Risk-bounded Formation of Fuzzy Coalitions among Service Agents

Bastian Blankenburg[1], Minghua He[2], Matthias Klusch[1], and Nicholas R. Jennings[2]

[1] German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3,
66123 Saarbrücken, Germany; {blankenb,klusch}@dfki.de
[2] School of Electronics and Computer Science, University of Southampton,
Southampton, SO17 1BJ, UK; {mh,nrj}@ecs.soton.ac.uk

Abstract. Cooperative autonomous agents form coalitions in order to share and combine resources and services to efficiently respond to market demands. With the variety of resources and services provided online today, there is a need for stable and flexible techniques to support the automation of agent coalition formation in this context. This paper describes an approach to the problem based on fuzzy coalitions. Compared with a classic cooperative game with crisp coalitions (where each agent is a full member of exactly one coalition), an agent can participate in multiple coalitions with varying degrees of involvement. This gives the agents more freedom and flexibility, allowing them to make full use of their resources, thus maximising utility, even if only comparatively small coalitions are formed. An important aspect of our approach is that the agents can control and bound the risk caused by the possible failure or default of some partner agents by spreading their involvement in diverse coalitions.

1 Introduction

In today's increasingly networked and competitive world, the appropriate utilization of pay per use Web services are considered as one major key to the success of commercial service oriented business applications in domains such as e-logistics, tourism, and entertainment. In the near future, intelligent service agents are not only supposed to search for, interact with, and compose, but also negotiate access to, and execute such Web services on behalf of its user, or other agents. In fact, they may exhibit some form of economically rational cooperation by forming coalitions to share the created joint monetary value while at the same time maximizing their own individual payoff. According to classical microeconomics, means and concepts of cooperative game theory are inherently well suited to this purpose. In this paper, we propose a protocol for resource-bounded computational rational agents to automatically form risk-bounded fuzzy coalitions in order to fulfill service requests with deadlines.

As opposed to traditional cooperative games, games with fuzzy coalitions allow the agents to be members of multiple coalitions with varying degrees of involvement. The notion of fuzzy coalitions was first introduced by Aubin and Butnariu (see [2, 5]) to overcome some problems of traditional cooperative games in real-world settings. For example, suppose that agent a_1 can independently benefit from cooperations with agent a_2 as well as with agent a_3. To realise both opportunities, a coalition of all three agents has to be formed, requiring a_2 and a_3 to agree on a coalition contract although they do not cooperate otherwise. Another drawback of non-overlapping coalitions emerges in the case of failure. If, in the above example, a_2 fails its task, thus reducing the coalition value, all members of the coalition are affected, including a_3, although it is not actually working together with a_2. In contrast, with fuzzy coalitions makes it is possible to form a coalition for each service request, without preventing other requests from being satisfied. This approach has the advantage that there are no unnecessary negotiations and contracts between agents which actually do not work together.

Additionally, using fuzzy coalitions allows the agents to lower their individual risk of monetary losses by participating in a number of coalitions, if coherent risk measure are considered. Assuming that agents are able to assess other agent's risk of failure in a coalition, we show how such risk-bounded coalition formation can be done. In particular, we consider the membership of an agent in a coalition as an investment, since the costly service execution takes place first. Rewards are received later only for successful and timely execution. We thus allow the agents to specify individual risk bounds in terms of the coherent financial risk measure *tail conditional expectation* (TCE). The adherence to these bounds is guaranteed by the proposed coalition formation protocol RFCF.

But as it turns out, we cannot directly use the existing solution concepts for cooperative games with fuzzy coalitions. The approaches taken by Aubin, Butnariu as well as Nishizaki and Sakawa (see [9]) all 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 [10]. Stearns transfer scheme can then be used to compute Kernel-stable solutions for the game (see [11]).

The remainder of this paper is organized as follows: in section 2 we introduce our service agent and coalition model. In section 3, we introduce our 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 4. Section 5 is concerned with the stability of risk-bounded fuzzy coalitions. We propose our coalition formation protocol RFCF in section 6. In section 7 we discuss related work and conclude in section 8.

2 Agent Model

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

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

Definition 1 *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* $r \in \mathbb{R}$ *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.*

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 2 *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.*
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*

   $$t_i = \frac{1}{r_i} \times t_i^{min}.$$

   (e) *spa might not be able to detemine t_i^{min} exactly in advance, but is able to specify a probability density function (PDF) $pdf_{t_i^{min}}$ over the values it might take.*
   (f) *there is a monetary cost for resource consumption of spa. We assume this is constant, so that because of Definition 2.2(d) 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.

