortunately, Myerson and Satterthwaite (1983) proved that no incentive-compatible, budget-balanced (i.e. all payments between agents sum to zero, which translates to efficiency in terms of cooperative game theory) and individually rational mechanism can exist (for an explanation, see also Osborne and Rubinstein, 1994).

In order to devise incentive-compatible mechanisms, the *revelation principle* is often employed in mechanism design. This principle states that for every mechanism, there is an equivalent one in which all agents report their required data directly at the start of the mechanisms execution and which motivates the agents to report these values truthfully. However, Conitzer and Sandholm (2004) criticize this approach for often being computationally intractable. Also, all the computational burden lies with the central implementor.

Now, please note that every coalition formation algorithm and contained protocol can easily be re-designed in a more mechanism-like and centralized way simply by introducing an additional "manager" agent, which will never join any coalition, and changing the interaction protocol such that this manager acts as a relay for all messages. That is, if the protocol specifies that agent a_1 should send a certain message to a_2 at a certain time, the protocol is changed such that a_1 sends this message to the manager, and the manager then forwards it to a_2. But this means that also no incentive-compatible, efficient and individually rational coalition formation algorithm can exist. However, for our proposed kernel-based trusted coalition formation algorithm introduced in chapter 6, we at least show that each agent cannot determine whether its non-truth-telling will lead to an increased payoff, and that indeed it always potentially decreases its payoff.

3.3 Privacy Preserving Coalition Formation

Another issue with coalition formation algorithms is the preservation of the participating agents' privacy. As we have seen in section 2.3, agents are often required to initially exchange their data which is relevant to determine all the coalition values. However, what if it is not acceptable for an agent to reveal all or parts of this data to certain other agents?

In a web service agent scenario, for example, the public revelation of the quantity and value of local service sales, as well as individual requests for particular services which required to play cooperative games with complete knowledge could lead to an unsolicited competitive advantage for other agents of competing web service oriented businesses. The problem is, how can certain kinds of local financial data be kept private while still successfully participating in coalition negotiations to maximize individual profits?

This question has received little attention in the literature so far; indeed, to our knowledge Blankenburg and Klusch (2004, 2005a) and Blankenburg and Klusch (2006) (on which chapter 7 is based), are the only works to address this problem so far.

However, we note that Ohta et al. (2006), Yokoo et al. (2005a,b) and Ohta and Conitzer (2008) tackle the complementary problem of *anonymity-proofness* for the core, nucleolus and Shapley value. They model the game such that each agent a_i has a set of skills $Skills_i$. If $Skills$ denotes the set of all skills in a game, they define the characteristic function in terms of sets of skills instead of coalition values:

$$v: \; 2^{Skills} \mapsto \mathbb{R}$$

The value of coalition is then determined by the union of the sets of skills of its members.

Note that if such a skill-based coalition value function is non-decreasing (i.e. adding more skills does not decrease the value), this model can be easily translated to our simplified service agent model that we use in chapter 7 by replacing the skills with appropriate requested and offered services.

They then define a solution concept to be anonymity proof iff it holds that if an agent hides some of its skills, its payoff does not increase. Anonymity-proof versions of the core and the Shapley value are discussed. The motivation behind this is thus not to enable anonymity or, more general, the hiding of private information in negotiations, but to ensure that no agent can gain unsolicited profit by doing so. Consequently, they do not cover the issues of how agents can hide their information effectively and ensure that it stays hidden over the complete process of coalition formation and coalitional executions, as we do for the BSCA-P. On the other hand, we briefly argue in section 7.5 that the BSCA-P is not anonymity proof.

Lastly, we point out that the computational complexity of the anonymity-proof core and Shapley value are very high, i.e. $O(n2^n)$ and $O(2^n)$.

3.4 Other Models of Coalition Formation

In this section, we take a brief look at alternative coalition formation approaches which are not directly applicable to the problems we consider in this thesis.

Dynamic Coalition Formation

Theories of building up of coalitions in a dynamic process have also been considered in the classical game theory literature, there named *dynamic coalition formation* (e.g. Arnold and Schwalbe, 2002, Aubin, 2005, Filar and Petrosjan, 2000). These are concerned with theoretical properties of coalition negotiation processes. For example, Arnold and Schwalbe (2002) study the reachability of the Core assuming that a specific bargaining process is used as the negotiation method. Aubin (2005) examines the evolution of core-stable coalitions in a game where fuzzy coalitions form over time. These works, however, take a mostly descriptive perspective. Their aim is not to actually design algorithms which are applicable in multiagent systems.

To this end, Klusch and Gerber (2002) introduced a generic coalition formation scheme which tackles dynamics in the environment such as a changing agent set. It simulates multiple possible evolving scenarios in parallel and thus enables the agents to react to changes in the game and world states. Khan et al. (2011) propose a dynamic coalition formation algorithm for transmitters and receivers in a wireless network. Their model includes both the dynamics of the negotiation and changes in the coalition value function. However, it does not account for changes in the agent set.

In this thesis, we do not consider dynamic changes of the game during coalition negotiation.

Coalition Formation in the Robot Domain

Coalition formation protocols have been widely applied to enable and implement cooperation in the robot domain, where teams of robots are required to achieve certain tasks (e.g. Chen and Sun, 2012, Guerrero and Oliver, 2012, Ramaekers et al., 2011, Rohrmüller et al., 2012, Service and Adams, 2011, Vig and Adams, 2007). For such approaches, achieving small computational and communication complexity is usually of great importance, as is the coping with dynamics and uncertainty in the environment. However, teams of robots are usually assumed to work together completely cooperatively, and agents are not assumed to be self-interested. Thus, only the problem of finding social-welfare maximizing coalition structure is relevant in this domain, but not the problem of obtaining a stable payoff distribution. Furthermore, specific assumptions are often made that are applicable only to the studied problem setting. Thus, these approaches are not directly relevant or comparable to the coalition formation approaches which we consider in this thesis.

Argumentation-based Coalition Formation

Amgoud (2006) proposed to utilise *argumentation theory* to reason about coalition formation. A formal model of coalitional conflict relationships is proposed for this purpose. It includes different types of semantics for coalition structures and a proof theory allowing an agent to identify *acceptable* coalitions. A coalition is deemed acceptable if it is either *not defeated* by any other coalition, according to some not further specified binary *defeat relation*, or it can *defend* itself against the defeating coalitions via another binary *preference relation*. An argumentation-based dialogue process is then proposed which is shown to arrive at coalition structures which are acceptable in this sense. Additionally, *stable* and *preferred* structures are defined in terms of the defeat and preference relationships. However, agent payoffs are not considered at all, and it is not made clear whether the proposed coalition structure semantics correspond to any of the classical solution concepts for cooperative games.

More recently, Riley et al. (2012) expanded on this idea and modeled an argumentation-based dialogue process which induces both a coalition structure and a core-stable payoff distribution. Therefore, it is applicable only for games with a non-empty core (they do provide a method for determining non-emptiness of the core, however). Also, the agents are expected to maximise social-welfare first and their individual payoffs only secondary. Unfortunately, no analysis of the computational or communication complexity of the proposed process is provided.

Chapter 4

Fuzzy-valued Coalition Formation

One apporach to tackle multiagent coalition formation in cooperative games with uncertain coalition values is to employ fuzzy set theory as introduced by Zadeh (1965). A fuzzy set extends the notion of traditional crisp sets such that membership is not a boolean property (i.e. an element either is a member of a given set, or it is not), but a *degree of membership*. Applying this notion to the set of real numbers yields the set of *fuzzy numbers*. The idea then is to extend solution concepts of cooprative game theory to operate on games with fuzzy coalition values.

The main idea behind fuzzy sets is to provide an intuitive and rather simple way of modeling vagueness or uncertainty. As opposed to probability theory, it does neither require unitarity nor additivity on membership. As we will see, this renders basic operations on fuzzy sets (or fuzzy numbers) to be simple as well, and in particular allows for their very efficient computation.

On the downside, however, the actual meaning of a fuzzy membership degree is not as easily established as e.g. in the case of probabilities. In the literature, there have traditionally been two main interpretations: vagueness and possibility. A vague intepretation presumes that the entities of interest are *inherently* imprecise. For example, the word “red” as used in day-to-day human language does not specify a specific colour with a precise wave-length, but a whole range of colours which are more or less red. Fuzzy membership degrees in this sense are thus also called *degrees of truth*.

In contrast, a *possibilistic* interpretation assumes that the considered entities do indeed have crisp and exact values, but there is uncertainty about these values. While this seems similar to the semantics of probabilities, it is more flexible (due to non-unitary and non-additivity) by allowing for a less exact and strict modeling of a problem instance. In particular, it allows for ignorance of membership degrees in cases which are not of interest, without introducing error. This, we argue, makes it easier for humans to model situations which they do not fully understand in a possibilistic rather than a probabilistic way. Additionally, there exist empirical evaluations indicating that possibilistic reasoning is generally more compatible with human concious reasoning than probabilistic reasoning is, as it was shown e.g. in Raufaste et al. (2003). They were able to confirm the claim in Zadeh (1978) that humans really conjure subjective possibilities rather than subjective probabilities. It means that a possibilistic system should be potentially able to better capture the essence of human-specified inputs than a probabilistic system.

Because we are specifically interested in modeling games with uncertain coalition values, we employ the possibilistic interpretation of fuzziness to devise an appropriate coalition formation algorithm. Its formal theory has been well developed in e.g. Zadeh (1978) and Dubois and Prade (1994). Furthermore, the extension of cooperative game theoretic concepts to use fuzzy coalition values with possibilistic interpretation is fairly straightforward because of the existence of appropriate arithmetic and ranking operators. This has already been done for some of these concepts in e.g. Mareš (2001) and Blankenburg et al. (2003).

The fuzzy coalition formation algorithm presented in this chapter is based on the fuzzy bilateral Shapley value. This is a combination of the bilateral Shapley values as introduced in Klusch (1997) with the notion of the fuzzy Shapley value introduced in Mareš (2001). While Mareš assumes the vagueness intrepretation of fuzzy memberships, his definition of the fuzzy Shapley value is completely compatible with the possibilistic interpretation employed here.

The proposed algorithm avoids the need to constrain coalition sizes while obtaining low computational and communication complexity, which we proove theoretically. In order to achieve this, we utilize the fuzzy bilateral Shapley value. This, however, implies that only subgame-stability is achieved. We however proove that of the possible coalitions, the most profitable one is always formed in every negotiation round of our algorithm.

We further point out that the possibilistic mean value (see Carlsson and Fullér (2001)) can reasonably be applied to defuzzify fuzzy payoffs in order to implement unambiguous coalition contracts.

The remainder of this chapter is organized as follows. In section 4.1 we introduce the necessary preliminaries with respect to fuzzy numbers and fuzzy-valued coalition formation. We then extend the bilateral Shapley value to a fuzzy version in section fbsv. In section 4.3, our coalition formation algorithm BSCA-F is presented. We discuss its properties theoretically in section sec:bscafprops. In section 4.5 show in detail how to apply the BSCA-F in an example real world scenario. Finally, we present some evaluation results in section 4.6 before concluding in section 4.7.

4.1 Preliminaries

We first introduce the basic concepts of fuzzy quantities and operations on them, and then continue to introduce fuzzy-valued coalition games.

Fuzzy Sets and Quantities

This section introduces basic notions of fuzzy sets and quantities, along with some operations on them that will be required in the following sections. The following definitions follow the theory of fuzzy sets in a possibilistic interpretation as established by Zadeh (1965) and Zadeh (1978). Additional concepts such as the ranking operators we employ and the possibilistic mean value have been established in Dubois and Prade (1983, 1987, 1994), and Carlsson and Fullér (2001).

Definition 4.1.1 Fuzzy subset
A *fuzzy subset* $\widetilde{s}$ of a set S is defined by its *membership function*

$$\mu_{\widetilde{s}} \ : \ S \mapsto [0, 1]$$

where $\mu_{\widetilde{s}}(x), x \in S$, is called the *degree of membership* or *membership value* of x in $\widetilde{s}$. Further, $\widetilde{s}$ is called *normalized* iff $\sup_{x \in \mathbb{R}}\{\mu_{\widetilde{x}}(x)\} = 1$. △

Definition 4.1.2 Membership, support, and modal member
Let $\widetilde{s}$ be a fuzzy subset of a set S, and x in S. We say x is a *member* or *element* of $\widetilde{s}$ and write $x \in \widetilde{s}$ if it has a positive membership value:

$$\mu_{\widetilde{s}}(x) > 0, \quad x \in S$$

and define the *support* of $\widetilde{s}$ as the set of all its members:

$$support(\widetilde{s}) := \{x \mid x \in \widetilde{s}\}$$

An element x with maximum membership in $\widetilde{s}$ is called a *modal element* of $\widetilde{s}$:

$$x \in \arg\max_{y \in S} (\mu_{\widetilde{s}}(y))$$

△

The definition of fuzzy subsets can be applied to any set. Since we are interested in modeling fuzzy coalition values, we particularly concern ourselves with fuzzy subsets of the real numbers:

Definition 4.1.3 Fuzzy quantity
Any fuzzy subset of $\mathbb{R}$ is called a *fuzzy quantity*, and $\widetilde{\mathbb{R}}$ denotes the set of all fuzzy quantities. △

Definition 4.1.4 α-level cut
For $\widetilde{x} \in \widetilde{\mathbb{R}}$ and $\alpha \in [0, 1]$, the α*-level cut* $L_\alpha(\widetilde{x})$ of $\widetilde{x}$ is the set of all real numbers with a membership of at least α in $\widetilde{x}$:

$$L_\alpha(\widetilde{x}) := \{x \,|\, x \in \mathbb{R}, \mu_{\widetilde{x}}(x) \geq \alpha\}$$

An α-level cut is also just called α*-cut*. We also define $\alpha\min(\widetilde{x})$ as the minimum member with membership level α:

$$\alpha\min(\widetilde{x}) := \inf\{L_\alpha(\widetilde{x})\}$$

And, analogously, the maximum member:

$$\alpha\max(\widetilde{x}) := \sup\{L_\alpha(\widetilde{x})\}$$

△

Definition 4.1.5 Fuzzy intervals
A *fuzzy interval* $\widetilde{x}$ is a normalized fuzzy quantity with finite support and convex alpha level cuts, i.e.

$$\forall \alpha_1, \alpha_2 \in \mathbb{R}, \alpha_1 < \alpha_2 \;:\; L_{\alpha_1}(\widetilde{x}) \supset L_{\alpha_2}(\widetilde{x})$$

Futher, a *trapezoid fuzzy interval* is defined as a fuzzy interval of the form

$$(x_1, \widehat{x_2, x_3}, x_4),\; x_1, x_2, x_3, x_4 \in \mathbb{R}$$

with

$$\mu_{(x_1,\widehat{x_2,x_3},x_4)}(r) = \begin{cases} 1 & \text{if } x_2 \leq r \leq x_3 \\ \frac{r-x_1}{x_2-x_1} & \text{if } x_1 < r < x_2 \\ \frac{x_4-r}{x_4-x_3} & \text{if } x_3 < r < x_4 \\ 0 & \text{otherwise} \end{cases}, \; r \in \mathbb{R}$$

△

Definition 4.1.6 Fuzzy numbers
A *fuzzy number* $\widetilde{x}$ is defined as a fuzzy interval for which there exists exactly one modal element:

$$|\{x \in \mathbb{R} \;:\; \mu_{\widetilde{x}}(x) = 1\}| = 1$$

A *triangular (shaped) fuzzy number* $(\widehat{x, y, z}), x, y, z \in \mathbb{R}$ is a trapezoid fuzzy interval with one modal value:

$$(\widehat{x, y, z}) := (x, \widehat{y, y}, z)$$

Finally, for a numeral num, $\widetilde{num}$ denotes the fuzzy quantity with

$$\mu_{\widetilde{num}}(x) = \begin{cases} 1 & \text{if } x = num \\ 0 & \text{otherwise} \end{cases}, \; x \in \mathbb{R}$$

△

For fuzzy quantities, arithmetic operations are defined following the *extension principle*, which provides a general means to transform functions on real numbers to functions on fuzzy quantities(Zadeh, 1965):

Definition 4.1.7 Extension principle
Let $\widetilde{x} \in \widetilde{\mathbb{R}}^n, n \in \mathbb{N}$. The function $\widetilde{f} : \widetilde{\mathbb{R}}^n \mapsto \widetilde{\mathbb{R}}$ is called a *fuzzy extension* of a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ iff $\forall x \in \mathbb{R}^n$:

$$\mu_{\widetilde{f}(\widetilde{x})}(x) = \begin{cases} \sup_{y \in \mathbb{R}^n}\{\min_{1 \leq i \leq n}\{\mu_{\widetilde{x}_i}(y_i)\} \mid f(y) = x)\} & \text{if } f^{-1}(x) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

△

Based on the extension principle, we are able to define some specific fuzzy arithmetic operations that we will require.

Definition 4.1.8 Fuzzy arithmetics
Let $\widetilde{x}, \widetilde{y} \in \widetilde{\mathbb{R}}, \; x, y, z, a \in \mathbb{R}$.

$$\begin{aligned} \mu_{\widetilde{x} \oplus \widetilde{y}}(x) &:= \sup\{\min(\mu_{\widetilde{x}}(y), \mu_{\widetilde{y}}(z)) \mid y + z = x\} \\ \mu_{-\widetilde{x}}(x) &:= \mu_{\widetilde{x}}(-x) \\ \mu_{\widetilde{x} \ominus \widetilde{y}}(x) &:= \mu_{\widetilde{x} \oplus (-\widetilde{y})}(x) \\ \mu_{a \cdot \widetilde{x}}(x) &:= \begin{cases} \mu_{\widetilde{x}}(x/a) & \text{if } a \neq 0 \\ \mu_{\widetilde{0}}(x) & \text{if } a = 0 \end{cases} \end{aligned}$$

We also use the symbol $\widetilde{\sum}$ for the addition of a set of fuzzy quantities analogously to the symbol $\sum$ for crisp numbers. △

When agents negotiate coalitions, they need to compare different fuzzy utilities and choose one. Thus, we need a *ranking method* for fuzzy quantities:

Definition 4.1.9 Fuzzy ranking operators and similarity relations
Let $\widetilde{x}, \widetilde{y} \in \widetilde{\mathbb{R}}$ and R be a fuzzy subset of $\widetilde{\mathbb{R}} \times \widetilde{\mathbb{R}}$. R is called a *fuzzy ranking operator* if $\mu_R(\widetilde{x}, \widetilde{y})$ denotes the degree to which $\widetilde{x}$ can be considered greater compared to $\widetilde{y}$. R is called a *fuzzy similarity relation* if $\mu_R(\widetilde{x}, \widetilde{y})$ denotes the degree to which $\widetilde{x}$ can be considered "similar" to $\widetilde{y}$. Further let G be a fuzzy ranking operator and S a fuzzy similarity relation. We define

$$(\widetilde{x} \widetilde{\geq}_G \widetilde{y}) \;:=\; \mu_G(\widetilde{x}, \widetilde{y})$$

and

$$(\widetilde{x} \widetilde{=}_S \widetilde{y}) \;:=\; \mu_S(\widetilde{x}, \widetilde{y})$$

△

Several of such methods have been proposed in the literature, for example by Bortolan and Degani (1985). Dubois and Prade (1983) introduced four fuzzy ranking operators and two similarity relations which are applicable in the setting of *possibility theory*, which is our employed interpretation of fuzziness in the BSCA-F algorithm.

Definition 4.1.10 Possibility distribution
Let $\widetilde{s}$ be a fuzzy subset of a set S. If $\mu_{\widetilde{s}}(x)$ models the *degree of possibility*[1] that a variable X in the domain S takes the value x, denoted $\Pi(X = x)$, then $\widetilde{s}$ is also called a possibility distribution for X. △

Definition 4.1.11 Possibilistic ranking operators
Let $\widetilde{x}, \widetilde{y} \in \widetilde{\mathbb{R}}$ be possibility distributions for variables $X, Y \in \mathbb{R}$, respectively. We define

1. the *possibility of dominance* $\widetilde{\geq}_P$ of X over Y as

$$\begin{aligned}\Pi(X \geq Y) &= \widetilde{x} \widetilde{\geq}_P \widetilde{y} \\ &= \sup\{\min(\mu_{\widetilde{x}}(x), \mu_{\widetilde{y}}(y)) \mid x, y \in \mathbb{R}, x \geq y\}\end{aligned}$$

[1] the meaning of “degree of possibility” is left open here, but see also section 3.1

2. the *necessity of dominance* $\widetilde{\geq}_N$ of X over Y as

$$
\begin{aligned}
N(X \geq Y) &= \widetilde{x}\widetilde{\geq}_N\widetilde{y} \\
&= \inf_x\{\sup_y\{\max(1 - \mu_{\widetilde{x}}(x), \mu_{\widetilde{y}}(y)) \mid x, y \in \mathbb{R}, x \geq y\}\}
\end{aligned}
$$

3. the *possibility of strict dominance* $\widetilde{>}_P$ of X over Y as

$$
\begin{aligned}
\Pi(X > Y) &= \widetilde{x} >_P \widetilde{y} \\
&= \sup_x\{\inf_y\{\min(\mu_{\widetilde{x}}(x), 1 - \mu_{\widetilde{y}}(y)) \mid x, y \in \mathbb{R}, x \leq y\}\}
\end{aligned}
$$

4. the *necessity of strict dominance* $\widetilde{>}_N$ of X over Y as

$$
\begin{aligned}
N(X > Y) &= \widetilde{x} >_N \widetilde{y} \\
&= \inf\{\max(1 - \mu_{\widetilde{x}}(x), 1 - \mu_{\widetilde{y}}(y)) \mid x, y \in \mathbb{R}, x \leq y\}
\end{aligned}
$$

5. the *possibility of equality* $\widetilde{=}_P$ of X and Y as

$$
\begin{aligned}
\Pi(X = Y) &= \widetilde{x}\widetilde{=}_P\widetilde{y} \\
&= \min(\Pi(X \geq Y), \Pi(Y \geq X))
\end{aligned}
$$

6. the *necessity of equality* $\widetilde{=}_N$ of X and Y as

$$
\begin{aligned}
N(X = Y) &= \widetilde{x}\widetilde{=}_N\widetilde{y} \\
&= \min(\\
&\qquad \min(N(Y \geq X), 1 - \Pi(Y > X)), \\
&\qquad \min(N(X \geq Y), 1 - \Pi(X > Y)))
\end{aligned}
$$

$\triangle$

Further we define a fuzzy set of maximal elements of a set of fuzzy quantities X, where the membership of an element of X is defined by pairwise comparions with all other members:

Definition 4.1.12 Maximum of fuzzy quantities
Let $\widetilde{X}$ be a set of fuzzy quantities and G a fuzzy ranking operator. The fuzzy subset $\widetilde{\max}^G\widetilde{X}$ of $\widetilde{X}$ is given by

$$
\forall \widetilde{x} \in \widetilde{X} : \ \mu_{\widetilde{\max}^G X}(\widetilde{x}) := \min_{\widetilde{y} \in \widetilde{X}, \widetilde{y} \neq \widetilde{x}} (\widetilde{x}\widetilde{\geq}_G\widetilde{y})
$$

$\triangle$

Thus, $\mu_{\widetilde{\max}^G \widetilde{X}}(\widetilde{x})$ denotes the degree to which $\widetilde{x}$ can be considered a maximal element of $\widetilde{X}$. We also define a crisp set of maximal elements of $\widetilde{X}$ containing those elements with maximal membership in $\widetilde{\max}^G \widetilde{X}$:

Definition 4.1.13 Crisp maximum set
Let $\widetilde{X}$ be a set of fuzzy quantities and G a fuzzy ranking operator. The (crisp) set $\max^G \widetilde{X}$ of maximal elements of $\widetilde{X}$ is defined as the set of modal members of $\widetilde{\max}^G \widetilde{X}$. △

Finally, we will need the logical operations "AND" and "OR" with operands $\in [0, 1]$.

Definition 4.1.14 Possibilistic mean value
Given a fuzzy interval $\widetilde{x} \in \widetilde{\mathbb{R}}$,with $\mu_{\widetilde{X}}$ representing a possibility distribution for a variable $X \in \mathbb{R}$,

$$E(X) := \int_0^1 \alpha(\alpha\min(\widetilde{x}) + \alpha\max(\widetilde{x}))d\alpha$$

is called the *possibilistic mean value* of X. Instead of $E(X)$, we also write $e(\widetilde{x})$. Note that it is additive: $e(\widetilde{x} \oplus \widetilde{y}) = e(\widetilde{x}) + e(\widetilde{y})$, $\widetilde{x}, \widetilde{y} \in \widetilde{\mathbb{R}}$ The possibilistic mean value is sometimes also called the *expected value* of the respective variable due to the similarity of the definition to the expectation in probability theory. △

Fullér and Majlender (2003) also introduced a weighted version which allows for adjustments of the importance of different possibility levels. We use the unweighted version here for simplicity. Since e maps fuzzy membership functions to crisp real values, it can be seen as a suitable defuzzification method for fuzzy quantities in the setting of possibility theory.

Remark 4.1.15. *For any trapezoid fuzzy interval* $\widetilde{I} = (x_1, \widehat{x_2, x_3}, x_4)$, x_1, x_2, x_3, $x_4 \in \mathbb{R}$,

$$e(\widetilde{I}) = \frac{x_1 + x_2 + x_3 + x_4}{4} = \frac{\frac{x_2+x_3}{2} + \frac{x_1+x_4}{2}}{2}$$

This form makes clear that for trapezoid fuzzy intervals, e *is the real value in* $\widetilde{I}$ *minimizing the average distance to the bounds of the most possible values* $[x_2, x_3]$ *of* $\widetilde{I}$ *and the bounds of the support* (x_1, x_4)*, i.e. the values that are possible at all. In this sense,* e *can also be considered as a possibilistic error minimizing defuzzification method.*

Fuzzy Cooperative Games

Combining the theories of cooperative games and fuzzy quantities, Mareš (2001) introduced *fuzzy cooperative games*, where coalitions have fuzzy values. Fuzzy versions of the core and Shapley value are therefore proposed. Here, we consider only the fuzzy Shapley value, as we will use it as a basis for our coalition formation algorithm for fuzzy cooperative games.

Definition 4.1.16 Fuzzy coooperative game
A *fuzzy cooperative game* in characteristic function form, also called *fuzzy-valued game*, is a pair $(\mathcal{A}, \widetilde{v})$ with the set of agents $\mathcal{A}$ and the fuzzy characteristic function

$$\widetilde{v} : 2^{\mathcal{A}} \mapsto \widetilde{\mathbb{R}}$$

$\widetilde{v}(C)$ is called the fuzzy value of the coalition C and represents a possibility distribution for the *real coalition value* $v(C) \in \mathbb{R}$. △

In the remainder of this chapter, we also say just "fuzzy game" instead of "fuzzy cooperative game". The real coalition values exhibit the same meaning as the coaltion values in crisp coalition games, but are generally unknown in fuzzy games.

Definition 4.1.17 Fuzzy configuration
A *fuzzy configuration* is a pair $(\mathcal{C}, \widetilde{u})$ with the (crisp) coalition structure $\mathcal{C}$ and the fuzzy payoff distribution $\widetilde{u} \ : \mathcal{A} \mapsto \widetilde{\mathbb{R}}$. With a fuzzy similarity relation $\widetilde{=}, \widetilde{u}$ is called $\widetilde{=}$- efficient to a degree of

$$\mu_{eff^{\widetilde{=}}}(\widetilde{u}) \ := \ \min_{C \in \mathcal{C}} \left\{ \widetilde{\sum_{a_i \in C}} \widetilde{u}(a_i) \widetilde{=} \widetilde{v}(C) \right\}$$

For $a \in \mathcal{A}$ and a fuzzy ranking operator $\widetilde{\geq}$ the degree of *individual* $\widetilde{\geq}$*-rationality* in $\widetilde{u}$ is defined as

$$\mu_{indrat^{\widetilde{\geq}}}(a) := \widetilde{u}(a) \widetilde{\geq} \widetilde{v}(a)$$

and the degree of overall *individual* $\widetilde{\geq}$*-rationality* of $\widetilde{u}$ is defined as

$$\mu_{indrat^{\widetilde{\geq}}}(\widetilde{u}) := \ \min_{a \in A} \left\{ \mu_{indrat^{\widetilde{\geq}}}(a) \right\}$$

△

Definition 4.1.18 Fuzzy shapley value
The fuzzy Shapley value $\widetilde{\sigma}(a)$ of agent $a \in \mathcal{A}$ in a fuzzy game $(\mathcal{A}, \widetilde{v})$ is defined as

$$\widetilde{\sigma}(a) = \widetilde{\sum_{C \subseteq \mathcal{A}}} \frac{(|\mathcal{A}| - |C|)!(|C| - 1)!}{|\mathcal{A}|!} (\widetilde{v}(C) \ominus \widetilde{v}(C \setminus \{a\}))$$

△

Remark 4.1.19. *Mareš (2001) also showed that a fuzzy Shapley value stable payoff distribution is $\widetilde{\geq}_P$-rational as well as $\widetilde{=}_P$-efficient with degree* 1 *if the coalition values are normalized fuzzy intervals.*

4.2 Fuzzy Bilateral Shapley-value

Since computing Shapley value-stable configurations is computationally intractable (see) The fuzzy bilateral Shapley value is accordingly defined.

Definition 4.2.1 Fuzzy bilateral Shapley value
Let a fuzzy game $(\mathcal{A}, \widetilde{v})$. The fuzzy bilateral Shapley value $\widetilde{\sigma_b}(C_1 \cup C_2, C_i, v)$,$C_i, i \in \{1, 2\}$ in the bilateral coalition $C_1 \cup C_2$ is defined as the fuzzy Shapley value of C_i in the game $(\{C_1, C_2\}, \widetilde{v})$.

$$\widetilde{\sigma_b}(C_i, \widetilde{v}) := \frac{1}{2} \cdot \widetilde{v}(C_i) \oplus \frac{1}{2} \cdot (\widetilde{v}(C_1 \cup C_2) \ominus \widetilde{v}(C_k)),\ k \in \{1, 2\}, k \neq i$$

△

By this procedure, all the uncertainty in $\widetilde{v}(C_1)$, $\widetilde{v}(C_2)$ and $\widetilde{v}(C_1 \cup C_2)$ is carried into the fuzzy bilateral shapley values. While this is fine from a theoretical point of view, it can be problematic at the time a coalition C_1 has to decide whether to merge with a coalition C_2. That is because when negotiations are finished and the agents carry out their tasks, crisp (side-) payments will have to be made, i.e. the fuzzy payoff distribution must be defuzzified. When e.g. the coalition $C_1 \cup C_2$ is an element of the final coalition structure of the negotiations, it might be possible to determine the real coalition value $v(C_1 \cup C_2)$, because it is possible that the actual costs and rewards are known after the agents carried out their actions. However, the real coalition values $v(C_1)$ and $v(C_2)$ will in general remain unkown and need to be defuzzified. As we mentioned earlier, the possibilistic mean value is an appropriate defuzzifier for fuzzy quantities in the setting of possibility theory, and thus in particular for our fuzzy coalition values. Thus, we will use a modified fuzzy bilateral shapley value for which the coalition values of the subcoalitions are defuzzified by the possibilistic mean. This implies that the fuzziness of the resulting payoffs is caused only by the fuzziness of the joint coalition value.

Definition 4.2.2 Fuzzy bilateral Shapley value with defuzzified values of subcoalitions
Let there be a fuzzy game $(\mathcal{A}, \widetilde{v})$. The *fuzzy bilateral Shapley value with defuzzified values of subcoalitions* $\widetilde{\sigma}_b^e(C_1 \cup C_2, C_i, v)$,$C_i, i \in \{1,2\}$ in the bilateral coalition $C_1 \cup C_2$ is defined as the fuzzy Shapley value of $C_i$in the game$(\{C_1, C_2\}, \widetilde{v})$.

$$\widetilde{\sigma}_b^e(C_i, \widetilde{v}) := \frac{1}{2} \cdot e(\widetilde{v}(C_i)) \oplus \frac{1}{2} \cdot (\widetilde{v}(C_1 \cup C_2) \ominus e(\widetilde{v}(C_k))),\ k \in \{1,2\}, k \neq i$$

△

In the following, when we just say "fuzzy bilateral Shapley value", we always refer to the latter definition. Similarly to the crisp case of definition 2.2.21, we define a recursive fuzzy payoff distribution:

Definition 4.2.3 Recursive fuzzy bilateral Shapley value
Given a recursively bilateral coalition C for a fuzzy game $(\mathcal{A}, \widetilde{v})$ a fuzzy payoff distribution $\widetilde{u}$ is called *recursively fuzzy bilateral Shapley value stable* iff for every non-leaf node

$$C^* \in T_C : u(C_i^*) = \widetilde{\sigma}_b^e(C^*, C_i^*, \widetilde{v}_{C^*}), i \in 1,2, \text{ with}$$

$$\forall C^{**} \subseteq \mathcal{A} : \widetilde{v}_{C^*}(C^{**}) = \begin{cases} \widetilde{\sigma}_b^e(C^p, C_k^p, \widetilde{v}_{C^p}) & \text{if } C^p \in T_C, C^* = C_k^p, k \in 1,2 \\ \widetilde{v}(C^{**}) & \text{otherwise} \end{cases}$$

△

In order to be able to actually utilize the above concepts, computational agents need a feasible means to compute the coalition values. We therefore introduce the fuzzy *local worth* of an agent:

Definition 4.2.4 Fuzzy local worth and coalition value
In a fuzzy game $(\mathcal{A}, \widetilde{v})$, $\forall C \subseteq \mathcal{A}$ and $a \in C$, the *fuzzy local worth* is defined analogously to its crisp counterpart of definition 2.2.27. That is, with fuzzy valuation and cost functions $\widetilde{w}_a$ and $\widetilde{c}_a$, respectively,

$$\widetilde{lw}_a(C) := \widetilde{\sum_{g \in G_a(C)}} \widetilde{w}_a(g) \ominus \widetilde{\sum_{\tau \in \mathcal{T}_a(C)}} \widetilde{c}_a(\tau)$$

We also write $\widetilde{lw}_C(C^*) := \widetilde{\sum}_{a\in C} \widetilde{lw}(C^*)$ with $C^* \subseteq \mathcal{A}, C \subseteq C^*$.

Finally, also analogously to the crisp case, the coalition value is the fuzzy sum of the fuzzy local worths:

$$\widetilde{v}(C) = \widetilde{\sum_{a\in C}} \widetilde{lw}_a(C)$$

△

4.3 Algorithm BSCA-F

In this section, we present the algorithm BSCA-F for the formation of coalitions with fuzzy coalition values. It is based on the algorithm BSCA (Klusch, 1997) for the formation of bilateral Shapley value stable for crisp games. Before we present the exact definition, we first provide an informal outline.

A negoatiation with the BSCA-F consists of several negotiation rounds in which new coalitions are formed. Every coalition is represented by one of its members. These agents do not have any priveledges, but are responsible for certain computations and communications. In one round, each coalition representative

1. determines the values of bilateral coalitions resulting from a merge with each other coalition;
2. identifies the most profitable merging options;
3. chooses one of these options and sends a proposal to the other coalition;
4. if the other coalition also sent a proposal, merge with it; otherwise, try to find another coalition to merge with (i.e. go to step 3);
5. if no new coalitions have been formed in this round, stop; otherwise, go to step 1.

With fuzzy coalition and Shapley values, the choice of a "most profitable" option becomes ambiguous as opposed to the crisp case. To achieve unambiguous coalition choices, the agents have to apply the same fuzzy ranking method to compare the fuzzy increase in profit for different coalitions. This prevents the agents' to implement a truly individual strategy: the chosen ranking operator determines whether rather optimistic or pessimistic coalition choices are made (this is demonstrated in section 4.5). But allowing such individual strategies requires not only a more complex proposal protocol. Also additional negotiations within coalitions in order to agree on a shared strategy are required. Because the main focus of this work is to provide an efficient coalition formation algorithm, we do not follow this line.

