

Article

Contextual and Possibilistic Reasoning for Coalition Formation

Antonios Bikakis ^{1,*}  and Patrice Caire ^{2,†}

¹ Department of Information Studies, University College London, London WC1E 6BT, UK

² Computer Science Department, New York University, New York, NY 10012, USA; patrice.caire@nyu.edu

* Correspondence: a.bikakis@ucl.ac.uk

† These authors contributed equally to this work.

Received: 18 June 2020; Accepted: 14 September 2020; Published: 19 September 2020



Abstract: In multi-agent systems, agents often need to cooperate and form coalitions to fulfil their goals, for example by carrying out certain actions together or by sharing their resources. In such situations, some questions that may arise are: Which agent(s) to cooperate with? What are the potential coalitions in which agents can achieve their goals? As the number of possibilities is potentially quite large, how to automate the process? And then, how to select the most appropriate coalition, taking into account the uncertainty in the agents' abilities to carry out certain tasks? In this article, we address the question of how to identify and evaluate the potential agent coalitions, while taking into consideration the uncertainty around the agents' actions. Our methodology is the following: We model multi-agent systems as Multi-Context Systems, by representing agents as contexts and the dependencies among agents as bridge rules. Using methods and tools for contextual reasoning, we compute all possible coalitions with which the agents can fulfil their goals. Finally, we evaluate the coalitions using appropriate metrics, each corresponding to a different requirement. To demonstrate our approach, we use an example from robotics.

Keywords: multi-agent systems; coalition formation; Multi-Context Systems; contextual reasoning; possibilistic reasoning; reasoning under uncertainty

1. Introduction

In multi-agent systems, agents often need to cooperate to fulfil their individual or common goals. For example, an agent may not be able to perform a task on its own or may lack a resource that is required for a task, which can be provided by another agent. Moreover, in most real-world settings, we cannot always be certain that agents will carry out their assigned tasks successfully.

Consider, for example, an environment where a group of robots with different capabilities are assigned a set of tasks, e.g., to move a set of objects between different locations. Assuming that only some of the robots know the exact location of each object, the robots need to cooperate by sharing information about the location of the objects to carry out the assigned tasks. If we also assume that more than one robot can carry each object, there is more than one way to assign the tasks to the robots, each involving cooperation between different robots. Finally, assuming that something may go wrong while the robots carry out their assigned tasks, for example a robot may fall on its way to pick up an object or run out of battery, the accomplishment of the tasks is uncertain, and the level of the uncertainty depends on the characteristics of the robots and how the tasks are assigned to them.

In such situations, agents need to decide which other agents to cooperate with. The problem of forming teams of agents, so that they can fulfil their individual or common goals is known as coalition formation. Several studies have proposed solutions to this problem using methods and techniques mostly from multi-agent systems, game theory, evolutionary computation and machine

learning (e.g., see [1–11] for the most recent approaches and Section 6 for a more comprehensive list). We propose a novel approach to the problem, which draws on methods and tools for contextual reasoning and Multi-Context Systems.

Multi-Context Systems (MCS) [12–14] is a logic-based formalization of distributed knowledge bases (*contexts*) interlinked via a set of *bridge rules*, which enable information flow between the contexts. It is among the formal models of context, which have been applied to several application domains, such as common sense knowledge bases [15], ontology languages [16,17], agent architectures [18,19], multi-agent negotiation [20], business process modeling [21], stream reasoning [22,23] and contextual reasoning in mobile networks [24] and ambient intelligence systems [25].

In this work, we use two variants of MCS. The first one is *non-monotonic MCS* [14], the main characteristics of which are that each context may use a different formalism to represent knowledge, and that bridge rules have the form of non-monotonic logic programming rules, which combine elements from different contexts in their body. The second variant is *possibilistic MCS* [26], an extension of non-monotonic MCS that explicitly models uncertainty in the contexts and the bridge rules. Four advantages of our proposed approach are: (a) It is able to handle agents that use different knowledge representation formalisms; (b) the format of the bridge rules allows modeling different types of relationships between agents such as inter-dependencies, conflicting goals and constraints; (c) the possibilistic extension of MCS enables modeling uncertainty in the agents' actions; (d) MCS is a well-studied model and there are both centralized and distributed reasoning algorithms and tools that can be used to reason with it.

Our main research question is:

Q *How to identify and evaluate agent coalitions while taking into consideration the uncertainty around the agents' actions?*

This breaks down into the following three sub-questions:

1. *How to identify all possible coalitions that agents can form to fulfil their goals?*
2. *How to evaluate the coalitions given a set of requirements?*
3. *How to compute and evaluate coalitions taking also into account the uncertainty in the agents' actions?*

To address these questions, we adopt the following methodology. We use the formalism proposed in [27] to represent the dependencies among agents. We then model multi-agent systems as non-monotonic MCS, i.e., we model agents as contexts, and their dependencies as bridge rules. We then compute the possible coalitions using existing algorithms for MCS equilibria. Finally, given a set of domain-specific or more general requirements such as efficiency, which in the robotics example described above can be associated with the total distance that the robots need to cover to carry out the assigned tasks, or conviviality, which is a measure of the cooperation among agents (roughly, more opportunities to work with others increases the conviviality of the system), we select and use appropriate metrics to evaluate the coalitions. We then extend our approach with features of possibilistic reasoning: We extend the definition of dependence relations with a certainty degree; we then use the model and algorithms of possibilistic MCS to compute the potential coalitions under uncertainty. In this case, we evaluate the different coalitions based on the certainty degree with which each coalition achieves the different goals, using multiple-criteria decision-making methods.

This article is an extended version of [28], where we presented our methodology for formalizing multi-agent systems and computing coalitions among agents under the perfect world assumption, i.e., actions are always carried out with success by the agents that they have been assigned to. Here, we provide more details about the computation and evaluation of coalitions in the perfect world case. We also present new results for the cases that the perfect world assumption does not hold due to uncertainty around the agents' actions.

The rest of the paper is structured as follows. Section 2 provides the necessary background information on dependence networks, coalition formation, non-monotonic MCS and possibilistic

MCS. Section 3 presents our example from robotics in more detail. Section 4 presents our approach in a setting without uncertainty: how we model multi-agent systems as MCS; and how we compute and evaluate all possible agent coalitions. Section 5 presents the possibilistic reasoning approach, which takes into account the uncertainty in the agents' actions. Section 6 discusses related work and Section 7 summarizes and presents future directions of this work.

2. Background

2.1. Dependence Networks and Coalition Formation

To model the dependencies among agents, we use dependence networks. As shown in [29], this model can represent different kinds of relationships among agents, including inter-dependencies among their goals and actions. A dependence network consists of a finite set of actors (people or organizations) and a set of relations among them. It can naturally be represented as a directed graph with nodes representing the agents and their actions, and directed edges linking agents and actions, representing the agents' goals and the ways that they can achieve them. Dependence networks have been applied to several domains including smart environments [30] and cyber-physical Systems [31].

Agents often form groups or coalitions to achieve their goals. As motivation for participating in a coalition, each agent receives some payoff. When participating in a coalition, the agents adopt the goals of the coalition, and in order to achieve their individual as well as their reciprocal goals, they cooperate and coordinate their actions and behaviors (see for example [29,32]). Coalitions can be represented as topological aspects of a dependence network.

2.2. Multi-Context Systems

Multi-Context Systems (MCS) [12–14] is a logic-based formalization of distributed knowledge bases, and has been one of the main efforts to formalize context and contextual reasoning in Artificial Intelligence. Here, we adopt the definition of non-monotonic MCS [14], the main characteristics of which are that each context may use a different knowledge representation formalism, and contexts may exchange information via the so-called bridge rules.

2.2.1. Formalization

According to the definition given in [14], a MCS is a set of contexts, each consisting of a knowledge base with an underlying logic, and a set of bridge rules. A logic $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$ consists of the following components:

- \mathbf{KB}_L is the set of well-formed knowledge bases of L . Each element of \mathbf{KB}_L is a set of formulae.
- \mathbf{BS}_L is the set of possible belief sets, where the elements of a belief set are a set of formulae.
- $\mathbf{ACC}_L: \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ is a function describing the semantics of the logic by assigning to each knowledge base a set of acceptable belief sets.

This definition is broad enough to capture the semantics of several different monotonic and non-monotonic logics such as propositional logic, description logics, modal logics, default logic, circumscription, defeasible logic and logic programs under the answer set semantics [14]. This feature (the generality of the representation model) is particularly important in open environments, where agents are typically heterogeneous with respect to their representation and reasoning capabilities (e.g., Ambient Intelligence systems).

A bridge rule has the form of a logic programming rule with default negation. The atoms in the body of the rule may refer to elements of different contexts. The aim of a bridge rule is to add information to the knowledge base of a context taking into account what is believed or what is not believed in other contexts. Let $\mathcal{L} = (L_1, \dots, L_n)$ be a sequence of logics. An L_k -bridge rule r over \mathcal{L} , $1 \leq k \leq n$, is of the form

$$r = (k : s) \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \text{not}(c_{j+1} : p_{j+1}), \dots, \text{not}(c_m : p_m). \quad (1)$$

where c_i , $1 \leq i \leq n$, denotes a context, p_i is an element of some belief set of L_i , and k refers to the context receiving information s . We write $h_b(r)$ to denote the belief formula s in the head of r .

A MCS $M = (c_1, \dots, c_n)$ is a set of contexts $c_i = (L_i, kb_i, br_i)$, $1 \leq i \leq n$, where $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ is a logic, $kb_i \in \mathbf{KB}_i$ a knowledge base, and br_i a set of L_i -bridge rules over (L_1, \dots, L_n) . For each $H \subseteq \{h_b(r) \mid r \in br_i\}$ it holds that $kb_i \cup H \in \mathbf{KB}_i$, meaning that the head of each bridge rule used to import information to context c_i , must be compatible with the knowledge base of c_i .

A belief state of a MCS is the set of the belief sets of its contexts. Formally, a *belief state* of $M = (c_1, \dots, c_n)$ is a sequence $\mathcal{S} = (S_1, \dots, S_n)$ such that $S_i \in \mathbf{BS}_i$. Intuitively, \mathcal{S} is logically derived from the knowledge bases of the contexts and the information that is imported to the contexts using the bridge rules that can be applied. A bridge rule of form (3) is applicable in a belief state \mathcal{S} iff for $1 \leq i \leq j$: $p_i \in S_i$ and for $j < l \leq m$: $p_l \notin S_l$.

Equilibrium semantics selects certain belief states of a MCS as acceptable. Intuitively, an equilibrium is a belief state $\mathcal{S} = (S_1, \dots, S_n)$ where each context c_i respects all bridge rules applicable in \mathcal{S} and accepts S_i . Formally, $\mathcal{S} = (S_1, \dots, S_n)$ is an equilibrium of M , iff for $1 \leq i \leq n$,

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) \mid r \in br_i \text{ applicable in } \mathcal{S}\}).$$

\mathcal{S} is a grounded equilibrium of M iff for $1 \leq i \leq n$, S_i is an answer set of logic program $P = kb_i \cup \{h_b(r) \mid r \in br_i \text{ applicable in } \mathcal{S}\}$. For a definite MCS (MCS without default negation in the bridge rules), its unique grounded equilibrium is the collection consisting of the least (with respect to set inclusion) Herbrand model of each context.

A characteristic of MCS is that even if each context is logically consistent, the information flow between the contexts via the bridge rules, may result in inconsistencies. In those cases, the MCS does not have an equilibrium. To resolve such inconsistencies, several different methods have been proposed, most of which are based on the following idea: to restore consistency, invalidate some of the bridge rules and/or apply unconditionally some others (see, for example, [33,34]). In [35] we proposed a different method for conflict resolution based on a property of multi-agent systems called *conviviality*.

Example 1. Consider a scholar social network through which software agents, acting on behalf of students or researchers, share information about research articles they find online. Consider three such agents, each one with its own knowledge base and logic exchanging information about a certain article. The three agents can be represented as contexts $c_1 - c_3$ in a MCS $M = \{c_1, c_2, c_3\}$. The knowledge bases of the three contexts are respectively:

$$kb_1 = \left\{ \begin{array}{l} \text{sensors} \leftarrow, \text{corba} \leftarrow, \\ \text{distributedComputing} \leftarrow \text{corba}, \text{not centralizedComputing} \end{array} \right\}$$

$$kb_2 = \{\text{prof}A\}$$

$$kb_3 = \{\text{ubiquitousComputing} \subseteq \text{ambientComputing}\}$$

kb_1 is a logic program stating that the article is about sensors and corba, and that articles about corba that are not classified in centralizedComputing can be classified in distributedComputing. kb_2 uses propositional logic to express its belief that profA has written the article. kb_3 is a description logic ontology, which includes the belief that ubiquitousComputing is a type of ambientComputing. The three agents share their beliefs about articles using bridge rules r_1-r_4 .

$$\begin{aligned}
r_1 &= (c_1 : \text{centralizedComputing}) \leftarrow (c_2 : \text{middleware}) \\
r_2 &= (c_1 : \text{distributedComputing}) \leftarrow (c_3 : \text{ambientComputing}) \\
r_3 &= (c_2 : \text{middleware}) \leftarrow (c_1 : \text{corba}) \\
r_4 &= (c_3 : \text{ubiquitousComputing}) \leftarrow (c_1 : \text{sensors}), (c_2 : \text{profB})
\end{aligned}$$

with r_1 and r_2 , the first agent classifies articles about middleware (as described in c_2) in the category of *centralizedComputing*, and articles about *ambientComputing* (as described in c_3) in *distributedComputing*. With r_3 , the second agent classifies articles about *corba* in *middleware*. Finally, with r_4 , the third agent classifies articles about *sensors*, which have been written by *profB*, in *ubiquitousComputing*. M has one equilibrium:

$$\mathcal{S} = \left(\begin{array}{l} \{\text{sensors}, \text{corba}, \text{centralizedComputing}\}, \\ \{\text{profA}, \text{middleware}\}, \emptyset \end{array} \right)$$

according to which, the first agent classifies the paper in *centralizedComputing*, and the second agent classifies it in *middleware*.