Note that because of the linear relationship assumed in Definition 2.2(d), it is easy to obtain the PDF of the execution time of a service instance i with a given fraction of resources per time r_i:

$$pdf_{t_i}(x) = pdf_{t_i^{min}}(r_i * x) \tag{1}$$

Example 1. As an example, we consider a medical service provider agent scenario. We assume that there are a number of these agents in the system, each offering medical information in one or more specific medical domains. A specific set of symptoms of a patient might have possible diagnosis in several domains. Thus, a full diagnosis as response to a request from e.g. a medical doctor might require a set of provider agents to collaborate. We assume that the medical personnel will request this information with specific deadlines to ensure the timely treatment of patients. Suppose that agent spa_1 gets a request from a doctor and realizes that it also needs spa_2 to provide a feasible diagnosis. spa_1 then estimates the runtime for its own service on the request and sends coalition proposal to spa_2. spa_2 then likewise estimates its runtime and sees that this coalition might actually fail, producing high costs. However, spa_2 has a further request from agent spa_3. While forming just the coalition spa_1 is too risky for spa_2, it is acceptable if the coalition with spa_3 is also formed.

3 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, a number of authors (most notably Aubin, Butnariu and Nishizaki and Sakawa [2, 5, 9]) 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 3 *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 $\widetilde{C}$.*

We also denote "fuzzy coalition" or just "coalition" instead of "fuzzy coalition of service provider agents" where the context is clear.

Because of the deadlines for service requests, $\widetilde{C}$ either earns the reward r for the successful and timely execution of $\mathcal{P}$ from the requesting agent, or nothing otherwise. To specify coalition values for fuzzy service provider agent coalitions, we thus need to consider its probabilities of failure and success. For simplicity, we assume that the execution times of service instances are independent of each other and that the services in a plan $\mathcal{P}$ are executed sequentially. Then, the total execution time of $\mathcal{P}$ is the sum of the execution times of the individual service instances:

$$t_{\mathcal{P}} = \sum_{i \in \mathcal{P}} t_i \tag{2}$$

The PDF of the sum of two independent random variables A and B is given by the *convolution integral* over their individual PDFs pdf_A and pdf_B(see, e.g., [8], p. 113). I.e., with $x \in \mathbb{R}$:

$$\begin{aligned} pdf_{A+B}(x) &= (pdf_A * pdf_B)(x) \\ &= \int_0^\infty pdf_A(y) pdf_B(x-y) dy \end{aligned} \tag{3}$$

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

$$pdf_{t_{\mathcal{P}}}(x) = (\cdots(pdf_{t_{i_1}} * pdf_{t_{i_2}}) \cdots * pdf_{t_{i_m}})(x) \tag{4}$$

For specific cases, there exist simple analytical solutions of the convolution. E.g., 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 F of the PDFs:

$$f * g = F^{-1}(F(f)F(g)) \tag{5}$$

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.

Suppose the agents executing a plan $\mathcal{P}$ agree to start the execution at time t_s. With the PDF of the execution time of a plan $\mathcal{P}$ and the deadline given for the respective service request, it is then easy to determine the probability that the plan execution exceeds this deadline, which we call the probability of failure (PoF):

$$PoF(\mathcal{P}, t_s, d) = \int_{d-t_s}^\infty pdf_{t_{\mathcal{P}}}(x) dx \tag{6}$$

Note that for $d < t_s$, we always have $PoF(\mathcal{P}, t_s, d) = 0$, since the plan execution time must be positive. Similarly, the probability of success (PoS) is:

$$PoS(\mathcal{P}, t_s, d) = 1 - PoF(\mathcal{P}, t_s, d) \tag{7}$$

Given the membership degrees, the PDF over the upper bound $\hat{t}_i$ for the execution time of a service instance $i \in \mathcal{P}$ of agent spa_k is, analogous to 1,

$$pdf_{\hat{t}_i}(x) = pdf_{t_i^{min}}(mem_k * x) \tag{8}$$

According to 4, we can then obtain the PDF of the upper bound $\hat{t}_{\mathcal{P}}$ for the execution time of the complete plan, and thus the probabilities of failure and success of the fuzzy coalition, denoted $PoF(\widetilde{C})$ and $PoS(\widetilde{C})$, resp. This enables us to determine a lower bound for the expected reward for $\widetilde{C}$, denoted $\underline{r}_{\widetilde{C}}$:

$$\underline{r}_{\widetilde{C}} = PoS(\widetilde{C}) \times r \tag{9}$$

To specify a value for the fuzzy coalitions, we further have to consider the costs that are generated by the service executions. The agents should reasonably stop the execution once the deadline is reached, since no additional reward can be obtained by any further work. However, to simplify things, we consider only the worst case, i.e. the case where maximum costs have been produced even if the coalition fails.