Now, if a coalition C assesses the option to form the coalition $C \cup C^*$ by means of the possible utility gain $\widetilde{\sigma}_b^e(C \cup C^*, C, \widetilde{v}) - \widetilde{v}(C)$, this value is different from the possible utility gain of C^* (assuming the coalition values are indeed fuzzy):

$$\widetilde{\sigma}_b^e(C \cup C^*, C, \widetilde{v}) - \widetilde{v}(C) \neq \widetilde{\sigma}_b^e(C \cup C^*, C^*, \widetilde{v}) - \widetilde{v}(C^*).$$

It might thus happen that there exists no global most profitable choice in this sense, and coalitions fail to agree to merge. We therefore base this decision on the *expected utility gain* of a bilateral coalition instead, which we show to be equal for both subcoalitions:

Definition 4.3.1 Expected utility gain
For a fuzzy game $(\mathcal{A}, \widetilde{v})$, the bilateral Shapley value based expected utility gain of a subcoalition C in the coalition $C \cup C^*, C, C^* \subset \mathcal{A}$ is

$$\widetilde{g}_{\widetilde{v}}(C, C \cup C^*) := \widetilde{\sigma}_b^e(C \cup C^*, C, \widetilde{v}) - e(\widetilde{v}(C))$$

△

Lemma 4.3.2. *Let there be a fuzzy game* $(\mathcal{A}, \widetilde{v})$ *and* $C_1, C_2 \subset \mathcal{A}$. *Then* $\widetilde{g}_{\widetilde{v}}(C_1, C_1 \cup C_2) = \widetilde{g}_{\widetilde{v}}(C_2, C_1 \cup C_2)$

Proof. By definitions 4.3.1 and 4.2.2, and because of the properties of $\oplus$ and $\ominus$ when applied to at least one crisp operand discussed e.g. in Dubois and Prade (1994), we rewrite

$$\begin{aligned}
\widetilde{g}_{\widetilde{v}}(C_1, C_1 \cup C_2) &= \frac{1}{2} \cdot e(\widetilde{v}(C_1)) \oplus \frac{1}{2} \cdot (\widetilde{v}(C_1 \cup C_2) \ominus e(\widetilde{v}(C_2))) \ominus e(\widetilde{v}(C_1)) \\
&= \frac{1}{2} \cdot \widetilde{v}(C_1 \cup C_2) \ominus \frac{1}{2} e(\widetilde{v}(C_2)) \ominus \frac{1}{2} e(\widetilde{v}(C_1)) \\
&= \frac{1}{2} \cdot \widetilde{v}(C_1 \cup C_2) \ominus \frac{1}{2} e(\widetilde{v}(C_1)) \ominus \frac{1}{2} e(\widetilde{v}(C_2)) \\
&= \widetilde{g}_{\widetilde{v}}(C_2, C_1 \cup C_2)
\end{aligned}$$

□

Finally, for cases with multiple coalitions with maximum expected utility gain, and also to choose coalition representatives in a simple way, we utilize an agent ordering function. This could, for example, be based on the agents' available computational resources.

Definition 4.3.3 Agent ordering function
Given an agent set $\mathcal{A}$, $o : \mathcal{A} \mapsto \mathbb{N}$, with $\forall a, a^* \in \mathcal{A}, a \neq a^* : o(a) \neq o(a^*)$ is called an agent ordering function. $\triangle$

Algorithm 4.3.4 BSCA-F
In the following, let $\mathcal{A}$, $(\mathcal{C}_0, \widetilde{u}_0)$ with $\forall C \in \mathcal{C}_0 : C = a, a \in \mathcal{A}$, $\widetilde{u}_0(a) = \widetilde{v}(a)$ and an agent ordering function o. Further, let a fuzzy ranking operator $\widetilde{\circ} \in \{\widetilde{\geq}_P, \widetilde{\geq}_N, \widetilde{>}_P, \widetilde{>}_N\}$ and a ranking threshold t. With the variables $r := 1$ and $\forall C, a, C = \{a\} : Rep_C := a$, each agent a performs: Let $C \in \mathcal{C}_r$ such that $a \in C$;

1. *Communication:* If $a \neq Rep_C$then go to step 3e; otherwise do for all $C^* \in \mathcal{C}_r, C^* \neq C$:
   (a) send $\widetilde{lw}_C(C \cup C^*)$ to Rep_{C^*}
   (b) receive $\widetilde{lw}_{C^*}(C \cup C^*)$ from Rep_{C^*}
   (c) compute $\widetilde{v}(C \cup C^*) = \widetilde{lw}_C(C \cup C^*) \oplus \widetilde{lw}_{C^*}(C \cup C^*)$
2. *Proposal Generation*
   (a) set $Cand_C := \left\{ C^* \;\middle|\; C^* \in \mathcal{C} \setminus C,\ (\widetilde{g}(C, C \cup C^*, \widetilde{v}) \widetilde{\circ} \widetilde{0}) \ \geq \ t \right\}$
   (b) if $Cand_C \neq \emptyset$, then set
   $$C^+ := \underset{C^* \in MaxCand_C}{\arg\min} \{o(Rep_{C^*})\} \text{ with}$$
   $$MaxCand_C = \left\{ C^* \;\middle|\; \widetilde{g}_{\widetilde{v}}(C, C \cup C^*) \in \underset{C^{**} \in Cand_C}{\arg\overset{\widetilde{\circ}}{\max}} \{\widetilde{g}_{\widetilde{v}}(C, C \cup C^{**})\} \right\}$$
   else set $C^+ := nil$
   (c) if $C^+ \neq nil$ then send a proposal to form $C \cup C^+$ to Rep_{C^+}
   (d) receive all proposals
3. Coalition Forming
   (a) set $New := \emptyset$ and $Obs := \emptyset$
   (b) if $C^+ \neq nil$ and a proposal was received from Rep_{C^+} , form the coalition $C \cup C^+$:

i. if $o(Rep_C) < o(Rep_{C^+})$ then set $Rep_{C \cup C^+} := Rep_C$; else set $Rep_{C \cup C^+} := Rep_{C^+}$

ii. inform all other Rep_{C^*}, $C^* \in \mathcal{C}_r, C^* \neq C, C^* \neq C^+$ and all $a^* \in C$, $a \neq Rep_C$ about the new coalition and $Rep_{C \cup C^+}$

iii. $New := \{C \cup C^+\}$, $Obs := \{C, C^+\}$, $Cand_C := \emptyset$

(c) receive all messages about each new coalition. For each new coalition $C_1 \cup C_2$ and $Rep_{C_1 \cup C_2}$, set $Cand_C := Cand_C \setminus \{C_1, C_2\}$, $New := New \cup \{C_1 \cup C_2\}$and $Obs := Obs \cup \{C_1, C_2\}$.

(d) send the sets New and Obs to all other coalition members $a^* \in C$, $a \neq Rep_C$

(e) if $a \neq Rep_C$ then receive the sets New and Obs from Rep_C.

(f) set $r := r + 1$, $\mathcal{C}_r := (\mathcal{C}_{r-1} \setminus Obs) \cup New$, and u_r according to the recursive fuzzy bilateral Shapley value based on the coalition structures $\mathcal{C}_r \ldots \mathcal{C}_0$.

(g) if $C_r = C_{r-1}$then stop; else go to step 1

$\triangle$

Remark 4.3.5. *In steps 2d and 3c, not all other representatives will have sent proposal or new coalition messages, respectively, so either a timeout or some other synchronisation method must be used here. This has been omitted here for clarity. However, the flow chart of the BSCA-F shown in figure 4.1 includes synchronisation steps by sending "no proposal" and "no new coalition" messages, respectively.*

Remark 4.3.6. *In the conference paper version (Blankenburg and Klusch, 2005b), there was a loop between steps 2c and the place just before step 3d to negotiate also with the rest of the candidates for the round if the first choice merged with another coalition. However, the newly formed coalitions might be better candidates than the rest of the candidates for the round. Therefore, we removed this loop. This also has the effect that the degree of the polynomial runtime complexity is reduced by one (the runtime complexity is stated in theorem 4.4.3 below).*

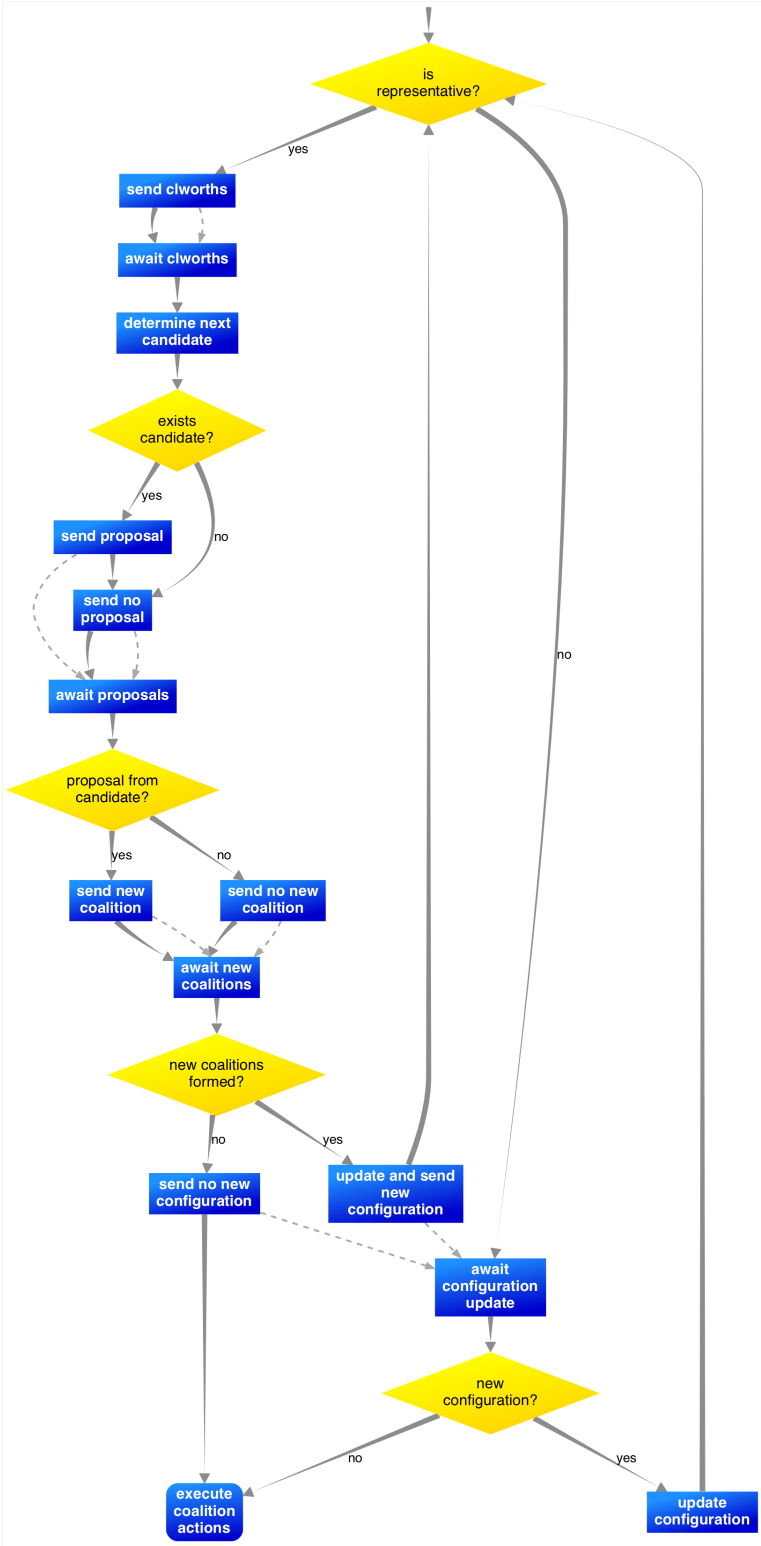

Figure 4.1: Coalition formation algorithm BSCA-F.

4.4 BSCA-F Properties

Proposition 4.4.1. *In any round $r \in \mathbb{N}$, the coalition $C_1 \cup C_2$, $C_1, C_2 \in \mathcal{C}_r$, which is (a) among the overall most profitable coalitions in the sense that*

$$\widetilde{g_{\widetilde{v}}}(C_1, C_1 \cup C_2) \in \overset{\widetilde{\circ}}{\max} \{\widetilde{g_{\widetilde{v}}}(C, C \cup C^{**}) \mid C \in \mathcal{C}_r, C^{**} \in Cand_C\}$$

, and (b) $o(Rep_{C_1 \cup C_2})$ is minimal as compared to o of other overall most profitable coalitions, is formed, or no proposals are sent at all.

Proof. Because of lemma 4.3.2, we have that if $C_1 \cup C_2$ is in the set $Cand_{C_1}$, it is also in the set $Cand_{C_2}$. From the properties of $\widetilde{\geq}_P, \widetilde{\geq}_N, \widetilde{>}_P$ and $\widetilde{>}_N$ discussed in Dubois and Prade (1983), it is clear that for a set of fuzzy quantities X, if $F_1 \in X$: $F_1 \in \max^G X$, then also $F_1 \in \max^G Y \subseteq X$ with $F_1 \in Y$. Further, $\widetilde{g_{\widetilde{v}}}(C_1, C_1 \cup C_2) = \widetilde{g_{\widetilde{v}}}(C_2, C_1 \cup C_2)$ because of lemma 4.3.2. Thus, with (a) it follows that $\widetilde{g_{\widetilde{v}}}(C_1, C_1 \cup C_2) \in \max^{\widetilde{\circ}} \{\widetilde{g_{\widetilde{v}}}(C_i, C_i \cup C^{**}) \mid C^{**} \in Cand_{C_i}\}$ for both $i = 1$ and $i = 2$. With the unambiguousness of the agent ordering function o and (b), it is then clear that in step 2.b C_1 and C_2 send proposals to each other and thus form $C_1 \cup C_2$ in step 3.c. □

Lemma 4.4.2. *The BSCA-F terminates after at most $|\mathcal{A}|$ rounds.*

Proof. In non-terminating each round $r \in \mathbb{N}$ of the BSCA-F at least one new coalition is formed, i.e. $|\mathcal{C}|_{r+1} \leq |\mathcal{C}|_r - 1$. Thus, after $|\mathcal{A}| - 1$ rounds, we have $|\mathcal{C}|_{|\mathcal{A}|-1} \leq 1$, which means that the BSCA-F terminates in round $|\mathcal{A}|$. □

Theorem 4.4.3. *The worst-case runtime of the BSCA-F for each agent is in $O(|\mathcal{A}|^3)$ assuming constant time for operations on fuzzy quantities.*

Proof. In step 2b, each C has to find the (crisp) maximum set of the fuzzy gains for coalitions in $Cand_C$, with $|Cand_C| \leq |\mathcal{C}_r|$. From definitions 4.1.12 and 4.1.13 it follows that this can be done in $O(|\mathcal{C}_r|^2)$. All other individual operations are of less complexity. Since $|\mathcal{C}_r| \leq |\mathcal{A}|$ and lemma 4.4.2, the overall runtime of the BSCA-F is then $O(|\mathcal{A}|^3)$. □

Remark 4.4.4. *Note that the constant time assumption for operations on fuzzy quantities holds at least for the use of trapezoid fuzzy intervals, some parametric representations (see e.g. Giachetti and Young (1997)) and by using e.g. α-level-cut based approximations* *with a fixed set of cuts. Otherwise, the runtime complexity becomes $O(|\mathcal{A}|^3) \cdot O(Comp_{fuzzy})$, where $Comp_{fuzzy}$ denotes the max. complexity of a single operation on fuzzy quantities.*

Theorem 4.4.5. *The total number of messages sent is in $O(|\mathcal{A}|^2)$.*

Proof. In each round $r \in \mathbb{N}$, each representative of a coalition C sends $|\mathcal{C}_r| - 1$ messages in step 1a. A single proposal message is sent in step 2c; if agents are synchronized via also sending "no proposal" messages as shown in figure 4.1, at most $|\mathcal{C}_r|$ such messages are sent. At most one time $|\mathcal{C}_r| - 2$ messages are sent in step 3(b)ii; again, at most $|\mathcal{C}_r|$ "no new coalition" messages must be sent if synchronized this way. Finally, $|C| - 1$ messages are sent in step 3d. So the number of messages per representative per round is bound by $|\mathcal{C}| \leq |\mathcal{A}|$. The number of messages sent by the $|\mathcal{A}| - |\mathcal{C}|$ non-representatives is zero. So with lemma 4.4.2, the overall number of messages sent is lower or equal $|\mathcal{A}|^2$. □

When negotiations with the BSCA-F are finished, we have a final recursive bilateral coalition structure $\mathcal{C}$ with a recursively fuzzy bilateral Shapley value stable configuration. Remember that the fuzzyness in the fuzzy payoffs comes only from the fuzzy coalition values of the coalitions in $\mathcal{C}$. All other fuzzy coalition values which are used in the computation of the recursive bilateral Shapley values are defuzzified by means of the possibilistic mean value. Thus, in order to obtain crisp payoffs, one only has to also defuzzify the coalition values of coalitions in $\mathcal{C}$. As we mentioned earlier, it seems plausible for a number of applications that real coalition values become known to the resp. coalition members after coalitions have been formed and the according actions have been carried out. If that is not the case, we can also use the possibilistic mean to obtain at least a good expectation of what the real coalition value is. So in both cases, we now have crisp coalition values for all coalitions in $\mathcal{C}$, and thus obtain crisp payoffs.

4.5 Example Application

Definition of the Game

In this section we show how the BSCA-F can be applied to negotiate cooperation of online magazines. We consider a situation where

Category	M_1	M_2	M_3	M_4
a)	politics	column	column 1	photography mag
b)	feature section	travel mag	column 2	feature section

Table 4.1: Content provided by magazines

I_1	$\widetilde{A}^1_{\cdot}$	I_2	$\widetilde{A}^2_{\cdot}$
M_2 a)	$(2.4, \widehat{3.6, 4.2}, 4.8)$	M_1 a)	$(8.4, \widehat{12, 14.4}, 16.8)$
M_2 b)	$(1.08, \widehat{1.2, 1.56}, 1.8)$	M_3 a)	$(6.6, \widehat{7.2, 7.8}, 8.4)$
M_3 a)	$(9.6, \widehat{12, 14.4}, 15.6)$	M_4a)	$(6, \widehat{9, 9.333}, 10)$
M_4 a)	$(3.6, \widehat{7.2, 12.8}, 16.4)$		

I_3	$\widetilde{A}^3_{\cdot}$	I_4	$\widetilde{A}^4_{\cdot}$
M_2 b)	$(13.2, \widehat{14.4, 15}, 15.6)$	M_1 b)	$(3.6, \widehat{3.72, 4.08}, 4.2)$
		M_2 b)	$(7.2, \widehat{7.56, 10}, 12.5)$

Table 4.2: Additional income for categories of interest

four such magazines, $M_1 - M_4$, are interested in exchanging content to extend their customer bases. We assume that cooperation is realized via coalition forming. Two magazines cooperate iff they are in the same coalition. Also, magazines are reluctant to give out more content to other magazines then they receive in return, even though they receive appropriate payoffs. Thus, the magazines agree to contribute exactly two categories of their content to their coalition, regardless whether the contributed content is actually used or not. Contributions will be submitted electronically to each coalition partner on a daily basis, while coalition contracts hold for one year, after which negotiations might be re-initiated. To prevent antitrust matters, coalitions with more than three members are ruled out. Each magazine M_i is represented by an agent a_i which carries out negotiation on behalf of M_i. The magazines provide content in categories which are summarized in table 4.1. Each magazine M_i would publish only such content provided by coalition partners which is in line with the general style of M_i. The set of categories M_i is interested in is called I_i. For each $x \in I_i$, M_i fuzzily estimates the number $\widetilde{A}^i_x$ of additional accesses it can achieve by publishing x for one year. For simplicity, we assume these estimations are independent of each other. The sets I_i and estimations $\widetilde{A}^i_x$ (in thousands) are given in table 4.2.

	$V_{M.a)}$	$V_{M.b)}$
M_1	$(1200, \widehat{1800, 2400}, 4200)$	$(600, \widehat{1200, 3000}, 6000)$
M_2	$(120, \widehat{360, 600}, 720)$	$(9600, \widehat{12000, 15600}, 20400)$
M_3	$(360, \widehat{840, 1200}, 1440)$	$(240, \widehat{360, 420}, 540)$
M_4	$(18000, \widehat{19200, 20400}, 21600)$	$(1800, \widehat{2400, 3000}, 3600)$

Table 4.3: Amount of data per category in MB

Any single access to content of a magazine M_i, $1 \leq i \leq 4$, is subject to a given price P_i determined by M_i. The P_i are given as follows: $P_1 := EUR$ 2.0, $P_2 := EUR$ 1.5, $P_3 := EUR$ 1.8 and $P_4 := EUR$ 2.0. Thus, the additional income produced by a magazine M_i by coalescing with a magazine M_k is $\widetilde{\sum}_{M_k x) \in I_i} \widetilde{A}^i_{M_k x)} \odot P_i$ and the total additional income $\widetilde{ai}_i(C)$ for M_i in coalition C is given with

$$\widetilde{ai}_i(C) = \widetilde{\sum_{M_k \in C, k \neq i}} \widetilde{\sum_{M_k x) \in I_i}} \widetilde{A}^i_{M_k x)} \odot P_i$$

The inner sum therein corresponds to the goal valuation term in definition 4.2.4. M_1 and M_2 arguably have the best cooperation opportunities in this game.

On the cost side, we consider only transfer costs and assume they are volume-based with a given transfer price T_i per MB depending on the internet connection of magazine M_i. The T_i are given as follows: $T_1 := 0.02\, EUR/MB$, $T_2 := 0.01\, EUR/MB$, $T_3 := 0.025\, EUR/MB$ and $T_4 := 0.012\, EUR/MB$. Based on experiences in the past, each M_i fuzzily estimates the amount of data V_x it would transfer for each contributed categorie x during the one year as shown in table 4.3. Every magazine has to pay for incoming as well as outgoing traffic. Thus, every magazine M_i has to pay $(V_{M_i a)} \oplus V_{M_i b)} \oplus V_{M_k a)} \oplus V_{M_k b)}) \odot T_i$ for the data transmitted to/from each coalition partner M_k. Therefore, the total cost $\widetilde{c}_i(C)$ for M_i in coalition C is given with

$$\widetilde{c}_i(C) = \widetilde{\sum_{M_k \in C, k \neq i}} (V_{M_i a)} \oplus V_{M_i b)} \oplus V_{M_k a)} \oplus V_{M_k b)}) \odot T_i$$

Having both the total additional income and costs for every magazine M_i in any coalition C, we can now obtain the local worths: $\widetilde{lw_{a_i}}(C)$ $= \widetilde{ai}_i(C) \ominus \widetilde{c}_i(C)$ Table 4.4 shows the resulting coalition values.

C	$\widetilde{v}(C)$	C	$\widetilde{v}(C)$
a_1, a_2	$(38060, \widehat{48552, 60739}, 70454)$	a_1, a_3	$(18835, \widehat{23789, 28674}, 31128)$
a_1, a_4	$(13338, \widehat{20976, 33022}, 40552)$	a_2, a_3	$(32967, \widehat{36185, 38293}, 40370)$
a_2, a_4	$(19010, \widehat{22932, 32981}, 39114)$	a_3, a_4	$(-815, \widehat{-751, -684}, -612)$
a_1, a_2, a_3	$(89564, \widehat{108332, 127576}, 141865)$	a_1, a_2, a_4	$(69646, \widehat{91830, 126204}, 149649)$
a_1, a_3, a_4	$(30690, \widehat{43458, 60529}, 70642)$	a_2, a_3, a_4	$(50331, \widehat{57648, 69967}, 78328)$
others	$(\widehat{0})$		

Table 4.4: Coalition values (rounded)

Negotiation with the BSCA-F

As ranking operator we chose the necessity of dominace $\widetilde{\geq}_N$ which, as we will see, leads to rather conservative choices which configurations are preferred by a coalition. As agent ordering function we use $o(a_i) := i$.

In the first round in step 2a, all coalitions put each other into their resp. set of candidates, with the exception of $\{3, 4\}$ which is clearly a non-profitable coalition. As $\{1, 2\}$ is the most profitable coalition, a_1 and a_2 bilaterally propose this coalition. Their payoffs are both half of the coalition value, as all coalition values of single-agent coalitions are equally 0:

$$\begin{aligned}\widetilde{u}(a_1) = \widetilde{u}(a_2) &= \frac{1}{2}\widetilde{0} \oplus \frac{1}{2}((38060, \widehat{48552, 60739}, 70454) \ominus \widetilde{0}) \\ &= (19030, \widehat{24276, 30369.5}, 35227)\end{aligned}$$

Because a_3 and a_4 are no candidates for each other, nothing more happens in the first round.

In the second round, for $\{1, 2\}$ both a_3 and a_4 are candidates. A proposal is sent to a_3 because the gain for $\{1, 2\}$ in this coalition is greater than the one in a coalition with a_4 according to $\widetilde{\geq}_N$. This is because $\widetilde{\geq}_N$ considers only the left sides of fuzzy interval (in the follwing, all real values are rounded): with
$e(\widetilde{v}(\{a_1, a_2\}) = \frac{38060+48552+60739+70454}{4} = 54451$, we have

$$\begin{aligned}\widetilde{g}_{\widetilde{v}}(\{1, 2\}, (\{1, 2\}) \cup \{3\}) &= \frac{1}{2}(\widetilde{v}(\{a_1, a_2, a_3\}) \ominus e(\widetilde{v}(\{a_1, a_2\}) \ominus e(\widetilde{v}(\{a_3\})) \\ &= \frac{1}{2}((89564, \widehat{108332, 127576}, 141865) \ominus 54451 \ominus 0) \\ &= (17556, \widehat{26940, 36562}, 43706)\end{aligned}$$

Because of lemma 4.3.2, we further have $= \widetilde{g}_{\widetilde{v}}(a_3, (\{1,2\}) \cup \{3\})$. Similarily,

$$\begin{aligned}\widetilde{g}_{\widetilde{v}}(\{1,2\},(\{1,2\}) \cup \{4\}) &= \widetilde{g}_{\widetilde{v}}(a_4,(a_1 \cup a_2) \cup \{4\}) \\ &= (7597, 186\widehat{89, 358}76, 47599)\end{aligned}$$

Please note that with $>_P$, the choice would have been a_4 because the expected utility gain is slightly better on the right side. In this sense, the choice of $>_P$ can indeed be viewed as a more optimistic approach. As $\{1, 2\}$ is the only candidate for both a_3 and a_4, coalition $\{1, 2, 3\}$ is formed and the round is completed. The payoff of the new coalition is distributed as follows:

$$\begin{aligned}\widetilde{u}(a_1) = \widetilde{u}(a_2) &= \widetilde{\sigma}^e_b(\{1,2\},\, a_1,\, \widetilde{\sigma}^e_b(C^*,\, \{1,2\},\, \widetilde{v}(C^*))) \\ &= \frac{1}{2}(\frac{1}{2}e(\widetilde{v}(\{1,2\})) \oplus \frac{1}{2}((89564, 1083\widehat{32, 12}7576, 141865) \ominus 0)) \\ &= \frac{1}{2}(\frac{1}{2}54451 \oplus \frac{1}{2}((89564, 1083\widehat{32, 12}7576, 141865) \ominus 0)) \\ &= (36004, 406\widehat{96, 45}507, 49079)\end{aligned}$$

and

$$\begin{aligned}\widetilde{u}(a_3) &= \frac{1}{2}((89564, 1083\widehat{32, 12}7576, 141865) \ominus 54451) \\ &= (17556, 269\widehat{40, 36}562, 43706)\end{aligned}$$

In the third round, the BSCA-F terminates, since the value of the grand coalition is zero and so is not a candidate for anyone.

Defuzzification

For this particular example, in which cooperation contracts for one year are negotiated, the magazines now have two options to obtain crisp coalition values:

1. wait for one year and then analyze the additional income and the costs that were realized in the period;

2. defuzzify the coalition values just when negotiations are finished, using the possibilistic mean value. We first look at the first case.

In the following, let $C^* := \{1, 2, 3\}$ the single non one-agent coalition that was formed. In the first case, the fuzzy value $\widetilde{v}(C^*) = \widehat{(89564, 108332, 127576, 141865)}$ tells us that the real value $v(C^*)$ is certainly in $(89564, 141865)$ and "most possibly" in $[108332, 127576]$. As indicated in section 4.1, the fuzziness of the fuzzy payoffs obtained by applying the recursive bilateral Shapley value comes only from the fuzziness of the values of the coalitions in the final configuration. This means that, shown for the example of a_1 (resp. M_1), the crisp payoffs that are obtained for each of these bounding values of $v(C^*)$ are:

1. For $v(C^*) = \inf\{support(\widetilde{v}(C^*))\} = 89564$, we have

$$\begin{aligned} u(a_1) &= \sigma_b(\{1,2\},\, a_1, \sigma_b(C^*,\, \{1,2\},\, 89564)) \\ &= \sigma_b(\{1,2\},\, a_1,\, \frac{54451 + 89564 - 0}{2}) \\ &= \sigma_b(\{1,2\},\, a_1,\, 72008) \\ &= 36004 \\ &= \inf\{support(\widetilde{u}(a_1))\}. \end{aligned}$$

2. For $v(C^*) = 108332$, the minimum modal value of $\widetilde{v}(C^*)$, we have

$$\begin{aligned} u(a_1) &= \sigma_b(\{1,2\},\, a_1,\, \sigma_b(C^*,\, \{1,2\},\, 108332)) \\ &= \sigma_b(\{1,2\},\, a_1,\, 81392) \\ &= 40696, \text{ the minimum modal value of } \widetilde{u}(a_1) \end{aligned}$$

3. For $v(C^*)$ taking the maximum modal value or the supremum of the support of $\widetilde{v}(C^*)$, we similarly get the maximum modal value or the supremum, resp., of the support of $\widetilde{u}(a_1)$ for $u(a_1)$.

4. For the above cases, the crisp payoffs of the other agents also take the resp. bounding values of their respective fuzzy payoffs.

This supports our impression that the recursively bilateral Shapley value stable payoffs using the partially defuzzified values for subcoalitions are "just as fuzzy as needed".

However, one year might be considered as too long a time to distribute payoffs in fast-paced internet businesses such as the publication of online magazines. Thus, we also consider the possibilistic mean value of

$$\widetilde{v}(C^*) : e(\widetilde{v}(C^*)) = \frac{89564 + 108332 + 127576 + 141865}{4} = 116834$$

Applying the recursive bilateral Shapley value to $e(\widetilde{v}(C^*))$, we obtain $u(a_1) = u(a_2) = 42821$ which turns out to be equal to $e(\widetilde{u}(a_1))$ $(= e(\widetilde{u}(a_2))$ due to the additiveness of e. Similarily, $u(a_3) = e(\widetilde{u}(a_3)) = 31192$. Thus, when negotiations are finished, one only has to compute the possibilistic mean values of the fuzzy payoffs to obtain the same result as if computing the recursively bilateral Shapley value stable payoffs for the possibilistic mean of the coalition values. The payoffs seem to be intuitively sound: a_1 and a_2, being the agents with the best cooperation opportunities, are assigned more payoff than a_3. Further, let us consider computing the fuzzy payoffs by recursively applying the non-defuzzyfying fuzzy bilateral Shapley value as defined in 4.2.1 instead. This means that the fuzzy payoffs now also contain the fuzzyness of the values of the subcoalitions. Then we obtain the fuzzy payoffs

$$\begin{aligned}\widetilde{u}^*(a_1) &= \widetilde{\sigma}_b(\{1,2\},\, a_1,\, \widetilde{\sigma}_b(C^*,\, \{1,2\},\, \widetilde{v}(C^*)))\\ &= (31906, \widehat{39221, 47079}, 53080)\ (= \widetilde{u}^*(a_2))\end{aligned}$$

and

$$\begin{aligned}\widetilde{u}^*(a_3) &= \widetilde{\sigma}_b(C^*,\, a_3,\, \widetilde{v}(C^*)))\\ &= (9555, \widehat{23797, 39512}, 51903)\end{aligned}$$

Again because of the additivity of e, it turns out that $e(\widetilde{u}^*(a_1)) = 42821 = e(\widetilde{u}(a_1))$, and accordingly $e(\widetilde{u}^*(a_2)) = e(\widetilde{u}(a_2))$ and $e(\widetilde{u}^*(a_3)) = e(\widetilde{u}(a_3))$. Thus, it can be said that by using $e(\widetilde{u})$ together with the partially defuzzified recursive bilateral Shapley values, not only is the possible error wrt. $v(C^*)$ possibilistically minimized in the sense of remark 4.1.15, but also the possible errors wrt. the values of the subcoalitions.

4.6 Evaluation

In this section, we present evaluation results that we obtained from simulating the BSCA-F with different fuzzy ranking operators and thresholds. The questions that we are interested in finding out by the simultation are:

- How do the average fuzzy payoffs generated by the BSCA-F compare to the fuzzy Shapley value payoffs, especially concerning guaranteed (i.e. minimum support) and possibilistic mean payoffs?
- How do the different possibilistic ranking operators compare to each other in these regards?
- To what degree are they BSCA-F payoffs correlated with the fuzzy Shapley value payoffs?
- What are the worst case behaviours of the different ranking operators, and how do they compare to the fuzzy Shapley value?
- Whith which degrees of possibility and necessity lie the fuzzy payoffs in the core?

We aim to get answers to these questions in a realistic service agent setting. To accomplish this, the games are randomly generated using a simple service model described in the following subsection. However, since we compute optimal service allocations and evaluate fuzzy Shapley value and core memberships, we have to restrict the number agent of agents to be rather small (5 agents) to obtain manageable runtime[2].

Simulation Setup

The simulation was run for 200 randomly generated service agent games with 5 agents. The games use a simple service agent model:

- Each agent may offer and request any number of services.
- There exists a number of globally known service composition plans. Each composition plan consists of a sequence of offered services.

[2]The simulation has been implemented in Clojure, a LISP dialect for the JVM. Each agent ran in its own thread, using a simple channel-based message system. For the simulated 5-agent games, each run of the BSAC-F took less than one second on a 4-core 2.2 GHz machine. However, the service allocation, fuzzy Shapley value and core memberships were computed brute-force, taking several seconds each

- Each plan satisfies exactly one request.
- Each offered service might only appear once in the same plan.
- Each offered service might be employed only once in coalitional plan execution.

Each game was then randomly generated as follows:

1. 10 service offers were randomly generated with random fuzzy cost with expected mean modal value between 100 and 1000.
2. 10 service requests were randomly generated with random fuzzy valuations with expected mean modal value between 100 and 1000.
3. 10 service composition plans with randomly drawn length of 2 or 3 and composed of randomly drawn offered services where generated.
4. All requests and offers were randomly but completely assigned to the agents.
5. For each coalition, an optimum service allocation was computed. For this purpose, the fuzzy net values of sets of plans allocatable in a coalition where compared using the same possibilistic ranking operator and threshold as the used by the respective BSCA-F invocation.
6. Coalition values where then determined via the agents' local worths resulting from their added fuzzy valuations and costs of satisfied and executed services, respectively, according to the optimum plan allocation.

Therein, the random assignments to agents where implemented using a uniform distribution over the assignable agents. The fuzzy costs and valuations where generated as follows: first, an expected mean value was drawn from a uniform distribution between 100 and 1000. Given this, four distinct random numbers from drawn from a normal distribution with mean 1 and standard deviation 0.25. The four numbers where then sorted and multiplied by the expected mean value, and the resulting numbers where then used to construct a trapezoid fuzzy interval, acting as minimum support, minimum modal value, maximum modal value and maximum support, respectively.