Consider now the case that *profB* is identified by c_2 as a second author of the paper:

$$kb_2 = \{\text{profA}, \text{profB}\}$$

Rules r_4 and r_2 would then become applicable, and as a result M would not have an equilibrium; it would therefore be inconsistent. To resolve the conflict, one of the four bridge rules r_1 - r_4 would have to be invalidated. For example, by invalidating rule r_1 , the system would have one equilibrium:

$$\mathcal{S}_1 = \left(\begin{array}{l} \{\text{sensors}, \text{corba}, \text{distributedComputing}\}, \\ \{\text{profA}, \text{profB}, \text{middleware}\}, \\ \{\text{ubiquitousComputing}, \text{ambientComputing}\} \end{array} \right)$$

2.2.2. Computational Complexity

An analysis of the computational complexity of MCS with logics that have poly-size kernels is presented in [14]. A logic L has poly-size kernels, if there is a mapping κ , which assigns to every $kb \in \mathbf{KB}$ and $S \in \mathbf{ACC}(kb)$ a set $\kappa(kb, S) \subseteq S$ of size (written as a string) polynomial in the size of kb , called the *kernel of S* , such that there is a one-to-one correspondence f between the belief sets in $\mathbf{ACC}(kb)$ and their kernels, i.e., $S = f(\kappa(kb, S))$. Examples of logics with poly-size kernels include propositional logic, default logic, auto-epistemic logic and non-monotonic logic programs. If furthermore, given any knowledge base kb , an element b , and a set of elements K , deciding whether (i) $K = \kappa(kb, S)$ for some $S \in \mathbf{ACC}(kb)$ and (ii) $b \in S$ is in Δ_k^p , then we say that L has kernel reasoning in Δ_k^p . For example, default logic and auto-epistemic logic have kernel reasoning in Δ_2^p .

According to their analysis, in a finite MCS M , i.e., a MCS with finite knowledge bases and bridge rules, and with logics from an arbitrary but fixed set, where all logics L_i have poly-size kernels and kernel reasoning in Δ_k^p , deciding whether a literal p is in a belief set S_i for some (or each) equilibrium of M is in Σ_{k+1}^p (respectively, $\Pi_{k+1}^p = co - \Sigma_{k+1}^p$).

2.3. Possibilistic Reasoning in MCS

Recently, Jin et al. proposed a framework for possibilistic reasoning in Multi-Context Systems, which they called *possibilistic MCS* [26]. This has been so far the only attempt to model uncertainty in MCS. It is based on *possibilistic logic* [36] and *possibilistic logic programs* [37], which are logic-based frameworks for representing states of partial ignorance using a dual pair of possibility and necessity measures. These frameworks are in turn based on ideas from Zadeh's possibility theory [38].

Below, we first provide some preliminary information on possibilistic logic programs, which will then help us to present possibilistic MCS.

2.3.1. Possibilistic Logic Programs

Possibilistic logic programs [37] use the notion of *possibilistic concept*, which is denoted by \bar{X} , where X denotes its classical counterpart. For example, in possibilistic logic programs, this notion is used in the definitions of *possibilistic atoms* and *poss-programs*:

Definition 1. Let Σ be a finite set of atoms. A *possibilistic atom* is $\bar{p} = (p, [\alpha])$, where $p \in \Sigma$ and $\alpha \in [0, 1]$.

The classical projection of \bar{p} is the atom p and $n(\bar{p}) = \alpha$ is called the *necessity degree* of \bar{p} .

Definition 2. A *possibilistic normal logic program* (or *poss-program*) \bar{p} is a set of *possibilistic rules* of the form:

$$\bar{r} = s \leftarrow p_1, \dots, p_m, \text{not } q_1, \dots, \text{not } q_n, [\alpha]. \quad (2)$$

where $m, n \geq 0$, $\{p_1, \dots, p_m, q_1, \dots, q_n, s\} \subseteq \Sigma$, and $n(\bar{r}) = \alpha \in [0, 1]$.

In (2), α represents the certainty level of the information described by rule \bar{r} . The head of \bar{r} is defined as $\text{head}(\bar{r}) = s$ and its body as $\text{body}(\bar{r}) = \text{body}^+(\bar{r}) \cup \text{not } \text{body}^-(\bar{r})$, where $\text{body}^+(\bar{r}) = \{p_1, \dots, p_m\}$ and $\text{body}^-(\bar{r}) = \{q_1, \dots, q_n\}$. The positive projection of \bar{r} is

$$\bar{r}^+ = \text{head}(\bar{r}) \leftarrow \text{body}^+(\bar{r}), [\alpha] \quad (3)$$

The classical projection of \bar{r} is the classical rule:

$$r = s \leftarrow p_1, \dots, p_m, \text{not } q_1, \dots, \text{not } q_n \quad (4)$$

If a poss-program \bar{p} does not contain any default negation, then \bar{p} is called a *definite poss-program*. The reduct of a poss-program \bar{p} regarding a set of atoms T is the definite poss-program defined as:

$$\bar{p}^T = \{\bar{r}^+ \mid \bar{r} \in \bar{p}, \text{body}^-(\bar{r}) \cap T = \emptyset\} \quad (5)$$

For a set of atoms $T \subseteq \Sigma$ and a rule $\bar{r} \in \bar{p}$, we say that \bar{r} is *applicable* in T if $\text{body}^+(\bar{r}) \subseteq T$ and $\text{body}^-(\bar{r}) \cap T = \emptyset$. $\text{App}(\bar{p}, T)$ denotes the set of rules in \bar{p} that are applicable in T .

P is said to be *grounded* if it can be ordered as a sequence $\langle \bar{r}_1, \dots, \bar{r}_n \rangle$ such that

$$\forall i, 1 \leq i \leq n, \bar{r}_i \in \text{App}(\bar{p}, \text{head}(\{\bar{r}_1, \dots, \bar{r}_{i-1}\})) \quad (6)$$

Given a poss-program \bar{p} over a set of atoms Σ , the semantics of \bar{p} is defined through possibility distributions on Σ (For more details about the semantics of poss-programs, see [37]).

2.3.2. Possibilistic MCS

A *possibilistic MCS* (or *poss-MCS*) [26] is a collection of possibilistic contexts. A *possibilistic context* \bar{c} is a triple $(\Sigma, \bar{P}, \bar{B})$ where Σ is a set of atoms, \bar{P} is a poss-program, and \bar{B} is a set of possibilistic bridge rules. A *possibilistic bridge rule* is defined as follows:

Definition 3. Let $\bar{k}, \bar{c}_1, \dots, \bar{c}_n$ be possibilistic contexts. A *possibilistic bridge rule* $\bar{p}\bar{r}$ for context \bar{k} is of the form

$$\begin{aligned} \bar{p}\bar{r} = & (\bar{k} : s) \leftarrow (\bar{c}_1 : p_1), \dots, (\bar{c}_j : p_j), \\ & \text{not}(\bar{c}_{j+1} : p_{j+1}), \dots, \text{not}(\bar{c}_m : p_m), [\alpha] \end{aligned} \quad (7)$$

where s is an atom in \bar{k} and each p_i is an atom in context \bar{c}_i . $1 \leq i \leq n$.

Intuitively, a rule of form (7) states that information s is added to context \bar{k} with necessity degree α if, for $1 \geq i \geq j$, p_i is provable in context \bar{c}_i and for $j+1 \geq l \geq n$, p_l is not provable in \bar{c}_l .

By pr (see Equation (3)) we denote the classical projection of \bar{pr} . The necessity degree of \bar{pr} is denoted by $n(\bar{pr})$.

Definition 4. A *possibilistic Multi-Context System*, or just *poss-MCS*, $\bar{M} = (\bar{c}_1, \dots, \bar{c}_n)$ is a collection of *possibilistic contexts* $\bar{c}_i = (\Sigma_i, \bar{P}_i, \bar{B}_i)$, $1 \geq i \geq n$, where each Σ_i is the set of atoms used in context \bar{c}_i , \bar{P}_i is a *poss-program* on Σ_i and \bar{B}_i is a set of *possibilistic bridge rules* over atom sets $(\Sigma_1, \dots, \Sigma_n)$.

A *poss-MCS* is *definite* if the *poss-program* and *possibilistic bridge rules* of each context are *definite*.

Definition 5. A *possibilistic belief state*, $\bar{S} = (\bar{S}_1, \dots, \bar{S}_n)$ is a collection of *possibilistic atom sets* \bar{S}_i , where each \bar{S}_i is a collection of *possibilistic atoms* \bar{p}_i and $p_i \in \Sigma_i$.

We will now describe the semantics for *poss-MCS*, starting with *definite poss-MCS*. The following definition specifies the possibility distribution of belief states for a given *definite poss-MCS*. It uses the notion of *satisfiability* of a rule r , which is based on its applicability regarding a belief state \mathcal{S} :

$$\mathcal{S} \not\models r \text{ iff } \text{body}^+(r) \subseteq \mathcal{S} \text{ and } \text{head}(r) \notin \mathcal{S} \quad (8)$$

Definition 6. Let $\bar{M} = (\bar{c}_1, \dots, \bar{c}_n)$ be a *definite poss-MCS* and $\bar{S} = (\bar{S}_1, \dots, \bar{S}_n)$ a *belief state*. The *possibility distribution* $\pi_{\bar{M}} : 2^\Sigma \rightarrow [0, 1]$ for \bar{M} is defined as:

$$\pi_{\bar{M}}(\mathcal{S}) = \begin{cases} 0, & \text{if } \mathcal{S} \not\subseteq \text{head}(\bigcup_i \text{App}_i(M, \mathcal{S})) \\ 0, & \text{if } \bigcup_i \text{App}_i(M, \mathcal{S}) \text{ is not grounded} \\ 1, & \text{if } \mathcal{S} \text{ is an equilibrium of } M \\ 1 - \max\{n(\bar{r}) \mid \mathcal{S} \not\models \bar{r}, \bar{r} \in \bar{B}_i \cup \bar{L}_i\}, & \text{otherwise} \end{cases} \quad (9)$$

The possibility distribution specifies the degree of compatibility of each belief set \mathcal{S} with the *poss-MCS* \bar{M} . Based on Definition 6 we can now define the possibility and necessity of an atom in a belief state \mathcal{S} .

Definition 7. Let \bar{M} be a *definite poss-MCS* and $\pi_{\bar{M}}$ be the *possibilistic distribution* for \bar{M} . The *possibility and necessity of an atom* p_i in a belief state \mathcal{S} are respectively defined as:

$$\Pi_{\bar{M}}(p_i) = \max\{\pi_{\bar{M}}(\mathcal{S}) \mid p_i \in \mathcal{S}_i\} \quad (10)$$

$$N_{\bar{M}}(p_i) = 1 - \max\{\pi_{\bar{M}}(\mathcal{S}) \mid p_i \notin \mathcal{S}_i\} \quad (11)$$

$\Pi_{\bar{M}}(p_i)$ represents the level of consistency of p_i regarding the *poss-MCS* \bar{M} , while $N_{\bar{M}}(p_i)$ represents the level at which p_i can be inferred from M . For example, whenever an atom p_i belongs to the equilibrium of M (the classical projection of \bar{M}), its possibility is equal to 1.

The semantics for *definite poss-MCS* is determined by its unique *possibilistic grounded equilibrium*.

Definition 8. Let \bar{M} be a *definite poss-MCS*. Then the following set of *possibilistic atoms* is referred to as the *possibilistic grounded equilibrium*:

$$\overline{MD}(\bar{M}) = \{\bar{S}_1, \dots, \bar{S}_n\} \quad (12)$$

where $\bar{S}_i = \{(p_i, N_{\bar{M}}(p_i)) \mid p_i \in \Sigma_i, N_{\bar{M}}(p_i) > 0\}$ for $i = 1, \dots, n$.

As proved in [26] (Proposition 5), the classical projection of $\overline{MD}(\bar{M})$ is the grounded equilibrium of M , where M is the classical projection of \bar{M} .

The definition of the semantics for normal poss-MCS is based on the notion of reduct for normal poss-MCS, which is in turn based on the definition of rule reduct (see Equation (5)):

Definition 9. Let $\overline{M} = (\overline{c}_1, \dots, \overline{c}_n)$ be a normal poss-MCS and $\mathcal{S} = (S_1, \dots, S_n)$ a belief state. The possibilistic reduct of \overline{M} regarding \mathcal{S} is the poss-MCS

$$\overline{M}^{\mathcal{S}} = (\overline{c}_1^{\mathcal{S}}, \dots, \overline{c}_n^{\mathcal{S}}) \quad (13)$$

where $\overline{c}_i^{\mathcal{S}} = (\Sigma_i, \overline{P}_i^{\mathcal{S}}, \overline{B}_i^{\mathcal{S}})$.

Please note that the reduct of \overline{P}_i relies only on S_i while the reduct of \overline{B}_i depends on the whole belief state \mathcal{S} .

Given the notion of reduct for normal poss-MCS, the equilibrium semantics of normal poss-MCS is defined as follows:

Definition 10. Let \overline{M} be a normal poss-MCS and $\overline{\mathcal{S}}$ a possibilistic belief state. $\overline{\mathcal{S}}$ is a possibilistic equilibrium of \overline{M} if $\overline{\mathcal{S}} = \overline{MD}(\overline{M}^{\mathcal{S}})$.