Definition 4 *Value of a Fuzzy Service Provider Agent Coalition*
Let there be a fuzzy coalition $\widetilde{C}$ with plan $\mathcal{P}$. The value $v(\widetilde{C})$ *of $\widetilde{C}$, also called* coalition value*, is defined as*

$$v(\widetilde{C}) = \underline{r}_{\widetilde{C}} - \sum_{i \in \mathcal{P}} cost_i$$

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 have

Definition 5 *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 *$\mathcal{S}$ for the agents in SPA is defined as a set of fuzzy coalitions with*

$$\forall spa \in SPA : \sum_{\widetilde{C} \in \mathcal{S}} mem_{spa}^{\widetilde{C}} \leq 1 \tag{10}$$

4 Risk of Fuzzy Coalition Structures

Given a variety of combination of coalitions that the agent can possibly join, rational agents will prefer coalitions with a high reward and a low PoF, i.e. a high expected value. But assume there is a coalition with a high expected value, but which also involves very high costs. If an agent cannot afford to lose more than some amount without compromising liquidity, even a low PoF 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 a recent overview, see [7] and references therein).

For the definitions in the remainder of this section, we follow Artzner et al.[1], 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 VaR and other measures in [1] include the reward of a reference investment (e.g. interest rates) as a scaling factor, which we omit here for simplicity.

Definition 6 *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 [7], 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 and the capital should be used to control the risk.

Definition 7 *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. also introduce the notion of *coherent* risk measures.

Definition 8 *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 $r(Y) \leq r(X)$*

As has also been shown in [1], VaR is not coherent, since it does not fulfill subadditivity. As it turns out (see below), 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 for continous distributions.

Definition 9 *Tail Conditional Expectation: given a probability measure P on Ω and a level α, the tail conditional expectation is defined by:*

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

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 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. The following follows directly from the independency of the PoF of different coalitions and the definition of VaR.

Corollary 1 *Let there be a fuzzy coalition structure S and let $S_{spa_i} \subseteq S$ be the subset of all coalitions involving spa_i. For each $S^*_{spa_i} \in 2^{S_{spa_i}}$ (including the empty set) let $cost_i(S^*_{spa_i})$ be the cost for spa_i if all coalitions in $S^*_{spa_i}$ fail, and let $u_i(S^*_{spa_i})$ be the net payoff obtained by spa_i from the coalitions in $S_{spa_i} \cup S^*_{spa_i}$ (i.e. the reward minus costs for the successful coalitions).*

The probability $PoF(S^*_{spa_i})$ that the coalitions in $S^*_{spa_i}$ fail while those in $S_{spa_i} \cap S^*_{spa_i}$ succeed is

$$PoF(S^*_{spa_i}) = \prod_{\widetilde{C} \in S^*_{spa_i}} PoF(\widetilde{C}) \times \prod_{\widetilde{C} \in S_{spa_i} \cap S^*_{spa_i}} PoS(\widetilde{C})$$

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

$$VaR^{\alpha_i}(S) = -min_{S^*_{spa_i} \in 2^{S_{spa_i}}} \{u_i(S^*_{spa_i}) : \sum_{\substack{S'_{spa_i} \in 2^{S_{spa_i}} \\ u_i(S'_{spa_i}) \leq u_i(S^*_{spa_i})}} PoF(S^*_{spa_i}) > \alpha_i\}$$

Having $VaR^{\alpha_i}(S)$, the computation of the $TCE^{\alpha_i}(S)$ is straight-forward. Please note that $VaR^{\alpha_i}(S)$ and thus also TCE^{α_i} depend on the agent's payoff. But as becomes clear in section 5, 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.