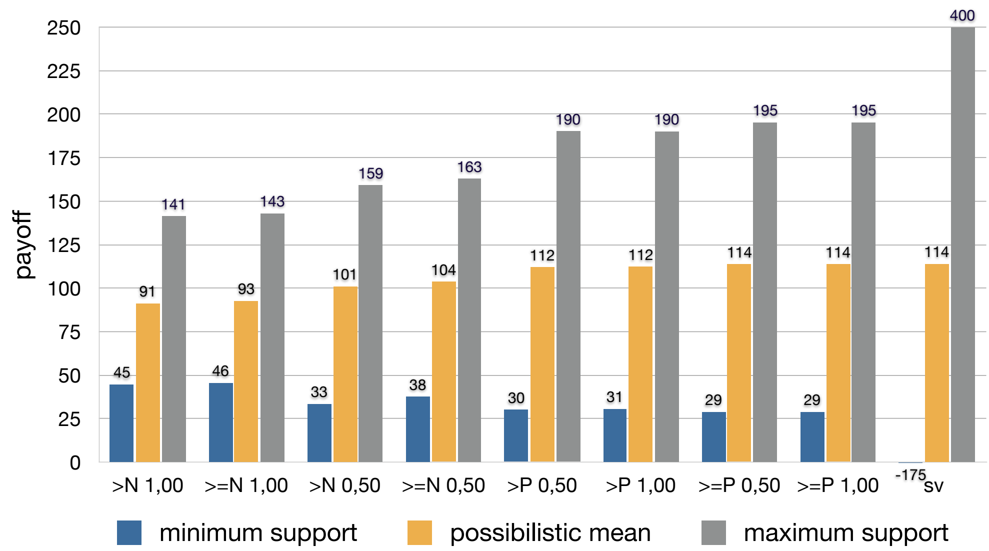

Figure 4.2: Mean payoff indicators

Note that the games generated in this are mostly superadditive. However, as it can happen with fuzzy games, a game might have high possibilities of both super- and subadditivity at the same time.

Simulation Results

The BSCA-F was run for the 200 generated games with each combination of the four possibilistic ranking operators (see 4.1.11) and thresholds 0.5 and 1.0 (remember that given a ranking operator $\widetilde{\circ}$, the BSCA-F considers a fuzzy payoff $\widetilde{u}_1$ beneficial to another one $\widetilde{u}_2$ if $\widetilde{u}_1 \widetilde{\circ} \widetilde{u}_2 \geq threshold$).

First, figure 4.2 shows the means of the payoffs minimum support, possibilistic mean and maximum support, respectively. In the chart's x-axis, ">N" denotes fg_N, ">=N" denotes fge_N etc., while "sv" denotes the Shapley value. The minimum and maximum supports lie at -175 and 400, respectively, far out of the chart, so we have cut them off for better readability.

Not surprisingly in our superadditive setting, the possibility-based ranking operators, i.e. fge_P and fg_P obtain the (almost) same mean possibilistic mean payoffs as the Shapley value. That is because if it is of any possible benefit to form the grand coalition, then the BSCA-

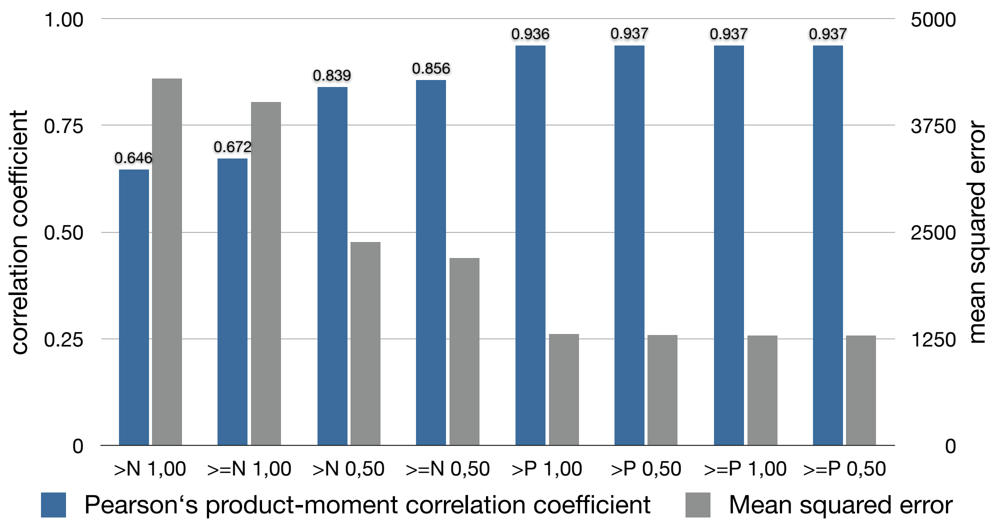

Figure 4.3: Correlation of Shapley values to BSCA-F payoffs

F will do so with these operators. The large mean support size of the Shapley value can be explained by the fact that we used the original Shapley value definition as provided in definition 4.1.18, which does no defuzzification of the values of subcoalitions. Therefore, the fuzziness of all the coalition values is caried over to the payoffs. In contrast, the bilateral Shapley value with defuzzified subcoalition values used by the BSCA-F retains only the fuzziness of the root coalitions.

Also not surprisingly, the fge_P and fg_P operators generally lead to the formation of coalitions whose values have a greater support size. Accordingly, they obtain a possibly higher payoff than fge_N and fg_N. The latter ones do miss out on some cooperation opportunities, as can be concluded from their lower posibilistic mean payoffs. However, they still fare quite well in this regard, especially with the 0.5 threshold. Also, their consistently higher minimum payoffs support the intuition that they might be preferred by risk-averse agents.

In figure 4.3, the linear correlation of degrees of the BSCA-F payoffs with respect to the Shapley value according to Pearson product-moment correlation coefficient are displayed, a standard correlation measure in statistics. A degree of 1 thereby implies full positive correlation. With a degree of 0.937, the possibility-based operators lead to a high correlation of BSCA-F payoffs with the Shapley value. The

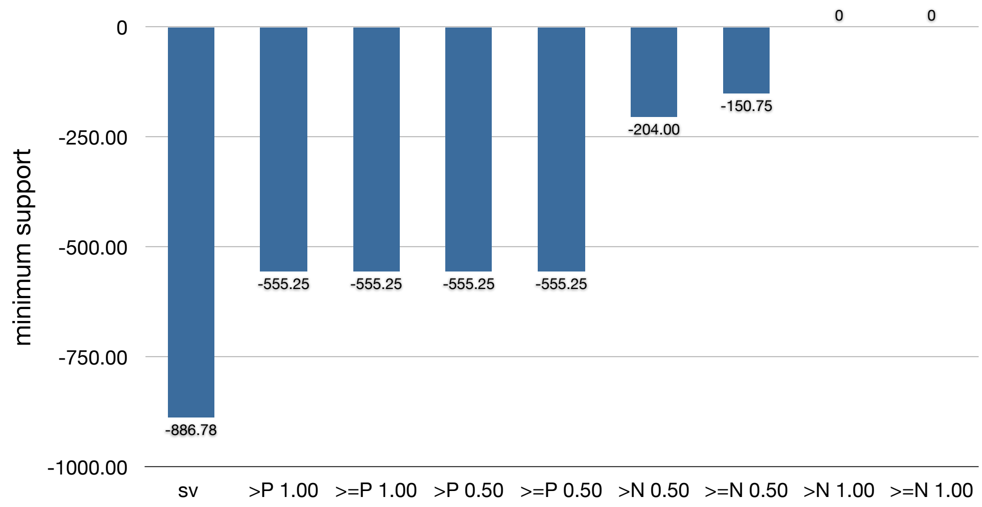

Figure 4.4: worst case losses

necessity-based operators, on the other hand, show less correlation, which is the obvious consequence of forming less coalitions.

Figure 4.4 shows the minimum supports of payoffs. It shows that all operators except $\tilde{\geq}_N$ and $\tilde{>}_N$ with threshold 1 lead to a possible loss for at least some agents. With $\tilde{\geq}_N$ and $\tilde{>}_N$ and threshold 1, however, coalitions leading to potential loss, i.e. with negative minimum support, are never formed. Thus, the N-operators might be suited for risk averse agents.

Next, we consider the possibility of core membership as defined by Mareš (2001) in figure 4.5. As it turns out, the possibility that the fuzzy payoffs obtained via the BSCA-F lie in the core is quite high for all operators, as is the number of payoffs which have possibility degree of 1 to lie in the core. However, the necessity of payoffs to be in the core is (naturally) much smaller, as can be seen in figure 4.6.

However, given that generally non-convex games were used to run the simulation, and that the BSCA-F does not consider core membership at all, we consider these results quite favourable. In particular, it shows that $\tilde{>}_N$ achieves quite high values for such a setting.

Therefore, we can conclude that generally the P-operators maximize the possibilistic expected payoff, the N-operators minimize worst case losses, and specifically $\tilde{>}_N$ manages to achieve a higher

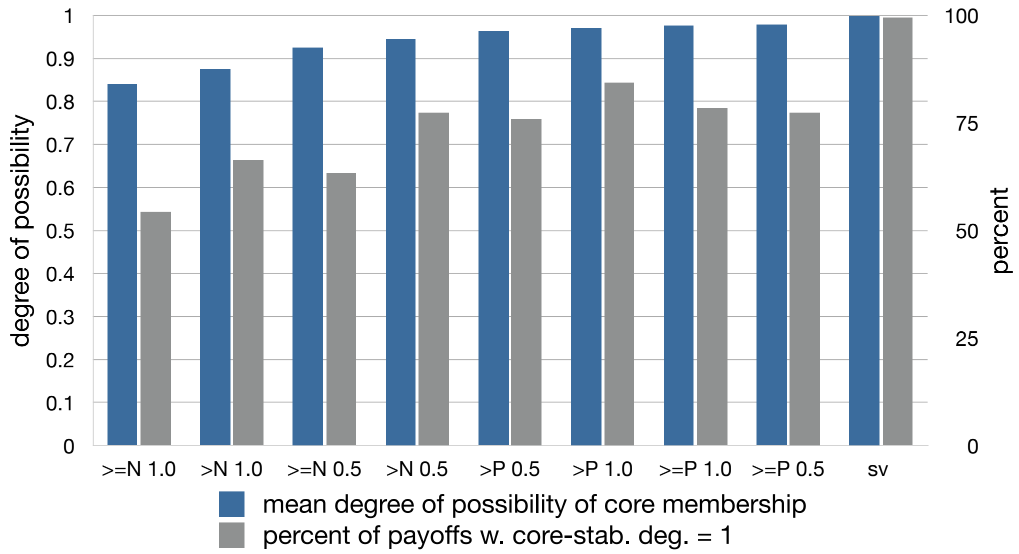

Figure 4.5: Possibility of core membership

degree of necessity of core membership.

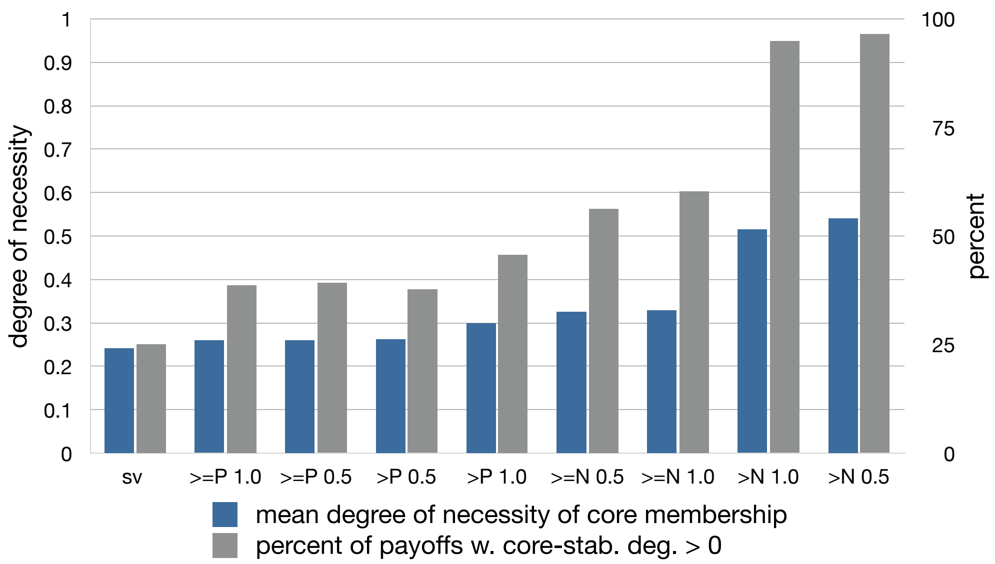

Figure 4.6: Necessity of core membership

4.7 Summary

In the setting of possibilistic fuzzy-valued cooperative game theory, we devised a new algorithm BSCA-F to form fuzzy recursiveley bilateral Shapley value stable coalitions. It was shown to be of low polynomial computational and communication complexity. The procedures of modeling a realistic economical situation as a fuzzy cooperative game and forming coalitions using the BSCA-F were illustrated with the help of an explanatory example. It was also demonstrated that using the possibilistic mean value to defuzzify the fuzzy payoffs is reasonable and consistent with the framework. Further, it was shown how the choice of an apropriate fuzzy ranking method can be utilized to implement optimistic or pessimistic strategies for the agents.

This was then confirmed in the evaluation section, were it was demonstrated that the N-operators might be a suitable choice for risk averse agents.

However, ranking operator needs to be agreed upon by all participating agents for the BSCA-F. Further work is thus needed to relax this requirement, such that each agent can deal with the uncertainties according to its individual preference in each step. The particular challenge of such an approach is to not seriously increase the communication complexity due to the required additional negotiations.

Chapter 5

On Risk-bounded Coalition Formation

In chapter 4, we introduced a coalition formation algorithm for fuzzy-valued games. There it was also demonstrated that choosing ranking operators based on necessity might lower the risk for agents to experience losses. However, the risk itself was not measured, and so has to be viewed as more of an ad-hoc method when it comes to risk control.

In this chapter, we introduce an approach that allows the agents to form stable coalitions such that strict individual risk bounds are adhered (this approach has been first outlined in Blankenburg et al., 2006). We therefore measure risk of coalitional failure via a standard financial measure of risk, assuming that the probability of failure of a coalition can be determined. We consider a setting of resource-bounded service provider agents forming coalitions to satisfy requests with deadlines. To do so, they are assumed to be able to compute service composition plans. The runtime of service executions therein is determined by the amount of resources spent by each agent.

To measure the risk of a coalition, we then employ the risk measure *tail conditional expectation* (TCE), which is a coherent risk measure. Coherent risk measures are subadditive, that is, given the independent risks of two options X and Y, then the sum of their risks $Risk(X) + Risk(Y)$ is greater or equal to $Risk(X + Y)$. Translated to coalitions, this means that the risk of two coalitions which are formed at the same time cannot be greater then the sum of their individual risks. We therefore allow agents to be part in multiple coalitions, each with some degree, where an agent's degree in the coalition determines how much of its resources it "invests" into the coalition.

Therefore, these coalitions can be viewed as fuzzy coalitions as introduced by Aubin (1979) and Butnariu (1980). Several solution concepts have been adapted for cooperative games with fuzzy coalitions, for instance by Li and Zhang (2009), Nishizaki and Sakawa (2001). But as it turns out, we cannot directly use these existing solution concepts, because they assume that the coalition value is a proportional function of the agents' membership degrees. As this assumption does not hold in our setting, we introduce appropriate extensions of the excess and surplus. We then show that it is possible to compute the surplus in polynomial time under some additional assumptions, similar to the approach taken in Shehory and Kraus (1999). As in the classic kernel, the transfer scheme introduced in Stearns (1968) can then be used to compute Kernel-stable solutions for the game. However, to obtain polynomial runtime, we show that not only the coalition size has to be bounded, but also the number of plans containing the same set of agents, and the number of coalitions in which an agent might simultaneously be a member of.

The remainder of this chapter is organized as follows: in section 5.1 we introduce some basic notions of continuous random variables and measures of risk which are employed in later sections, as well as our service agent and coalition model and finally the employed notion of fuzzy coalition games among service provider agents. We then show how to compute the risk of fuzzy coalitions and fuzzy coalition structures in section 5.2. Section 5.3 is concerned with the stability of risk-bounded fuzzy coalitions. We propose our coalition formation protocol RFCF in section 5.4. We conclude in section 5.5.

5.1 Preliminaries

Continuous random variables and measures of risk

Our service agent model which we use in this chapter is based on modelling service execution times as continuous random variables. In this section, we therefore first present their basic definitions and properties (see e.g. Grimmett and Stirzaker, 2001), although we assume that the reader is familiar with the basics of probability theory.

Definition 5.1.1 Continuous random variable and probability density function
A *random variable* X over values in $\mathbb{R}$ is called *continuous* if the probability $P(X \leq x)$ that $X \leq x$, $x \in \mathbb{R}$, is given by

$$P(X \leq x) = \int_{-\infty}^{x} pdf_X(y)dy$$

where pdf_X is the *probability density function* of X, an integrable function $pdf_X : \mathbb{R} \mapsto [0, \infty)$. △

We will also need to add up continuous random variables, for which purpose we require the *convolution of functions* (see e.g. Hirschman and Widder, 1955):

Definition 5.1.2 Convolution of functions
The *convolution* of two functions f and g on $\mathbb{R}$ is defined, with $x \in \mathbb{R}$, as

$$(f * g)(x) := \int_{0}^{\infty} f(y)g(x-y)dy$$

△

Theorem 5.1.3. *Some well-known properties of convolutions are that*

1. *convolutions are commutative and associative (again, see e.g. Hirschman and Widder, 1955), and that*
2. *the sum of the PDFs of two independent continuous random variables A and B is equal to their* convolution *(see e.g. Grimmett and Stirzaker, 2001, p. 113):*

$$pdf_{A+B}(x) = (pdf_A * pdf_B)(x)$$

These notions of continuous random variables are already sufficient for our purposes. Since the purpose of this chapter is to investigate risk-bounded coalition formation, we now introduce *measures of risk*. Intuitively, assume that there is a variety of combination of coalitions that an agent can possibly join, each of which has some random coalition value. A rational agent will then prefer the coalitions with a high expected value. However, considering only the expected value is said to be *risk-neutral* behaviour, because it is indifferent to the cost that is inflicted on the agent in case the coalition fails. But if an agent cannot afford to lose more than some amount, even a low probability of vailure of the coalition might be still too risky. To control and avoid such situations, a number of financial risk measures have been introduced in the literature (for an overview, see Cheng et al. (2004) and references therein).

For the definitions in the remainder of this section, we follow Artzner et al. (1999), omitting certain details which are not important in our setting. Also, where Artzner et al. speak of *positions* (meaning investment positions), we speak of *strategies*, meaning an agent's decision with whom to coalesce and service requests to work on. Lastly, note that the definitions of the *Value-at-Risk* and other measures in Artzner et al. (1999) include the reward of a reference investment (e.g. interest rates) as a scaling factor, which we omit here for simplicity.

Definition 5.1.4 Risk and Measure of Risk
Let Ω denote the set of states of nature, and assume it is finite. Considering Ω as the set of outcomes of an experiment, we compute the final net worth of a strategy for each element of Ω. Risk is the investor's future net worth, which is described by a random variable. Let G be the set of all risks, that is the set of all real valued functions on Ω. A measure of risk r is a mapping r: $G \mapsto \mathbb{R}$. △

According to Cheng et al. (2004), a widely known and used one is the Value-at-Risk (VaR), which also has become part of financial regulations. VaR calculates how much one may lose during a specified period given a probability of failure, and the amount of capital that should be used to control the risk.

Definition 5.1.5 Value-at-Risk (VaR)
Given $\alpha \in [0, 1]$, the Value-at-Risk VaR^α at level α of the final net worth $X \in G$ with distribution P is

$$VaR^\alpha(X) = -\inf\{x \in \mathbb{R} : \; P(X \leq x) > \alpha\}$$

△

Artzner et al. (1999) also introduce the notion of *coherent* risk measures.

Definition 5.1.6 Coherent risk measure
With $X, Y \in G, z \in \mathbb{R}$, a risk measure r is called coherent if it satisfies

1. subadditivity: for all $X, Y \in G$: $r(X + Y) \leq r(X) + r(Y)$
2. translation invariance: $r(X + z) = r(X) - z$
3. positive homogeneity: $\forall z \geq 0$, $r(zX) = zr(X)$

4. monotonicity: if $X \leq Y$ then $r(Y) \leq r(X)$

$\triangle$

As has also been shown in Artzner et al. (1999), VaR is not coherent, since it does not fulfill subadditivity. As it turns out (see section 5.2), this lack of superadditivity constitutes a major drawback in the design of a risk-bound coalition formation algorithm. Fortunately, a number of coherent measures which are derived from VaR have been proposed. Here, we employ the tail conditional expectation (TCE) which is coherent if the probability of failure is given by a continuous random variable:

Definition 5.1.7 Tail Conditional Expectation
Given a probability measure P on Ω and a level α, the tail conditional expectation is defined by:

$$TCE^{\alpha}(X) = -E_P\{X|X \leq -VaR^{\alpha}(X)\}$$

$\triangle$

Service Agent Model

In this section we specify more precisely the environment of service agents that we consider in this chapter.

We consider two types of agents: service request agents and service provider agents.

Definition 5.1.8 Service Request Agent
A *service request agent* sra requests exactly one (possibly complex) service s and some deadline d. It will pay a certain *monetary reward* $rw \in \mathbb{R}$ for a successful execution of s before d. Otherwise, no reward is paid.

SRA denotes the set of all service request agents in the system.
$\triangle$

On the other hand, service provider agents offer the execution of exactly one type of service. They are assumed to be computationally bounded, i.e. to have only limited resources per time for the execution of their service. For simplicity, we assume that the execution time for a service instance is a linear function of the resources devoted to it. This is reasonable in the case where the bounded resources are computing power and/or memory, for example.

Definition 5.1.9 Service Provider Agent
A *service provider agent* spa offers the execution of exactly one service s_{spa} and has the following properties:

1. Service Composition
   (a) spa is able to send *service advertisements* for s_{spa}.
   (b) given a requested service s and a set of service advertisements, spa has the ability to compute *service composition plans*; each such plan is a list of advertised services whose execution implements the requested service s.
   (c) each element of a plan $\mathcal{P}$ is called a *service instance* of the respective service.
   (d) all service instances in a plan $\mathcal{P}$ are to be executed sequentially.

2. Service Execution
   (a) spa can spend only some max. amount of resources per time in service executions.
   (b) the *minimum execution time* of an instance i of s_{spa} is denoted t_i^{min} (i.e. this is the execution time if spa devotes all its resources to it).
   (c) spa can split its resources and execute multiple instances of s_{spa} at the same time. The *fraction of resources per time* (wrt. the maximum) devoted to the execution of service instance i is denoted r_i.
   (d) the *execution time* t_i of service instance i is

   $$t_i = \frac{1}{r_i} \cdot t_i^{min}$$

   All execution times of services are assumed to be independent of each other.
   (e) spa might not be able to determine t_i^{min} exactly in advance. Instead, t_i^{min} (and therefore also t_i) is assumed to be a continuous random variable for which spa is able to determine its probability density function $pdf_{t_i^{min}}$.
   (f) there is a monetary *cost for resource consumption* of spa. We assume this is constant, so that because of point 2d. the cost $cost_i$ for executing service instance i is also constant and does not depend on r_i.

SPA denotes the set of all service provider agents in the system. △

Probability of plan execution failure

According to definitions 5.1.8 and 5.1.9, plan executions lead to a positive reward only if they are completed before their respective deadlines, and service provider agents might devote different amount of resources to different plan executions. So the question is, given a resource allocation of spas to a plan, what is the probability that it can be executed completely before the deadline, that is, the *probability of success*? Or conversely, what is the *probability of failure*. Formally, we define these concepts as follows:

Definition 5.1.10 Probability of success and failure
Given a plan $\mathcal{P}$ for a service request with deadline d, a set R of fractions r_i of resources per time for each service instance i in $\mathcal{P}$ and assuming that agents start executing $\mathcal{P}$ at time t_s and that the plan execution time is given by a random variable $t_{\mathcal{P}}$, subject to R, we define the *probability of success* $PoS(\mathcal{P}, t_s)$ that execution is finished before the deadline:

$$PoS(\mathcal{P}, R, t_s) := P(t_{\mathcal{P}} \leq d - t_s)$$

Accordingly, the *probability of failure* $PoF(\mathcal{P}, t_s)$ that the actual plan execution finishes later than the deadline is defined as:

$$PoF(\mathcal{P}, R, t_s) := 1 - PoS(\mathcal{P}, R, t_s)$$

△

To show how to compute these probabilities, we first establish that $t_{\mathcal{P}}$ is a continuous random variable and that we can deduce its PDF from the PDFs of the single service instance execution times:

Proposition 5.1.11. *For a plan $\mathcal{P}$ with $m \in \mathbb{N}$ service instances, the PDF of its execution time is an $m-1$-fold convolution over the individual service instance execution time PDFs. Formally, with $x \in \mathbb{R}^+$ (it is sufficient to consider only positive values since execution times are always positive),*

$$pdf_{t_{\mathcal{P}}}(x) = (\cdots(pdf_{t_{i_1}} * pdf_{t_{i_2}}) \cdots * pdf_{t_{i_m}})(x)$$

Proof. Because of the linear relationship assumed in definition 5.1.9, point 2d, the PDF of the execution time of a service instance i with fraction of resources per time r_i is:

$$pdf_{t_i}(x) = pdf_{t_i^{min}}(r_i \cdot x)$$

Further, because of the assumptions in definition 5.1.9 that the execution times of service instances are independent of each other and that the services in any plan are executed sequentially, the total execution time of $\mathcal{P}$ is the sum of the execution times of the individual service instances:

$$t_{\mathcal{P}} = \sum_{i \in \mathcal{P}} t_i$$

Since the PDF of the sum of independent random variables is equal to the convolution of the PDFs of the summands (see theorem 5.1.3), the proposition is proved. □

Having this, it is now easy to obtain the probabilities of success and failure of a plan:

Corollary 5.1.12. *From definitions 5.1.1 and 5.1.10, as well as proposition 5.1.11, it follows directly that*

$$PoS(\mathcal{P}, t_s) = \int_{-\infty}^{d-t_s} pdf_{t_{\mathcal{P}}}(x)dx$$

Remark 5.1.13. *While proposition 5.1.11 specifies how to theoretically compute the PDF of the execution time for a plan, it is quite not so obvious how to do it in practice. We note that for specific cases, there exist simple analytical solutions of the convolution. For example, the convolution of two normal PDFs is again normal, as is the convolution of a normal PDF with an exponential one. But this is not the case for arbitrary distribution types. Fortunately, there are alternative ways to obtain the convolution, such as the pointwise multiplication of the Fourier Transform of the PDFs (Bracewell, 1999). We do not go into its details here, but note that the Fast Fourier Transform algorithm efficiently approximates the Fourier Transform with complexity $k \log k$, where k is the number of sample points taken from the functions.*

Fuzzy Coalition Games of SPA Agents

In our setting, the capability of service provider agents to split their resources among different service instance executions makes it possible for them to take part in several service composition plan executions. This suggests to allow the agents to be a (partial) member of several coalitions. For this purpose, e.g. Aubin (1979), Butnariu (1980), Nishizaki and Sakawa (2001) extended concepts from cooperative game theory to allow for *fuzzy coalitions*, where each agent is a member only to a certain *membership degree*. In our model, each fuzzy coalition will execute exactly one service composition plan. The membership degree represents the relative amount of resources they spend for their respective service instance executions in the plan. If the same group of agents decides to execute an additional plan, it simply forms an additional fuzzy coalition. We also disallow any members that are not actually involved in the execution of P.

Definition 5.1.14 Fuzzy Coalition of Service Provider Agents Let there be a request for a service ws from a service request agent sra and a plan $\mathcal{P}$ whose execution satisfies ws.

1. $SPA_{\mathcal{P}} \subseteq SPA$ is the set of service provider agents involved in $\mathcal{P}$.

2. The *fuzzy coalition of service provider agents* $\widetilde{C}$ *for* sra *and* $\mathcal{P}$ is written as

$$\widetilde{C} = (spa_1/mem_1, \ldots, spa_k/mem_k, sra, \mathcal{P})$$

   with $k = |SPA_{\mathcal{P}}|$, $spa_j \in SPA_{\mathcal{P}}$, $1 \leq j \leq k$; $mem_j \in [0,1]$ is a *guaranteed minimum* for the fraction of resources per time r_i devoted by spa_j to any i of its service instances in $\mathcal{P}$.

3. $mem(spa, \widetilde{C})$ is agent spa's membership in $\widetilde{C}$.

4. We write $spa \in \widetilde{C}$ if spa is a member of $\widetilde{C}$ with some positive membership, i.e. $mem(spa, \widetilde{C}) > 0$.

5. $\widetilde{C} \widetilde{\subseteq} \widetilde{C}'$ if $\forall spa \in \widetilde{C} : mem(spa, \widetilde{C}) \leq mem(spa, \widetilde{C}')$, where $\widetilde{C}$ and $\widetilde{C}'$ are fuzzy coalitions for the same service request agent and plan.

6. $\widetilde{C}(sra, plan)$ denotes the set of all fuzzy coalitions $\widetilde{C} = (., sra, plan)$.

7. $|\widetilde{C}|$ is the number of agents in $\widetilde{C}$.

$\triangle$

In the remainder of this chapter, we also denote "fuzzy coalition" or just "coalition" instead of "fuzzy coalition of service provider agents" where the context is clear.

Corollary 5.1.15. *Since the membership degree* mem_i *of agent* spa_i *in a fuzzy coalition* $\widetilde{C}$ *is a lower bound on the fractions* r_i *of resources per time for each service instance* i *in the coalitional plan* $\mathcal{P}$*,* $PoS(\mathcal{P}, t_s)$ *for a given start time* t_s *is also a lower bound for the probability that the coalition fails to complete the service execution by the deadline.*

Although spas might actually devote more resources than guaranteed to a fuzzy coalition, we define the *probability of success of a fuzzy coalition* by assuming the worst-case scenario:

Definition 5.1.16
The *probability of success of a fuzzy coalition* $\widetilde{C}$ is defined as the probability of success of the coalitional plan $\mathcal{P}$ assuming that every $spa \in \widetilde{C}$ only devotes the resources which it guarantees to $\widetilde{C}$. That is, assuming plan execution start time t_s and that for all $spa_k \in \widetilde{C}$ and each service instance i of spa_k: $r_i = mem_k$, we define

$$PoS(\widetilde{C}) := PoS(\mathcal{P}, t_s)$$

Accordingly, the *probability of failure of a fuzzy coalition* is defined as

$$PoF(\widetilde{C}) := 1 - PoS(\widetilde{C}) = PoF(\mathcal{P}, t_s)$$

△

Corollary 5.1.17. *Given that* $PoS(\widetilde{C})$ *is a lower bound for the probability of success of a coalition, a lower bound on the expected reward of* $\widetilde{C}$ *is equal to* $PoS(\widetilde{C}) \cdot rw$.

Note that the monetary reward rw that is paid for a successful plan execution corresponds to the valuation $w(g)$ of goal g to satisfy the service request in the definition of the local worth 2.2.27. However, in this chapter, only the service provider agents form coalitions, and it is not important which of them receives the payment from the respective sra to distribute among the coalition members. In fact, we do not consider local worths here at all. Therefore, we define the coalition value directly in terms of rw.

We proceed analogously with the costs arising in a fuzzy coalition. Also, although agents should reasonably stop the execution once the deadline is reached, we consider only the worst case, i.e. the case where maximum costs have been produced even if the coalition fails:

Definition 5.1.18 Value of a Fuzzy Service Provider Agent Coalition
Let there be a fuzzy coalition $\widetilde{C}$ with plan $\mathcal{P}$ and reward rw. The *value* $v(\widetilde{C})$ of $\widetilde{C}$, also called *coalition value*, is defined as

$$v(\widetilde{C}) := PoS(\widetilde{C}) \cdot rw - \sum_{i \in \mathcal{P}} cost_i$$

△

Although fuzzy coalition structures allow the agents to be a member in several coalitions at the same time, we still have to require that each agent does not allocate more resources to coalitions than it can actually provide. Formally, we therefore define

Definition 5.1.19 Feasible Fuzzy Coalition Structure
For a fuzzy coalition $\widetilde{C}$, let $mem_{spa}^{\widetilde{C}}$ denote the membership degree of spa in $\widetilde{C}$, with $mem_{spa}^{\widetilde{C}} = 0$ if spa is not member of $\widetilde{C}$. A *feasible fuzzy coalition structure* $\widetilde{\mathcal{C}}$ for the agents in SPA is defined as a set of fuzzy coalitions with

$$\forall spa \in SPA : \sum_{\widetilde{C} \in \mathcal{C}} mem_{spa}^{\widetilde{C}} \leq 1$$

△

5.2 Risk of Fuzzy Coalition Structures

Using this measure, each agent spa_i may individually specify a parameter α_i and a TCE-threshold $tTCE_i$, expressing that it will only accept coalition structures which satisfy

$$TCE^{\alpha_i}(u_i) \leq tTCE_i$$

where u_i is agent spa_i's final net worth, i.e. the total net result from all coalitions it is involved in.

Proposition 5.2.1. *Let service provider agent spa_i be a member in a fuzzy coalition $\widetilde{C}$, let $cost_i$ be the cost for spa_i if $\widetilde{C}$ fails, and let $u_i(\widetilde{C}) > -cost_i$ be the payoff obtained by spa_i if $\widetilde{C}$ is successful. The $TCE^{\alpha_i}(\widetilde{C})$, i.e. the TCE^{α} restricted to consider only spa_i and $\widetilde{C}$, can be computed as follows:*

$$TCE^{\alpha_i}(\widetilde{C}) = \begin{cases} PoF(\widetilde{C})cost_i(\widetilde{C}) + PoS(\widetilde{C})(-u_i(\widetilde{C})) & PoF(\widetilde{C}) \leq \alpha_i \\ cost_i(\widetilde{C}) & PoF(\widetilde{C}) > \alpha_i \end{cases}$$

Proof. Let X_i be spa_i's net result from $\widetilde{C}$, with $X_i = u_i$ in case of success of $\widetilde{C}$ and $X_i = -cost_i$ in case of failure. Consider the first case, i.e. assume that $PoF(\widetilde{C}) \leq \alpha i$. Then the Value-at-Risk, i.e. the TCE^{α} restricted to consider only spa_i and $\widetilde{C}$, is $VaR^{\alpha_i}(\widetilde{C}) = -u_i$ because $P(X_i \leq -cost_i) = PoF(\widetilde{C}) \not> \alpha i$, but $P(X_i \leq u_i) = 1$ (since

$PoS(\widetilde{C}) = 1 - PoF(\widetilde{C})$). Thus, the set of relevant outcomes considered in TCE^{α} includes both $X_i = -cost_i$ and $X_i = u_i$. In the second case, with $PoF(\widetilde{C}) > \alpha i$, we have $VaR^{\alpha_i}(\widetilde{C}) = cost_i$ because $P(X_i \leq -cost_i) = PoF(\widetilde{C}) > \alpha i$. Thus, the set of relevant outcomes considered in TCE^{α} contains only $X_i = -cost_i$, and the case $X_i = u_i$ is disregarded. □

To obtain the TCE^{α_i} for a fuzzy coalition structure, we have to consider the probability of failure for each subset of fuzzy coalitions that spa_i is involved in, as well as the payoffs and costs for spa_i in these cases. From the independence of the PoF of different coalitions and the definition of VaR, it follows directly:

Corollary 5.2.2. *Let there be a fuzzy coalition structure $\widetilde{\mathcal{C}}$ and let $\widetilde{\mathcal{C}}_{spa_i} \subseteq \widetilde{\mathcal{C}}$ be the subset of all coalitions involving spa_i. For each $\widetilde{\mathcal{C}}^*_{spa_i} \in 2^{\widetilde{\mathcal{C}}_{spa_i}}$ (including the empty set) let $cost_i(\widetilde{\mathcal{C}}^*_{spa_i})$ be the cost for spa_i if all coalitions in $\widetilde{\mathcal{C}}^*_{spa_i}$ fail, and let $u_i(\widetilde{\mathcal{C}}^*_{spa_i})$ be the net payoff obtained by spa_i from the coalitions in $\widetilde{\mathcal{C}}_{spa_i} \setminus \widetilde{\mathcal{C}}^*_{spa_i}$ (i.e. the reward minus costs for the successful coalitions).*

The probability $PoF(\widetilde{\mathcal{C}}^*_{spa_i})$ that the coalitions in $\widetilde{\mathcal{C}}^*_{spa_i}$ fail while those in $\widetilde{\mathcal{C}}_{spa_i} \setminus \widetilde{\mathcal{C}}^*_{spa_i}$ succeed is

$$PoF(\widetilde{\mathcal{C}}^*_{spa_i}) = \prod_{\widetilde{C} \in \widetilde{\mathcal{C}}^*_{spa_i}} PoF(\widetilde{C}) \cdot \prod_{\widetilde{C} \in \widetilde{\mathcal{C}}_{spa_i} \setminus \widetilde{\mathcal{C}}^*_{spa_i}} PoS(\widetilde{C})$$

The $VaR^{\alpha_i}(\widetilde{\mathcal{C}})$, i.e. the VaR^{α} restricted to consider only spa_i and $\widetilde{\mathcal{C}}$, is then

$$VaR^{\alpha_i}(\widetilde{\mathcal{C}}) = -min_{\widetilde{\mathcal{C}}^*_{spa_i} \in 2^{\widetilde{\mathcal{C}}_{spa_i}}} \left\{ u_i(\widetilde{\mathcal{C}}^*_{spa_i}) : \sum_{\substack{\widetilde{\mathcal{C}}'_{spa_i} \in 2^{\widetilde{\mathcal{C}}_{spa_i}} \\ u_i(\widetilde{\mathcal{C}}'_{spa_i}) \leq u_i(\widetilde{\mathcal{C}}^*_{spa_i})}} PoF(\widetilde{\mathcal{C}}^*_{spa_i}) > \alpha_i \right\}$$

Having $VaR^{\alpha_i}(\widetilde{\mathcal{C}})$, the computation of the $TCE^{\alpha_i}(\widetilde{\mathcal{C}})$ is straight-forward. Please note that $VaR^{\alpha_i}(\widetilde{\mathcal{C}})$ and thus also TCE^{α_i} depend on the agent's payoff. But as becomes clear in section 5.3, computing a stable payoff depends on the risk. Also, we have to consider each element in the power-set of coalitions that spa_i is involved in, making the complexity of this computation exponential. However, by bounding the number of coalitions an agent might be involved in, we obtain polynomial complexity. This is also shown in section 5.3.