Jin et al. [26] present also a fixpoint theory for definite poss-MCS, which provides a way for computing the equilibrium for both definite and normal poss-MCS.

Example 2. In a different version of Example 1 all agents use possibilistic logic programs to encode their knowledge and bridge rules, forming a possibilistic MCS \overline{M} . The three agents are modeled as contexts \overline{c}_1 , \overline{c}_2 and \overline{c}_3 , respectively, with knowledge bases:

$$\begin{aligned} \overline{P}_1 &= \left\{ \begin{array}{l} \text{sensors} \leftarrow, [1], \text{ corba} \leftarrow, [1], \\ \text{distributedComputing} \leftarrow \text{ corba}, \text{ not centralizedComputing}, [0.8] \end{array} \right\} \\ \overline{P}_2 &= \{\text{prof A}, [1]\} \\ \overline{P}_3 &= \{\text{ambientComputing} \leftarrow \text{ ubiquitousComputing}, [0.9]\} \end{aligned}$$

and bridge rules:

$$\begin{aligned} \overline{B}_1 &= \left\{ \begin{array}{l} (c_1 : \text{centralizedComputing}) \leftarrow (c_2 : \text{middleware}), [0.7], \\ (c_1 : \text{distributedComputing}) \leftarrow (c_3 : \text{ambientComputing}), [0.6] \end{array} \right\} \\ \overline{B}_2 &= \{(c_2 : \text{middleware}) \leftarrow (c_1 : \text{corba}), [0.9]\} \\ \overline{B}_3 &= \{(c_3 : \text{ubiquitousComputing}) \leftarrow (c_1 : \text{sensors}), (c_2 : \text{prof B}), [0.8]\} \end{aligned}$$

Rules or facts with degree 1 indicate that the agent is certain about them, while rules with degree less than 1 indicate uncertainty about whether the rule holds.

\overline{M} is a normal poss-MCS. To compute its possibilistic equilibrium, we first have to compute its reduct with respect to \mathcal{S} , where \mathcal{S} is the grounded equilibrium of \overline{M} (the classical projection of \overline{M}):

$$\mathcal{S} = \left(\begin{array}{l} \{\text{sensors}, \text{corba}, \text{centralizedComputing}\}, \\ \{\text{prof A}, \text{middleware}\}, \emptyset \end{array} \right)$$

The reduct of \overline{M} with respect to \mathcal{S} , $\overline{M}^{\mathcal{S}}$, is derived from \overline{M} by replacing \overline{P}_1 with $\overline{P}_1^{\mathcal{S}}$:

$$\overline{P}_1^{\mathcal{S}} = \{\text{sensors}, [1], \text{ corba}, [1], \text{ distributedComputing} \leftarrow \text{ corba}, [0.8]\}$$

The next step is to compute the necessity of each atom in \mathcal{S} . Following Definition 6, $\pi_{\overline{M}}(\mathcal{S}) = 1$, as \mathcal{S} is the grounded equilibrium of M . For

$$\mathcal{S}' = (\{sensors, corba\}, \{profA, middleware\}, \emptyset)$$

it holds that $\pi_{\overline{M}}(\mathcal{S}') = 1 - \max\{0.8, 0.7\} = 0.2$, while for

$$\mathcal{S}'' = (\{sensors, corba\}, \{profA\}, \emptyset)$$

it holds that $\pi_{\overline{M}}(\mathcal{S}'') = 1 - \max\{0.8, 0.7, 0.9\} = 0.1$. Using Definition 7, the necessities of the atoms in \mathcal{S} are: $\Pi_{\overline{M}}(sensors) = 1$, $\Pi_{\overline{M}}(cobra) = 1$, $\Pi_{\overline{M}}(centralizedComputing) = 0.8$, $\Pi_{\overline{M}}(profA) = 1$ and $\Pi_{\overline{M}}(middleware) = 0.9$. The possibilistic equilibrium of \overline{M} is therefore:

$$\overline{\mathcal{S}} = \left(\begin{array}{l} \{(sensors, [1]), (corba, [1]), (centralizedComputing, [0.8])\}, \\ \{(profA, [1]), (middleware, [0.9])\}, \emptyset \end{array} \right)$$

3. Main Example

In this section, we describe in more detail the example scenario from robotics, which we introduced in Section 1. We use this scenario in the following sections of the paper to demonstrate our approach.

The scenario takes place in an office, where robots assist human workers in their tasks. The workers often need to share office supplies, to do their work. When they need certain supplies, they make requests to the robots to deliver the required supplies for them, while they keep working at their desks. We call each such request, a task.

The robots have partial knowledge of the environment. The location of each supply is only known by the last robot that delivered the supply. The robots therefore need to exchange information to be able to locate the supplies. We assume that the robots can communicate via the wireless network of the office. The scenario is depicted in Figure 1.

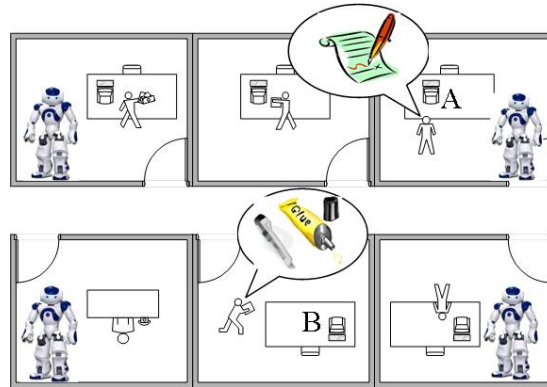


Figure 1. A depicted scenario of robots in office building.

Specifically, we consider a set of four robots $\mathcal{A}g = \{ag_1, ag_2, ag_3, ag_4\}$ and four tasks: $\mathcal{T} = \{t_1, t_2, t_3, t_4\}$, where t_1 is to deliver a pen to desk D_a , t_2 is to deliver a piece of paper to desk D_a , t_3 is to deliver a tube of glue to desk D_b , and t_4 is to deliver a cutter to desk D_b . A robot can carry out a task if it can handle the requested supply and it knows its current location and the location that the supply needs to be delivered to. We assume that the robots are not identical, and that each of them is able to handle a different subset of the supplies: ag_1 can handle the pen or the glue, ag_2 the paper, ag_3 the glue or the cutter, and ag_4 the pen or the cutter.

The robots know which other robot knows the current location of each supply and the location it has to be delivered to, but such information is only revealed to the robots that participate in the same coalition.

At a given point, we assume that the information that is available to each robot is the one presented in Table 1. For example, ag_1 does not know the current location of the supply associated with task t_1 (the pen), but knows where the pen needs to be delivered. On the other hand, it knows the current location of the supply associated with task t_2 (the paper), but does not know where this should be delivered. Table 2 presents the distances between each robot and each of the supplies and the distances between each desk and each supply at the given point.

Table 1. Robots' knowledge and capabilities.

Robot	ag_1				ag_2			
Task	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
Source		x			x			
Destination	x		x					x

Robot	ag_3				ag_4			
Task	t_1	t_2	t_3	t_4	t_1	t_2	t_3	t_4
Source				x			x	
Destination		x						

Table 2. Distances among locations.

Distances among Locations				
Robot	Pen	Paper	Glue	Cutter
ag_1	10	15	9	12
ag_2	14	8	11	13
ag_3	12	14	10	7
ag_4	9	12	15	11
Destination	Pen	Paper	Glue	Cutter
D_a	11	16	9	8
D_b	14	7	12	9

Based on the available information and their capabilities, the robots generate plans that will allow them to carry out the given tasks. For example, there are two alternative plans for carrying out t_1 , which is to deliver the pen to desk D_a . The pen can either be delivered by ag_1 , provided that it is informed about the current location of the pen by ag_2 ; or by ag_4 provided that it is informed about the location of the pen by ag_2 and about the location that it must be delivered to by ag_1 . After generating all possible plans for the four tasks, the robots need to decide what coalitions to form to carry out these plans. In this setting, a coalition is a group of robots that cooperate to carry out a task. For carrying out all four tasks, two alternative coalitions may be formed:

$$C_0 := \{(ag_1, t_3), (ag_2, t_2), (ag_3, t_4), (ag_4, t_1)\}$$

$$C_1 := \{(ag_1, t_1), (ag_2, t_2), (ag_3, t_3), (ag_4, t_4)\}$$

Both coalitions require the cooperation of all robots but differ in the assignment of the tasks to the robots. They also involve different smaller coalitions among the robots. For example, C_0 involves four smaller coalitions: (i) among ag_1 , ag_2 and ag_4 for carrying out t_1 ; (ii) among ag_1 , ag_2 and ag_3 for t_2 ; (iii) among ag_1 and ag_4 for t_3 and (iv) among ag_2 and ag_3 for t_4 . The two coalitions differ also in the total distance that the robots need to cover to carry out the four tasks. As we discuss in Section 4, such differences can be considered when deciding which coalition to form.

Finally, after forming a coalition, each robot has to make its own plan to carry out the assigned tasks, e.g., find the optimal route to deliver the supply from its current location to the required

destination. Typically, route planning programs split into two main parts, the representation of the environment and a method of searching possible route paths between the current robot position and some new location, avoiding the obstacles which are known. Hence, mobile robot navigation planning requires having sufficient reliable estimation of the current location of the robot, and a precise map of the navigation space. Path planning takes into consideration a model or a map of the environment or contexts, to determine what are the geometric path points for the mobile robots to track from a start position to the goal to be reached.

The most commonly used algorithms for these methods are the A^* algorithm, a global search algorithm giving a complete and optimal global path in static environments, and its optimization the D^* algorithm. Other examples in the literature include using distributed route planning methods for multiple mobile robots, using the Lagrangian decomposition technique, neural networks [39] and genetic algorithms [40]. One of the lessons which has been learned in this research area is that the need for optimal planning is outweighed by the need for quickly finding an appropriate plan [41].

In this paper, our focus is on finding and selecting among the possible coalitions with which a given set of goals will be reached, rather than on the individual plans of the agents to carry out their assigned tasks.

4. Computing and Evaluating Coalitions in the Perfect World

One of the problems that arise in scenarios like the one in our main example, is how to compute the possible coalitions with which the agents can carry out the given tasks, or, more generally, to fulfil their goals. We adopt a contextual reasoning approach to solve this problem, specifically using methods and tools for non-monotonic MCS [14]. The main advantages of such an approach are: (a) Using this model, we are able to represent heterogeneous agents that use different knowledge representation formalisms; (b) the format of the bridge rules allows modeling different types of relationships between agents such as inter-dependencies, conflicting goals and constraints; (c) Non-monotonic MCS is a well-studied model and there are both centralized and distributed reasoning algorithms and tools that can be used to reason with it. Our approach is, roughly, the following: We first model multi-agent systems as non-monotonic MCS, i.e., we model agents as contexts, and their dependencies and constraints as bridge rules. We then compute the possible coalitions using existing algorithms for MCS equilibria.

4.1. Modeling Dependencies

We model each agent in a multi-agent system as a context in a non-monotonic MCS. The knowledge base of the context describes the goals of the agent and the actions that it can perform. Goals and actions are represented as literals of the form g_k, a_j , respectively. The bridge rules of each context describe the dependencies of the corresponding agent on other agents to achieve its goal. We adopt the definition of [29] to model the dependence relations of the agents:

$$dp : \text{basic_dep}(ag_i, ag_j, g_k, p_l, a_m)$$

which has the following meaning: To achieve goal g_k using plan p_l , agent ag_i depends on agent ag_j performing action a_m .

A plan $p_l = (ag_1 : a_1, ag_2 : a_2, \dots, ag_n : a_n)$ that achieves goal g_k of agent a_i , where $ag_j : a_j$ denotes that action a_j is carried out by agent ag_j , involves the following dependencies:

$$dp_j : \text{basic_dep}(ag_i, ag_j, g_k, p_l, a_j), j = \{1, \dots, n\}$$

We write $DP(ag_i, g_k, p_l)$ to denote the set of all these dependencies.

A dependency can be written in the form of a logic programming rule of the form $Head \leftarrow Body$, where $Head$ represents the goal of the agent and $Body$ represents the plan p_l that achieves this goal, as a conjunction of literals, each representing an action included in this plan. Using this model, we use bridge rules to represent the dependencies among agents as follows:

Definition 11. For an agent ag_i with goal g_k achieved through plan $p_l = (ag_1 : a_1, ag_2 : a_2, \dots, ag_n : a_n)$, the set of dependencies $DP(ag_i, g_k, p_l)$ is represented by a bridge rule of the form:

$$(c_i : g_k) \leftarrow (c_1 : a_1), (c_2 : a_2), \dots, (c_n : a_n) \quad (14)$$

where $c_j, j = 1, \dots, i, \dots, n$ is the context representing agent ag_j .

Using this representation, we model multi-agent systems as MCS as follows:

Definition 12. A MCS $M(\mathcal{A})$ corresponding to a multi-agent system \mathcal{A} is a set of contexts $c_i = (L_i, kb_i, br_i)$, where $L_i = (KB_i, BS_i, ACC_i)$ is the logic of agent $ag_i \in \mathcal{A}$, $kb_i \in \mathbf{KB}_i$ is a knowledge base that includes the goals of ag_i and the actions it can perform, and br_i is a set of bridge rules, which include the rules that represent the dependencies of ag_i on other agents in \mathcal{A} for all the goals g_k of the agent and their associated plans p_l .

Two advantages of this model are that it enables agents using different logics to describe their actions and goals to form plans cooperatively by exchanging information through their bridge rules, and that it enables reasoning with inconsistencies that may arise e.g., by the agents' conflicting goals or by other types of conflicts among the agents' knowledge bases. Moreover, as discussed in the next section, there are both centralized and distributed reasoning algorithms for this model, which allow its use in different types of multi-agent systems and architectures.