5 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[2, 5, 9]. 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 (a) the execution time of a service instance is characterized by an $\frac{1}{x}$-relationship wrt. to the membership (see Definition 2.2(d)) and (b) 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 S with $C \notin S$ quantifies the difference in payoff that the agents in C obtain by forming C and leaving their resp. coalitions in S. 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 10 *Excess of a fuzzy coalition*
Let there be fuzzy coalition $\widetilde{C}$ and fuzzy coalition structures S and S' with $\widetilde{C} \in S'$, $\widetilde{C} \notin S$, S' is feasible, and $\forall \widetilde{C}' \in S', \widetilde{C}' \neq \widetilde{C} : \exists \widetilde{C}'' \in S : \widetilde{C}' \widetilde{\subseteq} \widetilde{C}''$. Further, let there be a payoff distribution u. We define

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

with

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

and

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

In crisp games, for a given configuration $(\mathcal{S}, u)$, the surplus of an agent a_i over another agent a_k with $a_i, a_k \in C \in \mathcal{S}$ 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 11 *Fuzzy coalition surplus*
Let there be a fuzzy coalition structure S and payoff distribution u and agents a_i and a_k.

1. *A feasible fuzzy coalition structure S' with $\forall \widetilde{C} \in S', \widetilde{C} \notin S : a_i \in \widetilde{C}, a_k \notin \widetilde{C}$, $\forall C \in S, a_k \notin C : C \in S'$ and $\nexists \widetilde{C} \in S' : a_i, a_k \in \widetilde{C}$ is called an* ik-fuzzy surplus structure.
2. *The set of all ik-fuzzy surplus structures wrt. S is denoted $SS_{ik}(S)$*
3. *The* fuzzy coalition surplus *of a_i over a_k is*

$$\widetilde{s}_{ik|TCE} := \max_{S' \in SS_{ik}(S)} \{ \sum_{a_i \in \widetilde{C} \in S'} \widetilde{e}(\widetilde{C}, \widetilde{u})_{|TCE} \}$$

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 12 *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 S, let $SS_{ik}(Q_{ik})$ denote the set of all ik-fuzzy surplus structures S' wrt. S such that for all pairs $(sra, \mathcal{P}) \in Q_{ik}$ there exists $\widetilde{C} \in \widetilde{C}(sra, \mathcal{P})$ with $\widetilde{C} \in S'$. We define the function $MaxS(Q_{ik}, S, u)$ to return $S^* \in SS_{ik}(Q_{ik})$ such that $\sum_{a_i \in \widetilde{C} \in S^*} \widetilde{e}(\widetilde{C}, \widetilde{u})_{|TCE}$ is maximized wrt. all other elements in $SS_{ik}(Q_{ik})$.*

Because the service instance runtime depends on the spent resources and thus the membership values by a $\frac{1}{x}$-relationship (see Definition 2.2(d)), $MaxS$ 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 size[10]. 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 2 *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 $\widetilde{s}_{ik|TCE}$. 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 $n^{(aMax \times \mathcal{P}Max)^{\widetilde{C}Max}}$.*

Proof. It was shown in [10] 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 [10], the number of sets of these coalitions with maximum size $\widetilde{C}Max$ is then bounded by $n^{(aMax \times \mathcal{P}Max)^{\widetilde{C}Max}}$.

In crisp games, the *kernel* of a cooperative game $(\mathcal{A}, v)$ with respect to a given coalition structure $\mathcal{S}$ is a set of configurations $(\mathcal{S}, u)$ wherein each pair of agents a_i, a_k in each coalition $C \in \mathcal{S}$ is in equilibrium wrt. their surplusses. That is the case if the agents cannot outweigh each other in $(\mathcal{S}, u)$ by having the option to get a better payoff in coalition(s) *not* in $\mathcal{S}$ excluding the opponent agent (agent i outweighs k, if $s_{ik} > s_{ki}$ and $u_k > w_i(C)$). 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 13 *Let there be a fuzzy coalition structure S and payoff distribution u. (S, u) is in the* kernel *of the fuzzy coalition game iff each pair of agents a_i, a_k in each fuzzy coalition $C \in \mathcal{S}$ is in equilibrium wrt. their fuzzy coalition surplusses.*

To make a payoff distribution kernel-stable for a given coalition structure, Stearns *transfer scheme* 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.