5.3 Stability of Fuzzy Coalitions Structures

In this section, we finally show how a coalition's payoff should be distributed among its members. Cooperative game theory traditionally deals with the question how this can be done in a *stable* way. Stable means that no agent has a reasonable incentive to break its coalition(s). For games with fuzzy coalitions, several such solution concepts, including the Core and the Shapley Value, have been introduced in the literature, such as Aubin (1979), Butnariu (1980) or Nishizaki and Sakawa (2001). Unfortunately, these assume a linear or even proportional relationship of the membership and coalition values. This does not hold in our case, because the coalition either gets the payoff or not, while the membership values determine the involved risk. But even considering the expected values does not help, since

1. the execution time of a service instance is characterized by an $\frac{1}{x}$-relationship wrt. to the membership and
2. the actual probability of failure also depends on the underlying distributions of the service instance runtimes which might be arbitrary.

We thus introduce a new variant of the *excess* which is compliant with out setting. Since the excess is the basis for a number of solution concepts including the Core, Kernel and Nucleolus, this allows us to use these concepts. In this paper, however, we consider only the Kernel.
In crisp games, the excess of a coalition C wrt. a given coalition structure $\widetilde{\mathcal{C}}$ with $C \notin \widetilde{\mathcal{C}}$ quantifies the difference in payoff that the agents in C obtain by forming C and leaving their resp. coalitions in $\widetilde{\mathcal{C}}$. Because each agent can be a member of only one coalition in a crisp coalition game, they then do not obtain any payoff from their former coalitions. But this is not the case in fuzzy coalition games. Here, it is possible to withdraw just some membership and put it into a new coalition. However, not all coalitions might be feasible wrt. the involved agents' individual risk bounds. We consider such coalitions not to be a feasible threat. Also, we exclude the case that an agent threatens to withdraw any amount membership from an existing coalition such that its own risk bound would be exceeded. While this makes sure that the hard risk bounds are taken into account, we also have to consider that more membership means

a better chance of success. Thus, we regard the expected coalition values.

Definition 5.3.1 Excess of a fuzzy coalition
Let there be fuzzy coalition $\widetilde{C}$ and fuzzy coalition structures $\widetilde{\mathcal{C}}$ and $\widetilde{\mathcal{C}}'$ with $\widetilde{C} \in \widetilde{\mathcal{C}}'$, $\widetilde{C} \notin \widetilde{\mathcal{C}}$, $\widetilde{\mathcal{C}}'$ is feasible, and $\forall \widetilde{C}' \in \widetilde{\mathcal{C}}', \widetilde{C}' \neq \widetilde{C} : \exists \widetilde{C}'' \in \widetilde{\mathcal{C}} : \widetilde{C}' \widetilde{\subseteq} \widetilde{C}''$. Further, let there be a payoff distribution u. We define

$$\widetilde{e}(\widetilde{C}, \widetilde{\mathcal{C}}', \widetilde{u})_{|TCE} := v_{|TCE}(\widetilde{C}, \widetilde{\mathcal{C}}') - \sum_{spa_i \in \widetilde{C}} d_i(\widetilde{\mathcal{C}}, \widetilde{\mathcal{C}}')$$

with

$$v_{|TCE}(\widetilde{C}, \widetilde{\mathcal{C}}') = \begin{cases} v(\widetilde{C}) & \text{if } \forall spa_i \in \widetilde{C} : TCE^{\alpha_i}(\widetilde{\mathcal{C}}' \cup \widetilde{C}) \leq tTCE_i \\ 0 & \text{otherwise} \end{cases}$$

and

$$d_i(\widetilde{\mathcal{C}}, \widetilde{\mathcal{C}}') = \sum_{\widetilde{C}^* \in \widetilde{\mathcal{C}}, \widetilde{C}' \in \widetilde{\mathcal{C}}', \widetilde{C}' \widetilde{\subseteq} \widetilde{C}^*} v(\widetilde{C}') - v(\widetilde{C}^*)$$

△

Remember from definition 2.2.14 that in crisp games, for a given configuration $(\mathcal{C}, u)$, the surplus of an agent a_i over another agent a_k with $a_i, a_k \in C \in \mathcal{C}$ is then defined as the maximum excess of all coalitions including agent a_i but without agent a_k. For games with fuzzy coalitions, however, it is possible to threaten with a number of alternative coalitions at the same time. Also, only a membership transfer from coalitions that include both a_i and a_k should be considered. Finally, we require that all membership of a_i from such coalitions is transferred.

Definition 5.3.2 Fuzzy coalition surplus
Let there be a fuzzy coalition structure $\widetilde{\mathcal{C}}$ and payoff distribution u and agents a_i and a_k.

1. A feasible fuzzy coalition structure $\widetilde{\mathcal{C}}'$ with $\forall \widetilde{C} \in \widetilde{\mathcal{C}}', \widetilde{C} \notin \widetilde{\mathcal{C}} : a_i \in \widetilde{C}, a_k \notin \widetilde{C}$, $\forall C \in \widetilde{\mathcal{C}}, a_k \notin C : C \in \widetilde{\mathcal{C}}'$ and $\nexists \widetilde{C} \in \widetilde{\mathcal{C}}' : a_i, a_k \in \widetilde{C}$ is called an *ik-fuzzy surplus structure*.

2. The set of all ik-fuzzy surplus structures wrt. $\widetilde{\mathcal{C}}$ is denoted $\widetilde{\mathcal{C}}S_{ik}(\widetilde{\mathcal{C}})$

3. The *fuzzy coalition surplus* of a_i over a_k is

$$\widetilde{sur}_{ik|TCE} := \max_{\widetilde{\mathcal{C}}' \in \widetilde{\mathcal{C}}S_{ik}(\widetilde{\mathcal{C}})} \{ \sum_{a_i \in \widetilde{C} \in \widetilde{\mathcal{C}}'} \widetilde{e}(\widetilde{C}, \widetilde{u})_{|TCE} \}$$

$\triangle$

To compute a fuzzy coalition surplus it is thus not only necessary to identify the best set of agents that should form alternative coalitions when excluding the other agent, but also to find the best membership values for them wrt. feasibility and the individual agent risk thresholds.

Definition 5.3.3
Let Q_{ik} denote a set of pairs $(sra, \mathcal{P})$ with $\mathcal{P}$ satisfies the request from sra, $a_i \in SPA_{\mathcal{P}}$ and $a_k \notin SPA_{\mathcal{P}}$. For a feasible coalition structure $\widetilde{\mathcal{C}}$, let $\widetilde{\mathcal{C}}S_{ik}(Q_{ik})$ denote the set of all ik-fuzzy surplus structures $\widetilde{\mathcal{C}}'$ wrt. $\widetilde{\mathcal{C}}$ such that for all pairs $(sra, \mathcal{P}) \in Q_{ik}$ there exists $\widetilde{C} \in \widetilde{C}(sra, \mathcal{P})$ with $\widetilde{C} \in \widetilde{\mathcal{C}}'$. We define the function $Max\widetilde{\mathcal{C}}(Q_{ik}, \widetilde{\mathcal{C}}, u)$ to return $\widetilde{\mathcal{C}}^* \in \widetilde{\mathcal{C}}S_{ik}(Q_{ik})$ such that $\sum_{a_i \in \widetilde{C} \in \widetilde{\mathcal{C}}^*} \widetilde{e}(\widetilde{C}, \widetilde{u})_{|TCE}$ is maximized wrt. all other elements in $\widetilde{\mathcal{C}}S_{ik}(Q_{ik})$. $\triangle$

Because the service instance runtime depends on the spent resources and thus the membership values by a $\frac{1}{x}$-relationship (see Definition 5.1.9.2(d)), $Max\widetilde{\mathcal{C}}$ has to solve a non-linear optimization problem. The complexity to compute a fuzzy coalition surplus is thus even worse than in the crisp case, where we have exponential complexity wrt. the number of agents in the system because of the exponential number of possible coalitions and excesses. Shehory and Kraus proposed to reduce this to a polynomial complexity by limiting the maximum coalition sizeShehory and Kraus (1999). We achieve the same effect for the fuzzy coalition surplus by not only bounding the number of agents in a coalition, but also the number of coalitions that an agent threatens to transfer membership to as well as the number of plans per set of agents.

Proposition 5.3.4. *Let $aMax \in \mathbb{N}$ be an upper bound for the number agents in a coalition and $\widetilde{C}Max \in \mathbb{N}$ be an upper bound for all sets $|Q_{ik}|$, i.e. the number of new coalitions including agent a_i and excluding agent a_k in the computation of $\widetilde{sur}_{ik|TCE}$. Let further $\mathcal{P}Max$ be an upper bound for the number of plans that involve the same set of agents and let $n \in \mathbb{N}$ be the number of agents. Then the number of sets Q_{ik}, constrained by $\widetilde{C}Max$ and $\forall (sra, \mathcal{P}) \in Q_{ik} : \mathcal{P} \in Plans$, is less or equal than $n^{(aMax \cdot \mathcal{P}Max)^{\widetilde{C}Max}}$.*

Proof. It was shown in Shehory and Kraus (1999) that the number of crisp coalitions with maximum size $aMax$ among n agents is bounded by n^{aMax}. Because each set of agents might be involved in multiple plans, this has to be multiplied $\mathcal{P}Max$ to obtain the upper bound for the number of considered coalitions. By the same argument as in the proof in Shehory and Kraus (1999), the number of sets of these coalitions with maximum size $\widetilde{C}Max$ is then bounded by $n^{(aMax \cdot \mathcal{P}Max)^{\widetilde{C}Max}}$. □

Remember from definition 2.2.14 that in crisp games, the *kernel* of a cooperative game $(\mathcal{A}, v)$ with respect to a given coalition structure $\mathcal{C}$ is a set of configurations $(\mathcal{C}, u)$ wherein each pair of agents a_i, a_k in each coalition $C \in \mathcal{C}$ is in equilibrium wrt. their surpluses: $\forall a_i, a_k \in C \in \mathcal{C}$:

$$\begin{aligned} &(sur_{ik} = sur_{ki}) \\ \vee &(sur_{ik} > s_{ki} \wedge u(a_k) = v(\{a_k\})) \\ \vee &(sur_{ki} > s_{ik} \wedge u(a_i) = v(\{a_i\})) \end{aligned}$$

Fortunately, having defined the surplus also for fuzzy coalitions, we can substitute it in this definition to obtain a definition for the kernel for games with fuzzy coalitions:

Definition 5.3.5
Let there be a fuzzy coalition structure $\widetilde{\mathcal{C}}$ and payoff distribution u. $(\widetilde{\mathcal{C}}, u)$ is in the *kernel* of the fuzzy coalition game iff each pair of agents a_i, a_k in each fuzzy coalition $C \in fCs$ is in equilibrium wrt. their fuzzy coalition surpluses. △

To make a payoff distribution kernel-stable for a given coalition structure, the *transfer scheme* introduced by Stearns (1968) can be used in the case of crisp games. The same can be applied here, since a side-payment from one agent to another will increase the former agent's payoff while lowering the latter agent ones.

5.4 Algorithm RFCF

In this section, we propose a fuzzy coalition formation protocol that guarantees to form coalitions which are compliant to the agents' individual risk bounds. The negotiation is to be finished in a fixed amount of time in order to ensure a timely start service executions. In order to achieve polynomial complexity in the negotiation, some compromises have to be made. In particular, upper bounds for the risk of a coalition structure can be obtained by either considering only the self-values of the agents instead the actual utilities or by computing the risk for subsets of the structure and utilizing the subadditivity of TCE. The main drawback of using upper bounds for the risk is that it might prevent the formation of some coalitions which are then considered too risky although they are acceptable. We thus propose to execute a parallel process to continually improve the bound as long as there is time.

Before we give the actual definition of RFCF, we here provide a short outline of the protocol to emphasize the main ideas of the individual steps. In RFCF, each agent performs multiple tasks in parallel:

- **Composition Planning** Every agent performs service composition planning. Since only agents that can execute a plan together will form coalitions, this step is necessary to identify possibly worthwhile coalitions.
- **Coalition Negotiation**
    1. *Proposal generation* - The agent computes fuzzy coalitions such that their formation certainly leads to a feasible coalition structure while minimising the membership values. This way, no more membership (i.e. resources) than necessary is used, allowing the involved agents to possibly form additional coalitions later. A proposal is then send to the agents of the fuzzy coalition which maximises the value per membership.
    2. *Proposal evaluation* - From the received proposals, form feasible coalitions with acceptable risk the and maximal value per membership
    3. *Payoff distribution and risk bound update* - Use the transfer scheme to compute the Kernel-stable payoff distribution. Compute the single-coalition TCE and add it to previous coalition structure TCE bound to obtain an updated bound on the coalition structure TCE.
- **Risk Measure Computation** - Compute TCE for a new random subset of coalitions to obtain a tighter bound for the coalition structure TCE.

In the following definition of the algorithm, we use the following functions and constants:

- $\mathcal{P}Max$: the maximum number of plans to be considered for a set of agents

- $aMax$: the maximum coalition size
- $\widetilde{C}Max$: the maximum number of coalitions that an agent threatens to transfer membership to in the surplus computation
- $sra(\mathcal{P})$ Returns the service request agent for whose request $\mathcal{P}$ was generated.
- $findFuzzyCoalition(\widetilde{\mathcal{C}}, \mathcal{P}, risk)$: Computes a fuzzy coalition $\widetilde{C}$ such that the membership degrees in $\widetilde{C}$ are minimized while $\widetilde{\mathcal{C}} \cup \widetilde{C}$ is acceptable for all agents wrt. risk. Use $\widetilde{C}(sra(\mathcal{P}), \mathcal{P})$ as a starting point. If $risk = nil$ then compute an upper bound for $TCE^{\alpha_a}(\widetilde{\mathcal{C}} \cup \widetilde{C}(sra(\mathcal{P}), \mathcal{P}))$, otherwise use $risk$ as this upper bound. It is possible to efficiently implement this function by exploiting the monotonicity of the TCE wrt. to the membership values. If this is not possible or $|\widetilde{C}| > MaxCSize$, return nil
- $makeStable(\widetilde{\mathcal{C}})$: Computes a new stable payoff distribution u^* for the fuzzy coalition structure $\widetilde{\mathcal{C}}$ using the transfer scheme (see 5.3) and the bounds $\mathcal{P}Max$, $aMax$ and $\widetilde{C}Max$.

Algorithm 5.4.1 RFCF
Each agent a performs:

Initialization:

1. set $Plans := \emptyset$
2. set $PPlans := \emptyset$
3. set $PPlansRisk := \emptyset$
4. set $Props :=$ new priority queue
5. set $risk_a := TCE(\{a\}/1)$

Parallel Execution:

- Composition plan generation: repeat (until terminated)
    1. Generate a new composition plan $\mathcal{P}$ for a random service request and for a set of agents for which the number of previously generated plans is less than $\mathcal{P}Max$.

2. $Plans := Plans \cup \mathcal{P}$

- Coalition negotiation: repeat (until terminated)
  1. Proposal generation
     (a) set $BestCoalition := nil$, $BestPayoffperMembership := 0$
     (b) for each $\mathcal{P}$ in $Plans$ do:
        i. $\widetilde{C} := findFuzzyCoalition(\widetilde{\mathcal{C}}, \mathcal{P}, nil)$
        ii. if $\widetilde{C} = nil$ then $Plans := Plans \setminus \mathcal{P}$;
            $PPlans := \cup\mathcal{P}$;next 1b
        iii. if $v(\widetilde{C})/|\widetilde{C}| > BestPayoffperMembership$ then
            $Plans := Plans \setminus \mathcal{P}$; $BestCoalition := \widetilde{C}$;
            $BestPayoffperMembership := |\widetilde{C}|$
     (c) if $BestCoalition = nil$ then for each $\mathcal{P}$ in $PPlans$ do:
        i. if $PPlansRisk$ contains $(\mathcal{P}, .)$ then
            $\widetilde{C} := findFuzzyCoalition(\widetilde{\mathcal{C}}, \mathcal{P}, PPlansRisk(\mathcal{P}))$
        ii. if $\widetilde{C} = nil$ then next 1b
        iii. if $v(\widetilde{C})/|\widetilde{C}| > BestPayoffperMembership$ then
            $PPlansRisk := PPlansRisk \setminus \mathcal{P}$;
            $BestCoalition := \widetilde{C}$; $BestPayoffperMembership := |\widetilde{C}|$
  2. send $(BestCoalition, BestPayoffperMembership)$ as a proposal to all other agents
  3. Proposal evaluation
     (a) receive coalition proposals from all other agents and self
     (b) for each non-nil proposal $(\widetilde{C}, ppm)$, put $\widetilde{C}$ in $Props$ with priority ppm.
     (c) set $\widetilde{\mathcal{C}}^* = \emptyset$
     (d) while $Props$ is not empty do
        i. get and remove the highest priority coalition $\widetilde{C}$ from $Props$
        ii. if $\widetilde{C}$ is feasible, set $\widetilde{\mathcal{C}}^* := \widetilde{\mathcal{C}}^* \cup \widetilde{C}$
  4. Payoff distribution and TCE update
     (a) set $u^* = makeStable(\widetilde{\mathcal{C}} \cup \widetilde{\mathcal{C}}^*)$
     (b) do atomically: set $\widetilde{\mathcal{C}} := \widetilde{\mathcal{C}} \cup \widetilde{\mathcal{C}}^*$ and $u := u^*$
     (c) set $risk_a := risk_a + \sum_{\widetilde{C} \in \widetilde{\mathcal{C}}^*_a}(TCE_a(\widetilde{C}))$

- Risk measure computation of current structure: repeat (until terminated)
  1. randomly choose a previously unconsidered subset $\mathcal{C}^*$ from $\mathcal{C}_a$
  2. $risk_a := risk_a - \sum_{\widetilde{C} \in \mathcal{C}^*} TCE_a(\widetilde{C}) + TCE_a(\mathcal{C}^*)$
- Risk measure computation of potential structures for postponed plans: repeat (until terminated)
  1. Randomly choose $\mathcal{P} \in PPlans$ such that $(\mathcal{P}, .) \notin PPlansRisk$
  2. Compute exact $TCE^{\alpha_a}(\widetilde{\mathcal{C}} \cup \widetilde{C}(sra(\mathcal{P}), \mathcal{P}))$ and put $(\mathcal{P}, TCE^{\alpha_a}(\widetilde{\mathcal{C}} \cup \widetilde{C}(sra(\mathcal{P}), \mathcal{P})))$ into $PPlansRisk$
- Termination of negotiation
  1. Wait(ExecutionStartTime)
  2. terminate all other tasks
  3. start service instance execution in my coalitions; terminate

△

Proposition 5.4.2. *The runtime of the coalition negotiation section of the RFCF is polynomial.*

Proof. In the proposal evaluation, each agent orders the coalition proposals in the same way in the priority queue since the priority is defined as payoff per membership which is a global measure. Because of the bounds used in the surplus computation, the payoff distribution is done in polynomial time (see 5.3). All other steps in the coalition negotiation section are of less complexity. □

5.5 Summary

We have studied a setting of cooperative service provider agents that form fuzzy coalitions in order to share and combine resources and services to efficiently respond to market demands while bounding individual risk. We showed how a coherent risk measure, the TCE, can be used to assess the risk for agents when taking part in coalitions to satisfy service requests with deadlines. By splitting resources among different coalitions, an agent might lower its overall risk. Despite previous work on fuzzy coalitions in the literature, we found it necessary to give our own definitions for the fuzzy coalition game, including the excess and surplus for fuzzy coalitions. This is because of unrealistic assumptions in the cited models that do not hold in our setting. In the surplus computation, sets of alternative fuzzy coalitions have to be considered. As a consequence, we had to bound not only the maximum coalition size, but also the number of coalitions in these sets as well as the number of plans for a set of agents to obtain a polynomial computation time for the fuzzy coalition surplus.

Finally, we point out that the RFCF is easily adaptable to settings with other agent models than the one used in this chapter, as long as one can determine the probability of success/failure of fuzzy coalitions.

Chapter 6

Trusted Kernel Stable Coalition Formation

In this chapter, we consider the problem of deceiving and unreliable agents in kernel-based coalition formation algorithms. Such matters are very relevant to multiagent coalition formation in open environments.

Recall that many coalition formation algorithms and their corresponding communication protocols rely on the exchange of data among the agents in order to compute solutions for a game (see 2.3). Thus, agents might try to unjustifiably increase their own payoffs by sending manipulated data to other agents. Building on results from Blankenburg and Klusch (2004), Blankenburg et al. (2005), we first show in the following section 6.2 how agents can achieve this in kernel-based coalition formation algorithms. We then show how a protocol might be extended to hinder such manipulations. This is done by employing a communication protocol using cryptographic techniques to ensure that all agents learn about the full game data at the same time. We show that this greatly reduces the possibility of an agent to determine how to manipulated the data that it sends to other agents in order to increase its own payoff unjustifiably.

Another aspect of multiagent coalitions that, to our best knowledge, has received very little attention in the literature, is the actual execution of side payments. In corollary 2.2.29 we saw how to determine an agent's net side payment $sp_u(a, C)$ in its coalition C in configuration $(\mathcal{C}, u)$. However, nothing is said so far to which exact other agent it should pay which amount. Or conversely from which agents it should expect to receive a certain amount, and how it can sure that it is indeed paid by these others. After all, the computation and negotiation of a stable solution is not quite that useful if it is then not actually implemented by the agents.

For example, consider a coalition of two agents $\{a_1, a_2\}$, where a_1 is assigned to execute the single service in the coalition for the satisfaction of a request of a_2. Then, a_1's local worth is just its cost of executing the service and thus it will have to receive some positive side payment a_2 (whose positive local worth is determined by its valuation of the service execution) to obtain a positive and individually rational utility. Now, shall a_1 just execute the service in hopes of eventually receiving its side payment from a_2, or should it wait until it has indeed received it? Conversely, should a_2 just make the side payment in hopes of a_1 actually executing the service to a_2's satisfaction, or should it wait until the execution has finished?

To solve this dilemma in repeated coalition formation in task (or service) allocation scenarios, we employ a two-folded approach:

1. We introduce a novel *side payment protocol* which is to be executed before the task execution. This protocol ensures that it is rational for each agent (except one) to make its side-payment at the specified time, since we prove that deviating from it would induce a loss for the agent. However, the agent which has to make the biggest side payment (and thus profits only from satisfaction of its requests) has to rely on a different means to ensure that it can expect the services to be executed faithfully, which is covered in step 2:

2. After making the side payments, the agents perform their assigned tasks while mutually tracking each others performance. In doing so, agents learn to assess the *probability of success* of other agents task executions. This is done with the help of a *trust measure* which is integrated into the coalition formation algorithm. Therein, trust is defined as the expectation that agents will perform reliably when defecting would prove more profitable. It is determined via an agent's own observation of another agent's behaviour as well as the other agent's *reputation* by considering also the reports of other agents about the agent in question. In this way, *expected* coalition values are derived and refined over repeated coalition negotiations.

6.1 Preliminaries

Agent Model

In this chapter, we are mainly interested in investigating the options of agents to manipulate the game by sending false data. We therefore use a generic task allocation model, but need to explicitly model valuations and costs of task executions.

Definition 6.1.1 Task Agent Model
With $\mathcal{T}$ denoting the set of all agent-executable tasks, agents can perform and/or request tasks $\tau \in \mathcal{T}$ to be performed. We define

1. $w_i(\tau)$ as the agent a_i's valuation for the execution of task τ,
2. $c_i(\tau)$ as the agent a's cost for the execution of task τ,
3. $Alloc_C$ as the set of all possible mappings from tasks to agents in coalition C,
4. $\tau_i^j \in \alpha$ with $\alpha_C \in Alloc_C$ as a task τ requested by agent a_i and to be executed by agent a_j in coalition C,
5. $$w_{\alpha_C} := \sum_{\tau_i^j \in \alpha_C} w_i(\tau) - c_j(\tau)$$
6. α_C^* as a task allocation which maximizes the achievable total payoff for C:
$$\forall\ \alpha_C \in Alloc_C : w_{\alpha_C^*} \geq w_{\alpha_C}$$

△

Remark 6.1.2. *Note that the same task might be requested by many agents which will all derive a positive value when the task is performed.*

Corollary 6.1.3. *From the general definition 2.2.27 of the local worth, it follows that the local worth in the task agent model for agent a_i is given by*
$$lw_i(C) := \sum_{\tau_i^j \in \alpha_C^*} w_i(\tau) - \sum_{\tau_j^i \in \alpha_C^*} c_i(\tau)$$

Trust Model

The coalition formation algorithm proposed in this chapter makes use of a *trust model* to let agents assess each others' trustworthiness. More precisely, if an agent announces that it is able to perform a task to the satisfaction of the requesting agents, i.e. such that the valuations of the requests for the tasks are realized, this might actually turn out to be not true. This might happen because the execution agent overestimated its own abilities, or because it maliciously misinformed the other agents. From a requester's perspective, however, it does not have to matter why exactly the task was not executed to its satisfaction.

Here, we employ a generic trust model, which was adapted from Dash et al. (2004) and also used in Blankenburg et al. (2005). Several computational trust models have been developed in the literature (see e.g. Ramchurn et al., 2004), but here we do not focus on a particular trust model. Instead, we concentrate on the abstract properties to keep the focus on the relationship between trust and the design of the coalition formation algorithm. We therefore ensure that the properties of the algorithm are independent of any specific trust model.

Definition 6.1.4 Generic Trust Model

1. The trust measure of an agent i in an agent j depends both on i's perception of j's POS and on the perception of other agents about j's *probability of success* (POS). This latter point encapsulates the concept of *reputation* whereby the society of agents generally attributes some characteristic to one of its members by aggregating some/all the opinions of its other members about that member. Thus, each agent can consider this societal view on other members when building up its own measure of trust in its counterparts Ramchurn et al. (2004). Specifically, the trust of agent i in its counterpart j, $trust_i^j \in [0,1]$, is given by a function, $g : [0,1]^{|\mathcal{I}|} \to [0,1]$, (which, in the simplest case, is a weighted sum) of all POS measures sent by other agents to agent i about agent j as shown below:

$$trust_i^j = g(\{\eta_1^j, \ldots, \eta_i^j, \ldots, \eta_N^j\})$$

where $\eta_i^j \in [0,1]$ is the POS of agent j as perceived by agent i and g is the function that combines both personal measures of POS and other agents' measures. In general, trust models compute the POS measures over multiple interactions. Thus, the level of success recorded in each interaction is normally averaged to give a representative value (see Ramchurn et al. (2004) for a general discussion on trust metrics).

2. Trust results from an analysis of an agent's POS in performing a given task. The more successful, the more trustworthy it is. Thus, the models assume that trust monotonically increases with POS. Therefore, the relationship between trust and POS is expressed as: $\frac{\partial trust_i^j}{\partial \eta_i^j} > 0$, where $trust_i^j$ is the trust of i in agent j and η_i^j is the actual POS of agent j as perceived by i.

△

Given the above, agents can update their trust rating for another agent each time they perceive the execution of a task (both by recording their view of the success of their counterpart and by gathering new reports from other agents about it). Thus, if an agent's POS does not change, the trust measure in it should become more precise as more observations are made and received from other agents.

Since the trust model provides an estimation of the probability of success of other agents' task executions, we can use it to obtain according expected local worths and coalition values:

Definition 6.1.5 Expected Local Worth

$$\underline{lw}_k(C) := \sum_{\tau_k^j \in \alpha_C^*} trust_k^j \cdot w_k(\tau) - \sum_{\tau_j^k \in \alpha_C^*} c_k(\tau)$$

△

Corollary 6.1.6. *It follows that the* expected coalition value *is accordingly*

$$\underline{v}(C) = \sum_{k \in C} \underline{lw}_k(C)$$

6.2 Preventing Manipulation of the Kernel

In our model, the coalition values are defined as the sum of the local worths of the agents in a coalition. Each agent's local worth is determined by its reported valuation of requested tasks, the costs of its offered tasks and its trust values.

Even though agents are supposed measure each other's performance and compute resulting expected coalition values using the trust value, they might still try to manipulate their reported values so as to get a better payoff.

Therefore, in this section, we investigate how agents might try to manipulate their net outcome if taking part in kernel-based coalition negotiations. But first, we establish a general condition for a successful coalition negotiation manipulation, independent of the solution concept.

Profitability of a Game Manipulation

In this section, we take a step back again from the task agent model involving trust as defined in section 6.1 and consider the generic local worths and coalition values of definitions 2.2.27 and 2.2.28, respectively.

Therefore, remember from definition 2.2.27 that the local worth $lw_a(C)$ is the sum of a's valuations $w_a(g)$ of satisfied goals and a's cost $c_a(\tau)$ of its assigned tasks. But suppose that agent a manipulates the coalition negotiation by misleading the other agents about its true goal satisfaction valuations and/or task execution costs. Then, these *modified valuations/costs* induce a *modified game* with respect to the *original game*. We capture these notions in the following definition:

Definition 6.2.1 Original and Modified Game
Let $(\mathcal{A}, v)$ be a game for which a solution is to be found via some coalition formation algorithm, but let an agent $a \in \mathcal{A}$ mislead the other participating agents such that *modified goal valuations*, denoted $\overleftrightarrow{w}_a$ and/or *modified task execution costs*, denoted $\overleftrightarrow{c}_a$, instead of their *true* counterparts w_a and c_a are employed in the negotiation.

We then generally write $\overleftrightarrow{x}$ for any entity x of the *modified game* $(\mathcal{A}, \overleftrightarrow{v})$. Particularly, this includes (but is not limited to) the *modified local worths* $\overleftrightarrow{lw}$ and the *modified solution* $(\overleftrightarrow{\mathcal{C}}, \overleftrightarrow{u})$.

Further, if for an entity x agent a privately knows its value to be really x^t which might be different from $\overleftrightarrow{x}$, we also call x^t the *true* value of x.