Example 3. In our main example, described in Section 3, we assume, for the sake of simplicity, that all robots use propositional logic. We model the four robots, ag_1 - ag_4 , as contexts c_1 - c_4 , respectively, with the following knowledge bases:

$$\begin{aligned} kb_1 &= \{a_{2s}, a_{1d}, a_{3d}, a_{1c} \vee a_{3c}\} \\ kb_2 &= \{a_{1s}, a_{4d}, a_{2c}\} \\ kb_3 &= \{a_{4s}, a_{2d}, a_{3c} \vee a_{4c}\} \\ kb_4 &= \{a_{3s}, a_{1c} \vee a_{4c}\} \end{aligned}$$

where a_{ij} represent the actions that a robot can perform, with i denoting the supply (1 for the pen, 2 for the paper, 3 for the glue and 4 for the cutter) and j the type of action that the agent can perform: c denotes carrying, s denotes providing the current location (source) of a supply, and d denotes providing the location that the supply needs to be delivered to (destination). For example, ag_1 can

- provide the current location of the paper (a_{2s})
- provide the location that the pen needs to be delivered to (a_{1d})
- provide the location that the glue needs to be delivered to (a_{3d})
- carry the pen or the glue ($a_{1c} \vee a_{3c}$)

The four tasks in our main example are represented as goals. For example, t_1 , the task to deliver the pen to desk D_a is represented by g_1 . We assume that a robot can achieve a goal g_i , i.e., to deliver object i to the requested location, if it can carry the object, denoted as a_{ic} . For example, only robots ag_1 and ag_4 can fulfil goal g_1 , because these are the only robots that can carry the pen.

Given the information in Table 1, the four goals can be fulfilled as follows. There are two alternative plans for fulfilling g_1 :

$$\begin{aligned} p_{11} &= (ag_2 : a_{1s}, ag_1 : a_{1c}) \\ p_{12} &= (ag_2 : a_{1s}, ag_1 : a_{1d}, ag_4 : a_{1c}) \end{aligned}$$

According to p_{11} , robot ag_2 must provide the current location of the pen ($ag_2 : a_{1s}$) and ag_1 must carry the pen to its destination ($ag_1 : a_{1c}$). According to p_{12} , robot ag_2 must provide the current location of the pen

($ag_2 : a_{1s}$), ag_1 must provide the location that it has to be delivered to ($ag_1 : a_{1d}$), and ag_4 must carry the pen to its destination ($ag_4 : a_{1c}$).

For g_2 there is only one plan, p_{21} ; for g_3 there are two alternative plans: p_{31} and p_{32} ; and for g_4 there are two plans: p_{41} and p_{42} :

$$p_{21} = (ag_1 : a_{2s}, ag_3 : a_{2d}, ag_2 : a_{2c})$$

$$p_{31} = (ag_4 : a_{3s}, ag_1 : a_{3c})$$

$$p_{32} = (ag_4 : a_{3s}, ag_1 : a_{3d}, ag_3 : a_{3c})$$

$$p_{41} = (ag_2 : a_{4d}, ag_3 : a_{4c})$$

$$p_{42} = (ag_3 : a_{4s}, ag_2 : a_{4d}, ag_4 : a_{4c})$$

Each plan involves some dependencies among robots. For example, p_{11} involves the following dependency:

$$dp_1 : \text{basic_dep}(ag_1, ag_2, g_1, p_{11}, a_{1s})$$

i.e., to achieve goal g_1 , ag_1 depends on ag_2 providing the current location of the pen (a_{1s}). Figure 2 depicts the dependencies involved in all plan, abstracting from the plan, like in [27]. The figure should be read as follows: The pair of edges pointing from node ag_1 to the rectangle a_{1s} and then from this rectangle to ag_2 , indicates that to fulfil goal g_1 , ag_1 depends on ag_2 performing action a_{1s} .

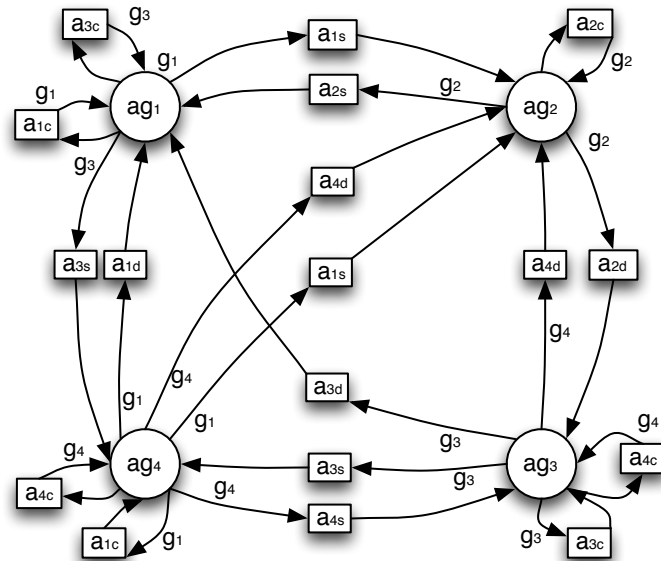


Figure 2. Dependencies among the four robot agents.

The same dependencies can also be represented as bridge rules. Each of these rules describes the dependencies involved in a single plan. For example, r_1 corresponds to plan p_{11} and represents dependency dp_1 .

$$r_1 = (c_1 : g_1) \leftarrow (c_1 : a_{1c}), (c_2 : a_{1s})$$

$$r_2 = (c_4 : g_1) \leftarrow (c_4 : a_{1c}), (c_2 : a_{1s}), (c_1 : a_{1d})$$

$$r_3 = (c_2 : g_2) \leftarrow (c_2 : a_{2c}), (c_1 : a_{2s}), (c_3 : a_{2d})$$

$$r_4 = (c_1 : g_3) \leftarrow (c_1 : a_{3c}), (c_4 : a_{3s})$$

$$r_5 = (c_3 : g_3) \leftarrow (c_3 : a_{3c}), (c_4 : a_{3s}), (c_1 : a_{3d})$$

$$r_6 = (c_3 : g_4) \leftarrow (c_3 : a_{4c}), (c_2 : a_{4d})$$

$$r_7 = (c_4 : g_4) \leftarrow (c_4 : a_{4c}), (c_3 : a_{4s}), (c_2 : a_{4d})$$

A constraint of the system is that an object can only be carried by at most one robot at any time. To represent this constraint, we expand the body of each of the bridge rules with a predicate that describes the fact that another robot carries the supply that the rule refers to:

$$\begin{aligned}
r'_1 &= (c_1 : g_1) \leftarrow (c_1 : a_{1c}), (c_2 : a_{1s}), \text{not}(c_1 : \text{carriesElse}_1) \\
r'_2 &= (c_4 : g_1) \leftarrow (c_4 : a_{1c}), (c_2 : a_{1s}), (c_1 : a_{1d}), \text{not}(c_4 : \text{carriesElse}_1) \\
r'_3 &= (c_2 : g_2) \leftarrow (c_2 : a_{2c}), (c_1 : a_{2s}), (c_3 : a_{2d}), \text{not}(c_2 : \text{carriesElse}_2) \\
r'_4 &= (c_1 : g_3) \leftarrow (c_1 : a_{3c}), (c_4 : a_{3s}), \text{not}(c_1 : \text{carriesElse}_3) \\
r'_5 &= (c_3 : g_3) \leftarrow (c_3 : a_{3c}), (c_4 : a_{3s}), (c_1 : a_{3d}), \text{not}(c_3 : \text{carriesElse}_3) \\
r'_6 &= (c_3 : g_4) \leftarrow (c_3 : a_{4c}), (c_2 : a_{4d}), \text{not}(c_3 : \text{carriesElse}_3) \\
r'_7 &= (c_4 : g_4) \leftarrow (c_4 : a_{4c}), (c_3 : a_{4s}), (c_2 : a_{4d}), \text{not}(c_4 : \text{carriesElse}_4)
\end{aligned}$$

For example, $c_1 : \text{carriesElse}_1$, which appears in r'_1 , represents that an agent, other than ag_1 , carries supply 1 (the pen).

For each of the four robots ag_l , and for each supply i that the robot can carry, we also add bridge rules of the form:

$$(c_l : \text{carriesElse}_i) \leftarrow (c_k : a_{ic})$$

where $i, k, l = \{1..4\}$ and $k \neq l$, which describe the cases that an agent, other than ag_l , carries supply i . For example, the following rules describe the cases that a robot, other than ag_1 , carries the pen.

$$\begin{aligned}
r'_8 &= (c_1 : \text{carriesElse}_1) \leftarrow (c_2 : a_{1c}) \\
r'_9 &= (c_1 : \text{carriesElse}_1) \leftarrow (c_3 : a_{1c}) \\
r'_{10} &= (c_1 : \text{carriesElse}_1) \leftarrow (c_4 : a_{1c})
\end{aligned}$$

We should note here that the same example can also be represented using dynamic MCS [42], which use schematic bridge rules that are instantiated at run time. For example, using dynamic MCS, we could replace the static predicates $(c_1 : \text{carriesElse}_1)$ in rule r'_1 and $(c_4 : \text{carriesElse}_1)$ in r'_2 by a dynamic predicate $(c_x : a_{1c})$, which would be instantiated at run time by any context c_x , which contains a_{1c} in its knowledge base; in other words by any robot that can carry supply 1. Rules such as r'_8 , r'_9 and r'_{10} would then be unnecessary and would be omitted. Such an approach is more appropriate for open and dynamic environments, where any agent may join or leave the system at any time and without prior notice (e.g., Ambient Intelligence environments).

In the running example, we assumed, for the sake of simplicity, that all robots use propositional logic. However, any knowledge representation formalism captured by Definition 12 can be used and each robot may use a different formalism. Note also that we used a very simple notation for the plans, as we do not aim to reason with plans; we are only interested in the dependencies that each plan involves.

4.2. Computing Coalitions

In non-monotonic MCS, an equilibrium represents an acceptable belief state of the system. Each belief set in this state is derived from the knowledge base of the corresponding context and is compatible with the applicable bridge rules. For a multi-agent system \mathcal{A} and the corresponding MCS $M(\mathcal{A})$ it holds that every coalition with which the agents in \mathcal{A} can fulfil their goals corresponds to a different equilibrium $\mathcal{S} = \{S_1, \dots, S_n\}$ of $M(\mathcal{A})$. Each belief set S_i in \mathcal{S} contains the actions that ag_1 can perform and the goals that the agent can fulfil within this coalition. If a goal does not appear in any of the equilibria of $M(\mathcal{A})$, then there is no coalition that can fulfil this goal.

The problem of computing all possible coalitions with which the agents in a multi-agent system \mathcal{A} can fulfil their goals can therefore be solved by creating the corresponding MCS $M(\mathcal{A})$ and computing the equilibria of $M(\mathcal{A})$.

Equilibria in non-monotonic MCS can be computed using any of the following algorithms/implementations depending on the requirements of the specific system or application, e.g., with respect to the representation and reasoning capabilities of the agents:

- The MCS-IE system [43] implements a centralized reasoning approach that is based on the translation of MCS into HEX-programs [44] (an extension of answer set programs with external atoms), and on their execution in the dlw-hex system (<http://www.kr.tuwien.ac.at/research/systems/dlvhex/>).
- The algorithms proposed in [25] implement a distributed computation, which however assumes that all contexts are homogeneous with respect to the logic that they use (defeasible logic).
- The three algorithms proposed in [45] enable distributed computation of equilibria. DMCS assumes that each agent has minimal knowledge about the world, namely the agents that it is connected through the bridge rules, but does not have any further metadata, e.g., topological information, of the system. Its computational complexity is exponential to the number of literals used in the bridge rules. DMCS-OPT uses graph theory techniques to detect any cycle dependencies in the system and avoid them during the evaluation of the equilibria, improving the scalability of the evaluation. DMCS-STREAMING computes the equilibria gradually (k equilibria at a time), reducing the memory requirements for the agents. The three algorithms have been implemented in a system prototype (<http://www.kr.tuwien.ac.at/research/systems/dmcs>).

Dao-Tran et al. [45] present a performance evaluation of the algorithms they propose, as well as a comparison with previous approaches. They conclude that the algorithms in [25] are in general much faster as they are based on a low-complexity logic, while among the others there is no clear winner, as their performance depends on the memory capacity of the agents, and the topology of the system.

Choosing the best approach for the computation of coalitions depends on several parameters, such as the targeted environment, the available means of communication, the computational, communication and knowledge representation capabilities of the agents, and the specific needs and requirements of the use cases that we want to support. For small-scale systems, such as the one in our running example, a centralized approach, e.g., the one proposed in [43], is probably more appropriate as it achieves better performance when the total number of agents is small. For larger-scale systems, a distributed scalable approach is probably more appropriate. In that case, if the main requirement is to compute all possible coalitions as fast as possible, DMCS and DMCS-OPT should be preferred; DMCS for systems with simple topology, i.e., fewer dependencies among the agents, and DMCS-OPT for more complex systems with possible cycle dependencies among the agents. Finally, in cases that the memory capacity of the available agents is limited, or that we are only interested in computing fast some (and not all) of the possible coalitions, DMCS-STREAMING seems to be the most appropriate approach.