6 Coalition Formation Protocol RFCF

In this section, we propose a fuzzy coalition formation protocol that guarantees to form coalitions which are in compliance with 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** - Composition plans are generated. 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.
- $findFuzzyCoalition(S, \mathcal{P}, risk)$: Computes a fuzzy coalition $\widetilde{C}$ such that the membership degrees in $\widetilde{C}$ are minimized while $S \cup \widetilde{C}$ 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 $TCE^{\alpha_a}(S \cup \widetilde{C}(sra(\mathcal{P}), \mathcal{P}))$, 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
- $makeStable(S)$: Computes a new stable payoff distribution u^* for the fuzzy coalition structure S using the transfer scheme (see 5) and the bounds $\mathcal{P}Max$, $aMax$ and $\widetilde{C}Max$.

Algorithm 1 *RFCF*

Each agent a performs:

```
Initialization:

  1. $setPLANS := \emptyset$
  2. $setPPPLANS := \emptyset$
  3. $setPPPLANSRISK := \emptyset$
  4. $setPROPS :=$ new priority queue
  5. $setrisk_a := TCE(\{a\}/1)$

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. $\widetilde{C} := findFuzzyCoalition(S, \mathcal{P}, nil)$
               ii. if $\widetilde{C} = nil$ then $PLANS := PLANS \setminus \mathcal{P}$;
                   $POSTPONEDPLANS := \cup\mathcal{P}$ ;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 $POSTPONEDPLANS$ do:
                i. if $PPPLANSRISK$ contains $(\mathcal{P}, .)$ then
                   $\widetilde{C} := findFuzzyCoalition(S, \mathcal{P}, PPPLANSRISK(\mathcal{P}))$
               ii. if $\widetilde{C} = nil$ then next 1b
              iii. if $v(\widetilde{C})/|\widetilde{C}| > BestPayoffperMembership$ then
                   $PPPLANSRISK := PPPLANSRISK \setminus \mathcal{P}$;
                   $BestCoalition := \widetilde{C}$; $BestPayoffperMembership := |\widetilde{C}|$
      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 $S^* = \emptyset$
          (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 $S^* := S^* \cup \widetilde{C}$
      4. Payoff distribution and TCE update
          (a) set $u^* = makeStable(S \cup S^*)$
          (b) do atomically: set $S := S \cup S^*$ and $u := u^*$
          (c) set $risk_a := risk_a + \sum_{\widetilde{C} \in S^*_a}(TCE_a(\widetilde{C}))$
```

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

Proposition 3 *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). All other steps in the coalition negotiation section are of less complexity.

7 Related Work

In the research field of fuzzy coalition formation, Nishizaki and Sakawa in [9] 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. Also, as we have pointed out in section 1, they assume that the coalition value is a proportional function of the agents' membership degrees, which does not hold in our case.

Shehory and Kraus 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.

There also exist approaches for the formation of non-overlapping coalitions which take uncertainty in the coalition values into account. These are also suitable to tackle the problem of reduced coalition values due to (partial) coalition failure in some cases. Probabilistic approaches, such as [6], usually consider the expected values of coalitions. This might lead to the case that a number of risk-neutral agents decide to form a high-risk coalition, excluding risk-averse agents to cooperate with them because overlapping coalitions are not allowed. In contrast, our approach allows for such cooperations by forming additional coalitions. Approaches that employ fuzzy coalition values, such as [3], account for a range of possible coalition values. However, the fuzzy coalition values are assumed to actually be fuzzy numbers or intervals. But this assumption is not compatible with our setting where a coalition value either produces a specific profit or a specific loss.

8 Conclusions

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.

References

1. P. Artzner, F. Delbaen, S. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, pages 203–228, 1999.
2. J.-P. Aubin. *Mathematical Methods of Game and Economic Theory*. North-Holland, 1979.
3. B. Blankenburg, M. Klusch, and O. Shehory. Fuzzy kernel-stable coalitions between rational agents. In *Proc. 2nd Int. Conference on Autonomous Agents and Multiagent Systems, Melbourne, Australia*, 2003.
4. R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Science/Engineering/Math, New York, 3rd edition, 1999.
5. D. Butnariu. Stability and shapley value for an n-persons fuzzy game. *Fuzzy Sets and Systems*, 4:63–72, 1980.
6. G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proc. 3rd Int. Conference on Autonomous Agents and Multiagent Systems, New York, USA*, 2004.
7. S. Cheng, Y. Liu, and S. Wang. Progress in risk management. *Advanced Modelling and Optimization*, 6(1):1–20, 2004.
8. G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.
9. 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.
10. O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents. *Computational Intelligence*, 15(3):218–251, 1999.
11. R. E. Stearns. Convergent transfer schemes for n-person games. *Transactions of the American Mathematical Society*, 134:449–459, 1968.