In this context, $(\mathcal{A}, v)$ is also called the *original game*, and similarly we speak of the *original local worths* etc. Also, we call agent a a *manipulator.* △

For such a modified game, we can immediately state the following implications:

Corollary 6.2.2. *Assume the negotiation of the modified game* $(\mathcal{A}, \overleftrightarrow{v})$ *leads to the formation of coalition* $\overleftrightarrow{C}$*, with the manipulator* $a \in \overleftrightarrow{C}$*. Then,*

- $\overleftrightarrow{C}$ *might be equal or not equal to the coalition* C*,* $a \in C$*, which would have been formed if the original game* $(\mathcal{A}, v)$ *had been negotiated and*
- *the modified task allocation in* $\overleftrightarrow{C}$*, based on the modified valuations* $\overleftrightarrow{w}_a(g)$ *and costs* $\overleftrightarrow{c}_a(\tau)$*, might be different to the original task allocation that would have resulted from considering the* true *valuations* $w_a(g)$ *and costs* $c_a(\tau)$ *in* $\overleftrightarrow{C}$*.*

The fact that the modified task allocation for the coalition $\overleftrightarrow{C}$ might be different to its original task allocation implies that in the modified game, we need to consider different kinds of local worths of $a \in \overleftrightarrow{C}$:

Corollary 6.2.3. *In the modified game* $(\mathcal{A}, \overleftrightarrow{v})$*, the* true *local worth in the modified game of agent* a *in coalition* $\overleftrightarrow{C} \subseteq \mathcal{A}$ *is the sum of* a*'s* true *goal valuations and costs for the modified task allocation in* $\overleftrightarrow{C}$*:*

$$\overleftrightarrow{lw}^t_a(\overleftrightarrow{C}) = \sum_{g \in \overleftrightarrow{G}_a(\overleftrightarrow{C})} w_a(g) - \sum_{\tau \in \overleftrightarrow{\mathcal{T}}_a(\overleftrightarrow{C})} c_a(\tau)$$

Since the true *valuations and costs of* a *are private knowledge of* a*, so is its* true *local worth. From the other agents' point of view,* a*'s local worth in* $\overleftrightarrow{C}$ *is its modified local worth*

$$\overleftrightarrow{lw}_a(\overleftrightarrow{C}) = \sum_{g \in \overleftrightarrow{G}_a(\overleftrightarrow{C})} \overleftrightarrow{w}_a(g) - \sum_{\tau \in \overleftrightarrow{\mathcal{T}}_a(\overleftrightarrow{C})} \overleftrightarrow{c}_a(\tau)$$

Also note that both the true *and the modified local worth of* a *might be different to* a*'s original local worth in* $\overleftrightarrow{C}$*:*

$$lw_a(\overleftrightarrow{C}) = \sum_{g \in G_a(\overleftrightarrow{C})} w_a(g) - \sum_{\tau \in \mathcal{T}_a(\overleftrightarrow{C})} c_a(\tau)$$

Now, remember from corollary 2.2.29 that for a game $(\mathcal{A}, v)$ and solution $(\mathcal{C}, u)$, an agent a's side payment is determined by $sp_u(a, C) = u(a) - lw_a(C)$, $a \in C$, $C \in \mathcal{C}$. Hence it follows:

Corollary 6.2.4. *In the modified game $(\mathcal{A}, \overleftrightarrow{v})$, $a \in \mathcal{A}$ and modified solution $(\overleftrightarrow{\mathcal{C}}, \overleftrightarrow{u})$, a's modified side payment is*

$$\overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, \overleftrightarrow{C}) = \overleftrightarrow{u}(a) - \overleftrightarrow{lw}_a(\overleftrightarrow{C})$$

Further, note that an agent's payoff can also be stated as the sum of its local worth and its side payment:

Corollary 6.2.5. *In a game $(\mathcal{A}, v)$ and solution $(\mathcal{C}, u)$,*

$$u(a) = lw_a(C) + sp_u(a, C)$$

and similarly, in the modified game $(\mathcal{A}, \overleftrightarrow{v})$ and modified solution $(\overleftrightarrow{\mathcal{C}}, \overleftrightarrow{u})$,

$$\overleftrightarrow{u}(a) = \overleftrightarrow{lw}_a(\overleftrightarrow{C}) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, \overleftrightarrow{C})$$

Proposition 6.2.6. *In the modified game $(\mathcal{A}, \overleftrightarrow{v})$ and modified solution $(\overleftrightarrow{\mathcal{C}}, \overleftrightarrow{u})$, the* true *utility $u^t(a)$ of the manipulator $a \in \overleftrightarrow{C} \subseteq \mathcal{A}$ is*

$$u^t(a) = lw^t_a(\overleftrightarrow{C}) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, \overleftrightarrow{C})$$

Proof. Remember that from the other agents' point of view, $\overleftrightarrow{lw}_a(\overleftrightarrow{C})$ is indeed a's local worth, and likewise $\overleftrightarrow{u}(a)$ is a's utility. Therefore, they will indeed expect a to make the side payment of $\overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, \overleftrightarrow{C})$. However, we already pointed out in corollary 6.2.3 that if a is the manipulator, a's *true* local worth might be different from its modified local worth. Hence, a's modified side payment but *true* local worth constitute a's *true* utility of the modified solution. □

Having thus disentangled the *modified* and *true* payoffs and local worths, and which agents believe which to be true, it is clear that:

Corollary 6.2.7. *An agent a's manipulation of a game $(\mathcal{A}, v)$ leading to the modified game $(\mathcal{A}, \overleftrightarrow{v})$is profitable for a iff*

$$u^t(a) \geq u(a)$$

Corollary 6.2.8. *From corollary 6.2.5 and proposition 6.2.6, it also follows that*

$$\overleftrightarrow{u}(a) - u^t(a) = \overleftrightarrow{lw}_a(\overleftrightarrow{C}) - lw_a^t(\overleftrightarrow{C})$$
$$\Leftrightarrow u^t(a) = \overleftrightarrow{u}(a) + lw_a^t(\overleftrightarrow{C}) - \overleftrightarrow{lw}_a(\overleftrightarrow{C})$$

Thus, we can assert that in order to determine whether a game manipulation by agent a will be profitable for a, it must ensure that the right part of the last equation in corollary 6.2.8 is greater or equal than $u(a)$, the payoff it would obtain in the unmodified game. It therefore follows:

Proposition 6.2.9. *If a manipulator a does not have sufficient prior knowledge of the negotiated game $(\mathcal{A}, v)$, $a \in \mathcal{A}$, a is in general not able to determine how to modify the game such that it is profitable for a.*

Proof. From corollary 6.2.8, it follows that a needs to at least approximately be able to compute $u(a)$, $\overleftrightarrow{u}(a)$, $lw_a^t(\overleftrightarrow{C})$ and $\overleftrightarrow{lw}$. In order to compute (bounds on) $lw_a^t(\overleftrightarrow{C})$ and $\overleftrightarrow{lw}$, a is required to at least approximately know the task allocation in modified game. And in order to compute (bounds on) $u(a)$ and $\overleftrightarrow{u}(a)$, it is additionally required to at least approximately know the task allocation in the original game, as well as the goal valuations and task execution costs of the other agents in coalitions C and $\overleftrightarrow{C}$. □

However, we also point out the special case where $\overleftrightarrow{C} = C$ and $lw_a^t(C) = lw_a(C)$. That is, neither the manipulator's assigned coalition C nor the task allocation for C changes as a result from the game manipulation. In this case, it is sufficient for the manipulator to ensure that its side payments are increased:

Proposition 6.2.10. *In the modified game $(\mathcal{A}, \overleftrightarrow{v})$ and modified solution $(\overleftrightarrow{C}, \overleftrightarrow{u})$ with $\overleftrightarrow{C} = C$ and $lw_a^t(C) = lw_a(C)$, the* true *utility $u^t(a)$ of the manipulator $a \in \overleftrightarrow{C} \subseteq \mathcal{A}$ is*

$$u^t(a) = u(a) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, C) - sp_u(a, C)$$

Proof. According to proposition 6.2.6,

$$\begin{aligned} u^t(a) &= lw^t_a(\overleftrightarrow{C}) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, \overleftrightarrow{C}) \\ &= lw_a(C) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, C) \\ &= lw_a(C) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, C) + u(a) - u(a) \\ &= u(a) + lw_a(C) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, C) - lw_a(C) - sp_u(a, C) \\ &= u(a) + \overleftrightarrow{sp}_{\overleftrightarrow{u}}(a, C) - sp_u(a, C) \end{aligned}$$

□

Game Manipulation and the Kernel

Having established when a game manipulation is profitable for an agent in general, we now look more specifically at kernel-stable coalitions.

Therefore, recall from definition 2.2.14 that for a configuration $(\mathcal{C}, u)$, the excess of a coalition $C^* \notin \mathcal{C}$ is given by

$$e(C^*, u) := v(C^*) - \sum_{a \in C^*} u(a)$$

and that the surplus of agent a_i with respect to agent a_k with $a_i, a_k \in C \in \mathcal{C}, a_i \neq a_k$, is defined as

$$sur_{ik} := max\{e(C^*, u) \mid C^* \notin \mathcal{C}, a_i \in C^*, a_k \notin C^*\}$$

Lastly, recall that for a configuration $(\mathcal{C}, u)$ to be kernel-stable, it must hold that $\forall a_i, a_k \in C \in \mathcal{C}$:

$$\begin{aligned} &(sur_{ik} = sur_{ki}) \\ \vee&(sur_{ik} > s_{ki} \wedge u(a_k) = v(\{a_k\})) \\ \vee&(sur_{ki} > s_{ik} \wedge u(a_i) = v(\{a_i\})) \end{aligned}$$

Proposition 6.2.11. *An agent a_i in coalition $C \subseteq \mathcal{A}$ might increase its kernel-stable payoff by modifying the game in such a way that its surplus with respect to some other agent $a_k \in C$ is increased.*

Proof. The described manipulation implements the special case of corollary 6.2.7, and it is directly clear from the definition of the kernel that an increased surplus of sur_{ik} leads to a higher payoff for i. From the definitions of the surplus and excess, it is also clear that a_i has therefore to increase the value of a coalition C^*, such that $a_i \in C^*$ and $a_k \notin C^*$ and that C^* becomes (or stays) the coalition which yields the excess which is the modified surplus. □

While proposition 6.2.11 suggests a way to successfully manipulate kernel-stable coalition formation, the manipulator must be careful not to increase the value of its assigned coalition. This is shown in by following proposition:

Proposition 6.2.12. *Let* $(\mathcal{A}, v)$ *be a game,* $(\mathcal{A}, \overleftrightarrow{v})$ *be a corresponding modified game,* $r \in \mathbb{R}^+$*, and let* $C^* \subseteq \mathcal{A}$ *with*

$$\forall C \subseteq \mathcal{A} : \overleftrightarrow{v}(C) := \begin{cases} v(C) + r & \text{for } C = C^* \\ v(C) & \text{otherwise} \end{cases}$$

Further, let $(\mathcal{C}, u)$ *be a kernel stable configuration for* $(\mathcal{A}, v)$*, let* $a^* \in C^* \in \mathcal{C}$*, and let* $\overleftrightarrow{u}$ *be the modified payoff distribution such that* $\overleftrightarrow{u}(a) = u(a) + r$*,* $a \in \mathcal{A}$*, if* $a = a^*$*, and* $\overleftrightarrow{u}(a) = u(a)$ *otherwise.*

Then $(\mathcal{C}, \overleftrightarrow{u})$ *is not kernel stable for* $(\mathcal{A}, \overleftrightarrow{v})$ *if there exists an agent* $a^+ \in C^*, a^+ \neq a^*$*, such that* $sur_{a^*,a^+} - sur_{a^+,a^*} < r$ *holds in* $(\mathcal{A}, v)$

Proof. Let $\overleftrightarrow{sur}_{a_1,a_2}$ be the modified surplus of agent a_1 over agent a_2 in the modified game $(\mathcal{A}, \overleftrightarrow{v})$ and $Z := \{C | C \subseteq 2^{\mathcal{A}}, a^* \in C, a^+ \notin C\}$. Further, let $\overleftrightarrow{e}(C)$ be the modified excess of coalition C in the modified game $(\mathcal{A}, \overleftrightarrow{v})$. Then

$$\begin{aligned}
\overleftrightarrow{sur}_{a^*,a^+} &= \max_{C \in Z}\{\overleftrightarrow{e}(C)\} \\
&= \max_{C \in Z}\{v(C) - \sum_{a' \in C, a' \neq a^*} u(a') - \overleftrightarrow{u}(a^*)\} \\
&= \max_{C \in Z}\{v(C) - \sum_{a' \in C, a' \neq a^*} u(a') - (u(a^*) + r)\} \\
&= \max_{C \in Z}\{v(C) - \sum_{a' \in C} u(a')\} - r \\
&= sur_{a^*,a^+} - r < sur_{a^+,a^*}
\end{aligned}$$

But since $\overleftrightarrow{u}(a^*) > \overleftrightarrow{v}(\{a^*\}) = v(\{a^*\})$, configuration $(\mathcal{C}, \overleftrightarrow{u})$ is not kernel stable for $(\mathcal{A}, \overleftrightarrow{v})$. □

Corollary 6.2.13. *The inequality at the end of the proof of proposition 6.2.12 implies that* a^* *will have to make a side payment to* a^+ *in order to restore kernel stability. In other words,* a^* *is not "paid back" the full amount of* r*.*

Corollary 6.2.14. *If* r *is taken to be negative in proposition 6.2.12, the proof shows that* a^* *might indeed increase its surplus* sur_{a^*,a^+}*, thus obtaining a higher side-payment and increasing its profit.*

Theorem 6.2.15. *For a game $(\mathcal{A}, v)$ and solution $(\mathcal{C}, u)$, a manipulator $a \in \mathcal{A}$ profits from modifying the game in kernel-based coalition negotiations such that $(\mathcal{C}, \overleftrightarrow{u})$ results, i.e. the coalition structure remains unchanged, only if it either*

- *increases its surplus by increasing the value of an appropriate other coalition $C^* \notin C$ or*
- *decreases the value of its own coalition $C \in \mathcal{C}$, $a \in C$.*

Otherwise, a loss might be incurred by a.

Proof. The theorem follows directly from proposition 6.2.12 and corollaries 6.2.13 and 6.2.14. □

Proposition 6.2.9 and theorem 6.2.15 make it clear that a manipulator needs both prior knowledge about the game being negotiated and rather fine-grained control over coalition values to manipulate in order to manipulate profitably. Prior knowledge can only be ensured by assuming that agents do not possess it at the beginning of negotiations, and that they do not obtain such information during the negotiation and before they have to make their moves. As for the second constraint, we show in the next section that in the task agent model defined in section 6.1, only limited control over coalition value manipulation is available to the agents.

Manipulating Coalition Values in the Task Agent Model

In this section, we consider again the task agent model as defined in section 6.1 to investigate the possibilities of coalition value manipulation. Therefore, in the following, we assume that task valuations, costs and trust values (in the form of probability of success) are communicated among the agents, and that the task agent model from definition 6.1.1 applies to the participating agents. We now consider which modified local worths and coalition values might result when agents misreport any of those values.

We first consider which effect reporting modified task valuations have on the resulting modified game and the reporting agent's local worth:

Lemma 6.2.16. *Let agent i report an overstated task valuation for a task τ:*

$$\overleftrightarrow{w}_i(\tau) := w_i(\tau) + r, r \in \mathbb{R}^+$$

Then for the resulting coalition values of the modified game it holds that

$$\forall C,\ i \in C:\ \exists r_C \in [0, r]:\ \overleftrightarrow{\underline{v}}(C) = \underline{v}(C) + r_C \textit{ and}$$
$$\exists r' \in [0, r]:\ \overleftrightarrow{\underline{lw}}_i(C) = \underline{lw}_i(C) + r'$$

Proof. Let α^*_C and $\overleftrightarrow{\alpha}^*_C$ be the task allocations established by the TKCF for C, $i \in C$, in $(\mathcal{A}, v)$ and $(\mathcal{A}, \overleftrightarrow{\underline{v}})$, respectively. Then, if $\tau_i \in \alpha^*_C$, increasing the valuation of τ will not have an effect on the allocation as it was already assumed to be optimal. Formally, in this case,

$$\begin{aligned}
\alpha^*_C &= \overleftrightarrow{\alpha}^*_C \text{ with} \\
\overleftrightarrow{\underline{w}}_i(C) &= \sum_{\pi^j_i \in \alpha^*_C \setminus \tau} trust^j_i \cdot w_i(\pi) + trust^j_i \cdot \overleftrightarrow{w}_i(\tau) \\
&= \sum_{\pi^j_i \in \alpha^*_C \setminus \tau} trust^j_i \cdot w_i(\pi) + trust^j_i \cdot (w_i(\tau) + r) \\
&= \sum_{\pi^j_i \in \alpha^*_C} trust^j_i \cdot w_i(\pi) + trust^j_i \cdot r \\
&= \underline{w}_i(C) + trust^j_i \cdot r \\
&\leq \underline{w}_i(C) + r \text{ (because } trust^j_i \leq 1\text{)}
\end{aligned}$$

However, if $\tau_i \notin \alpha^*_C$, then we have two possible cases:

1. r is too small to let τ_i be included in $\overleftrightarrow{\alpha}^*_C$, and thus

$$\alpha^*_C = \overleftrightarrow{\alpha}^*_C \text{ with } \overleftrightarrow{\underline{w}}_i(C) = \underline{w}_i(C)$$

2. r is big enough to let τ_i be included in $\overleftrightarrow{\alpha}^*_C$. Then it might either be the case that no other tasks are affected, which might happen if $v_i(\tau) < 0$ and the coalition possesses still enough resources to execute τ_i additionally, such that again

$$\alpha^*_C = \overleftrightarrow{\alpha}^*_C \text{ with } \overleftrightarrow{\underline{w}}_i(C) \leq \underline{w}_i(C) + r$$

Or there exist some other tasks assignments in α^*_C but not in $\overleftrightarrow{\alpha}^*_C$, whose valuations therefore do not contribute to $\overleftrightarrow{\underline{v}}(C)$. However, their sum must be $\leq r$ or otherwise $\overleftrightarrow{\alpha}^*_C$ would not be optimal. Therefore (the inequality again comes from the trust values)

$$\overleftrightarrow{\underline{v}}(C) \leq \underline{v}(C) + r_C$$

In particular, this might include also tasks that i additionally requests. Thus it holds that

$$\exists r^{'} \in [0, r] : \overleftrightarrow{\underline{w}}_i(C) \leq \underline{w}_i(C) + r^{'}$$

From the above cases combined and the fact that the coalition value is the sum of the local worths, it then follows that the lemma is true. □

Remark 6.2.17. *Note that in lemma 6.2.16, each coalition value (of coalitions including i) might change differently, depending on the optimal task allocations, in case where resources are bounded or there exist multiple agents offering the same task at different costs.*

Corollary 6.2.18. *From lemma 6.2.16 it follows that if agent i reports an understated task valuation, i.e. $r \in \mathbb{R}^-$, then*

$$\forall C, \ i \in C : \ \exists r_C \in [r, 0] : \ \overleftrightarrow{\underline{v}}(C) = \underline{v}(C) + r_C$$

The arguments in the proof just have to be reversed: if $\tau_i \in \alpha^*_C$, either r is not small enough for τ_i to be not included in $\overleftrightarrow{\alpha}^*_C$, such that $\overleftrightarrow{\underline{w}}_i(C) = \underline{w}_i(C)$. Or it is small enough such that $\tau_i \notin \overleftrightarrow{\alpha}^*_C$. In that case, either analogously (r being negative) $\overleftrightarrow{\underline{w}}_i(C) = \underline{w}_i(C) + trust^j_i \cdot r \geq \underline{w}_i(C) + r$, if no other tasks can then be included in $\overleftrightarrow{\alpha}^*_C$. Or other tasks can then be allocated, e.g. due to freed resources from the removal of τ_i, such that they contribute a value $r \in [r0]$ to $\overleftrightarrow{\underline{v}}(C)$. Finally, if $\tau_i \notin \alpha^*_C$, then also $\tau_i \notin \overleftrightarrow{\alpha}^*_C$.

Corollary 6.2.19. *From lemma 6.2.16 and corollary 6.2.18 it further follows symmetrically for costs: if agent i understates its cost for a task assignment τ^i, i.e. $\overleftrightarrow{c}_i(\tau) := c_i(\tau) - r, r \in \mathbb{R}^+$ then*

$$\forall C, \ i \in C : \ \exists r_C \in [0, r] : \ \overleftrightarrow{\underline{v}}(C) = \underline{v}(C) + r_C$$

And, similarly, if it overstates the cost with $r \in \mathbb{R}^-$, then

$$\forall C, \ i \in C : \ \exists r_C \in [r, 0] : \ \overleftrightarrow{\underline{v}}(C) = \underline{v}(C) + r_C$$

To summarize, the results of this section show that an agent has only limited control over single coalition values. However, it is also clear that some such control exists. Therefore, in the next section, the proposed coalition formation algorithm employs cryptographic techniques in order to make it hard for the agents to obtain an unfair information advantage which could help agent to profitable manipulate the negotiations.

6.3 Algorithm TKCF

In this section we describe the coalition formation algorithm TKCF. It employs the task agent model from definition 6.1.1 and the generic trust model from definition 6.1.4. It consists of four parts, which we describe in turn.

Communication

This covers the protocol agents use to exchange valuations, costs, and trust values with one another so that no information asymmetry can exist among them such that one agent can find exploit another (which would make the mechanism unattractive to potential participants). Perhaps the easiest way of achieving this is to ensure that all agents get information about these variables at the same time. Otherwise, agents can simply wait for messages about other agents' valuations and costs, analyse these and, in turn, transmit their own valuations and costs such that the latter exploit the agents that have already transmitted their private information. To achieve such simultaneous information revelation, we adapt the common Data Encryption Standard (DES) cryptographic technique Schneier (1996) to build our communication protocol (any other encryption method can be easily substituted, which might be advisable because DES is known to have weaknesses). Specifically, we assume that each agent has a unique key e_i (randomly chosen) that allows it to encrypt a message (e.g. containing information about valuations and costs) using a commonly known function enc. The message can only be decrypted using that key and inverting the function enc^{-1}. The protocol is as follows:

1. All agents transmit $enc(\langle \boldsymbol{\tau}, v_i, c_i, \boldsymbol{\eta}_i, g_i(.)\rangle, e_i)$. This means that they encrypt their private information with their key e_i. Then, this encrypted message is sent to all agents directly (it is reasonable to assume here that all agents are directly connected to each other).

2. All agents confirm to all other agents that they have received all encrypted messages from all the other agents. This means that for I agents, each one needs to receive $I-1$ encrypted messages and send a confirmation of this to all others.

3. When $I-1$ confirmations (of the reception of $I-1$ messages) have been received by each agent, all agents send their key e_i to all agents in the population. Then all agents can use this key to decrypt received messages simultaneously using $enc^{-1}(enc(\langle \boldsymbol{\tau}, v_i, c_i, \boldsymbol{\eta}_i, g_i(.)\rangle), e_i)) = \langle \boldsymbol{\tau}, v_i, c_i, \boldsymbol{\eta}_i, g_i(.)\rangle$.

The above protocol guarantees that there is no information asymmetry between any pair of agents in the population. Note that the agents need to obtain $I-1$ confirmations before sending their keys since, doing otherwise, results in an information asymmetry that could lead to agents being exploited. For example, let agent A send its (encrypted) private information to agents B and C, while B sends its private information to A and C, and C only sends its private information to B. Then, let A send its key to B, and B responds by sending its key to A. C then sends its key to B and gets B's key in return. Now, C can analyse its own information and B's information in order to select valuations, costs, and trust vectors that could allow it to exploit unfairly both A and B. This happens because C can calculate what it can profitably reveal (i.e. its valuations and costs) to A since A does not already possess C's encrypted private information while C already has A's private information which it can no more change (i.e. there exists an information asymmetry). To avoid this, our protocol forces agents to wait for $I-1$ confirmations each time private information is shared, and ensures that all agents have the same information.

Kernel Stable Solution Computation

We now provide a protocol that lets the agents achieve a Kernel-stable configuration given the information they obtained by executing the communication protocol of section 6.3. As has been stated in section 6.1, there generally exist multiple coalition structures for which Kernel-stable solutions can be found. In the proposed protocol, a coalition structure which maximizes the sum of the values (sometimes also called *social welfare*) of the formed coalitions is chosen. We consider this a favorable approach with respect to the experimental evaluation (see section 6.5), because it enables us to compare the quality of the generated coalition structures to a theoretical optimum. But there exist also other, more individual agent or coalition centric coalition structure generation approaches (as e.g. proposed in Shehory and Kraus (1998)).

Now, since there might exist multiple optimal coalition structures, task assignments in individual coalitions, and kernel stable payoff distributions for a given coalition structure we introduce a function to allow the agents to jointly make unambiguous choices. We therein assume that each agent possesses a strictly ordered list $L^{\mathcal{A}}$ of all agents in $\mathcal{A}$. This list could, for example, be obtained by the agents' joining order in the system, but since the exact method is not important here, we simply consider it as given.

Let p be a task assignment, coalition structure, or payoff distribution, let p_i denote that p was computed by agent a_i and let $P := \langle p_1, \ldots, p_{|\mathcal{A}|}\rangle$. Then let $Choose(P)$ return p which was computed by the greatest number of agents. If there are more than one such elements, among them choose the one which was computed by an agent which is considered lowest by $L^{\mathcal{A}}$. To achieve a kernel stable configuration which maximizes the sum of the coalition values, each agent $a_i \in \mathcal{A}$ performs:

1. Determine trust values $trust_i^j$ of a_i in other agents a_j where $a_i, a_j \in \mathcal{A}$.
2. Compute expected coalition values; for each $C \subseteq \mathcal{A}$ do:
   (a) Compute an optimal task allocation α_C^i for C and send it to each other agent $a_j \in C$; receive all α_{C_j}. $P^\alpha := \langle \alpha_{C_1}, \ldots, \alpha_{C_{|\mathcal{A}|}}\rangle$; determine $\alpha_C^* := Choose(P^\alpha)$.
   (b) Given the trust values about agents, for each coalition $C \in \mathcal{C}$ assess the expected local worths $\underline{lw}_k(C)$ according to definition 6.1.5 for all agents in $a_k \in C$.
   (c) Compute the overall expected coalition value $\underline{\underline{v}}(C)$.
3. Find a coalition structure $\mathcal{C}_i$ such that $\sum_{C \in \mathcal{C}_i} \underline{\underline{v}}(C)$ is maximised. Send $\mathcal{C}_i$ to each other agent $a_j \in \mathcal{A}$ and receive all other $\mathcal{C}_j$. Let $P^{\mathcal{C}} := \langle \mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{A}|}\rangle$; determine $\mathcal{C} := Choose(P^{\mathcal{C}})$
4. Compute a kernel stable payoff distribution $\boldsymbol{u}_i$ for $\mathcal{C}$.
5. Send $\boldsymbol{u}_i$ to all other agents j and receive all $\boldsymbol{u}_j$.
6. Determine $\boldsymbol{u} := Choose(P^{\boldsymbol{u}})$

After completing the execution of the TKCF, each agent is assigned to a coalition and a payoff, which completes the coalition formation process. However, these coalitions and payoffs still have to be implemented in order to actually realize the solution. While the task execution performance of the agents is measured via the trust values, it is still unclear how to enforce the execution of the side-payments resulting from the solution. This is covered in the following section.

Payment Execution

We now develop a payment protocol which provides the incentives to the agents to faithfully implement it so as to ensure that each agent $a_i \in \mathcal{A}$ derives the payment m_i.

Our protocol initially involves the creation of $|\mathcal{C}|$ strictly ordered lists for each coalition in the stable configuration computed in section 6.3. With $PL^C = \{a_1, \ldots, a_k, \ldots, a_K\}$ we denote the list of all agents in a coalition C (hence $K = |C|$) with agents sorted in descending order of the difference, $u(a_i) - lw_i(C)$. Ties are broken such that an agent in PL^C gets a higher index if it has a higher index in the list $L^{\mathcal{A}}$. Thus agent a_1 in PL^C corresponds to the agent which has the maximum $u(a_i) - lw_i(C)$. Since all information required to form this list has already been transmitted in the communication stage (described in section 6.3), PL^C is commonly known to all agents in $|C|$. Now, our protocol intuitively works by cascading payments between agents, with an agent providing a payment before it receives one. The sorted list allows us to condition payments such that agents always make positive transfers to each other. The transfer m^k_{k+1} each agent a_{k+1} makes to agent a_k is computed as:

$$m^k_{k+1} = u(a_k) - lw_k(C) + m^{k-1}_k \tag{6.1}$$

The following specification of the payment protocol is designed for the case when $|C| \geq 3$ (figure 6.2 graphically depicts the protocol when all agents implement it faithfully with each step below corresponding to the labelled steps in the figure). Note that the payment protocols for the cases when $|C| \leq 2$ are trivial. When $|C| = 1$, no transfers occur and when $|C| = 2$ a single transfer occurs between the two agents.

1. The protocol is initiated by agent a_K sending an encrypted but verifiable payment, $enc(m^{K-1}_K, e'_K)$, to agent a_{K-1}. That is, agent a_{K-1} can check the amount but cannot access it. This is what secure digital cash achieves and can be intuitively seen as an

unbreakable glass safe Schneier (1996). Agent a_{K-1} then broadcasts the message $\langle start\_payment \rangle$ to all agents in the list if the value of the encrypted transfer from agent a_K to a_{K-1} is according to equation 6.1. Otherwise, agent a_{K-1} transmits $\langle \widehat{m}_K^{K-1}\_rec \rangle$ (which means payment $\widehat{m}_K^{K-1}$ has been received) and the coalition dissolves and a new coalition structure is computed without agent a_K.

2. Each agent a_{k+1} ($a_k \in PL^C \setminus a_K$) then pays agent a_k according to equation 6.1 if it receives the message $\langle m_k^{k-1}\_rec \rangle$ from agent a_{k-1}. Otherwise, if it receives message $\langle \widehat{m}_k^{k-1}\_rec \rangle$ where $\widehat{m}_k^{k-1} \neq m_k^{k-1}$, it then decides according to whether it has also received a message $\langle \widehat{m}_{k-1}^{k-2}\_rec \rangle$ from agent a_{k-2}. If it has received such a message and $m_{k-1}^{k-2} - \widehat{m}_{k-1}^{k-2} + \delta = m_k^{k-1} - \widehat{m}_k^{k-1}$, it then implements the transfer according to equation 6.1. Otherwise, it then implements the following transfer:

$$m_{k+1}^k = u_k - w_k(C) + \widehat{m}_k^{k-1} - \delta \tag{6.2}$$

where $\delta \in \Re^+$ is a penalty applied for wrong payment (which may happen if the agent is irrational). The transfer $m_2 1$ is initialised to be $u(a_1) - \underline{lw}_1(C)$.

3. Upon receipt of payment m_{k+1}^k, each agent a_k ($a_k \in PL^C \setminus a_K$) transmits message $\langle m_k^{k-1}\_ \rangle$ to agent a_{k+2}. However, if the payment received is $\widehat{m}_{k+1}^k$ where $\widehat{m}_{k+1}^k \neq m_k^{k-1}$, agent a_k then transmits $\langle \widehat{m}_{k+1}^k\_rec \rangle$ to both agents a_{k+2} and a_{k+3}.

4. The protocol is different for the last three agents since these agents control the message which will start the task execution stage. If agent a_K receives the message $\langle m_{K-1}^{K-2}\_rec \rangle$ (or if it receives message $\langle \widehat{m}_k^{k-1}\_rec \rangle$ from a_{K-2} and it also receives $\langle \widehat{m}_{K-2}^{K-3}\_rec \rangle$ from agent a_{K-3} and $m_{K-1}^{K-2} - \widehat{m}_{K-1}^{K-2} - \delta = m_{K-2}^{K-3} - \widehat{m}_{K-2}^{K-3}$) it then transmits the key and broadcasts the message $\langle key\_sent \rangle$. If $lw_{K-1}(C) \geq 0$, agent a_K transmits the key to agent a_{K-1} who then broadcasts the message $\langle start\_tasks \rangle$. Otherwise, it then transmits the key to the agent a_n such that $lw_n(C) \geq 0$ and has the highest index in PL^C . This agent then transmits the key to agent a_{K-1} and broadcasts the message $\langle start\_tasks \rangle$. If ever agent a_K receives $\langle \widehat{m}_k^{k-1}\_rec \rangle$ and it detects a deviation by a_{K-1}, agent a_K then broadcasts the message $\langle no\_key\_sent \rangle$.

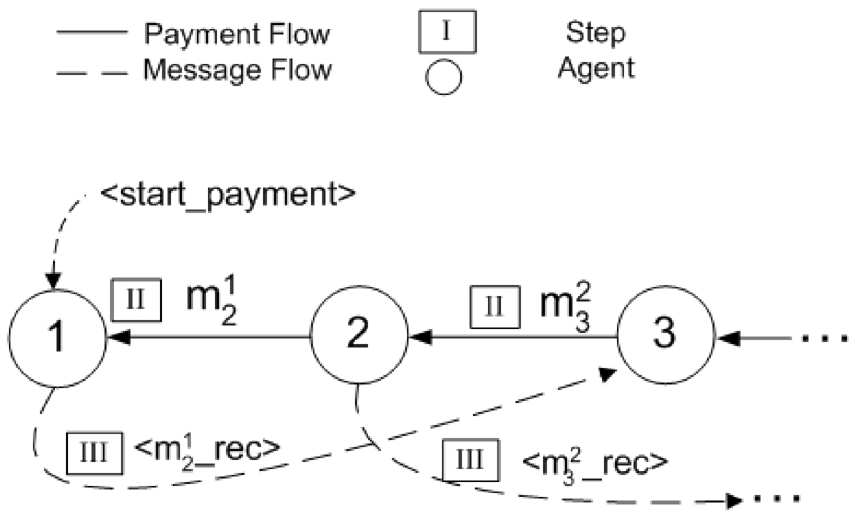

Figure 6.1: For agents $a_1, \ldots, a_{K-3}$

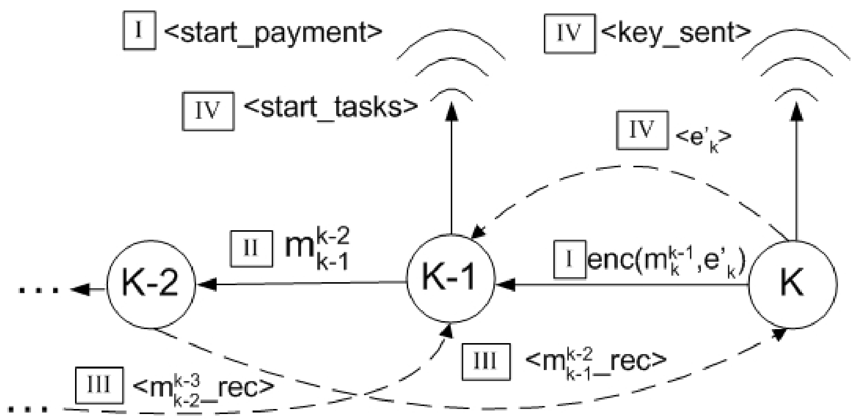

Figure 6.2: For agents a_{K-2}, a_{K-1} and a_K

In theorem 6.4.3, it is shown that a rational agent would not find it in its best interest to deviate from the payment protocol, i.e. it will implement the payments specified by equation 6.1 and would not send erroneous messages once it has received the payments.

For now, however, we first complete the TKCF protocol with specifying the task execution phase.

Task Execution

Once the payment execution phase is completed (i.e. after the agents have received the two broadcasted messages $\langle key\_sent \rangle$ and $\langle start\_tasks \rangle$ or the single message $\langle no\_key\_sent \rangle$), the agents start performing their tasks. All agents deriving value from a task $\tau \in \mathcal{T}$ then measure the POS values of the respective executing agents, and the next round of CF starts.

6.4 TKCF Properties

Theorem 6.4.1. *Assuming no prior knowledge of the agents about each other, in the TKCF, no agent a is able to determine values $\overleftrightarrow{x} \neq x$ to report that will unjustifiably increase a's payoff with certainty with respect to the original game $(\mathcal{A}, \underline{v})$.*

Proof. The encrypted communication protocol at the start of the TKCF ensures that all agents have to report their values before they learn about reported values from other agents. Therefore, with no prior knowledge about each other, proposition 6.2.9 and theorem 6.2.15 it is clear that no agent is able to determine how to profitably manipulate the game. □

Remark 6.4.2. *We further argue that proposition 6.2.9, theorem 6.2.15, lemma 6.2.16 and corollaries 6.2.18 and 6.2.19 together suggest that agents might actually be allowed some limited prior knowledge without being able to determine how to profitably manipulate the game. However, the exact amount or kind of prior knowledge which is sufficient to make profitable manipulation certainly or at least expectedly achievable remains unknown for now.*

Theorem 6.4.3. *It is rational for each agent to follow the payment scheme.*

Proof. (Sketch) We prove the above theorem by comparing the utility that an agent derives when following the protocol to that when it deviates.

The net utility an agent derives when following the protocol faithfully is its payoff which can be rewritten from equation 6.1 as :

$$u_k(C) = w_k + \begin{cases} m_2^1, & k = 1 \\ m_{k+1}^k - m_k^{k-1}, & \forall k \in PL^C \setminus \{K, 1\} \\ -m_K^{K-1}, & k = K \end{cases} \tag{6.3}$$

Now consider each agent's opportunity to defect as the protocol proceeds (assuming all other agents have followed it till that point). At the beginning, agent a_K can deviate by not sending the correct value in the encrypted payment. Then, this is detected by agent a_{K-1} and thus the coalition does not start. Agent a_K then derives a utility of $v(\{a_K\})$ ($v(\{a_K\}) \leq u(a_K)$ by the definition of kernel stability) and thus will not deviate.

On the second step, agent a_{K-1} may deviate by not acknowledging the payment and not sending the $\langle start\_payment \rangle$ message. But then, it will not be able to decrypt and make use of the received payment. Thus its net payoff would be $v(\{a_K\})$, not beneficial for agent a_{K-1}.

On the third step, agent a_2 can deviate by sending an incorrect payment, $\widehat{m}_2^1$, to agent a_1. In this case, agent a_1 sends to agent a_3 and agent a_4 the message $\langle \widehat{m}_2^1\_rec \rangle$ and agent a_3 then pays agent a_2 the amount $u(a_k) - lw_k(C) + \widehat{m}_2^1 - \delta$. As a result, the net transfer to agent a_2 is $u(a_k) - lw_k(C) - \delta$ which is strictly less than in equation 6.3. Thus, agent a_2 cannot benefit by providing a payment $\widehat{m}_2^1 \neq m1_2$. Notice also that by the protocol, agent a_3 derives a benefit of δ when applying the correct penalty and will not get charged by agent a_4 who has been informed of agent a_2's deviation. However, if agent a_3 deviates and does not apply the correct penalty, then agent a_4 will also penalise it. Notice also that if agent a_1 receives the correct payment, it can still deviate by misreporting this payment. Furthermore, the agent is indifferent between all the messages it can send (in a scenario where the coalition game is run only once) once it has received its correct payment. However, in a repeated coalition game (which is the case we consider here), this would amount to penalising a good payer or not penalising a bad payer, which is clearly not what an agent would like to do here. The same argument as used for agents a_1 and a_2 can now be used for all other payments between agents until agent a_{K-1}.