Example 4. Returning to our main example, the MCS that corresponds to the system of the four robots, $M(\mathcal{A})$, has two equilibria: \mathcal{S}^0 and \mathcal{S}^1 :

$$\mathcal{S}^0 = \left(\begin{array}{l} \{a_{2s}, a_{1d}, a_{3d}, a_{3c}, g_3\}, \\ \{a_{1s}, a_{4d}, a_{2c}, g_2\}, \\ \{a_{4s}, a_{2d}, a_{4c}, g_4\}, \\ \{a_{3s}, a_{1c}, g_1\} \end{array} \right)$$

$$\mathcal{S}^1 = \left(\begin{array}{l} \{a_{2s}, a_{1d}, a_{3d}, a_{1c}, g_1\}, \\ \{a_{1s}, a_{4d}, a_{2c}, g_2\}, \\ \{a_{4s}, a_{2d}, a_{3c}, g_3\}, \\ \{a_{3s}, a_{4c}, g_4\} \end{array} \right)$$

\mathcal{S}^0 represents coalition C_0 , according to which ag_1 delivers the glue to desk D_b (g_3), ag_2 delivers the paper to desk D_a (g_2), ag_3 delivers the cutter to desk D_b (g_4) and ag_4 delivers the pen to desk D_a (g_1). \mathcal{S}^1 represents

coalition C_1 , according to which ag_1 delivers the pen to desk D_a (g_1), ag_2 delivers the paper to desk D_a (g_2), ag_3 delivers the glue to desk D_b (g_3) and ag_4 delivers the cutter to desk D_b (g_4). Figure 3 graphically represents the two coalitions.

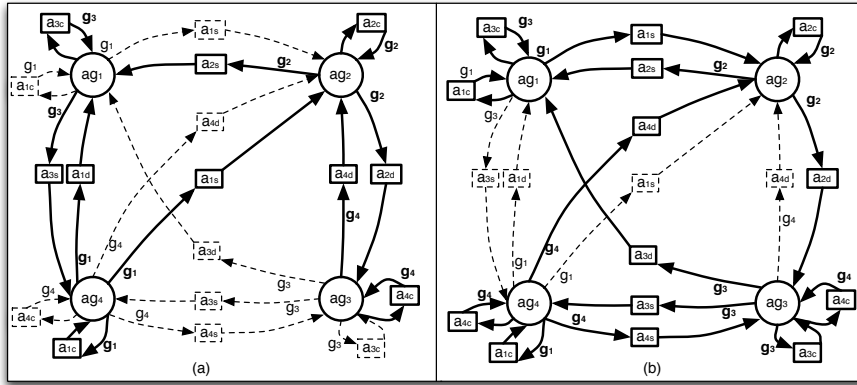


Figure 3. Coalitions C_0 (a), and C_1 (b) in bold; remaining dependencies in dotted lines.

To fulfil their goals in a given coalition, the agents need to perform the actions contained in the plans associated with these goals. For example, the plans associated with coalition C_0 are: p_{12} (for goal g_1), p_{21} (for g_2), p_{31} (for g_3) and p_{41} (for g_4), while the plans associated with C_1 are p_{11} , p_{21} , p_{32} and p_{42} .

4.3. Evaluating the Coalitions

Once the possible coalitions are identified, the problem then is to evaluate the alternative coalitions and select the best one. To address this problem, which is orthogonal to the computation of the possible coalitions, we need to identify the requirements of the domain and apply the associated metrics. In this section, we provide some examples of such requirements and their associated metrics and demonstrate how they can be applied to our motivating example.

Efficiency and stability metrics are commonly used metrics to evaluate coalitions (e.g., see [46–50]). Efficiency refers to the gain the agents receive by being in a coalition, while stability refers to the certainty that the coalition is viable in the longer term.

Specifically, the efficiency of a coalition is the relation between what the agents can achieve in this coalition compared to what they would achieve alone or in a different coalition. Furthermore, a coalition is economically efficient iff (i) no one can be made better off without making someone else worse off, (ii) no additional output can be obtained without increasing the amount of inputs, (iii) production proceeds at the lowest possible per-unit cost [46].

Example 5. In our running example, efficiency can be associated with the total distance that the four robots must cover to carry out the tasks. Using the information in Table 2, we can compute the cost of carrying out the tasks in a given coalition as the sum of the distances that the robots have to cover in order to perform the actions involved in this coalition:

$$cost(C) = \sum_{i=1}^4 dist(ag_i, j) + (dist(j, dest(j)))$$

where $dist(ag_i, j)$ denotes the distance between the robot (ag_i) and the supply (j) it has to carry in coalition C , and $dist(j, dest(j))$ denotes the distance between the material (j) and its destination ($dest(j)$). Based on this formula, the costs of C_0 and C_1 are:

$$cost(C_0) = (9 + 12) + (8 + 16) + (7 + 9) + (9 + 11) = 81$$

$$cost(C_1) = (10 + 11) + (8 + 16) + (10 + 12) + (11 + 9) = 87$$

If we compare C_0 and C_1 , C_0 is more economically efficient as at least one agent is better off without making anyone worse off, all else being equal (the distance that ag_3 has to cover is 16 in C_0 and 22 in C_1 , while all other robots have to cover the same distance in both coalitions). C_0 is also more cost efficient than C_1 , as its overall cost is smaller.

Other approaches could take into consideration the time it takes for each robot to accomplish its task, and the best coalition would in this case be the one which executes the fastest all its tasks. To do this we would need to integrate information about the task duration for each robot and task, along with the distances, and compute the times similarly as we did for the distances. Moreover, other variables could be included such as the resources being actually used by each robot to accomplish its task, and the potential obstacles to avoid to reach their goals.

The stability of a coalition is related to the potential gain in staying in the coalition or quitting the coalition for more profit (i.e., free riding). To evaluate the stability of a coalition several different factors need to be considered. First, the outcome of the coalition should be better than the individual ones accumulated. This is usually computed via a characteristic function such as the one proposed in [51]. Second, the benefits of a coalition should be distributed fairly among the agents that participate in it. Several *sharing rules* have been proposed for this purpose, such as the Shapley value [47,50], nucleolus [48] and Satisfactory Nucleolus [49]. The common idea of all such rules is to take into account both the individual contribution and the free rider's value when sharing the benefits.

Depending on the specific characteristics of the domain, several other requirements may be taken into account such as security, user-friendliness or conviviality. In [52], we proposed evaluating coalitions in terms of *conviviality*. Defined by Illich as "individual freedom realized in personal interdependence" [53], conviviality was introduced as a social science concept for multi-agent systems to highlight soft qualitative requirements like user-friendliness of systems. As explained in [54], this property is more relevant in social systems (e.g., web communities, social networks, digital cities, etc.) where the cohesion of a group can be reinforced by sharing knowledge or resources and through reciprocal interactions. Conviviality has recently been applied to several domains such as education [55], scientific theory building [56] and social networks [57] involving either face-to-face or virtual interactions. Although the system in our motivating example consists only of artificial agents (robots), we use it below to illustrate how this concept can be applied to any kind of multi-agent systems, including those that involve human agents. In [52], we proposed measures for conviviality to evaluate coalitions in dependence networks based on the following rough idea: more opportunities to work with others increases the system's conviviality. Specifically, we measured conviviality by the number of reciprocity-based coalitions that can be formed within an overall coalition. In our measures we used the notion of *cycles* in the dependence graph, which denote the smallest graph topology expressing interdependence, and therefore conviviality. Given the dependence network (DN) that corresponds to a given coalition, the conviviality of the coalition $conv(DN)$ can be computed as follows:

$$conv(DN) = \frac{\sum coal(a, b)}{\Omega}, \quad (15)$$

$$\Omega = |\mathcal{A}|(|\mathcal{A}| - 1) \times \Theta, \quad (16)$$

$$\Theta = \sum_{l=2}^{l=|\mathcal{A}|} Perm(|\mathcal{A}| - 2, l - 2) \times |G|^l, \quad (17)$$

where $|\mathcal{A}|$ is the number of agents in the system, $|G|$ is the number of goals, $Perm$ is the usual permutation defined in combinatorics, $coal(a, b)$ for any distinct pair of agents $a, b \in \mathcal{A}$ is the number of cycles that contain the ordered pair (a, b) in DN , l is the cycle length, and Ω denotes the maximal number of pairs of agents in cycles.

Example 6. Back to our running example, abstracting from plans and actions, Figure 4a,b represent the dependence networks for coalitions C_0 and C_1 respectively. For both dependence networks, by applying formulae (16) and (17), we get $\Theta = 656$ and $\Omega = 7872$. By applying formula (15), we can then compute the conviviality of the two coalitions:

$$\text{conv}(C_0) = 0.00089, \text{conv}(C_1) = 0.00127$$

Based on the above, C_1 is therefore preferred to C_0 in terms of conviviality.

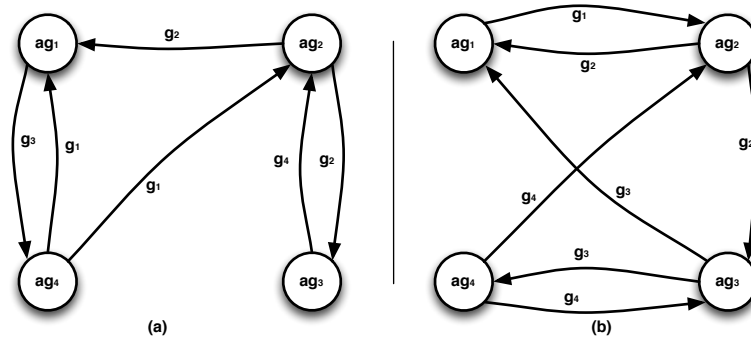


Figure 4. Goal dependencies in coalitions C_0 (a), and C_1 (b).

In the case where the agents' goals are in conflict, the selection of a coalition should depend on the priorities among the agents' goals. It is among our plans for future work, to integrate in our model a preference relation on goals and to develop methods for preference-based coalition formation.

The problem of selecting the best coalition has also been seen as tied to finding the optimal coalition structure, and therefore as a search problem which consists of performing a search on the coalition structure graph [58]. A search through a subset of the coalition structure space is required: first, the bottom two levels of the coalition structure graph, then a time bounded breadth first search from the top of the coalition structure graph. The best coalition structure that is found is selected. This was optimized by [59], who analyzed fewer coalitions to establish small bounds from the optimal. In [60], coalition sizes are limited and a greedy heuristic to yield a coalition structure is used; i.e., based on a limited number of agents.

5. Computing Coalitions under Uncertainty

The models for representing and computing coalitions that we presented in Section 4 are based on the *perfect world assumption*, according to which it is always certain that agents will carry out the actions that are specified in the associated plans. However, in real-world settings, such assumption is not always valid. For instance, in our running example, there is uncertainty associated with the robots carrying the supplies to their destinations. A robot may fall while on its way to pick up the supply or towards its final destination; it may also run out of battery and fail to carry out its action.

In this section, we extend the representation and computational models that we presented in Section 4 to take into account the uncertainty in the agents' actions. Similarly as in Section 4, we first present a rule-based representation model for dependencies, based on Possibilistic Multi-Context Systems [26]. We then present the algorithms for computing the alternative coalitions and for evaluating the coalitions under uncertainty using multiple-criteria decision-making methods.

5.1. Modeling Uncertainty

We extend the definition of [29] for dependence relations to take into account the uncertainty of actions:

Definition 13. The dependency of an agent ag_i on agent ag_j performing action a_m , which is part of the plan p_l to achieve goal g_k , is denoted as:

$$dp : poss_dep(ag_i, ag_j, g_k, p_l, a_m, \alpha_{a_m})$$

where α_{a_m} represents the possibility that ag_j will carry out action a_m with success.

We denote the set of dependencies of agent ag_i to achieve goal g_k using the plan $p_l = (ag_1 : a_1, ag_2 : a_2, \dots, ag_n : a_n)$ as $DP_{poss}(ag_i, g_k, p_l, \alpha_{p_l})$, where α_{p_l} represents the aggregated possibility that agents ag_i, \dots, ag_n will successfully perform actions a_1, \dots, a_n , respectively. Assuming that actions a_1, \dots, a_n are mutually independent:

$$\alpha_{p_l} = \prod_{i=1}^n \alpha_{a_i}$$

In the general case, where dependencies among the actions in p_l may exist:

$$\alpha_{p_l} = \left(\prod_{i=1}^{n-1} \alpha_{[a_i | a_{i+1} \wedge \dots \wedge a_n]} \right) \times \alpha_{a_n}$$

where $\alpha_{[a_i | a_{i+1} \wedge \dots \wedge a_n]}$ denotes the possibility that a_i will be carried out successfully given that actions a_{i+1}, \dots, a_n have been successful.

To model the uncertainty in the agents' actions, we model a multi-agent system as a possibilistic MCS (or poss-MCS) with possibilistic bridge rules that represent the dependencies of each agent on other agents to achieve its goals.

Definition 14. For an agent ag_i with goal g_k achieved through plan $p_l = (ag_1 : a_1, ag_2 : a_2, \dots, ag_n : a_n)$, the set of dependencies $DP_{poss}(ag_i, g_k, p_l, \alpha_{p_l})$ is represented as a possibilistic bridge rule of the form:

$$(c_i : g_k) \leftarrow (c_1 : a_1), \dots, (c_n : a_n), [\alpha_{p_l}]$$

where $c_j, j = 1, \dots, i, \dots, n$ is the context representing agent ag_j .

Based on the above representations, we can now define the following representation for multi-agent systems with uncertainty.

Definition 15. A poss-MCS $\overline{M}(\mathcal{A})$ representing a multi-agent system \mathcal{A} with uncertainty is a set of possibilistic contexts $\bar{c}_i = (\Sigma_i, \overline{P}_i, \overline{B}_i)$, where Σ_i is a set of atoms representing the goals of the agent and the actions it can perform; \overline{P}_i represents the local theory of the agent as a possibilistic logic program; and \overline{B}_i is a set of possibilistic bridge rules representing the dependencies of the agent on other agents to fulfil its goals.