Now if agent a_{K-1} deviates when paying, then agent a_{K-2} will report this deviation to agent a_K who will withhold the key. Then agent a_{K-1} will derive a net payment of $-\widehat{m}_{K-1}^{K-2}$ which is less than the amount it derives in equation 6.3. Agent a_K can also deviate by sending the wrong key. In this case agent a_K does not derive any higher utility by so doing. Finally the agent who has to send the message $\langle start\_tasks \rangle$ can deviate by not sending it. However, the agent sending it (either agent a_{K-1} or some other agent) would not find any utility in doing so since it gains a positive $lw_k(C)$ when the coalition tasks are performed. □

6.5 Experimental Evaluation

Having ensured that TKCF incentivises agents to reveal their true costs and valuations and that they execute the payments that are due, agents have to rely on the trust model to assess each other's performance correctly. As mentioned before, the trust model itself is not the focus of this chapter. Nevertheless, we include some evaluation results (which have been obtained collaboratively with the other authors of Blankenburg et al. (2005)) for completeness.

The aim is to see whether the TKCF can use the trust model defined in section 6.1.4 in order to evaluate the reliability of agents over multiple interactions. Here we consider a super-additive game, but restrict the maximum coalition size (which remains non-trivial) in order to analyze the TKCF's behaviour when finding an optimal coalition structure. This size is fixed to half of the number of agents in our case. The agents' valuations and costs are taken from a uniform distribution between 0 and 1. The agents' POS are determined *a priori* and their actual success after each coalition executes tasks is taken from a uniform distribution whose mean is equal to their POS. Then, according to our trust model, the agents' reported POS in each other are summed using a weight vector to give the actual trust values. Given this, a number of agents, six in this case, are allowed to form coalitions of a maximum of 3 agents. To simplify the analysis, each agent is allowed to execute more than one task and asks for only one task to be completed. However, agents might request different tasks and vary valuations and costs in each game. Thus, in each iteration, a solution to a possibly different game is to be found. Although this might increase the number of iterations until the correct POS are determined, and thus the correct solutions are found, we consider this a more realistic situation than the case of repeating just one game all the time.

Given that the payoffs described in section 6.3 are calculated according to the expected value (resulting from the trustworthiness of agents) of coalitions (see definition 6.1.5), we postulate the following hypothesis:

H1:*The agents' payoffs converge to those reflecting their actual POS in the long run. Given this, the coalition structure $\mathcal{C}$ chosen converges to $\mathcal{C}^*$ which maximises the overall value $\boldsymbol{v}(\mathcal{C}) = \sum_{C \in \mathcal{C}} v(C)$.* To test this hypothesis we performed an experiment given the above settings and recorded each agent's payoff and determined the ratio $\frac{|u_i^* - u_i|}{u_i}$ which indicates the distance of the calculated payoff u_i from the exact payoff u_i^*. We also recorded the ratio $\frac{\boldsymbol{v}(\mathcal{C})}{\boldsymbol{v}(\mathcal{C}^*)}$ to check whether we actually chose the most valuable coalition structures. We repeated the CF game 200 times over which trust measures were refined each time the tasks were executed. The results are shown on figure 6.3. As can be seen, the difference between the payoffs converge to 0 indicating that the exact payoffs are chosen in the long run. Moreover, an optimal coalition structure ($\mathcal{C}^*$) is chosen well before the payoffs stabilise (when the trust is exactly determined after 200 interactions). This means that even though an optimal coalition structure has been chosen (after around 167 interactions in this case), the payoffs are still affected by slight deviations of the trust perceived by agents.We used ANOVA (Analysis Of VAriance) to determine whether there were any significant differences between means of $\frac{|u^* - u|}{u^*}$ of the agents. Thus, it was found that for 10 samples of 200 games that $p = 0.5534$ for $\alpha = 0.5$ such that $p > \alpha$ and the null hypothesis is validated. Also, for the value of $\frac{\boldsymbol{v}(\mathcal{C})}{\boldsymbol{v}(\mathcal{C}^*)}$, it was found that $p = 0.0182$ for $\alpha = 0.1$. This validates the null hypothesis in this case since $p > \alpha$, which tells us there is no significant difference between the means of the samples.

We can also note that the results show that as trust is being learnt by all agents, the agents' payoffs may, at times, significantly diverge from the optimal ones (the spikes in the graph), though the size of this occasional divergence decreases over time (due to more precises trust values). In such cases, the spikes are due to u_i^* being very low compared to the difference $u_i^* - u_i$. These, in turn, are due to the sensitivity of the kernel-based payoffs to slight changes in the trust values.

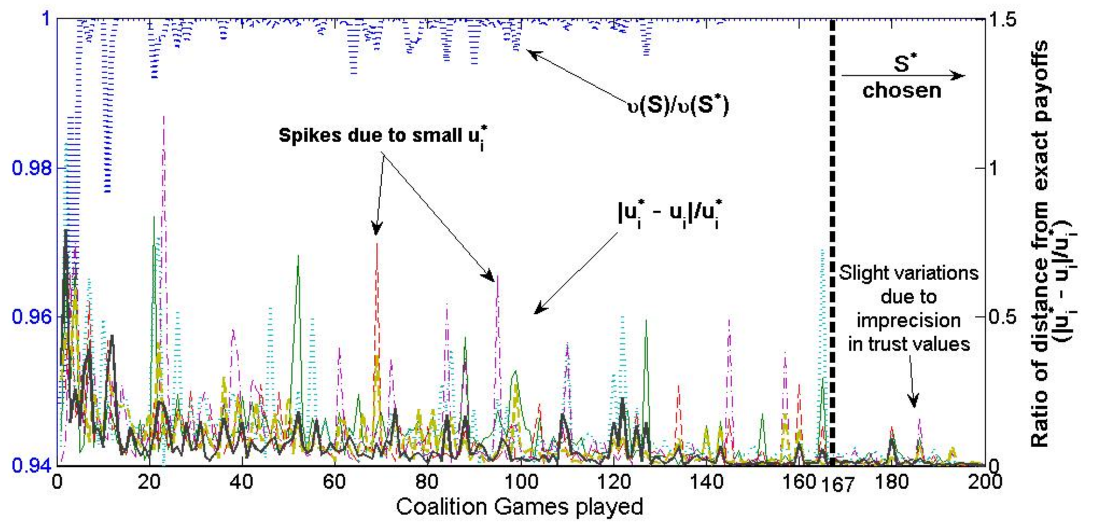

Figure 6.3: Computing stable payoffs as trust values converge.

The convergence of the TKCF might seem slow, but taking into account that different games are played and thus different coalitions are formed in each iteration, we consider the result at least reasonable.

6.6 Summary

In the task allocation via coalition formation domain, we proposed a novel model to compute expected coalition values that account for agents' trust in each others' ability to execute tasks with satisfactory reliability. Instead of specifying a particular trust model, we identified necessary properties of trust models in general in order that they can be soundly applied within this context. Thus, any trust model exhibiting these properties can be used.

We further presented a protocol that allows the agents, based on the expected coalition values, to form kernel stable coalitions. The protocol accounts for every step in the coalition formation process from the communication of individual valuations and costs to the actual execution of side payments and tasks, as well as updating of the trust values. It was experimentally shown that for the realistic case of repeated games with varying task requests, valuations and costs, the computed solutions over time converge to their theoretical optimum. Moreover, it was formally shown that for all communications and payments required by this protocol, it is not rational for any agent to deviate from what we specify. To achieve this, we applied encryption-based communication techniques and developed a sequential payment protocol.

In our proposed mechanism, expected optimal coalition structures and kernel stable solutions are computed and this involves exponential complexities. This was done in order to demonstrate convergence to the theoretical optimum in the experiments. However, we believe that none of our results actually depend on this property, and that polynomial kernel based coalition formation can equally be applied. However further work is needed to confirm this conjecture.

Chapter 7

Privacy Preserving Coalition Formation

In the previous chapters, we covered the general issues of uncertainty and trust of multiagent coalition formation. Here, we tackle a problem which has not received much attention yet in the literature: privacy preservation in coalition formation.

When agents representing different, unrelated, independent and unacquainted entities such as individual users and service-offering companies engage in online negotiation, it might be unacceptable for an agent that others learn which services are accessed and which utilities are achieved. This is especially a problem for algorithms based on cooperative game theory since the computation of a classic solution requires complete knowledge of the game. Example applications that would benefit or even require a privacy preserving coalition formation protocol include health care web service agents, which form coalitions e.g. to automatically handle insurance issues, transportation, hospital and medical personal assignments. But an agent responsible for transportation should not need to know which patients are assigned to which doctors.

In this chapter, we thus present a coalition formation algorithm and protocol BSCA-P for service agents which overcomes such privacy issues. More precisely, we show that distributing the coalition value by application of the recursively bilateral Shapley value, agents are not required to reveal their payoffs and particularly their self values. However, as it turns out, the existence of service requests might not be completely hidden in certain cases. As a resort to this problem, we propose to anonymize service accesses, as well as a simple means to measure this anonymity. This allows the agents to individually specify minimum degrees for these anonymities, which might also account for particular other agents.

An additional effect of the utilized anonymous routing protocol is that all input/output data for service accesses can also be hidden from all agents except the recipient, since the protocol relies on message encryption. This ensures that the even during the execution phase, the established anonymity degrees can be preserved.

Preliminary versions of the BSCA-P have been previously published in Blankenburg and Klusch (2005a) and Blankenburg and Klusch (2006).

Additionally, we provide a section on some basic properties of the kernel which are relevant for privacy preservation. In particular, we show that agents can theoretically hide their self values completely if an appropriate coalition formation algorithm and protocol are employed. This section is based on Blankenburg and Klusch (2004).

The remainder of this chapter is organized as follows: in section 7.1 we briefly introduce a simplified service agent model and coalition game that we use in this chapter. In section 7.2 we show how this model can be exploited to negotiate coalitions while hiding information about coalition values, local worths and payoffs, but also show that the existence of service requests might not be hidden completely in general. In section 7.3 we then alliviate this problem by introducing means to let the agents at least stay anonymous to some degrees, and adopt an anonymous routing protocol to enable anonymous service access. Based on these results, we propose and discuss the coalition formation protocol BSCA-P in section 7.4. In section 7.6 we additionally show some basic properties of the kernel which are relevant for privacy preservation. Finally, we summarize in section 7.7.

7.1 Preliminaries

Since our focus is the design of a coalition formation algorithm and protocol that preserves privacy, we employ a service agent model where requests and offers are explicitly modeled. We therein assume that only primitive services are offered, and no composed ones. However, the requesting agents might still compute and employ compositions of the offered services.

Definition 7.1.1 Simplified Service Agent Model
We define

- OS_a as the the set of (primitive) services offered by agent a,
- $OS_C := \bigcup OS_a,\ a \in C$, as the the set of (primitive) services offered by all agents in coalition C,
- R_a as the the set of all services requested by agent a,
- $KR_a(C) \subseteq \bigcup_{a^* \in C} R_{a'}$ as the subset of requests by agents a' in coalition C which agent $a \in C$ *knows about*,
- $E_a(C) \subseteq OS_a \cap KR_a(C)$ as the the set of services that are to be executed by a in coalition C,
- $R_a(C) \subseteq OS_C \cap R_a$ as the set of services which are executed by specific agents in coalition C, for which we also say that these services are *accessed* by a,
- $w_a(ws)$, $ws \in R_a$, as the agent a's valuation for the execution of service ws and
- $c_a(ws)$, $ws \in OS_a$, as the agent a's cost for the execution of service ws.

It is further assumed that

- $\forall ws \in R_a \cap E_{a'}(C) :\ w_a(ws) \geq c_{a'}(ws)$, i.e. the agents abstain from service execution with negative net value.
- the agents have no resource constraints, i.e. that each offered service might be executed multiple times in a coalition.

△

Corollary 7.1.2. *From the general definition 2.2.27 of the local worth, it follows that the local worth in the simplified service agent model is given by*

$$lw_a(C) = \sum_{ws \in R_a(C)} w_a(ws) - \sum_{ws \in E_a(C)} c_a(ws)$$

We provide a simple example game which will be used as a running example throughout this chapter:

Example 7.1.3
Consider a 3-agent coalition game in the simplified service agent model as shown in figure 7.1. Service agent a_1, for example, offers its own web service ws_1 to any other known agent of the game, that

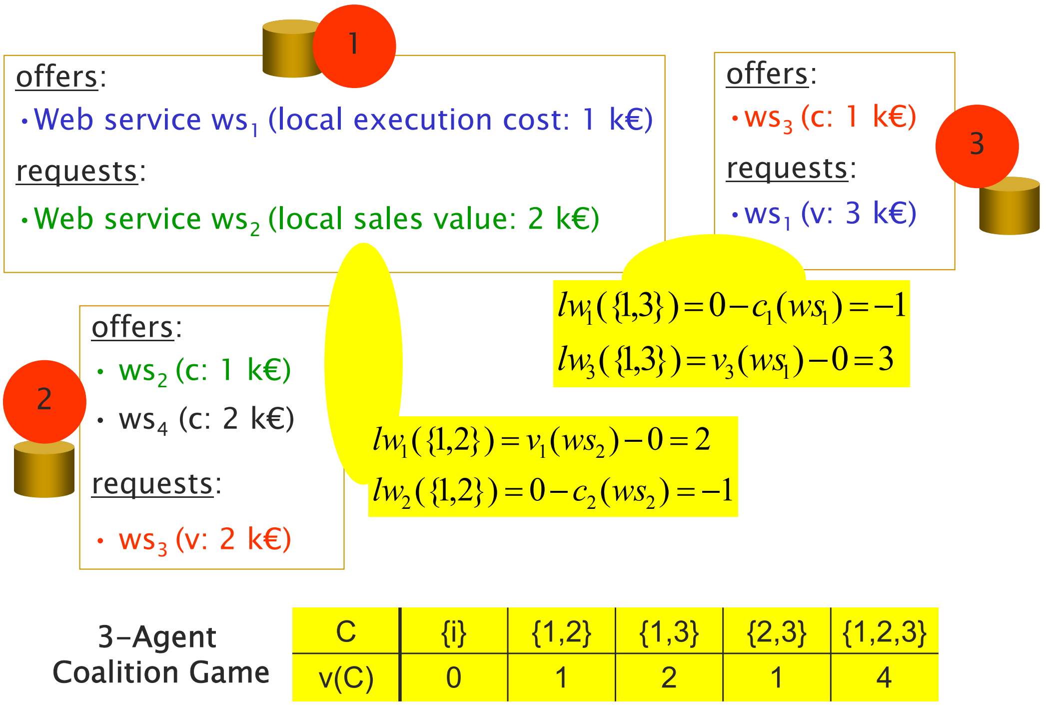

Figure 7.1: Example coalition game for three web service agents.

are service agents a_2 and a_3. Each local execution of its service would cost a_1 an amount of 1ke, but produces no monetary income as it is of no relevance for its own users. Hence, its self value $v(a_1)$ is zero.

Agent a_3 is requesting access to service ws_1 from a_1, as it can charge its local users with an total amount of 3ke per use, but does not offer any service of interest for users of a_1 in turn. As a consequence, the local worth of a_1 in a joint coalition with a_3 is $lw_{a_1}(C_1) = -c_{a_1}(ws_1) = -1$ whereas that of a_3 is $lw_{a_3}(C_1) = -c_{a_3}(ws_1) = 3$. $\triangle$

As stability concept, we employ the recursively bilateral Shapley value. Remember that the bilateral Shapley value, as introduced in definition 2.2.18, is

$$\sigma_b(C_i, v) = \frac{1}{2}v(C_i) + \frac{1}{2}(v(C_1 \cup C_2) - v(C_k)),\ k \in \{1,2\}, k \neq i$$

Further, recall that the recursively bilateral as presented in defini-

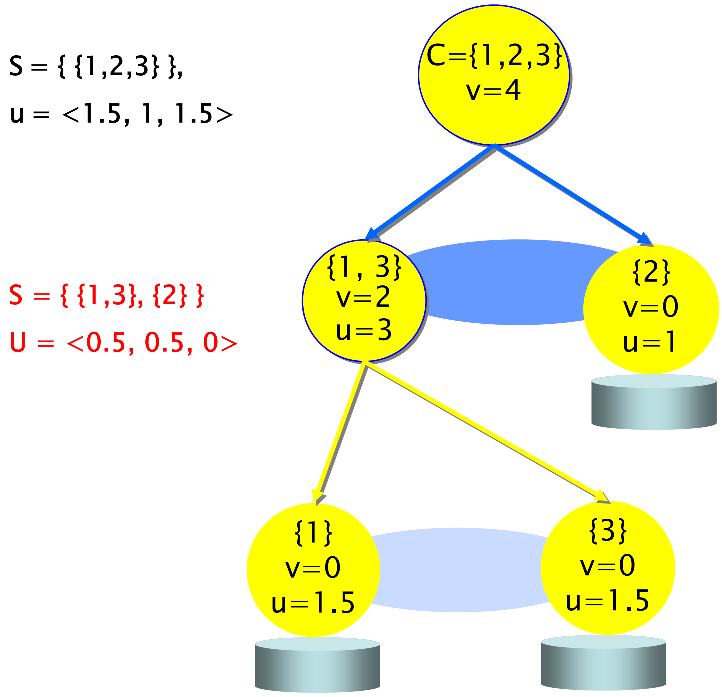

Figure 7.2: Binary tree of bilateral coalitions for the example game.

tion 2.2.21, is defined as

$$u(C_i^*) = \sigma_b(C^*,\, C_i^*,\, v_{C^*}),\;\; i \in 1,2, \text{ with}$$

$$\forall C^{**} \subseteq \mathcal{A}: \; v_{C^*}(C^{**}) = \begin{cases} \sigma_b(C^p,\, C_k^p,\, v_{C^p}) & \text{if } C^{**} = C^*, C^p \in T_C, \\ & C^* = C_k^p, k \in 1,2 \\ v(C^{**}) & \text{otherwise} \end{cases}$$

Example 7.1.4
In the game of example 7.1.3, consider the bilateral coalition $C_1 = \{1\} \cup \{3\}$. Since $v(\{1\}) = v(\{3\}) = 0$, it holds that $\sigma_b(\{1\}, \{1\} \cup \{3\}, v) = \sigma_b(\{1\}, \{1\} \cup \{3\}, v) = 0 + \frac{1}{2}(2-0) = 1$. Merging of C_1 with $C_2 = \{2\}$ ($C = C_1 \cup C_2$) yields $v(C) = 4$ and $v(C_2) = 0$, thus $\sigma_b(C_1, C, v) = 2 + \frac{1}{2}(4-2) = 3$ and $\sigma_b(C_2, C, v) = 0 + \frac{1}{2}(4-2) = 1$. Recursively replacing the coalition value $v(C_i)$ in (5) with the bilateral Shapley value of C_i then leads to the following payoff distribution (cf. figure 7.2): $u(a_1) = \sigma_b(\{1\}, \{1\} \cup \{3\}, v^*) = 0 + \frac{1}{2}(3-0) = 1.5$ and $u(a_3) = \sigma_b(\{3\}, \{1\} \cup \{3\}, v^*) = 0 + \frac{1}{2}(3-0) = 1.5$. $\triangle$

Like the BSCA-F algorithm presented in chapter 4, the coalition formation algorithm BSCA-P introduced in this chapter is based on the classic BSCA algorithm for the formation of bilateral Shapley value stable coalitions. The BSCA protocol restricts negotiation to pairs of voted leaders of coalitions of given maximum size, thereby reducing the communication complexity. Each coalition leader recursively distributes the potential joint coalition value to those agents that are members of its current coalition according to the bilateral Shapley values (cf. figure 7.2). Coalitions are formed bilaterally per round based on coalition proposals that are mutually accepted based on the expected maximum of individually rational payoffs for the agents involved. However, to determine these potential payoffs, the BSCA protocol requires each agent to reveal its local worth to every potential coalition partner per round.

From the defition of the local worths in section 7.1 it is clear that knowing an agent's local worths, one is able to deduce at least

- its self value, i.e. the local income of the agent from selling its own services exclusively to local users.
- stronger interest of the agent in certain services offered by particular agents than by others.

But such knowledge could lead to an unwanted competitive advantage, in particular if a broader context than just the game at hand is considered. Fortunately, as we show in the following section, it turns out that the recursively bilateral Shapley value is very well-suited to hide local worths and coalition values from other agents participating in the game.

7.2 Hiding Local Worths and Coalition Values

In this section, we show that in order to implement recursively bilateral Shapley value stable payoff distributions by repeatedly merging coalitions, local worths and coalition values do not need to be commicated at all. Instead, for every merge of two coalitions, it is sufficient to compute the updated side payments using only the previous side payment and the *additional local worths* (defined below) of some subcoalitions.

To see how this can be done, we first consider an alternative view of the recursively bilateral Shaply value stable payoffs. It makes clearer that each non-root node gets its own coalition's value plus half of the additional payoff allocated to its parent node (its sibling getting the other half):

Lemma 7.2.1. *If C is a recursively bilateral coalition and u a recursively bilateral Shapley value stable payoff distribution, it follows directly from the definitions 2.2.18 and 2.2.21 that for non-root nodes C_i^* in T_C, $i \in \{1,2\}$,*

$$u(C_i^*) = v(C_i^*) + \frac{1}{2}(u(C^*) - v(C_1^*) - v(C_2^*)),\ k \in \{1,2\}, k \neq i$$

Proof. Because C_i^* is a non-root node, it must have the parent $C^* \in T_C$. Thus, by applying definitions 2.2.18 and 2.2.21 we get

$$\begin{aligned} u(C_i^*) &= \sigma_b(C^*, C_i^*, v_{C^*}) \\ &= \frac{1}{2}v_{C^*}(C_i^*) + \frac{1}{2}(v_{C^*}(C^*) - v_{C^*}(C_k^*)) \\ &= \frac{1}{2}v(C_i^*) + \frac{1}{2}(v_{C^*}(C^*) - v(C_k^*)) \end{aligned}$$

The last equation holds because $C_i^* \neq C^*$ and $C_k^* \neq C^*$. To show that $v_{C^*}(C^*) = u(C^*)$, we have to consider two cases: either C^* is the root node, or it has a parent $C^p \in T_C$.

In the first case, definition 2.2.21 asserts that $v_{C^*}(C^*) = v(C^*)$ and that u is efficient, so $v(C^*) = u(C^*)$.

In the second case, because $C^* = C_m^p$, $m \in \{1,2\}$, by definition 2.2.21

$$u(C^*) = \sigma_b(C^p, C^*, v_{C^p})$$

while at the same time (because $C^* = C_m^p$, $m \in \{1,2\}$)

$$v_{C^*}(C^*) = \sigma_b(C^p, C^*, v_{C^p}) \overset{2.2.21}{=} u(C^*)$$

Finally, we can thus rewrite

$$\begin{aligned} u(C_i^*) &= \frac{1}{2}v(C_i^*) + \frac{1}{2}(v_{C^*}(C^*) - v(C_k^*)) \\ &= \frac{1}{2}v(C_i^*) + \frac{1}{2}(u(C^*) - v(C_k^*)) \\ &= v(C_i^*) - \frac{1}{2}v(C_i^*) + \frac{1}{2}(u(C^*) - v(C_k^*)) \\ &= v(C_i^*) + \frac{1}{2}(u(C^*) - v(C_k^*) - v(C_i^*)) \\ &\overset{k\neq i}{=} v(C_i^*) + \frac{1}{2}(u(C^*) - v(C_1^*) - v(C_2^*)) \end{aligned}$$

□

The last term in the proof thus signifies the additional payoff obtained by C_1^* and C_2^* joining together. If C^* is the root node in T_C and thus $u(C^*) = v(C^*)$, we also call it the *additional coalition value*:

Definition 7.2.2 Additional coalition value
For a bilateral coalition C, its *additional coalition value* (in the context of v) is defined as the difference of C's value and the values of its subcoalitions C_1 and C_2:

$$av(C_1, C_2) := v(C_1 \cup C_2) - v(C_1) - v(C_2)$$

$\triangle$

Now we consider the case where two coalitions are joined together from one configuration to another. Together with lemma 7.2.1 we can show that the recursively bilateral Shapley value payoff of each non-root node can be stated as a function of its payoff in the first configuration, the additional coalition value of the new root and the node's depth in the tree:

Lemma 7.2.3. *Let $(\mathcal{C}_1, u_1)$ and $(\mathcal{C}_2, u_2)$ be configurations for a game $(\mathcal{A}, v)$, with u_1 and u_2 being recursively bilateral Shapley value stable, and $\exists C_1, C_2 \in \mathcal{C}_1 : C = C_1 \cup C_2 \in \mathcal{C}_2$. Then*

$$\forall C^* \in T_C,\ d(C^*, T_C) > 0 : u_2(C^*) = u_1(C^*) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}}$$

Proof. By induction over $d(C^*, T_C)$:

$d(C^*, T_C) = 1$**:** In this case, it holds that $C^* = C_i,\ i \in \{1, 2\}$ (one of C's direct children). Because of lemma 7.2.1 and because u_2 is efficient (by definition 2.2.18), we can write

$$\begin{aligned} u_2(C^*) &= \sigma_b(C^*, C, v) \\ &\overset{7.2.1}{=} v(C^*) + \frac{1}{2}(u(C) - v(C_1^*) - v(C_2^*)) \\ &= v(C^*) + \frac{1}{2}(v(C) - v(C_1^*) - v(C_2^*)) \\ &\overset{7.2.2}{=} v(C^*) + \frac{1}{2}av(C_1, C_2) \end{aligned}$$

Because also u_1 is efficient, and because $C^* \in \mathcal{C}_1$, we have

$v(C^*) = u_1(C^*)$ and thus

$$\begin{aligned} v(C^*) + \frac{1}{2}av(C_1, C_2) &= u_1(C^*) + \frac{1}{2}av(C_1, C_2) \\ &= u_1(C^*) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} \end{aligned}$$

$d(C^*, T_C) = k > 1$**:** In this case, we assume lemma 1 holds for all C^{**} with $d(C^{**}, T_C) < k$. Because C is recursively bilateral, C^* is a bilateral coalition at a depth > 1 in T_C. Formally, this implies that

$$\begin{aligned} C^* &= C_i^p,\ i \in \{1, 2\}, \\ C^p &\in T_C \text{ and} \\ d(C_i^p, T_C) &= d(C^p, T_C) + 1 \end{aligned}$$

Because u_2 is recursively bilateral Shapley value stable and thus also efficient, and applying the induction hypothesis, we can write

$$\begin{aligned} u_2(C_i^p) &\overset{2.2.21}{=} \sigma_b(C_i^p, C^p, v_{C_i^p}) \text{ with} \\ v_{C_i^p}(C^p) &= u_2(C^p) \\ &= u_1(C^p) + \frac{av(C_1, C_2)}{2^{d(C^p, T_C)}} \end{aligned}$$

Therefore, applying lemma 7.2.1 and the induction hypothesis, we can then rewrite

$$\begin{aligned} u_2(C_i^p) &\overset{7.2.1}{=} v(C_i^p) + \frac{1}{2}(u_2(C^p) - v(C_1^p) - v(C_2^p)) \\ &\overset{ind.h.}{=} v(C_i^p) + \frac{1}{2}(u_1(C^p) + \frac{av(C_1, C_2)}{2^{d(C^p, T_C)}} - v(C_1^p) - v(C_2^p)) \\ &= v(C_i^p) + \frac{1}{2}(u_1(C^p) - v(C_1^p) - v(C_2^p)) + \frac{av(C_1, C_2)}{2^{d(C^p, T_C)+1}} \\ &\overset{7.2.1}{=} u_1(C_i^p) + \frac{av(C_1, C_2)}{2^{d(C^p, T_C)+1}} \\ &= u_1(C_i^p) + \frac{av(C_1, C_2)}{2^{d(C_i^p, T_C)}} \end{aligned}$$

□

For a merge of C_1 and C_2 to form $C = C_1 \cup C_2$, we further define the *additional local worth* of agents and subcoalitions with respect to a coalition:

Definition 7.2.4 Additional local worth
Let C_1 and C_2 be coalitions, and $a \in C_i$, $i \in \{1, 2\}$. Then a's *additional local worth* is defined as

$$alw_a(C_i, C) := lw_a(C) - lw_a(C_i)$$

We also define the additional local worth for a subcoalition $C^* \in T_{C_i}$:

$$alw(C^*, C_i, C) := \sum_{a \in C^*} alw_a(C_i, C)$$

△

Corollary 7.2.5. *Given a recusively bilateral coalition $C = C_1 \cup C_2$ and a non-leaf coalition $C^* \in T_{C_i}$, $i \in \{1, 2\}$, then with $C^* = C_1^* \cup C_2^*$*

$$\begin{aligned} alw(C^*, C_i, C) &= \sum_{a \in C^*} alw_a(C_i, C) \\ &= \sum_{a \in C_1^*} alw_a(C_i, C) + \sum_{a \in C_2^*} alw_a(C_i, C) \\ &= alw(C_1^*, C_i, C) + alw(C_2^*, C_i, C) \end{aligned}$$

Therefore, if the representative agent a of C^* is the agent to to compute $alw(C^*, C_i, C)$ and receives the additional local worths of C^*'s subcoalitions from their respective representatives, a has simply to sum up the two values. Additionally, the representatives of C_1 and C_2, having thus obtained their respective $alw(C_i, C_i, C)$, can do analogously to compute the additional coalition value by exchanging $alw(C_i, C_i, C)$ among each other and adding them up, because:

Corollary 7.2.6. *The additional coalition value $av(C_1, C_2)$ for a potential merge to form coalition $C = C_1 \cup C_2$ is equal to the sum of C_1's and C_2'2 additional local worth with respect to C:*

$$\begin{aligned} av(C_1, C_2) &= \sum_{a \in C} lw_a(C) - \sum_{a \in C_1} lw_a(C_1) - \sum_{a \in C_2} lw_a(C_2) \\ &= \sum_{a \in C_1} lw_a(C_1 \cup C_2) - \sum_{a \in C_1} lw_a(C_1) \\ &\quad + \sum_{a \in C_2} lw_a(C_1 \cup C_2) - \sum_{a \in C_2} lw_a(C_2) \\ &= alw(C_1, C_1, C) + alw(C_2, C_2, C) \end{aligned}$$

Finally, the following theorem shows that in order to compute its side payment when merging coalitions C_1 and C_2, each subcoalition $C^* \in T_{C_i}$ only needs to consider its sidepayment for the case without the merge and the additional local worths of C_1, C_2 and C^*:

Theorem 7.2.7. *Let* $(\mathcal{C}_1, u_1)$ *and* $(\mathcal{C}_2, u_2)$ *be configurations for a game* $(\mathcal{A}, v)$*, with* u_1 *and* u_2 *being recursively bilateral Shapley value stable, and* $\exists C_1, C_2 \in \mathcal{C}_1 : C = C_1 \cup C_2 \in \mathcal{C}_2$*. Then* $\forall C^* \in T_{C_i}$*,* $i \in \{1, 2\}$*:*

$$\begin{aligned} sp_{u_2}(C^*, C) =& sp_{u_1}(C^*, C_i) - alw(C^*, C_i, C) \\ &+ \frac{alw(C_1, C_1, C) + alw(C_2, C_2, C)}{2^{d(C^*, T_C)}} \end{aligned}$$

Proof. Remember that by definition 2.2.29, for any u it holds that

$$sp_u(C^*, C) = \sum_{a \in C^*} u(a) - lw_a(C) = u(C^*) - \sum_{a \in C^*} lw_a(C)$$

Because of lemma 7.2.3, definition 7.2.4 and corollary 7.2.5, we can rewrite

$$\begin{aligned} sp_{u_2}(C^*, C) &= u_2(C^*) - \sum_{a \in C^*} lw_a(C) \\ &\overset{7.2.3}{=} u_1(C^*) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} - \sum_{a \in C^*} lw_a(C) \\ &\overset{7.2.4}{=} u_1(C^*) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} \\ &\quad - \sum_{a \in C^*} (lw_a(C_i) + alw_a(C_i, C)) \\ &\overset{7.2.4}{=} sp_{u_1}(C^*, C_i) - alw(C^*, C_i, C) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} \\ &\overset{7.2.5}{=} sp_{u_1}(C^*, C_i) - alw(C^*, C_i, C) \\ &\quad + \frac{alw(C_1, C_1, C) + alw(C_2, C_2, C)}{2^{d(C^*, T_C)}} \end{aligned}$$

□

Please note that in the particular case of $C^* = C_i$, it holds that $sp_{u_1}(C^*, C_i) = 0$ because of corollary 2.2.29. Hence, in order to implement recursively bilateral Shapley value stable payoff distributions by repeatedly merging coalitions, all subcoalitions in the tree T_C of a coalition $C = C_1 \cup C_2$ to be formed can compute their side payments directly by using only

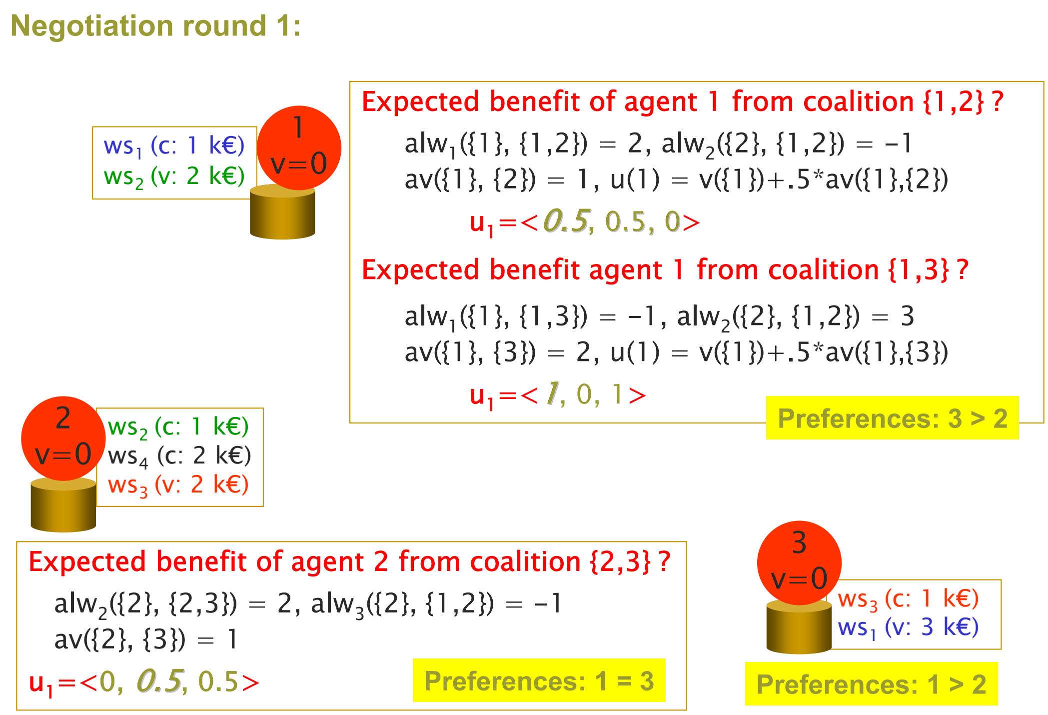

Figure 7.3: Privacy preserving negotiation of coalitions (round 1).

1. their side payment in their current coalition C_1 or C_2,

2. their own as well as C_1 and C_2's additional local worths for the merge, which can be computed recursively from the additional local worths of their respective subcoalitions, or directly for single-agent coalitions.

Therefore, local worths as well as coalition values do not have to be revealed at all. This is in contrast to the traditional way of negotiating stable coalitions with complete prior knowledge about local worth and coalition values that constitute the game to be solved. We acknowledge that this does hold in particular for the bilateral Shapley value but not necessarily for other game-theoretic stability concepts.

Example 7.2.8
Consider, again, our example coalition game (cf. figure 7.1). During the first negotiation round, it turns out that agents a_1 and a_3 would prefer each other as a coalition partner, since both of them could obtain a higher individually rational payoff in a joint coalition than

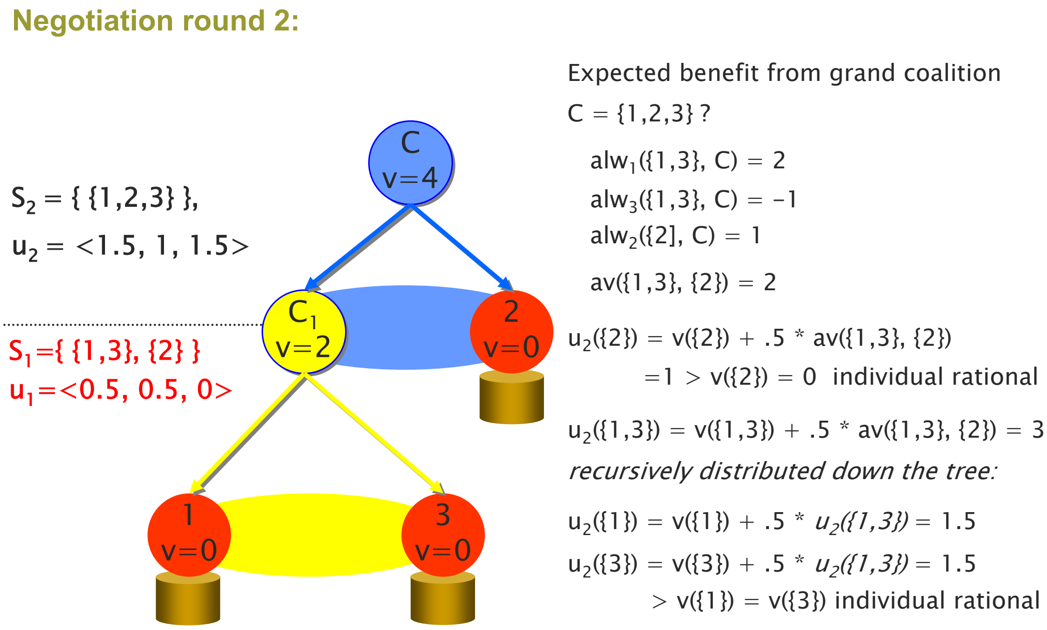

Figure 7.4: Privacy preserving negotiation of coalitions (round 2).

each could get in a separate coalition with agent a_2 (cf. figure 7.3). Agent a_2 is even indifferent in respect to the coalition it would prefer.

More concrete, $\{1\}$ and $\{3\}$ form a coalition C_1, with $alw_{a_1}(\{1\}, C_1) = -1 - 0 = -1$ and $alw_{a_3}(\{3\}, C_1) = 3 - 0 = 3$. According to theorem 7.2.7 we get

$$sp_u(\{1\}) = 0 + \frac{(-1) + 3}{2^1} - (-1) = 2$$

and

$$sp_u(\{2\}) = 0 + \frac{(-1) + 3}{2^1} - 3 = -2.$$

Thus, the net amount received by a_1 and a_3 are

$$\begin{aligned} u(a_1) &= lw_{a_1}(C_1) + sp_u(\{1\}) = -1 + 2 = 1 \\ &= \sigma_b(\{1\}, \{1\} \cup \{3\}, v) \end{aligned}$$

and

$$\begin{aligned} u(a_3) &= lw_{a_3}(C_1) + sp_u(\{3\}) = 3 - 2 = 1 \\ &= \sigma_b(\{2\}, \{1\} \cup \{2\}, v). \end{aligned}$$

In the second round (cf. figure 7.4), agent a_2 negotiates with the leader of the newly formed coalition C_1 for joining as it is individually rational to do so: Its expected payoff in a potential grand coalition amounts to 1ke, that is it may obtain more by means of cooperation than it would by staying alone. On the other hand, forming of coalition C_1 is consent with this proposal for the same reason: Its bilateral Shapley value of 3ke, recursively distributed down the coalition tree to agents a_1 and a_3, yields a rational expected payoff for both members.

More concrete, their additional local worths in the grand coalition C are

$$\begin{aligned} alw_{a_1}(\{1\}, C) &= 1 - (-1) = 2, \\ alw_{a_3}(\{3\}, C) &= 2 - 3 = -1 \\ alw(C_1, C_1, C) &= alw_{a_1}(\{1\}, C) + alw_{a_3}(\{2\}, C) = 1 \\ alw(C_2, C_2, C) &= 1 - 0 = 1 \end{aligned}$$

The additional coalition value is thus

$$av(C_1, C_2) = alw(C_1, C_1, C) + alw(C_2, C_2, C) = 2$$

Applying theorem 7.2.7 again, we get the new payoff distribution u^* with

$$\begin{aligned} sp_{u^*}(C_1) = 0 + \frac{1+1}{2^1} - 1 = 0 \\ (= sp_{u^*}(C_2)) \end{aligned}$$

The net payoffs of C_1 and C_2 are equal to their bilateral Shapley values:

$$\begin{aligned} u^*(C_1) &= lw_{a_1}(C) + lw_{a_3}(C) + sp_{u^*}(C_1) \\ &= 1 + 2 + 0 = 3 = \sigma_b(C_1, C, v) \\ u^*(C_2) &= lw_{a_2}(C) + sp_{u^*}(C_2) \\ &= 1 + 0 = 1 = \sigma_b(C_2, C, v) \end{aligned}$$

For sidepayments within C_1, we again apply theorem 7.2.7:

$$\begin{aligned} sp_{u^*}(\{1\}, C) &= sp_u(\{1\}, C_1) + \frac{1+1}{2^2} - 2 \\ &= 2 + 0.5 - 2 = 0.5 \\ sp_{u^*}(\{3\}, C) &= sp_u(\{3\}, C_1) + \frac{1+1}{2^2} + 1 \\ &= -2 + 0.5 + 1 = -0.5 \end{aligned}$$

Consequently, the net payoffs of a_1 and a_3 are equal to their recursively bilateral Shapley value stable payoffs:

$$\begin{aligned} u^*(a_1) &= lw_{a_1}(C) + sp_{u^*}(a_1) \\ &= 1 + 0.5 = 1.5 = \sigma_b(\{1\}, C, v) \\ u^*(a_3) &= lw_{a_3}(C) + sp_{u^*}(a_2) \\ &= 2 + (-0.5) = 1.5 = \sigma_b(\{3\}, C, v) \end{aligned}$$

△

Now, the question is, to which extent can the agents' privacy be preserved by negotiating coalitions while hiding their local worths and coalition values as outlined above?

Proposition 7.2.9. *Let there be an additional local worth $alw(C^*, C_i, C)$, $i, k \in \{1, 2\}, k \neq i$. Further assume that the agents in C_i know about all requests in C_i, i.e. that $\forall a \in C_i : KR_a(C_i) = \bigcup_{a' \in C_i} R_{a'}$. Then, if*

1. *$alw(C^*, C_i, C) < 0$, it can be deduced that an agent $a \in C^*$ executes some service $ws \in OS_a$ for an agent $a' \in C_k$.*
2. *$alw(C^*, C_i, C) > 0$, it can be deduced that*
   - *(a) an agent in C^* requests some service in C_k, or*
   - *(b) that a service which was to be executed in C^* previous to the merge with C_k is to be executed by an agent in C_k after the merge.*

   Both cases might also be true at the same time.

Proof. We proof the different cases according to their numbering in the proposition:

1. $alw(C^*, C_i, C) < 0$ implies that

$$\begin{aligned} & \sum_{a \in C^*} alw_a(C_i, C) < 0 \\ \Leftrightarrow & \sum_{a \in C^*} (lw_a(C) - lw_a(C_i)) < 0 \\ \Leftrightarrow & \sum_{a \in C^*} lw_a(C) < \sum_{a \in C^*} lw_a(C_i) \end{aligned}$$

   and therefore that for at least one agent $a \in C^*$: $lw_a(C) < lw_a(C_i)$. But remember the requirement of definition 7.1.1 that excludes service executions with negative net value. It then follows that a must execute at least one additionally requested service in coalition C. Then, considering the assumption that the agents in C_i know about all requests in C_i, it further follows that this request cannot come from an agent in C_i.

2. Conversely, $alw(C^*, C_i, C) > 0$ implies that for at least one agent $a \in C^*$: $lw_a(C) > lw_a(C_i)$. Therefore, reversing the argument of the first case, it follows that this incease of a's payoff can only be induced by either one or both of these two possible cases:

   (a) At least one more of a's requested services is executed in C with respect to C_i. This additional execution cannot come from an agent in C_i, since otherwise the assigned service executions in C_i would have already contained it because of the assumption of perfect knowledge of service requests in C_i. Note that this might be true only transitively if there is at least one other agent $a' \in C_i$ requesting the same service, and for which the service was to be executed by an agent $a^* \in C_i$. Then it might happen in C that a^* executes the service for a instead of a'. But then, service needs to be executed by yet another agent for a', so ultimately an agent in C_k must provide an additional execution.

   (b) In C, a does not have to execute a service which it is to execute in C_i. It follows that an agent in C_k must take over this execution in C, at a lower or equal cost. Similar to the case 2a, this might be true only transitively, but in which case the same argument can be used to show that ultimately an agent in C_k must take over an additional execution of the service in C.

□

It is interesting to regard some corner cases of proposition 7.2.9:

Corollary 7.2.10. *Consider case 1 of proposition 7.2.9 and assume that the sets of service offers OS_a are known for all $a \in \mathcal{A}$. Then to all beholders of $alw(C^*, C_i, C)$,*

- *if $|C_k| = 1$, the identity of agent a' with an additional request is immediately known;*

- *if* $|OS_a| = 1$*, the identity of the additionally requested service is immediately known;*
- *if* $|OS_{C_*}| = 1$*, the identity of the additionally requested service as well as the net profit of its execution are immediately known.*

In the case 2, if $OS_{C_i} \cap OS_{C_k} = \emptyset$*, then case 2a must be true. Then, the same above statements for case 1 hold in in reverse for* C^* *instead of* C_k*, and vice versa.*

7.3 Anonymous Service Requests and Access

With the result of proposition 7.2.9 in the previous section, it is clear that the direct computation of side payments based on additional local worths is not quite enough for the agents to preserve their privacy, or at least compromise it in a controlled way. However, proposition 7.2.9 and corollary 7.2.10 also show that it depends on the sizes of coalitions and offer sets what exactly can be deduced, and with which certainty.

For example, it might be unacceptable for an agent a to be idenfiable as the certain originator of a service request. But it might be acceptable for a if the request can only be known to originate from any one of $k \in \mathbb{N}$ agents. Likewise, it might be also acceptable for a to be identifiable as the originator of a service request if the service can only be identified to be one of k possible services.

In other words, agents might constrain their service requests to adhere to some degree of anonymity, thereby achieving some weaker notion of privacy at least. This is the approach that we outline in this section.

To measure degrees of anonymity, different notions have been proposed in the literature, such as total, or group anonymity, under possibilistic or probabilistic interpretations Halpern and O'Neill (2003), Pfitzmann and Köhntopp (2001).

For simplicity, we employ the concept of possibilistic *k-anonymity*, which requires only that there exists some set of agents K with size k, such that each $a \in K$ is a possible sender. If we assume that each (sub-)coalition communicates with other (sub-)coalition only via their respective representatives, then from the perspective of agents in C_2, any agent in C_1 might be the originator of the service request. The determination of the agent k-anonymity for agents in C_1 wrt. agents in C_2 and a matching minimum-constraint are thus straightforwardly defined:

Definition 7.3.1 Agent anonymity constraint
Given a coalition C_1, for every agent $a \in C_1$ its *agent anonymity aa* with respect to (agents in) any other coalition C_2, $C_1 \cap C_2 = \emptyset$, is defined as

$$aa_{C_1} = |C_1|$$

Each agent a_i might then specify *agent anonymity (minimum) contraints* $aa^i_{min}(ws)$ with respect to requests for service ws. With a_i in coalition C_1, sending a request for service ws to coalition C_2, $C_1 \cap C_2 = \emptyset$, is said to adehre to a_i's agent anonymity constraints iff

$$aa^i_{min}(ws) \leq aa_{C_1}$$

△

Thus, the adherence to agent anonymity constraints enables agents in a coalition to maintain a degree of privacy when negotiating a merge with another coalition. However, this doesn't yet help an agent to also preserve its privacy with respect to other agents in its own coalition: for a merge of coalition C_1 with coalition C_2 to form $C = C_1 \cup C_2$, the additional local worth $alw(C_1, C)$ has to be computed. Therefore, each subcoalition $C^* \in T_{C_1}$ has to compute $alw(C^*, C_1, C)$ first. In particular, agent a has in general to inform some other agent in C_1 about $alw(\{a\}, C_1, C)$. For example, assume that the additional local worths are computed recursively as outlined in the previous section. Then they will be propagated up the coalition tree T_{C_1} via the respective subcoalition representatives, though being accumulated with the additional local worth of the respective sibling subcoalition at each step.

Thus, we additionally use the concept of *service anonymity*, expressing that an agent accesses any one of a number of possible services:

Definition 7.3.2 Service anonymity constraint
Let OS_a denote the set of offered services of agent a. Then the service anonymity for requested services offered by a coalition C_1 with respect to (agents in) any other coalition C_2, $C_1 \cap C_2 = \emptyset$, is defined as the number of unique services offered by C_1's members:

$$sa(C_1) = \left| \bigcup_{a \in C_1} OS_a \right|$$

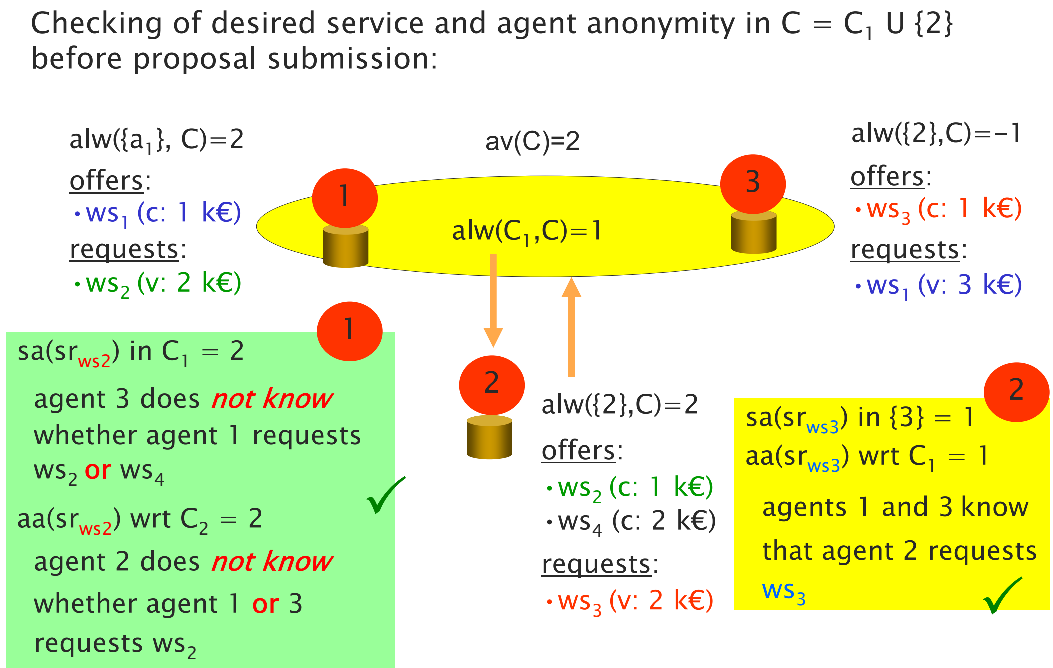

Figure 7.5: Individual service request anonymities.

Analogously to the agent anonymity contraints, each agent a_i might then additionally specify *service anonymity (minimum) contraints* $sa^i_{min}(ws)$ with respect to requests for service ws. With a_i in coalition C_2, sending a request for service ws to coalition C_1, $C_1 \cap C_2 = \emptyset$, is said to adehre to a_i's service anonymity constraints iff

$$sa^i_{min}(ws) \leq sa_{C_1}$$

$\triangle$

For an example demonstrating agent and service anonymity, see figure 7.5.

Having outlined how to maintain the minimum anonymity conraints during the coalition negotiation, there is still one part missing: after the negotiation, requesting agents will need to actually engage in communication with the executing agents (assuming that there is some input/output involved). But an agent sending a message to another agent in another coalition can then still be identified by the receiving agent (and possibly others) simply as the originator of that message.

Therefore, to maintain the above mentioned types of anonymity also at the communication level for the service execution phase, we

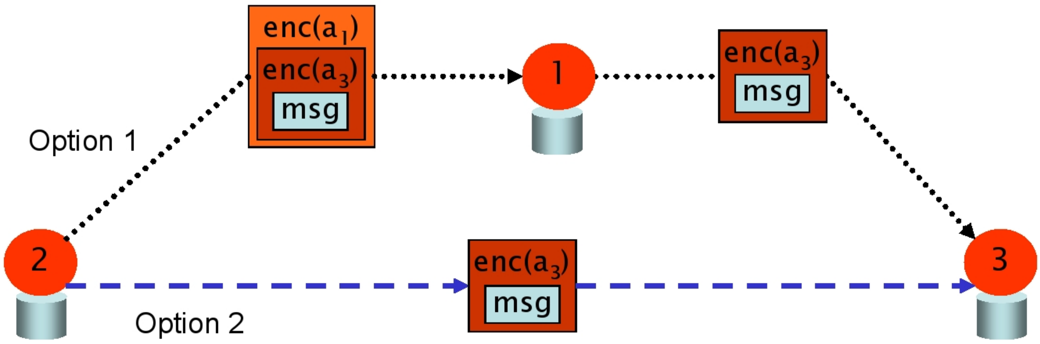

Figure 7.6: Options of encrypted service request message "onion" routing from agent a_2 to agent a_3.

Figure 7.7: Two ways of a_2 contacting a_3 via Onion Routing.

adopt the simple onion routing protocol Syverson et al. (1997), which is based on rerouting.

In a rerouting protocol, a message is not directly sent to the receiver, but travels over intermediate network nodes, or agents in our case. The onion routing protocol was originally defined for HTTP-connections, but we adopt it here for our agent coalition formation setting, by looking only at high-level messages sent between the agents instead of technical details of an underlying protocol. Our focus is to enable the agents to request and access services within their coalition anonymously. We thus also do not bother about problems like possible eavesdropper agents or traffic analysis, as such problems are out of scope of this thesis. The basic idea of the onion routing protocol is to wrap a message in several layers of encryption and reroute it over several rerouting nodes such that no single node is able to determine the sender and receiver of a message. Also, when one agent contacts another, the nodes over which messages are sent are chosen randomly. Figure 7.7 illustrates this for a three-agent case. It incorporates a public/private key encryption method, such as the well-known RSA method (originally proposed in Rivest et al. (1978)). Thus, we extend our agent model such that every agent a is required to possess a private key $privkey_a$ and a matching public key $pubkey_a$ for the chosen encryption method. Further, a needs to be able to execute according encryption/decryption functions. In the following, $enc(pubkey, m)$ denotes a function that encrypts message m

using the public key $pubkey$, and $dec(privkey, em)$ denotes the corresponding decryption function for the encrypted message em using the private key $privkey$. To let agent a_1 send an encrypted message m to agent a_2, a_1 encrypts m by executing $enc(pubkey_{a_2}, m)$, sends the result em to a_2 which decrypts it by executing $dec(privkey_{a_2}, em)$. Thus, the agents need to perform an initial public key exchange.

In the onion protocol, actually only a part of a message is encrypted with the public key method. This part contains a key for a symmetric encryption method, i.e. one that uses the same key for encryption and decryption. The remainder of the message is encrypted with this method. This is done because of performance reasons, since symmetric encryption methods usually are much faster than public key methods. However, we go not into those details here.

Once an initial onion routed *circuit*, i.e. the overall route from the originator to the receiver, is established, it might be kept open to allow for further mutual messages. This is achieved by assigning randomly generated identifiers to each part of the route. Therefore, by employing onion routing, each agent might establish a circuit with each other agent to send anonymized messages which can also be replied to. But since the list of relay nodes is generated randomly, its length should be bounded by a max. value, which we call $cmax$. Otherwise, circuits of arbitrary length could be constructed, making it impossible to put a bount on the additional communication complexity that is induced by employing onion routing.

Thus, having a means to also anonymize the service execution phase in place, we are finally ready to present our privacy preserving coalition formation algorithm and protocol, BSCA-P, in the following section.

7.4 Algorithm BSCA-P

In this section, we finally propose the coalition formation protocol BSCA-P that makes use of all concepts and means that have been introduced in the previous sections. In the algorithm, we assume that

- service offers along with service execution costs are known in prior (they could be distributed among the agents in a simple initial step of offer exchange).

- all agents are capable of sending, receiving and replying to anonymous messages to all other agents via onion routing with constant max. circuit length $cmax$.
- there exists a function $Rep : 2^{\mathcal{A}} \mapsto \mathcal{A}$ which unambiguously identifies the representative of a given coalition.

Algorithm 7.4.1
For a game $(\mathcal{A}, v)$, $\mathcal{C}_0 := \{\{a\}|a \in \mathcal{A}\}$, $r := 0$ and $\forall C \in \mathcal{C}_0 :\ sp_0(C) := 0$. In every coalition $C \in \mathcal{C}_r$, every agent $a \in C$ performs:

1. Let $C \in \mathcal{C}_r,\ a \in C$ and $\mathcal{C}^* := \mathcal{C} \setminus C$.
2. *Communication:*
   (a) For all $C^* \in \mathcal{C}^*$ do:
      i. Determine the set $R_a(C^*)$ of requests, subject to the set OS_{C^*} of offers, service execution costs and minimum anonymity constraints $aa^i_{min}(ws)$ and $sa^i_{min}(ws)$
      ii. For each service request which is in $R_a(C) \cap R_a(C^*)$ keep the one with minimum costs, thereby determining set $R_a(C \cup C^*)$ of a's requests in the merged coalition.
      iii. Inform all agents in $C \cup C^*$ about a's accessing their offered services according to $R_a(C \cup C^*)$ via an anonymously sent message.
      iv. Receive anonymously sent messages about a's offered services being accessed from all agents in C^*, thereby determining $E_a(C \cup C^*)$.
      v. Set $alws_a(C^*) := alw_a(C, C^*)$.
      vi. For each bilateral coalition $C^a,\ C^a \in T_C,\ a \in C^a,\ a = Rep(C_a)$, wait for a message from $Rep(C^a_i),\ i \in 1, 2, a \notin C^a_i$ containing
      $alws_{Rep(C)}(C^*)$ and set
      $$alws_a(C^*) := alws_a(C^*) + alws_{Rep(C)}(C^*)$$
      vii. If $a = Rep(C)$ then send $alws_a(C^*)$ to $Rep(C^*)$; else send $alws_a(C^*)$ to $Rep(C^+)$ with $C^+ \in T_C,\ a = Rep(C^+_i),\ i \in 1, 2, a \neq Rep(C^+)$.

   (b) If $a = Rep(C)$ then receive $alws_{Rep(C^*)}(C)$ and set
   $alws(C^*) := alws_{Rep(C^*)}(C) + alws_a(C^*)$ for all $C^* \in \mathcal{C}^*$; else go to step 3i.

3. *Coalition Proposals:*

   (a) Set $Candidates := \mathcal{C}^*$, $New := \emptyset$ and $Obs := \emptyset$

   (b) Determine a coalition $C^+ \in Candidates$ with

   $$\forall C^* \in Candidates: \ alws_a(C^+) \geq alws_a(C^*)$$

   (c) Send a proposal to $Rep(C^+)$ to form coalition $C \cup C^+$.

   (d) Receive all coalition proposals from other agents.

   (e) If no proposal from $Rep(C^+)$ is received and $Candidates \neq \emptyset$, set $Candidates := Candidates \setminus \{C^+\}$ and go to step 3b.

   (f) If a proposal from $Rep(C^+)$ is received, then form the coalition $C \cup C^+$:

      i. If $o(Rep(C)) < o(Rep(C^+))$ then set $Rep(C \cup C^+) := Rep(C)$; else set $Rep(C \cup C^+) := Rep(C^+)$.

      ii. Inform all other $Rep(C^*), C^* \in \mathcal{C}^* \setminus C^+$ and all $a^* \in C$, $a^* \neq a$ about the new coalition and $Rep(C \cup C^+)$

      iii. $New := \{C \cup C^+\}$, $Obs := \{C, C^+\}$

   (g) Receive all messages about new coalitions. For each new coalition $C_1 \cup C_2$ and $Rep(C_1 \cup C_2)$, set $New := New \cup \{C_1 \cup C_2\}$, $Obs := Obs \cup \{C_1, C_2\}$ and $Candidates := Candidates \setminus \{C_1, C_2\}$.

   (h) Send the sets New and Obs to all other coalition members $a^* \in C$, $a^* \neq a$

   (i) If $a \neq Rep(C)$ then receive the sets New and Obs from $Rep(C)$.

   (j) Set $r := r + 1$, $\mathcal{C}_r := (\mathcal{C}_{r-1} \setminus Obs) \cup New$.

   (k) For each (sub-)coalition $C^* \in T_C$ with $Rep(C^*) = a$, determine $sp_r(C^*)$ according to theorem 7.2.7 (applying $sp_{r-1}(C^*)$ as the original side payment).

   (l) If $C_r = C_{r-1}$then stop; else go to step 2

△

7.5 BSCA-P Properties

Theorem 7.5.1. *The BSCA-P maintains the privacy of each agent a, with a^* denoting the agent with whom a first forms a coalition, by*

1. *adhering to a's anonymity constraints,*

2. *hiding a's local worths, and in particular its self value, from all other agents, although a^* might establish lower bounds, and*

3. *hiding a's additional local worths from all other agents but a^*.*

Proof. We first show that anonymity contraints are adhered to by $R_a(C \cup C^*)$ via induction over the rounds:

- In round 0, $R_a(C) = R_a(\{a\})$, which trivially adheres to the constraints.

- In round $k > 0$, assume that $R_a(C)$ adheres to the anonymity constraints. Also observe that if $R_a(C)$ and $R_a(C^*)$ adhere to the constraints, then so does $R_a(C) \cup R_a(C^*)$, because the service accesses are bound to the offering agents by definition 7.1.1. Then

  1. in step 2(a)i, including only those service requests in the set $R_a(C^*)$ for which the anonymity constraints hold prevents the access of services in OS_{C^*} for which the constraint do not hold.
  2. in step 2(a)ii, the set $R_a(C \cup C^*)$ is constructed such that $R_a(C \cup C^*) \subseteq R_a(C) \cup R_a(C^*)$.

Having established that $R_a(C \cup C^*)$ adheres to the constraints, we now show that anonymity is upheld also in all other steps of the BSCA-P:

1. in step 2(a)iii, the anonymity constraints are upheld by informing the agents which are to execute the accessed services via an anonymously sent message.

2. Elements of $R_a(C \cup C^*)$ are not sent to any additional agents in any other step.

3. Since the additional local worths are computed using only valuations of accessed services $R_a(C \cup C^*)$, analysis of additional local worths by an agent in C to deduce information about service requests (as in 7.2.9) cannot lead to the discovery of service requests for which the anonymity contraints do not hold.

4. Additional local worths of subcoalitions in T_C, except for the root node, involving requests $R_a(C) \cap R_a(C \cup C^*)$ might be analyzed by agents in C^* to deduce information about service requests within T_C, thereby possibly violating anonymity constraints. But these additional local worths are communicated only within T_C in step 2(a)vii. Only $alw(C, C^*)$ (which is built up recursively by the representatives of subcoalitions in T_C) is sent to $Rep(C^*)$. From $alw(C, C^*)$, agents in C^* might deduce that agents in C request services from agents in C^* and vice versa (see 7.2.9). But if this is the case, then the anonymity contraints of the respective agents allow for this, because otherwise they would not have been included in their respective sets of accessed services and thus also not be involved in the computation of additional local worths.

5. The same argument holds also for other coalitions observing that $C \cup C^*$ forms (or might form), thereby being able to deduce that $av(C, C^*) > 0$: from their perspective, the degree of agent anonymity of agents in $C \cup C^*$ is $|C \cup C^*| > |C|$, and the degree of service anonymity is $|OS_{C \cup C^*}| \geq |OS_{C^*}|$. Therefore, the anonymity constraints are also upheld in this case.

6. No other steps or effects of the BSCA-P involve information about service requests (the actual service execution obviously does involve service requests, but this is seen as a part external to the BSCA-P and outlined below, again using anonymous messages to uphold anonymity also during service execution).

It is thus proved that the anonymity constraints are upheld throught all steps of the BSCA-P. It now remains to be shown that each agent can hide its

1. local worths, including its self value, from all other agents, while a^* might establish a lower bound, and

2. its additional local worths from all other agents but a^*.

For the first case, remember that only additional local worths are communicated within the BSCA-P. Thus, no agent can determine a's local worths or self value. However, if agent a^* services are accessed by a in the coalition $\{a, a^*\}$, then a^* can establish a lower bound on a's self value: if service ws is accessed in coalition C, this implies by definition 7.1.1 that $w_a(ws) - c_{a^*}(ws) > 0$, and thus $lw_a(C) \geq w_a(ws) -$

$c_{a^*}(ws)$. If more than one of a^*'s services are accessed, then the lower bound is the sum of their differences of valuations and costs. Having this, for the next merge with a coalition C^{Next} to form $\{a, a^*\} \cup C^{Next}$, a^* can add the lower bound on the self value to $alw_a(\{a, a^*\}, C^{Next})$, thus obtaining a lower bound for $lw_a(C^{Next})$, because a's self value is equal to $lw_a(\{a\})$ and definition 7.2.4. For subsequent additional local worths received from a, a^* might successively update its lower bounds for a's local worths in the same way.

For the second case, remember that the additional local worths of each subcoalition in T_C are computed recursively by summing up the additional local worths of each direct subcoalition of a node in T_C. Thus, the fact that each additional local worth is sent only to the representative of the immediate parent node C^+ in T_C ensures that only the representative of $\{a, a^*\}$ (which might be a^* or a itself) will ever know a's additional local worths. □

Corollary 7.5.2. *The fact that service requests might be withhold within the BSCA-P to adhere to anonymity constrains implies that the agents solve a game G which might be different to the corresponding game G_0 which is equal to G except that anonymity constrains are ignored. In particular, there might exist coalitions which are more profitable in G_0 than in G.*

Theorem 7.5.3. *With $n = |\mathcal{A}|$ and $maxR := max_{a\in\mathcal{A}}\{|R_a|\}$, the computational complexity of the BSCA-P is in $O(n^3 maxR^2)$.*

Proof. In any round r, $\mathcal{C}_r \leq n$. The iteration in step 2a is thus done at most n times. In step 2(a)i, for each service in R_a, a has to find an agent in the potential partner coalition which offers this service at the least cost. The conditions 7.3.1 and 7.3.2 only have to be checked once for each service, for which we assume negligible complexity. Thus, at most $n\ maxR$ operations are required in this step. Step 2(a)ii can be done in less than $maxR^2$ steps. All other steps within and outside of the iteration in step 2a are of less complexity. Thus, the complexity of one round of the BSCA-P is in $O(n)(O(nmaxR) + O(maxR^2)) = O(n^2 maxR^2)$. Since the maximum number of coalition merges is smaller than n (because after at most $n-1$ merges, the grand coalition is formed), the number of rounds is also bound by n. The overall computational complexity of the BSCA-P is thus $O(n)O(n^2 maxR^2) = O(n^3 maxR^2)$. □

Theorem 7.5.4. *In the BSCA-P, the number of messages caused by each agent is in $O(n^2)$.*

Proof. In each round of the BSCA-P, the iteration in step 2a is performed $|\mathcal{C}_r| - 1 < n$ times. Therein,

1. in step 2(a)iii, an anonymous message is sent to each agent in $C \cup C^*$, and thus relayed via at most $cmax$ agents. Including the relayed messages, this therefore amounts to max. $cmax\ n$ messages in the worst case. But with $\mathcal{C}' := \mathcal{C} \cup \{C \cup C^*\} \setminus \{C, C^*\}$ being a partition of $\mathcal{A}$, $\sum_{C^* \in \mathcal{C}'} |C^*| = |\mathcal{C}'| \leq n$. Therefore, and since $cmax$ is constant, the number of messages caused by this step over all iterations is in $O(n)$.

2. in step 2(a)vii, a message to the agent's subcoalition representative or to $Rep(C^*)$ is sent. Assuming that agents which are representatives of several subcoalitions omit sending messages to themselves, and with less than n iterations, the number of messages sent in this step over all iterations is in $O(n)$.

The loop between steps 3b and 3e is executed at most $|Candidates| < n$ times. In each iteration, in step 3c one message is sent, and therefore the overall number is in $O(n)$.

In step 3(f)ii, the number of messages sent is $< |\mathcal{C}| \leq n$ and so is also in $O(n)$.

Finally, in step 3h, $|C| - 1 < n$ messages are sent, again being in $O(n)$.

Thus, in each message-sending step, $O(n)$ messages are sent per round. With at most n rounds, as shown in the proof of 7.5.3, the overall number of messages caused by each agent in the BSCA-P is thus in $O(n^2)$. □

When the protocol is finished and thus coalitions are formed, agents still have to execute the following steps in order to implement the coalitions:

1. Each agent accesses its fulfilled services from other agents via anonymous routing.

2. All (sub-)coalition representatives execute their respective side-payments sp_r for their (sub-)coalitions. Each representative only makes/receives payments to/from representatives of immediate parent and child coalitions, such that no additional information about payments is gained by any agent.

The last step ensures that only a representative of a two-agent coalition is informed about individual side-payments, and only about two of them: its own, and the other agent of the two-agent coalition. Therefore, only the first partner agent that an agent a coalesces with might ever know a's exact side payment.

Lastly, we point out that while corollary 7.5.2 might seem to indicate that anonymity proofness, as briefly discussed in section 3.3, also holds for the BSCA-P, this is actually not the case: because recursively bilateral Shaply value stable payoffs depend on the joining order of the subcoalitions (as opposed to the classic Shapley value), an agent a might be better off if it joins the coalition later rather than earlier. But exactly this can happen if a is at first prevented to join the building-up coalition due to its anonymity constraints. Later, when coalitions merged several times, joining the bigger coalition might then be allowed by a's anonymity constraints.

Having thus shown how to form an execute privacy preserving bilateral Shapley value stable coalition formation, we now briefly consider privacy preservation in kernel-stable coalitions before concluding this chapter.

7.6 On Privacy Preservation in Kernel-stable Coalitions

In this section we show that agents involved in the negotiations of kernel-stable coalitions can hide their local data and information used to compute their self-values, as well as the self values themselves, from other agents. This can be done without even risking any loss of profit in the final coalition configuration.

This property of the kernel is an inherent property of the definition of kernel stability, which is stated in the following lemma.

Lemma 7.6.1. *Let* $(\mathcal{A}, v)$ *and* $(\mathcal{A}, v^*)$ *with*

$$\exists a^* \in \mathcal{A}, r \in \mathbb{R} : v^*(C) := \begin{cases} v(C) + r & \text{for } a^* \in C \\ v(C) & \text{otherwise} \end{cases}$$

Then it holds that the configuration $(\mathcal{C}, u^*)$ *with* $u^*(a^*) = u(a^*) + r$ *and* $\forall a \in \mathcal{A}, a \neq a^* : u^*(a) = u(a)$ *is kernel-stable with respect to the game* $(\mathcal{A}, v^*)$ *iff* $(\mathcal{C}, u)$ *is kernel-stable with respect the game* $(\mathcal{A}, v)$.