As it becomes obvious, the above definition is more restrictive compared to Definition 12, as it only allows agents that use possibilistic logic programs as their knowledge representation formalism. However, by explicitly representing the degree of certainty in the rules, it offers a more elaborate manipulation of uncertainty, which is desirable when such values are available (e.g., in sensor-enriched environments).

Example 7. In our running example, the uncertainty is associated with the robots carrying the supplies to their destinations. In the robotics-related literature, different approaches may be found to compute uncertainty in (robotic) motion planning. Typically, uncertainty is considered in two ways: in sensing, where the current state of the robot and workspace may not be known with certainty, or in predictability, where the future state of the robot and workspace cannot be deterministically predicted even when the current state and future actions are known [61]. Factors that may affect predictability are uncertainty in the workspace, e.g., moving obstacles [62], uncertainty in the configuration of the robot [63]; or uncertainty in the robot's motion [64].

In our running example, the assumption is that uncertainty is associated with the motion of the robots and the handling of a supply. Furthermore, it is a function of the distance that the robot must cover with and without the supply. For our purpose, the following very simple function will suffice to compute the possibility that robot ag_i carries out action a_{jc} ; i.e., carries supply c to its destination:

$$\text{poss}(ag_i, a_{jc}) = 1 - (0.001 \times \text{dist}(ag_i, j) + 0.002 \times (\text{dist}(j, \text{dest}(j))) \quad (18)$$

where $\text{dist}(ag_i, j)$ denotes the distance between the robot (ag_i) and the supply (j), and $\text{dist}(j, \text{dest}(j))$ denotes the distance between the supply (j) and its destination ($\text{dest}(j)$). The intuition is that the uncertainty of a robot carrying a supply to its destination is proportional to the distance that the robot has to cover to get to the supply, and the distance that it has to cover after it grabs the supply to deliver it to its destination. As many more things may go wrong while carrying the supply, the second element has a double weight.

Using this function and the distances between the different locations as they appear in Table 2, the dependencies among the robots are represented by the following possibilistic bridge rules:

$$\begin{aligned} \overline{pr}_1 &= (c_1 : g_1) \leftarrow (c_1 : a_{1c}), (c_2 : a_{1s}), \text{not}(c_1 : \text{carriesElse}_1), [0.968] \\ \overline{pr}_2 &= (c_4 : g_1) \leftarrow (c_4 : a_{1c}), (c_2 : a_{1s}), (c_1 : a_{1d}), \text{not}(c_4 : \text{carriesElse}_1), [0.969] \\ \overline{pr}_3 &= (c_2 : g_2) \leftarrow (c_2 : a_{2c}), (c_1 : a_{2s}), (c_3 : a_{2d}), \text{not}(c_2 : \text{carriesElse}_2), [0.960] \\ \overline{pr}_4 &= (c_1 : g_3) \leftarrow (c_1 : a_{3c}), (c_4 : a_{3s}), \text{not}(c_1 : \text{carriesElse}_3), [0.967] \\ \overline{pr}_5 &= (c_3 : g_3) \leftarrow (c_3 : a_{3c}), (c_4 : a_{3s}), (c_1 : a_{3d}), \text{not}(c_3 : \text{carriesElse}_3), [0.966] \\ \overline{pr}_6 &= (c_3 : g_4) \leftarrow (c_3 : a_{4c}), (c_2 : a_{4d}), [0.975], \text{not}(c_3 : \text{carriesElse}_3), \\ \overline{pr}_7 &= (c_4 : g_4) \leftarrow (c_4 : a_{4c}), (c_3 : a_{4s}), (c_2 : a_{4d}), \text{not}(c_4 : \text{carriesElse}_4), [0.971] \end{aligned}$$

For example, the necessity degree of rule \overline{pr}_1 is associated with the uncertainty of action ($ag_1 : a_{1c}$): robot ag_1 carrying the pen to desk D_a . According to function (18), the possibility that this action will be carried out with success is:

$$\text{poss}(ag_1, a_{1c}) = 1 - (0.001 \times 10 + 0.002 \times 11) = 0.968$$

where 10 is the distance between ag_1 and the pen, and 11 is the distance between the pen and desk D_a . As in the perfect world case, the negative atoms in the bridge rules (e.g., $\text{not}(c_1 : \text{carriesElse}_1)$) represent that the supply that a rule refers to (e.g., the pen) is not carried by another robot, and are derived from bridge rules such as the following:

$$\begin{aligned} \overline{pr}_8 &= (c_1 : \text{carriesElse}_1) \leftarrow (c_2 : a_{1c}), [1] \\ \overline{pr}_9 &= (c_1 : \text{carriesElse}_1) \leftarrow (c_3 : a_{1c}), [1] \\ \overline{pr}_{10} &= (c_1 : \text{carriesElse}_1) \leftarrow (c_4 : a_{1c}), [1] \end{aligned}$$

5.2. Computing Coalitions under Uncertainty

In Section 2, we showed that the possibilistic equilibrium of a poss-MCS \overline{M} is a collection of possibilistic atom sets \overline{S}_i , where each \overline{S}_i is a collection of possibilistic atoms \overline{p}_i from the system contexts. In the context of multi-agent systems, an equilibrium represents a coalition in which agents can fulfil their goals and \overline{p}_i represents an action or a goal of agent ag_i and its necessity degree. In the case of goals, the necessity degree represents the level of certainty with which the coalition will fulfil the goal.

To compute the potential coalitions under uncertainty in a multi-agent system \mathcal{A} , we compute the possibilistic equilibria of $\overline{M}(\mathcal{A})$. This is possible using a fixpoint theory proposed in [26], which is based on the use of a consequence operator that gradually computes the consequences of a possibilistic MCS using its applicable rules.

Example 8. In our example, the poss-MCS that corresponds to the system of the four robots, $\overline{M}(\mathcal{A})$, has two possibilistic equilibria: \overline{S}^0 and \overline{S}^1

$$\bar{S}^0 = \begin{pmatrix} \{(a_{2s}, [1]), (a_{1d}, [1]), (a_{3d}, [1]), (a_{3c}, [0.967]), (g_3, [0.967])\}, \\ \{(a_{1s}, [1]), (a_{4d}, [1]), (a_{2c}, [0.960]), (g_2, [0.960])\}, \\ \{(a_{4s}, [1]), (a_{2d}, [1]), (a_{4c}, [0.975]), (g_4, [0.975])\}, \\ \{(a_{3s}, [1]), (a_{1c}, [0.969]), (g_1, [0.969])\} \end{pmatrix}$$

$$\bar{S}^1 = \begin{pmatrix} \{(a_{2s}, [1]), (a_{1d}, [1]), (a_{3d}, [1]), (a_{1c}, [0.968]), (g_1, [0.968])\}, \\ \{(a_{1s}, [1]), (a_{4d}, [1]), (a_{2c}, [0.960]), (g_2, [0.960])\}, \\ \{(a_{4s}, [1]), (a_{2d}, [1]), (a_{3c}, [0.966]), (g_3, [0.966])\}, \\ \{(a_{3s}, [1]), (a_{4c}, [0.971]), (g_4, [0.971])\} \end{pmatrix}$$

\bar{S}^0 represents coalition C_0 and \bar{S}^1 represents coalition C_1 ; the two coalitions are graphically represented in Figure 3. C_0 will fulfil goal g_1 with certainty degree 0.969, g_2 with 0.96, g_3 with 0.967 and g_4 with 0.975. On the other hand, C_1 will fulfil goal g_1 with certainty degree 0.968, g_2 with 0.96, g_3 with 0.966 and g_4 with 0.971.

5.3. Evaluating Coalitions under Uncertainty

In Section 4.3 we presented different approaches for evaluating coalitions in the *perfect world*, where there is certainty about the ability of the agents to carry out their tasks. All these approaches can also be applied to environments with uncertainty. However, in such environments, one may also want to take into account the uncertainty in the agents' actions, and the degree of certainty with which each coalition can fulfil the agents' goals, which can be an alternative criterion for evaluating the coalitions.

The problem of evaluating coalitions taking into account the degree of certainty with which the coalitions fulfil the different goals is essentially a *multi-attribute decision-making* (MCDM) problem. We use the following definition for MCDM problems [65]:

Definition 16. Let $D = \{D_i\}, i = 1, \dots, m$ be a set of decision alternatives and $R = \{R_j\}, j = 1, \dots, n$ a set of decision criteria according to which the desirability of an action is judged. Determine the optimal decision alternative D^* with the highest degree of desirability with respect to all relevant decision criteria R_j .

In our case, the decision alternatives, D , are the alternative coalitions that the agents may form, and the criteria, R , are the certainty degrees with which the goals will be reached. In the literature one can find many different alternatives for solving this problem, most of which are based on functions that aggregate the scores for the different criteria. Some characteristic examples are:

- the *Weighted Sum Method* [66], according to which each criterion R_j is given a weight w_j , so that the sum of all weights is 1 ($\sum_{j=1}^n w_j = 1$), and the overall score of each alternative D_i is the weighted sum of q_{ij} , i.e., the scores of D_i for each criterion R_j :

$$WS(D_i) = \sum_{j=1}^n w_j q_{ij}$$

The optimal decision alternative is the one with the highest overall score (WS). This method can only be used in single-dimensional cases, in which all the score units are the same.

- the *Weighted Product Method* [67], which aggregates the individual scores using their product instead of their sum. Specifically, each decision alternative is compared with the others by multiplying several ratios, one for each criterion. Each ratio is raised to the power equivalent to

the relative weight of the corresponding criterion. In general, to compare alternatives D_1 and D_2 , the following product must be calculated:

$$\text{Ratio}(D_1/D_2) = \prod_{j=1}^n (q_{1j}/q_{2j})^{w_j}$$

where q_{1j} and q_{2j} are the scores of D_1 and D_2 respectively for criterion R_j , and w_j is the weight of criterion R_j . If the ratio is greater than one, then D_1 is more desirable than D_2 . The best alternative is the one that is better, with respect to this ratio, than all other alternatives. This method is sometimes called *dimensionless analysis* because its structure eliminates any units of measure. It can, therefore, be used in both single- and multi-dimensional decision-making problems.

- *TOPSIS (Technique for Order Preference by Similarity to Ideal Solution [68])*, which is based on the concept that the chosen alternative should have the shortest geometric distance from the positive ideal solution and the longest geometric distance from the negative ideal solution. TOPSIS assumes that each alternative has a tendency of monotonically increasing or decreasing utility. Therefore, it is easy to locate the ideal and negative ideal solutions.

A review and comparative study of the most prominent methods for multi-criteria decision-making is available in [69].

Evaluating coalitions in multi-agent systems, as this is described in this paper, is a single-dimension problem, as the different criteria are the certainty degrees with which the goals are achieved, and are, therefore, of the same form and take values in the same range ($[0, 1]$). For such cases, the Weighted Product Method is considered to be the preferred method because of its simplicity.

Example 9. In our running example, there are two alternative coalitions with which the agents can fulfil their goals: C_0 and C_1 . If we apply the Weighted Sum Method, and considering that the four criteria represent the certainty degrees with which the four goals will be reached (α_{g_i}), the weighted sum of a coalition is:

$$WS(C) = \sum_{i=1}^4 w_{g_i} \alpha_{g_i}$$

where w_{g_i} is the weight of goal g_i . Assuming that the four goals are equally important, their weights are equal, therefore the weighted sums of the coalitions are:

$$WS(C_0) = 0.25 \times 0.967 + 0.25 \times 0.960 + 0.25 \times 0.975 + 0.25 \times 0.969 = 0.96775$$

$$WS(C_1) = 0.25 \times 0.968 + 0.25 \times 0.960 + 0.25 \times 0.966 + 0.25 \times 0.971 = 0.96625$$

C_0 is preferable as it achieves the system's goals with greater certainty.

Assuming that goals g_1 and g_4 (delivering the pen to desk D_a and the cutter to desk D_b) are more important than g_2 and g_3 (delivering the piece of paper to desk D_a and the glue to desk D_b), and the weights of the four goals are: $w_{g_1} = w_{g_4} = 0.4$ and $w_{g_2} = w_{g_3} = 0.1$, the weighted sums of the two coalitions are:

$$WS(C_0) = 0.4 \times 0.967 + 0.1 \times 0.960 + 0.1 \times 0.975 + 0.4 \times 0.969 = 0.9679$$

$$WS(C_1) = 0.4 \times 0.968 + 0.1 \times 0.960 + 0.1 \times 0.966 + 0.4 \times 0.971 = 0.9682$$

and in this case C_1 is considered to be the best coalition.

6. Related Work

This article does not present the first attempt to bring together agents and formal logics of context. Two earlier studies used MCS to specify and implement agent architectures [18,19]. The main idea in both studies is that the logical description of an agent can be broken into components, each component is modeled as a context, and the interactions among the components as bridge rules. This approach

was used in [18] to simplify the construction of a BDI agent, and was then extended in [19] to handle more efficiently implementation issues such as grouping together contexts in modules, and enabling inter-context synchronization. The main difference with our work, is that while these studies focus on the internal representation of an agent, we focus on the coalitions among agents in multi-agent systems.

As we also mention in the Introduction, this study is built on previous work on modeling multi-agent systems as MCS and computing coalitions using MCS computational methods and tools, which we first presented in [28]. Here we give more details about the proposed methodology. We also extend it with modeling and reasoning methods from possibilistic reasoning to handle uncertainty in the agents' actions. In another previous study, we evaluated information exchange in distributed information systems, based on modeling MCS as dependence networks where bridge rules are represented as dependencies [35]. Here we do the opposite: we use bridge rules to represent dependencies among agents, and model agents as contexts in MCS.

There are also several previous studies that deal with the problem of coalition formation. The most notable approaches include variants of the contract net protocol [70,71], the main idea of which is that agents break down their composite tasks into simpler ones and subcontract the subtasks to other agents via bidding systems; formal methods from multi-agent systems and game theory, e.g., [1–4,27,60,72–76]; solutions from the area of robotics based on schema theory [77,78] or synergy [79]; approaches inspired by the formation of coalitions in politics [80]; approximate solutions using genetic [5,6], swarm [7,8] or dynamic programming algorithms [9,10], and adaptive approaches based on machine learning algorithms [11,81–83].

Compared to all previous approaches, the solution that we propose here is the only one that combines the following four characteristics: (a) It is able to handle heterogeneous agents that use different knowledge representation formalisms; (b) it enables reasoning with agents with conflicting goals; (c) by integrating features from possibilistic reasoning, it also allows handling uncertainty in the agents' actions; (d) there are both centralized and distributed algorithms that can be used for computing the coalitions, and can therefore be used in several different settings with different privacy and information security requirements.

There are also several studies that focus on the problem of forming coalitions under uncertainty. There are different kinds of uncertainty that they consider, such as uncertainty in the agent type, i.e., the knowledge about the agent's abilities and motivations [84,85]; uncertainty in the agents' costs, which in a multi-robot system can be energy, hardware cost or processing time [86]; uncertainty in the resources that an agent can contribute to the coalition [87,88]; or uncertainty in the value of a coalition, i.e., the costs and payoff of a coalition for each agent taking part in it [89,90]. Most of the proposed solutions are based on game theory and machine learning algorithms. Our approach, on the other hand, focuses on another kind of uncertainty, i.e., uncertainty in the agents' actions, and our proposed solution combines elements of logic-based and possibilistic reasoning.

Another line of research that is related to our work focuses on the problem of coalition formation taking into account the trust and reputation of agents. In this domain, trust has been defined as "the subjective probability that an agent will perform a particular task as expected" [91], and reputation as "the expectation of someone's behavior based on previous interactions indicated by others" [92]. As is obvious from the definitions, both concepts are associated with the uncertainty that the agent will perform a certain action. A recent survey reviewed different models of trust and reputation in the context of different types of agent interactions including coalition formation [93]. For the problem of coalition formation, they identified in the literature seven different models [94–100] and analyzed them with respect to various dimensions of trust, such as the paradigm type, i.e., the method used to build the model (cognitive, numeric or hybrid), the information sources (direct interaction, direct observation or witness information), the use of risk measures to decide on the formation of coalitions and their cheating assumptions. Although our approach does not directly address trust and reputation, the model we propose in Section 5 can be used to model trust by associating the possibility that an agent will carry out a certain action with the trust that the dependent agents have in this agent,

adopting the numeric paradigm for representing and calculating trust. The method for evaluating coalitions that we propose in the same section can then be employed as a risk measure for forming coalitions. Although we do not make any assumptions on the source of the trust information and the ability of agents to cheat, by adopting ideas from previous work in this area, it should be possible to develop a complete model of trust for multi-agent systems based on our proposed approach.

7. Summary and Future Work

In multi-agent systems, agents often need to form coalitions to fulfil their goals. In this article, we propose a novel approach for computing all possible coalitions using the model, algorithms and tools for Multi-Context Systems. We also propose different ways of evaluating the coalitions, taking into account several functional or non-functional requirements, such as efficiency, stability and conviviality. Finally, we extend our model and algorithms with features from possibilistic reasoning, and more specifically possibilistic MCS, in order to model, compute and evaluate the different coalitions taking also into account the uncertainty in the agents' actions. To demonstrate our approach, we use an example from robotics.

It is among our plans to test and analyze the performance of the proposed methods in more complex scenarios, i.e., with a much larger number of agents and tasks and, therefore, with a much larger number of possible coalitions, and to compare it with the performance of some of the other methods for coalition formation discussed in Section 6. We also plan to integrate preferences on agents and goals into our methods, and by drawing on previous research in preference-based inconsistency resolution in MCS [25,33,34], to develop algorithms for preference-based coalition formation, which can be used in systems with agents with conflicting goals. Another plan is to extend our approach using dynamic MCS [42], which use schematic bridge rules that are instantiated at run time with concrete contexts, to deal with dynamic environments, where agents may join or leave the system at any point and without prior notice. Finally, we want to apply and test our methods in domains characterised by the heterogeneity of the agents and uncertainty, such as Ambient Intelligence systems. For this purpose, we will extend existing MCS tools, such as DMCS [101], a distributed solver for MCS, and MCS-IE [43], a tool for explaining inconsistencies in MCS, with features from possibilistic MCS.

Author Contributions: Conceptualization, A.B. and P.C.; methodology, A.B. and P.C.; formal analysis, A.B. and P.C.; investigation, A.B. and P.C.; writing—original draft preparation, A.B. and P.C.; writing—review and editing, A.B. and P.C.; visualization, A.B. and P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Anglano, C.; Canonico, M.; Castagno, P.; Guazzone, M.; Sereno, M. Profit-aware coalition formation in fog computing providers: A game-theoretic approach. *Concurr. Comput. Pract. Exp.* **2019**. [[CrossRef](#)]
2. Bullinger, M. Computing Desirable Partitions in Coalition Formation Games. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems; International Foundation for Autonomous Agents and Multiagent Systems, AAMAS '20, Richland, SC, USA, 9–13 May 2020; pp. 2185–2187.
3. Kashevnik, A.; Teslya, N. Blockchain-Oriented Coalition Formation by CPS Resources: Ontological Approach and Case Study. *Electronics* **2018**, *7*, 66. [[CrossRef](#)]
4. Liu, C.; Zhu, E. A New Modeling of Cooperative Agents from Game-theoretic Perspective. In Proceedings of the 4th International Conference on Mathematics and Artificial Intelligence—ICMAI 2019, Chengdu, China, 12–15 April 2019; pp. 133–136.
5. Rizk, T.; Awad, M. A quantum genetic algorithm for pickup and delivery problems with coalition formation. In Proceedings of the 23rd International Conference KES-2019, Budapest, Hungary, 4–6 September 2019; pp. 261–270.

6. Guo, M.; Xin, B.; Chen, J.; Wang, Y. Multi-agent coalition formation by an efficient genetic algorithm with heuristic initialization and repair strategy. *Swarm Evol. Comput.* **2020**, *55*, 100686. [[CrossRef](#)]
7. Bhateja, N.; Sethi, N.; Kumar, D. Study of Ant Colony Optimization Technique for Coalition Formation in Multi Agent Systems. In Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 21–22 December 2018; pp. 1–4.
8. Zhang, K.; Hu, Y.; Tian, F.; Li, C. A coalition-structure's generation method for solving cooperative computing problems in edge computing environments. *Inf. Sci.* **2020**, *536*, 372–390. [[CrossRef](#)]
9. Su, X.; Wang, Y.; Jia, X.; Guo, L.; Ding, Z. Two Innovative Coalition Formation Models for Dynamic Task Allocation in Disaster Rescues. *J. Syst. Sci. Syst. Eng.* **2018**, *27*, 215–230. [[CrossRef](#)]
10. Changder, N.; Aknine, S.; Dutta, A. An Effective Dynamic Programming Algorithm for Optimal Coalition Structure Generation. In Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 721–727.
11. Pei, Z.; Piao, S.; Souidi, M. Coalition Formation for Multi-agent Pursuit Based on Neural Network. *J. Intell. Robot. Syst.* **2019**, *95*, 887–899. [[CrossRef](#)]
12. Giunchiglia, F.; Serafini, L. Multilanguage hierarchical logics, or: How we can do without modal logics. *Artif. Intell.* **1994**, *65*, 29–70. [[CrossRef](#)]
13. Ghidini, C.; Giunchiglia, F. Local Models Semantics, or contextual reasoning=locality+compatibility. *Artif. Intell.* **2001**, *127*, 221–259. [[CrossRef](#)]
14. Brewka, G.; Eiter, T. Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In Proceedings of the 22nd AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22–26 July 2007; pp. 385–390.
15. Lenat, D.B.; Guha, R.V. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989.
16. Borgida, A.; Serafini, L. Distributed Description Logics: Assimilating Information from Peer Sources. *J. Data Semant.* **2003**, *1*, 153–184.
17. Bouquet, P.; Giunchiglia, F.; van Harmelen, F.; Serafini, L.; Stuckenschmidt, H. C-OWL: Contextualizing Ontologies. In Proceedings of the International Semantic Web Conference, Sanibel, FL, USA, 20–23 October 2003; pp. 164–179.
18. Parsons, S.; Sierra, C.; Jennings, N.R. Agents that reason and negotiate by arguing. *J. Log. Comput.* **1998**, *8*, 261–292. [[CrossRef](#)]
19. Sabater, J.; Sierra, C.; Parsons, S.; Jennings, N.R. Engineering Executable Agents using Multi-context Systems. *J. Log. Comput.* **2002**, *12*, 413–442. [[CrossRef](#)]
20. de Mello, R.R.P.; Gelaim, T.Â.; Silveira, R.A. Negotiating Agents: A Model Based on BDI Architecture and Multi-Context Systems Using Aspiration Adaptation Theory as a Negotiation Strategy. In Proceedings of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018), Matsue, Japan, 4–6 July 2018; pp. 351–362.
21. Lagos, N.; Mos, A.; Vion-Dury, J.Y. Multi-Context Systems for Consistency Validation and Querying of Business Process Models. In Proceedings of the 21st International Conference KES-2017, Marseille, France, 6–8 September 2017; pp. 225–234.
22. Dao-Tran, M.; Eiter, T. Streaming Multi-Context Systems. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 1000–1007.
23. Ellmauthaler, S. Multi-Context Reasoning in Continuous Data-Flow Environments. *KI-Künstliche Intell.* **2018**, *33*, 101–104. [[CrossRef](#)]
24. Antoniou, G.; Papatheodorou, C.; Bikakis, A. Reasoning about Context in Ambient Intelligence Environments: A Report from the Field. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 12th Intl. Conference, KR 2010, Toronto, ON, Canada, 9–13 May 2010*; AAAI Press: Palo Alto, CA, USA, 2010; pp. 557–559.
25. Bikakis, A.; Antoniou, G.; Hassapis, P. Strategies for contextual reasoning with conflicts in Ambient Intelligence. *Knowl. Inf. Syst.* **2011**, *27*, 45–84. [[CrossRef](#)]
26. Jin, Y.; Wang, K.; Wen, L. Possibilistic Reasoning in Multi-Context Systems: Preliminary Report. In *PRICAI 2012: Trends in Artificial Intelligence, Proceedings of the 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, 3–7 September 2012*; Springer: Berlin, Germany, 2012; pp. 180–193.
27. Sichman, J.S.; Conte, R. Multi-agent dependence by dependence graphs. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, AAMAS 2002, Bologna, Italy, 15–19 July 2002; ACM: New York, NY, USA, 2002; pp. 483–490.

28. Bikakis, A.; Caire, P. Computing Coalitions in Multiagent Systems: A Contextual Reasoning Approach. In *Multi-Agent Systems, Proceedings of the 12th European Conference, EUMAS 2014, Prague, Czech Republic, 18–19 December 2014*; Revised Selected Papers; Springer: Berlin, Germany, 2014; pp. 85–100.
29. Sichman, J.S.; Demazeau, Y. On Social Reasoning in Multi-Agent Systems. *Rev. Iberoam. Intel. Artif.* **2001**, *13*, 68–84.
30. Harman, H.; Simoens, P. Action graphs for proactive robot assistance in smart environments. *J. Ambient Intell. Smart Environ.* **2020**, *12*, 79–99. [[CrossRef](#)]
31. Brings, J.; Daun, M.; Weyer, T.; Pohl, K. Goal-Based Configuration Analysis for Networks of Collaborative Cyber-Physical Systems. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing, SAC '20, Brno, Czech Republic, 30 March–3 April 2020*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1387–1396.
32. Sauro, L. Formalizing Admissibility Criteria in Coalition Formation among Goal Directed Agents. Ph.D. Thesis, University of Turin, Turin, Italy, 2006.
33. Eiter, T.; Fink, M.; Schüller, P.; Weinzierl, A. Finding Explanations of Inconsistency in Multi-Context Systems. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, ON, Canada, 9–13 May 2010*; AAAI Press: Palo Alto, CA, USA, 2010.
34. Eiter, T.; Fink, M.; Weinzierl, A. Preference-Based Inconsistency Assessment in Multi-Context Systems. In *Logics in Artificial Intelligence, Proceedings of the 12th European Conference, JELIA 2010, Helsinki, Finland, 13–15 September 2010*; Springer: Berlin, Germany, 2010; Volume 6341, pp. 143–155.
35. Caire, P.; Bikakis, A.; Traon, Y.L. Information Dependencies in MCS: Conviviality-Based Model and Metrics. In *Principles and Practice of Multi-Agent Systems, Proceedings of the 16th International Conference, Dunedin, New Zealand, 1–6 December 2013*; Springer: Berlin, Germany, 2013; pp. 405–412.
36. Dubois, D.; Lang, J.; Prade, H. Possibilistic Logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming–Nonmonotonic Reasoning and Uncertain Reasoning (Volume 3)*; Gabbay, D.M., Hogger, C.J., Robinson, J.A., Eds.; Clarendon Press: Oxford, UK, 1994; pp. 439–513.
37. Nicolas, P.; Garcia, L.; Stéphan, I.; Lefèvre, C. Possibilistic uncertainty handling for answer set programming. *Ann. Math. Artif. Intell.* **2006**, *47*, 139–181. [[CrossRef](#)]
38. Zadeh, L. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* **1978**, *1*, 3–28. [[CrossRef](#)]
39. Somhom, S.; Modares, A.; Enkawa, T. Competition-based neural network for the multiple travelling salesmen problem with minmax objective. *Comput. Oper. Res.* **1999**, *26*, 395–407. [[CrossRef](#)]
40. Cai, Z.; Peng, Z. Cooperative Co-evolutionary Adaptive Genetic Algorithm in Path Planning of Cooperative Multi-Mobile Robot Systems. *J. Intell. Robot. Syst.* **2002**, *33*, 61–71. [[CrossRef](#)]
41. Koenig, S.; Likhachev, M. Fast replanning for navigation in unknown terrain. *IEEE Trans. Robot.* **2005**, *21*, 354–363. [[CrossRef](#)]
42. Dao-Tran, M.; Eiter, T.; Fink, M.; Krennwallner, T. Dynamic Distributed Nonmonotonic Multi-Context Systems. In *Nonmonotonic Reasoning, Essays Celebrating Its 30th Anniversary, Lexington, KY, USA, 22–25 October 2010*; College Publications: London, UK, 2011; Volume 31, pp. 63–88.
43. Bögl, M.; Eiter, T.; Fink, M.; Schüller, P. The mcs-ie System for Explaining Inconsistency in Multi-Context Systems. In *Logics in Artificial Intelligence, Proceedings of the 12th European Conference, JELIA 2010, Helsinki, Finland, 13–15 September 2010*; Springer: Berlin, Germany, 2010; pp. 356–359.
44. Eiter, T.; Ianni, G.; Schindlauer, R.; Tompits, H. A Uniform Integration of Higher-Order Reasoning and External Evaluations in Answer-Set Programming. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI-05, Edinburgh, UK, 30 July–5 August 2005*; pp. 90–96.
45. Dao-Tran, M.; Eiter, T.; Fink, M.; Krennwallner, T. Distributed Evaluation of Nonmonotonic Multi-context Systems. *J. Artif. Intell. Res. (JAIR)* **2015**, *52*, 543–600. [[CrossRef](#)]
46. O’Sullivan, A.; Sheffrin, S.M. *Economics: Principles in Action*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2006.
47. Shapley, L.S. A Value for n-person Games. *Ann. Math. Stud.* **1953**, *28*, 307–317.
48. Schmeidler, D. The nucleolus of a characteristic functional game. *SIAM J. Appl. Math.* **1969**, *17*, 1163–1170. [[CrossRef](#)]
49. Kronbak, L.G.; Lindroos, M. Sharing Rules and Stability in Coalition Games with Externalities. *Mar. Resour. Econ.* **2007**, *22*, 137–154. [[CrossRef](#)]
50. Magaña, A.; Carreras, F. Coalition Formation and Stability. *Group Decis. Negot.* **2018**, *27*, 467–502. [[CrossRef](#)]