Proof. Let $sur^*_{a^*,a^\circ}(C)$ be the surplus of agent a^* over agent a°, $a^*, a^\circ \in C \in \mathcal{C}$ in configuration $(\mathcal{C}, u^*)$. Then it holds

$$\begin{aligned}
sur^*_{a^*,a^\circ}(C) &= \max_{C^+ : a^* \in C^+, a^\circ \notin C^+} \{v^*(C^+) - \sum_{a \in C^+} u^*(a)\} \\
&= \max_{C^+ : a^* \in C^+, a^\circ \notin C^+} \{v(C^+) + r(- \sum_{a \in C^+} u(a) + r)\} \\
&= \max_{C^+ : a^* \in C^+, a^\circ \notin C^+} \{v(C^+) + r - \sum_{a \in C^+} u(a) - r\} \\
&= sur_{a^*,a^\circ}(C) \text{(in configuration}(\mathcal{C}, u).\text{)}
\end{aligned}$$

□

From this property of the kernel, we can make the following conclusion:

Corollary 7.6.2. *Assume that an agent a can modify coalition values as stated as in lemma 7.6.1 in a coalition game to form kernel-stable coalitions. Then a can completely hide its self value from other agents without loss of utility, under the assumption that they are not revealed by other means, by applying lemma 7.6.1 with $r = -v(\{a\})$.*

Example 7.6.3
Let a_1 offer a service ws at cost 2 and request the same service once with valuation 3. Let a_2 also offer the same service ws at cost 1, and request no service. Further assume that agents inform each other about their local worths in order to compute coalition values, but not about individual service offers and requests (it is not important here to consider how the local worths are actually obtained since we are only interested in what can be inferred from local worths and the game itself).

In the unmodified game, we thus have the following local worths and coalition values:

$$\begin{aligned}
lw_1(\{1\}) &= 3 - 2 = 1 & v(\{1\}) &= 1 \\
lw_2(\{2\}) &= 0 & v(\{2\}) &= 0 \\
lw_1(\{1,2\}) &= 3 & & \\
lw_2(\{1,2\}) &= -1 & v(\{1,2\}) &= 3 - 1 = 2
\end{aligned}$$

The kernel-stable configuration including the grand coalition of this game is $(\{\{1,2\}\}, u)$ with $u(1) = 1.5$ and $u(2) = 0.5$.

Now suppose that a likes to hide the fact that it accesses its own service which can be easily inferred from the fact that $lw_1(\{1\}) = 1$ (in this example, we ignore other means of getting this information that might be given in the specific negotiation protocol that is used). a_1 thus induces a modified game, resulting in different local worths for a_1 and different coalition values:

$$\begin{aligned} lw_1^*(\{1\}) &= 1 - 1 = 0 & v^*(\{1\}) &= 0 \\ lw_2^*(\{2\}) &= 0 & v^*(\{2\}) &= 0 \\ lw_1^*(\{1,2\}) &= 3 - 1 = 2 & & \\ lw_2^*(\{1,2\}) &= -1 & v^*(\{1,2\}) &= 2 - 1 = 1 \end{aligned}$$

The kernel-stable configuration including the grand coalition of this game is $(\{\{1,2\}\}, u^*)$ with $u^*(1) = 0.5$ and $u(2) = 0.5$. Now, observe that the side payments resulting from this are

$$\begin{aligned} sp_1^*(\{1,2\}) &= u^*(1) - lw_1^*(\{1,2\}) = 0.5 - 2 = -1.5 \\ sp_2^*(\{1,2\}) &= u^*(2) - lw_2^*(\{1,2\}) = 0.5 - (-1) = 1.5 \end{aligned}$$

But since a_1's valuation of ws is 3, after making the side payment it obtains still the same net payoff of

$$3 - 1.5 = 1.5 = u(1)$$

as in the unmodified game. Now, a_2 might infer that a_1's valuation of its request for ws must be $\geq lw_1^*(\{1,2\}) = 2$, because ws is the only service executed in the grand coalition. But it cannot infer anything about a_1's true self value. △

On the other hand, it was shown in Blankenburg and Klusch (2004) that in the classic kernel-based coalition formation protocol KCA, agents might establish a lower bound on an agent a's self value if it is not independent of the coalition that a ends up being in. This is because the KCA requires the public announcement of requests and offers including valuations and costs, respectively, among the participating agents. If we change our example to make a similar requirement, the same problem arises:

Example 7.6.4

Consider again example 7.6.4, but this time agents are expected to inform each other about their mutual requests and offers including valuations and costs, respectively. Then agent a_1 might still try to hide its self value by again modifying the game as in example 7.6.4. Additionally, it might not inform a_2 about a_1's own offer of ws, which is not used outside a_1's single-agent coalition and thus this omitment has no impact on the modified game.

However, in this scenario, a_2 is informed about a_1's valuation of its request for ws, namely that it is 3. But this does not match the fact that $lw_1^*(\{1,2\}) = 2$ because, again, ws is the only executed service in the grand coalition. Thus, a_2 might infer that a_1's self value is at least 1 ("at least" because a_1 potentially executes yet more services for its own requests in $\{1\}$ which a_2 doesn't know about). △

The two examples demonstrate once more that the exact design of the communication protocol within a coalition formation algorithm is crucial when privacy preservation is concerned. However, it is not quite clear whether the kernel theoretically allows for more privacy preservation at all. Since in the kernel, all agents in a coalition need to be in quilibrium with respect to their surpluses, this seems rather unlikely.

Also, note that the kernel of a two-agent game is equal to the Shapley value of that game (see e.g. Peleg and Sudhölter, 2007). Therefore, if one devised a "recursively bilateral kernel" analogously to the recursively bilateral Shapley value in order to again exploit the tree structure, these two solution concepts would simply be the same.

7.7 Summary

We have proven that the recursive nature of the recursively bilateral Shapley value can be exploited to enable iterative bilateral coalition formation while keeping agents' local worths and coalition values hidden. In fact, agents can compute their stable side payments at each round of coalition formation directly using only their side payments from the previous round and some additional local worths, the latter modeling only the difference of local worths to the local worths of the previous round.

We then also proved that this is however not enough to enable privacy preserving coalition formation, since it is still possible to deduce the existence of service requests among certain agents from additional local worths in particular situations.

To remedy this, we outlined how minimum conctraints on agent and service anonymity can be used together with an anonymous routing protocol to allow coalition formation with at least some degree of privacy.

Based on these results, we proposed the algorithm and protocol for privacy preserving and stable coalition formation among rational service agents, BSCA-P. It allows the participating agents to hide their service requests and payoffs while having low polynomial computational and communication complexities. Specifically, the BSCA-P was proved to

1. adhere to agents' and service anonymity constraints,
2. hide the agents' local worths, and in particular its self values, from all other agents,
3. hide the agents' additional local worths from all other agents but the one with whom they first form a coalition and

To our best knowledge, the BSCA-P is the first coalition formation algorithm to have this set of properties, and also the first privacy preserving coalition formation algorithm in general, at least when similar notions of privacy are regarded.

Additionally, it was outlined how services and side payments should be executed such that the level of privacy and anonymity established by the BSCA-P is maintained also over these activities.

In summary, the BSCA-P allows service agents in open environments to keep personal financial data private, while increasing their individual profits by means of rational cooperation with others in coalitions.

Finally, we showed that agents can theoretically hide their self values completely also in kernel-based coalition formation if an appropriate coalition formation algorithm and protocol are employed. However, as we demonstrated by an example, this might not be easy to achieve.

Chapter 8

Conclusion

8.1 Research Questions Answered

Fuzzy-valued coalition formation

1. *How can stable coalitions be formed efficiently when the coalition values are fuzzy?*

   To form fuzzy-valued coalitions efficiently, we devised a new coalition formation algorithm BSCA-F by extending an existing one for crisp games, the BSCA, to cope with fuzzy coalition values. Therefore, we provided a fuzzified version of the underlying solution concept, the (recursively) bilateral Shapley value with defuzzified subcoalition values. We showed that this induces coalitions to have unambiguous preferences for other coalitions to merge with, provided that the same possibilistic ranking operator is used by all agents. Thus, no additional communication steps are necessary in the proposal generation and evaluation steps. We could therefore prove that polynomial computation and communication complexities could be maintained also in the BSCA-F.

2. How can a resulting fuzzy solution be used to determine concrete, non-fuzzy side-payments in a stable manner?

   It was shown that employing the possibilistic mean defuzzification method is generally applicable to fuzzy payoffs obtained with the BSCA-F. In particular, such defuzzified values were shown to gave a high correlation to the fuzzy Shapley value's possibilistic mean, in particular for the possibility-based ranking operators.

3. What impact does the choice of possibilistic ranking operators have on the resulting payoffs?

   It was shown by evaluation that the different operators do lead to different solutions with different characteristics. Generally, the possibility based operators maximize the possibilistic expected payoff, while the necessity based operators minimize worst case losses, and specifically $\widetilde{>}_N$ manages to achieve a higher degree of necessity of core membership. Therefore, agents might choose one based on their preferences.

4. Are the resulting fuzzy payoffs core-stable?

   As the evaluation demonstrated, the possibility that the fuzzy payoffs obtained via the BSCA-F lie in the core is quite high for all operators, as is the number of payoffs which have possibility degree of 1 to lie in the core. However, the necessity of payoffs to be in the core was shown to be much smaller generally, but the $\widetilde{>}_N$-operator achieves much better degrees than the other operators in this case.

Risk-bounded coalition formation

1. How can resource-bounded agents reduce the risk of suffering losses due to failing coalitions according to an approved measures of risk?

   Using a coherent risk measure, it was shown that an agent can reduce its risk by being a member of multiple fuzzy coalitions. The coherence of the risk measure ensures that if an agent a is a member of a set of fuzzy coalitions whose combined risk is acceptable for a and additionally becomes a member of another coalition whose risk is acceptable, then the overall risk is also acceptable for a.

2. How can stability of risk-bounded coalitions be ensured and what is the computational cost of such an approach?

   The kernel was extended to adhere to agents' individual risk bounds. The extension was necessary because the risk depends on the payoffs, and a standard kernel stable payoff distribution might thus be unacceptable to some agents given their risk bounds. We showed that the computational complexity to compute a risk-bounded kernel stable coalitions can be reduced to be polynomial if each of the coalition size, the number of plans containing the same set of agents, and the number of coalitions in which an agent might simultaneously be a member of are bounded by constants. Using this, we provided an algorithm which forms risk-bounded kernel stable coalitions in polynomial time.

Coalition formation with deceiving agents

How can stable coalitions be formed while preventing rational agents from deceiving during

1. coalition negotiations and
2. side payment executions?

1. A protocol to communicate initial data about requests and offers was devised. Using cryptographic techniques, it was shown that this prevents the agents to determine manipulated valuations and costs to send in order to unjustifiably increase their payoffs in kernel stable coalitions. Specifically, for the kernel it was proved that an agent has to manipulate data in a certain way in order to receive a higher net payoff. But with the proposed protocol, it was shown that no agent is able to determine such values before it has to send them to the other agents.

   For repeated coalition formation, a trust model was additionally integrated in the coalition formation algorithm. This is used to compute corrected expected coalition values if agents over- or understate their capabilities. Agents also exchange trust values among each other to assess one another's reputation. Since it was also shown that reporting over- or understated trust values has a similar effect on kernel stable payoffs as sending manipulated valuations and costs, an agent's chances to gain a profit from defrauding are further decreased.
2. A protocol for the execution of side payments was devised, again using cryptographic techniques. The protocol orders the agents into a sequence such that successive side payments are made. It was shown that for all except one agents in the sequence, it is rational to stick to the protocol

and execute the side payment at the specify time. The last agent, however, is shown to have to execute some assigned tasks (or services), and therefore will be subject to trust measure evaluation by other agents.

Privacy-preserving coalition formation

1. How can profitable coalitions efficiently be formed while adhering to privacy constraints?

   A privacy preserving coalition formation algorithm BSCA-P was proposed and shown to let agents hide their private financial data (almost) completely. Further, it was shown that service requests can be kept private to given minimum degrees of anonymities. In particular, it was proved that by exploiting the coalition tree structure induced by the recursively bilateral Shapley value, the side payments can be computed without direct reference to private values such as local worths and service requests, but by using only the additional local worths. But it was also shown that private information might be deduced from additional local worths in certain situations, depending on the size of the involved coalitions and sets of offered services. Therefore, we proposed to employ possibilistic k-anonymity in combination with minimum anonymity constraints.

   Finally, we also briefly considered the kernel in terms of privacy preservation, and proved that agents can hide their self values completely if the coalition formation protocol is designed appropriately.

2. How can adherence to these privacy constraints be maintained also during the side payment and execution phases?

   It was shown that the coalition tree structure can be reused to not reveal any further information in the side payment execution phase.

   For the service execution phase, it was shown that accessing services via the anonymous message routing protocol used during the coalition negotiation, adherence to the anonymity constraints is maintained.

8.2 Outlook

In this thesis, we provided algorithms and mostly theoretical results for coalition formation in uncertain and untrustworthy environments. Although these results indicate that coalition formation algorithms and protocols can be adapted to such environments, there is still much to be done. In particular, the questions of truth-telling and privacy preservation should be further explored. While these issues have been addressed in other, also related, areas like mechanism design, they have not received much attention in the coalition formation research community. The approaches outlined in this thesis are solid, but can only be first steps. For instance, while the TKCF encourages hinders deception and ensures execution of side payments, the coalition formation phase itself was kept very simple. Also, it always forms the social welfare coalition structure. It would thus be beneficial to devise a more flexible algorithm which still gives the same guarantees.

Concerning privacy preserving coalition formation, it is as of now unclear which other solution concepts might be employed while keeping the agents' information private. Also, a very simple measure of anonymity was used, which might be extended in order to model more elaborate anonymity constraints. Furthermore, the combination of trusted and privacy preserving coalition formation seem to pose a major obstacle, since their goals seem to contradict each other: while trusted coalitions formation aims to make agents reveal their true private values, the goal of privacy preserving coalition formation is to allow agents to hide as much information as possible.

Finally, while coalition formation under uncertainty has received the greatest attention of these problems in the literature, algorithms that at the same time are fast, lead to stable results, and are adaptable to agents' needs yet have to be devised.

Bibliography

S. Airiau and S. Sen. On the stability of an optimal coalition structure. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 203–208. IOS Press, 2010. ISBN 978-1-60750-605-8.

L. Amgoud. An argumentation-based model for reasoning about coalition structures. In S. Parsons, N. Maudet, P. Moraitis, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems*, volume 4049 of *Lecture Notes in Computer Science*, pages 217–228. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-36355-2. doi: 10.1007/11794578_14.

S. Arib and S. Aknine. A plan based coalition formation model for multi-agent systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 2, pages 365–368. IEEE Computer Society, 2011. doi: 10.1109/WI-IAT.2011.222.

S. Arib and S. Aknine. Enhancing coalition formation in multi-agent systems when agents plan their activities. In K. Kersting and M. Toussaint, editors, *STAIRS*, volume 241 of *Frontiers in Artificial Intelligence and Applications*, pages 11–22. IOS Press, 2012. ISBN 978-1-61499-095-6.

T. Arnold and U. Schwalbe. Dynamic coalition formation and the core. *Journal of Economic Behavior & Organization*, 49(3):363 – 380, 2002. ISSN 0167-2681. doi: 10.1016/S0167-2681(02) 00015-X.

P. Artzner, F. Delbaen, S. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, pages 203–228, 1999. ISSN 1467-9965.

J.-P. Aubin. *Mathematical Methods of Game and Economic Theory*. North-Holland, 1979.

J.-P. Aubin. Dynamic core of fuzzy dynamical cooperative games. In A. Nowak and K. Szajowski, editors, *Advances in Dynamic Games*, volume 7 of *Annals of the International Society of Dynamic Games*, pages 129–162. Birkhäuser Boston, 2005. ISBN 978-0-8176-4362-1. doi: 10.1007/0-8176-4429-6_7.

R. Aumann. Some non-superadditive games, and their shapley values, in the talmud. *International Journal of Game Theory*, 39:3–10, 2010. ISSN 0020-7276. doi: 10.1007/s00182-009-0191-4.

R. J. Aumannn. *Collected Papers*, volume 1. MIT Press, Cambridge, MA, USA, 1997.

R. J. Aumannn. *Collected Papers*, volume 2. MIT Press, Cambridge, MA, USA, 1999.

R. J. Aumannn and S. Hart, editors. *Handbook of Game Theory With Economic Applications*, volume 1 of *Handbooks in Economics*. Elsevier, 1992.

Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2*, AAMAS '08, pages 1023–1030, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-1-6.

P. Bartalos and M. Bieliková. Automatic dynamic web service composition: A survey and problem formalization. *Computing and Informatics*, 30(4):793–827, 2011.

L. Barton and V. Allan. Adapting to changing resource requirements for coalition formation in self-organized social networks. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, volume 2, pages 282 –285. IEEE Computer Society, dec. 2008. doi: 10.1109/WIIAT. 2008.121.

B. Blankenburg and M. Klusch. On safe kernel stable coalition forming among agents. In *Proceedings of the Third International Joint*

Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '04, pages 580–587. IEEE Computer Society, 2004. ISBN 1-58113-864-4. doi: 10.1109/AAMAS.2004.201.

B. Blankenburg and M. Klusch. Bsca-p: Privacy preserving coalition formation. In T. Eymann, F. Klügl, W. Lamersdorf, M. Klusch, and M. Huhns, editors, *Multiagent System Technologies*, volume 3550 of *Lecture Notes in Computer Science*, pages 47–58. Springer Berlin Heidelberg, 2005a. ISBN 978-3-540-28740-7. doi: 10.1007/11550648_5.

B. Blankenburg and M. Klusch. Bsca-f: E.cient fuzzy valued stable coalition forming among agents. *Intelligent Agent Technology, IEEE / WIC / ACM International Conference on*, 0:732–738, 2005b. doi: http://doi.ieeecomputersociety.org/10.1109/IAT.2005.48.

B. Blankenburg and M. Klusch. Privacy-preserving coalition forming between rational service agents. *KI - Zeitschrift Künstliche Intelligenz, Organ des Fachbereiches "Künstliche Intelligenz" der Gesellschaft für Informatik e.V. (KI)*, 1/2006:18–24, 2006.

B. Blankenburg, M. Klusch, and O. Shehory. Fuzzy kernel-stable coalitions between rational agents. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 9–16, New York, NY, USA, 2003. ACM. ISBN 1-58113-683-8. doi: 10.1145/860575.860578.

B. Blankenburg, R. K. Dash, S. D. Ramchurn, M. Klusch, and N. R. Jennings. Trusted kernel-based coalition formation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 989–996, New York, NY, USA, 2005. ACM. ISBN 1-59593-093-0. doi: 10.1145/1082473.1082623.

B. Blankenburg, M. He, M. Klusch, and N. R. Jennings. Risk-bounded formation of fuzzy coalitions among service agents. In *Proceedings of the 10th international conference on Cooperative Information Agents*, CIA'06, pages 332–346. Springer Berlin Heidelberg, 2006. ISBN 3-540-38569-X, 978-3-540-38569-1. doi: 10.1007/11839354_24.

G. Bortolan and R. Degani. A review of some methods of ranking fuzzy subsets. *Fuzzy Sets and Systems*, 15(1):1–19, 1985. ISSN 0165-0114.

R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Science/Engineering/Math, New York, 3rd edition, 1999.

S. Breban and J. Vassileva. Using inter-agent trust relationships for efficient coalition formation. In R. Cohen and B. Spencer, editors, *Advances in Artificial Intelligence*, volume 2338 of *Lecture Notes in Computer Science*, pages 221–236. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-43724-6. doi: 10.1007/3-540-47922-8_19.

D. Butnariu. Stability and shapley value for an n-persons fuzzy game. *Fuzzy Sets and Systems*, 4:63–72, 1980. ISSN 0165-0114.

C. Carlsson and R. Fullér. On possibilistic mean and variance of fuzzy numbers. *Fuzzy Sets and Systems*, 122:315–326, 2001. ISSN 0165-0114.

G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '04, pages 1090–1097. IEEE Computer Society, 2004. ISBN 1-58113-864-4. doi: 10.1109/AAMAS.2004.74.

G. Chalkiadakis and C. Boutilier. Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents and Multi-Agent Systems*, 24(3):441–484, 2012. ISSN 1387-2532.

G. Chalkiadakis, E. Markakis, and C. Boutilier. Coalition formation under uncertainty: bargaining equilibria and the bayesian core stability concept. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 64:1–64:8, New York, NY, USA, 2007. ACM. ISBN 978-81-904262-7-5. doi: 10.1145/1329125.1329203.

G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. Jennings. Cooperative games with overlapping coalitions. *Journal of Artificial Intelligence Research (JAIR)*, 39:179–216, September 2010. URL http://eprints.soton.ac.uk/271574/.

G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Synthesis Lectures on Artificial

Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

G. Chalkiadakis, E. Markakis, and N. R. Jennings. Coalitional stability in structured environments. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 779–786, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 0-9817381-2-5, 978-0-9817381-2-3.

J. Chen and D. Sun. Coalition-based approach to task allocation of multiple robots with resource constraints. *IEEE Transactions on Automation Science and Engineering*, 9(3):516–528, 2012.

S. Cheng, Y. Liu, and S. Wang. Progress in risk management. *Advanced Modelling and Optimization*, 6(1):1–20, 2004. ISSN 1841-4311.

S. Chowdhury, P. Champagne, and P. J. McLellan. Uncertainty characterization approaches for risk assessment of dbps in drinking water: A review. *Journal of Environmental Management*, 90(5): 1680 – 1691, 2009. ISSN 0301-4797. doi: 10.1016/j.jenvman.2008.12.014.

V. Conitzer and T. Sandholm. Computational criticisms of the revelation principle. In *Proceedings of the 5th ACM conference on Electronic commerce*, EC '04, pages 262–263, New York, NY, USA, 2004. ACM. ISBN 1-58113-771-0. doi: 10.1145/988772.988824.

J. Contreras, M. Klusch, and J. Yen. Multi-agent coalition formation in power transmission planning: A bilateral shapley value approach. In *Artificial Intelligence Planning Systems*, pages 19–26. AAAI Press, 1998.

P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Blackwell, 1998.

R. K. Dash, S. D. Ramchurn, and N. R. Jennings. Trust-based mechanism design. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, pages 748–755. IEEE Computer Society, 2004. ISBN 1-58113-864-4. doi: 10.1109/AAMAS.2004.287.

M. Davis and M. Maschler. The kernel of a cooperative game. *Naval Research Logistics Quarterly*, 12:223–259, 1965. ISSN 1931-9193.

M. Delgado, J. L. Verdegay, and M. A. Vila. Fuzzy numbers, definitions and properties. *Mathware & Soft Computing*, 1:31–43, 1994. ISSN 1134-5632).

X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematical Operations Research*, 19(2):257–266, May 1994. ISSN 0364-765X. doi: 10.1287/moor.19.2.257.

D. Dubois and H. Prade. Ranking fuzzy numbers in the setting of possibility theory. *Information Sciences*, 30:183–224, 1983. ISSN 0020-0255.

D. Dubois and H. Prade. The mean value of a fuzzy number. *Fuzzy Sets and Systems*, 24:279–300, 1987. ISSN 0165-0114.

D. Dubois and H. Prade. Fuzzy numbers: an overview. In J. C. Bezdek, editor, *Analysis of fuzzy information*, volume I Mathematics and Logic, pages 3–39. CRC Press, Boca Raton, Florida, 1994.

E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2*, AAMAS '08, pages 1007–1014, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-1-6.

J. A. Filar and L. A. Petrosjan. Dynamic cooperative games. *International Game Theory Review*, 2(1):47–65, 2000.

S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL'96)*, volume 1193. Springer Berlin Heidelberg, 1996.

R. Fullér and P. Majlender. On weighted possibilistic mean and variance of fuzzy numbers. *Fuzzy Sets and Systems*, 136:363–374, 2003. ISSN 0165-0114.

R. E. Giachetti and R. E. Young. A parametric representation of fuzzy numbers and their arithmetic operators. *Fuzzy Sets and Systems*, 92(2):207–219, 1997. ISSN 0165-0114.

N. Griffiths and M. Luck. Coalition formation through motivation and trust. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 17–24, New York, NY, USA, 2003. ACM. ISBN 1-58113-683-8. doi: 10.1145/860575.860579.

G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.

J. Guerrero and G. Oliver. Multi-robot coalition formation in real-time scenarios. *Robotics and Autonomous Systems*, 60(10):1295–1307, 2012. ISSN 0921-8890.

J. Halpern and K. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, Special Edition on CSFW 16:75–88, 2003.

I. Hirschman and D. Widder. *The Convolution Transform*. Princeton Mathematical Series. Princeton University Press, 1955. ISBN 9780486154565.

B. Hoelz and C. Ralha. A coalition formation mechanism for trust and reputation-aware multi-agent systems. In L. Barros, M. Finger, A. Pozo, G. Gimenénez-Lugo, and M. Castilho, editors, *Advances in Artificial Intelligence - SBIA 2012*, Lecture Notes in Computer Science, pages 162–171. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-34458-9. doi: 10.1007/978-3-642-34459-6_17.

L. Hurwicz. On informationally decentralized systems. In C. B. McGuire and R. Radner, editors, *Decision and Organization: A volume in Honor of Jacob Marschak*, volume 12 of *Studies in Mathematical and Managerial Economics*, chapter 14, pages 297–336. North-Holland, 1972.

E. Hüllermeier. Possibilistic instance-based learning. *Artificial Intelligence*, 148(1–2):335 – 383, 2003. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(03)00019-5. <ce:title>Fuzzy Set and Possibility Theory-Based Methods in Artificial Intelligence</ce:title>.

S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *ACM Conference on Electronic Commerce*, pages 193–202, New York, NY, USA, 2005. ACM.

S. Ieong and Y. Shoham. Bayesian coalitional games. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 1*, AAAI'08, pages 95–100. AAAI Press, 2008. ISBN 978-1-57735-368-3.

C. Jones and K. Barber. Combining job and team selection heuristics. In J. Hübner, E. Matson, O. Boissier, and V. Dignum, editors, *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, volume 5428 of *Lecture Notes in Computer Science*, pages 33–47. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-00442-1. doi: 10.1007/978-3-642-00443-8_3. URL http://dx.doi.org/10.1007/978-3-642-00443-8_3.

J. P. Kahan and A. Rapoport. *Theories of coaltion formation.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1984.

S. P. Ketchpel. Coalition formation among autonomous agents. In C. Castelfranchi and J.-P. Müller, editors, *From Reaction to Cognition*, volume 957 of *Lecture Notes in Computer Science*, pages 73–88. Springer Berlin Heidelberg, 1995. ISBN 978-3-540-60155-5. doi: 10.1007/BFb0027057.

Z. Khan, S. Glisic, L. DaSilva, and J. Lehtomäki. Modeling the dynamics of coalition formation games for cooperative spectrum sharing in an interference channel. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(1):17 –30, march 2011. ISSN 1943-068X. doi: 10.1109/TCIAIG.2010.2080358.

M. Klusch. *Kooperative Informationsagenten im Internet.* Verlag Dr. Kovač, 1997.

M. Klusch, editor. *On Agent-Based Semantic Service Coordination.* Universität des Saarlandes, 2008.

M. Klusch and A. Gerber. Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, 17(3):42–47, May 2002. ISSN 1541-1672. doi: 10.1109/MIS.2002.1005630.

M. Klusch and O. Shehory. Coalition formation among rational agents. In *Proceedings of the European Workshop Modelling Autonomous Agents in a Multi-agent World (MAAMAW)*, Lecture Notes in Artificial Intelligence, pages 204–217. Springer Berlin Heidelberg, 1996.

M. Klusch and O. Shehory. A polynomial kernel-oriented coalition algorithm for rational information agents. In *Proceedings of the second International Conference on Multi-Agent Systems*. AAAI Press, 1996.

S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2):79–97, 1997. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(97)00025-8.

S. Kraus, O. Shehory, and G. Taase. Coalition formation with uncertain heterogeneous information. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 1–8, New York, NY, USA, 2003. ACM. ISBN 1-58113-683-8. doi: 10.1145/860575.860577.

S. Li and Q. Zhang. A simplified expression of the shapley function for fuzzy game. *European Journal of Operational Research*, 196(1): 234 – 245, 2009. ISSN 0377-2217. doi: 10.1016/j.ejor.2008.02.034.

M. Mareš. *Fuzzy cooperative games: cooperation with vague expectations*, volume 72 of *Studies in fuzziness and soft computing*. Physica Verlag, Heidelberg, 2001.

T. Michalak, A. Dowell, P. McBurney, and M. Wooldridge. Preprocessing techniques for anytime coalition structure generation algorithms. In J.-J. Meyer and J. Broersen, editors, *Knowledge Representation for Agents and Multi-Agent Systems*, volume 5605 of *Lecture Notes in Computer Science*, pages 99–113. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05300-9. doi: 10.1007/978-3-642-05301-6_7.

R. B. Myerson. mechanism design. In S. N. Durlauf and L. E. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, Basingstoke, 2008.

R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29 (2):265–281, April 1983. ISSN 0022-0531. available at http://ideas.repec.org/a/eee/jetheo/v29y1983i2p265-281.html.

I. Nishizaki and M. Sakawa. *Masatoshi Fuzzy and multiobjective games for conflict resolution*, volume 64 of *Studies in Fuzziness and Soft Computing*. Physica-Verlag, Heidelberg, 2001.

N. Ohta and V. Conitzer. Anonymity-proof shapley value: extending shapley value for coalitional games in open environments. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2*, pages 927–934, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-1-6.

N. Ohta, A. Iwasaki, M. Yokoo, K. Maruono, V. Conitzer, and T. Sandholm. A compact representation scheme for coalitional games in open anonymous environments. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*. AAAI Press, 2006.

M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, Aug. 2003. ISBN 0195128958.

M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, USA, 1994.

G. Owen. *Game Theory*. Academic Press, New York, dritte edition, 1995.

OWL-Services-Coalition. Owl-s: Semantic markup for web services. White Paper, 2004. URL `http://www.w3.org/Submission/OWL-S`.

A. Patil, S. Oundhakar, A. Sheth, and K. Verma. Meteor-s web service annotation framework. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW2004)*, New York, NY, USA, May 2004. LSDIS Lab, University of Georgia, ACM.

B. Peleg and P. Sudhölter. *Introduction to the Theory of Cooperative Games*, volume 34 of *Theory and Decision Library C*. Springer US, 2 edition, 2007.

A. Pfitzmann and M. Köhntopp. Anonymity, unobservability and pseudonymity: a proposal for terminology. In *International Workshop on Designing Privacy Enhancing Technologies*, pages 1–9. Springer US, 2001.

T. Rahwan. *Algorithms for Coalition Formation in Multi-Agent Systems*. PhD thesis, University of Southampton, August 2007.

T. Rahwan, S. D. Ramchurn, N. R. Jennings, and A. Giovannucci. An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research*, 34(1):521–567, Apr. 2009. ISSN 1076-9757.

T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, 186:95–122, 2012. ISSN 0004-3702.

Z. Ramaekers, R. Dasgupta, V. Ufimtsev, S. G. M. Hossain, and C. Nelson. Self-reconfiguration in modular robots using coalition games with uncertainty. In *Automated Action Planning for Autonomous Mobile Robots*, volume WS-11-09 of *AAAI Workshops*. AAAI Press, 2011.

S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multiagent systems. *Knowedge Engineering Review*, 19(1):1–25, Mar. 2004. ISSN 0269-8889. doi: 10.1017/S0269888904000116.

A. Rapoport. *N-Person Game Theory: Concepts and Applications*. Dover Publications, Inc., Dover, 2001.

E. Raufaste, R. D. S. Neves, and C. Mariné. Testing the descriptive validity of possibility theory in human judgments of uncertainty. *Artificial Intelligence*, 148(1-2):197–218, 2003. ISSN 0004-3702.

L. Riley, K. Atkinson, and T. R. Payne. A dialogue game for coalition structure generation with self-interested agents. In B. Verheij, S. Szeider, and S. Woltran, editors, *Computational Models of Argument - Proceedings of COMMA 2012*, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 229–236. IOS Press, 2012. ISBN 978-1-61499-110-6.

R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21 (2):120–126, Feb. 1978. ISSN 0001-0782. doi: 10.1145/359340.359342.

F. Rohrmüller, D. Wollherr, and M. Buss. Muroco: A framework for capability- and situation-aware coalition formation in cooperative multi-robot systems. *Journal of Intelligent and Robotic Systems*, 67(3-4):339–370, 2012. ISSN 0921-0296.

D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Appl. Ontol.*, 1(1):77–106, Jan. 2005. ISSN 1570-5838.

T. Sandholm. Distributed rational decision making. In *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, pages 201–258. MIT Press, Cambridge, MA, USA, 1999.

B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

T. Service and J. Adams. Coalition formation for task allocation: theory and algorithms. *Autonomous Agents and Multi-Agent Systems*, 22(2):225–248, 2011. ISSN 1387-2532. doi: 10.1007/s10458-010-9123-8.

L. S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pages 307–317. Princeton University Press, Princeton, 1953.

O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, 1998. ISSN 0004-3702.

O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents in nonsuperadditve environments. *Computational Intelligence*, 15:218–251, 1999. http://citeseer.nj.nec.com/shehory99feasible.html.

Y. Shoham and K. Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009. ISBN 978-0-521-89943-7.

T. Shrot, Y. Aumann, and S. Kraus. On agent types in coalition formation problems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 757–764, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9826571-1-9. URL `http://dl.acm.org/citation.cfm?id=1838206.1838307`.

R. E. Stearns. Convergent transfer schemes for n-person games. *Transactions of the American Mathematical Society*, 134:449–459, 1968.

J. Suijs. *Cooperative Decision Making in a Stochastic Environment*. PhD thesis, Tilburg University, Tilburg: CentER Dissertation Series, 1998.

J. Suijs, A. De Waegenaere, and P. Borm. Stochastic cooperative games in insurance. *Insurance: Mathematics and Economics*, 22 (3):209–228, July 1998.

J. Suijs, P. Borm, A. D. Waegenaere, and S. Tijs. Cooperative games with stochastic payoffs. *European Journal of Operational Research*, 113:193–205, 1999. ISSN 0377-2217.

P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 4–7 1997. ISBN 0-8186-7828-3. URL `citeseer.ist.psu.edu/syverson97anonymous.html`.

J. Vassileva, S. Breban, and M. Horsch. Agent reasoning mechanism for long–term coalitions based on decision making and trust. *Computational Intelligence*, 18(4):583–595, 2002. ISSN 1467-8640. doi: 10.1111/1467-8640.t01-1-00205.

L. Vig and J. Adams. Coalition formation: From software agents to robots. *Journal of Intelligent and Robotic Systems*, 50:85–118, 2007. ISSN 0921-0296. doi: 10.1007/s10846-007-9150-0.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. ISBN 0691119937.

M. Wooders. Small group effectiveness, per capita boundedness and nonemptiness of approximate cores. *Journal of Mathematical Economics*, 44(7-8):888–906, July 2008.

M. Wooldridge and P. E. Dunne. On the computational complexity of coalitional resource games. *Artifical Intelligence*, 170(10):835–871, July 2006. ISSN 0004-3702. doi: 10.1016/j.artint.2006.03.003.

L. Xiao, C. K. Chang, H.-I. Yang, K.-S. Lu, and H. yi Jiang. Automated web service composition using genetic programming. *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, 0:7–12, 2012. doi: http://doi.ieeecomputersociety.org/10.1109/COMPSACW.2012.12.

B. Xiao-fei, Y. Yang, and X. song Qiu. Multi-task overlapping coalition formation mechanism in wireless sensor network. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 635–638, 2012. doi: 10.1109/NOMS.2012.6211974.

M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki. Coalition games in open anonymous environments. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 509–514. AAAI Press, 2005a.

M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki. Coalitional games in open anonymous environments. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1668–1669, 2005b. ISBN 0938075934.

L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965. ISSN 0019-9958. doi: http://dx.doi.org/10.1016/S0019-9958(65)90241-X.

L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978. ISSN 0165-0114.