51. Mesterton-Gibbons, M. *An Introduction to Game-Theoretic Modelling*; Addison-Wesley: Redwood, CA, USA, 1992.
52. Caire, P.; Alcade, B.; van der Torre, L.; Sombattheera, C. Conviviality Measures. In Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, 2–6 May 2011.
53. Illich, I. *Deschooling Society*; Marion Boyars Publishers, Ltd.: London, UK, 1971.
54. Caire, P. New Tools for Conviviality: Masks, Norms, Ontology, Requirements and Measures. Ph.D. Thesis, Luxembourg University, Luxembourg, 2010.
55. Arenas, F.J.; Connelly, D.A. The Claim on Human Conviviality in Cyberspace. In *Integrating an Awareness of Selfhood and Society into Virtual Learning*; Andrew, S., Cynthia Calongne, B.T., Arenas, F., Eds.; IGI Global: Hershey, PA, USA, 2017; Chapter 3, pp. 29–39.
56. Ellemers, N.; Fiske, S.T.; Abele, A.E.; Koch, A.; Yzerbyt, V. Adversarial alignment enables competing models to engage in cooperative theory building toward cumulative science. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 7561–7567. [[CrossRef](#)]
57. Cabitza, F.; Simone, C.; Cornetta, D. Sensitizing concepts for the next community-oriented technologies: Shifting focus from social networking to convivial artifacts. *J. Community Inform.* **2015**, *11*, 11.
58. Sandholm, T.; Larson, K.; Andersson, M.; Shehory, O.; Tohmé, F. Coalition Structure Generation with Worst Case Guarantees. *Artif. Intell.* **1999**, *111*, 209–238. [[CrossRef](#)]
59. Dang, V.D.; Jennings, N.R. Generating Coalition Structures with Finite Bound from the Optimal Guarantees. In Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), New York, NY, USA, 19–23 August 2004; pp. 564–571.
60. Shehory, O.; Kraus, S. Methods for Task Allocation via Agent Coalition Formation. *Artif. Intell.* **1998**, *101*, 165–200. [[CrossRef](#)]
61. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: New York, NY, USA, 2006.
62. Vasquez, D.; Large, F.; Fraichard, T.; Laugier, C. High-speed autonomous navigation with motion prediction for unknown moving obstacles. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004; pp. 82–87.
63. Hsiao, K.; Kaelbling, L.P.; Lozano-Pérez, T. Grasping POMDPs. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, ICRA 2007, Roma, Italy, 10–14 April 2007; pp. 4685–4692.
64. Alterovitz, R.; Siméon, T.; Goldberg, K.Y. The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty. In Proceedings of the Robotics: Science and Systems III, Atlanta, GA, USA, 27–30 June 2007.
65. Zimmermann, H. *Fuzzy Set Theory and Its Applications*; Kluwer Academic: Dordrecht, The Netherlands, 1991.
66. Fishburn, P.C. Additive Utilities with Incomplete Product Sets: Applications to Priorities and Assignments. *Oper. Res.* **1967**, *15*, 537–542. [[CrossRef](#)]
67. Miller, D.; Starr, M. *Executive Decisions and Operations Research*; Prentice-Hall, Inc.: Englewood Cliffs, NJ, USA, 1969.
68. Hwang, C.L.; Yoon, K. *Multiple Attribute Decision Making Methods and Applications: A State-of-the-Art Survey*; Springer: Berlin, Germany, 1981.
69. Triantaphyllou, E. *Multi-Criteria Decision Making Methods: A Comparative Study*; Springer: Berlin, Germany, 2000.
70. Gerkey, B.P.; Matarić, M.J. Sold!: Auction Methods for Multi-Robot Coordination. *IEEE Trans. Robot. Autom.* **2002**, *18*, 758–768. [[CrossRef](#)]
71. Lemaire, T.; Alami, R.; Lacroix, S. A Distributed Tasks Allocation Scheme in Multi-UAV Context. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, New Orleans, LA, USA, 26 April–1 May 2004; pp. 3622–3627.
72. Sichman, J.S. DEPINT: Dependence-based coalition formation in an open multi-agent scenario. *J. Artif. Soc. Soc. Simul.* **1998**, *1*, 1–3.
73. Klusch, M.; Gerber, A. Dynamic Coalition Formation among Rational Agents. *IEEE Intell. Syst.* **2002**, *17*, 42–47. [[CrossRef](#)]
74. Boella, G.; Sauro, L.; van der Torre, L. Algorithms for finding coalitions exploiting a new reciprocity condition. *Log. J. IGPL* **2009**, *17*, 273–297. [[CrossRef](#)]
75. Grossi, D.; Turrini, P. Dependence theory via game theory. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, ON, Canada, 10–14 May 2010; pp. 1147–1154.

76. Caire, P.; Villata, S.; Boella, G.; van der Torre, L. Conviviality masks in multiagent systems. In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, 12–16 May 2008; Volume 3, pp. 1265–1268.
77. Tang, F.; Parker, L.E. ASyMTRe: Automated Synthesis of Multi-Robot Task Solutions through Software Reconfiguration. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, New Orleans, LA, USA, 26 April–1 May 2004; pp. 1501–1508.
78. Zhang, Y.; Parker, L.E. IQ-ASyMTRe: Forming Executable Coalitions for Tightly Coupled Multirobot Tasks. *IEEE Trans. Robot.* **2013**, *29*, 400–416. [[CrossRef](#)]
79. Liemhetcharat, S.; Veloso, M.M. Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents. *Artif. Intell.* **2014**, *208*, 41–65. [[CrossRef](#)]
80. Chella, A.; Sorbello, R.; Ribaudo, D.; Finazzo, I.V.; Papuzza, L. A Mechanism of Coalition Formation in the Metaphor of Politics Multiagent Architecture. In *AI*IA 2003: Advances in Artificial Intelligence, Proceedings of the 8th Congress of the Italian Association for Artificial Intelligence, Pisa, Italy, 23–26 September 2003*; Cappelli, A., Turini, F., Eds. Springer: Berlin, Germany, 2003; pp. 410–422. [[CrossRef](#)]
81. Abdallah, S.; Lesser, V.R. Organization-Based Coalition Formation. In Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), New York, NY, USA, 19–23 August 2004; pp. 1296–1297.
82. Soh, L.; Li, X. Multiagent Coalition Formation for Distributed, Adaptive Resource Allocation. In Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, Las Vegas, NV, USA, 21–24 June 2004; CSREA Press: Sterling, VA, USA, 2004; Volume 1, pp. 372–378.
83. Sen, S. *An Intelligent and Unified Framework for Multiple Robot and Human Coalition Formation*; AAAI Press: Palo Alto, CA, USA, 2015; pp. 4395–4396.
84. Chalkiadakis, G.; Markakis, E.; Boutilier, C. 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, Honolulu, HI, USA, 14–18 May 2007; ACM: New York, NY, USA, 2007; pp. 64:1–64:8. [[CrossRef](#)]
85. Shehory, O.; Kraus, S. Feasible Formation of Coalitions Among Autonomous Agents in Non-Super-Additive Environments. *Comput. Intell.* **1999**, *15*, 218–251. [[CrossRef](#)]
86. Kargin, V. Uncertainty of the Shapley Value. In *Game Theory and Information*; University Library of Munich: Munich, Germany, 2003; Number: 0309003.
87. Mamakos, M.; Chalkiadakis, G. Probability Bounds for Overlapping Coalition Formation. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017; pp. 331–337.
88. Mittal, V.; Maghsudi, S.; Hossain, E. Distributed Cooperation Under Uncertainty in Drone-Based Wireless Networks: A Bayesian Coalitional Game. *arXiv* **2020**, arXiv:2009.00685.
89. Soh, L.; Tsatsoulis, C. Utility-based multiagent coalition formation with incomplete information and time constraints. In Proceedings of the IEEE International Conference on Systems, Man & Cybernetics, Washington, DC, USA, 5–8 October 2003; pp. 1481–1486.
90. Vig, L.; Adams, J.A. Multi-robot coalition formation. *IEEE Trans. Robot.* **2006**, *22*, 637–649. [[CrossRef](#)]
91. Gambetta, D. Can we trust? In *Trust: Making and Breaking Cooperative Relations*; Basil Blackwell: New York, NY, USA, 1988; pp. 213–237.
92. Abdul-Rahman, A.; Hailes, S. Supporting Trust in Virtual Communities. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7 January 2000; p. 6007.
93. Granatyr, J.; Botelho, V.; Lessing, O.R.; Scalabrin, E.E.; Barthès, J.P.; Enembreck, F. Trust and Reputation Models for Multiagent Systems. *ACM Comput. Surv.* **2015**, *48*. [[CrossRef](#)]
94. Griffiths, N.; Luck, M. Coalition Formation through Motivation and Trust. In Proceedings of the AAMAS03: Second International Conference on Autonomous Agents and Multiagent Systems, Melbourne Australia, 14–18 July 2003; Association for Computing Machinery: New York, NY, USA, 2003; pp. 17–24. [[CrossRef](#)]
95. Reháč, M.; Foltýn, L.; Pechoucek, M.; Benda, P. Trust Model for Open Ubiquitous Agent Systems. In Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Compiegne, France, 19–22 September 2005; IEEE Computer Society: Washington, DC, USA, 2005; pp. 536–542.

96. Ghanea-Hercock, R.; Ipswich, A.P. Dynamic Trust Formation in Multi-Agent Systems. In Proceedings of the Tenth International Workshop on Trust in Agent Societies at the Autonomous Agents and Multi-Agent Systems Conference, Honolulu, HI, USA, 15 May 2007.
97. Tong, X.; Huang, H.; Zhang, W. Agent Long-Term Coalition Credit. *Expert Syst. Appl.* **2009**, *36*, 9457–9465. [[CrossRef](#)]
98. Guo, L.; Wang, X.; Zeng, G. Trust-Based Optimal Workplace Coalition Generation. In Proceedings of the 2009 International Conference on Information Engineering and Computer Science, Wuhan, China, 19–20 December 2009; pp. 1–4.
99. Zhou, Q.; Wang, C.; Xie, J. CORE: A Trust Model for Agent Coalition Formation. In Proceedings of the Fifth International Conference on Natural Computation, ICNC 2009, Tianjian, China, 14–16 August 2009; IEEE Computer Society: Washington, DC, USA, 2009; Volume 6, pp. 541–545.
100. Erriquez, E.; van der Hoek, W.; Wooldridge, M. An Abstract Framework for Reasoning about Trust. In Proceedings of the The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, 2–6 May 2011; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2011; Volume 3, pp. 1085–1086.
101. Bairakdar, S.E.; Dao-Tran, M.; Eiter, T.; Fink, M.; Krennwallner, T. The DMCS Solver for Distributed Nonmonotonic Multi-Context Systems. In *Logics in Artificial Intelligence, Proceedings of the 12th European Conference, JELIA 2010, Helsinki, Finland, 13–15 September 2010*; Springer: Berlin, Germany, 2010; pp. 352–355.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).