Strengthening Admissible Coalitions

Guido Boella and **Luigi Sauro**[1] and **Leendert van der Torre**[2]

Abstract. We develop a criterion for coalition formation among goal-directed agents, the indecomposable do-ut-des property. The indecomposable do-ut-des property refines the do-ut-des property (literally *give to get*) by considering the fact that agents prefer to form coalitions whose components cannot be formed independently. A formal description of this property is provided as well as an analysis of algorithms and their complexity.

1 Introduction

In this work we study the formation of coalitions among goal-directed agents. We start from a theory of social power and dependence introduced by Castelfranchi et al. [3]. In this context a coalition is intended as a group of agents which agree to cooperate for the achievement of a shared goal or to exchange with each other the achievement of their own goals. The second case is of particular interest as different networks of exchanges can be considered but not all of them are realizable.

We assume that the formation of a coalition is supported by unanimous and enforced agreements, i.e. a coalition is effectively formed only when all its members agree to it (unanimousness) and they cannot deviate from what established in the agreement, once they decide to enter it (enforcement). Under these assumptions, we develop a criterion of admissibility, the indecomposable do-ut-des property (i-dud in the following), establishing which coalitions cannot be formed under the assumption that the agents are self-interested. The i-dud property is a refinement of the do-ut-des property [2] which describes a condition of reciprocity: an agent *gives* a goal only if this fact enables it to *obtain*, directly or indirectly, the satisfaction of one of its own goals. The i-dud property refines the do-ut-des property by taking into account also the fact that a coalition formation process can itself be costly and usually the costs involved in a coalition formation process increase with the number of agents involved. Furthermore, being a coalition agreed unanimously, the more agents are involved in it, the larger is the risk of defections which can jeopardize the formation of the coalition. Thus, agents prefer to form coalitions which are as small as possible.

In Section 2 we define the i-dud property and provide some examples of this notion. In Section 3, we provide an algorithm to search all the sub-coalitions of a given coalition which satisfies the i-dud property. Even if this problem is not computationally tractable, we show in Section 4 that the problem to verify if a single coalition satisfies the i-dud property is tractable and, in several cases, also the complexity of the first problem may decrease considerably. Conclusions end the paper.

[1] Università di Torino, Italy, email: {guido,sauro}@di.unito.it
[2] University of Luxembourg, Luxembourg, email: leon.vandertorre@uni.lu

2 From do-ut-des to i-dud coalitions

Depending on the problem, a multiagent system can be represented at different levels of abstraction. For example, if you want to study how agents have to coordinate in order to achieve a goal or how agents should optimally use their resources, then a multiagent system can involve resources, actions, plans [1, 6]. Instead, if you want to study which coalitions are strategically admissible to be formed, you usually do not need such fine-grained descriptions of a multiagent system, coalitions are directly described by means of the consequences that they can attain collaborating, without any description about which joint plans agents have to perform [7, 10]. Following this idea, we describe potential coalitions abstracting from actions, plans or resources. However, in contrast with these approaches, we do not represent a coalition just indicating the set of goals that it can attain. We want to use the topology of goal exchanges inside a coalition to define our admissibility criterion.

Thus, inspired by Conte and Sichman [4] we represent a (potential) coalition as a labeled AND-graph of dependencies among agents. A labeled AND-graph consists of a set of nodes $\mathcal{V}$ - which denotes the agents involved in the coalition - and a set $\mathcal{E}$ of labeled AND-arcs. Denoting with Gl the goals exchanged in the coalition, a labeled AND-arc connects an agent ag_i to a nonempty set of agents Q and it is labelled with a goal $g \in Gl$, so it can be represented as a triple (ag_i, Q, g). The meaning of such an arc is that the agent ag_i desires the goal g and the achievement of g is delegated to the set of agents Q. In order to represent a coalition a labeled AND-graph has to satisfy two further conditions. Since a coalition is intended as the result of an agreement process, the first condition is that only those agents that contribute to the achievement of some goals are admitted in this process. The second condition establishes that a coalition formation process does not involve private commitments that do not require any form of collaboration.

Definition 1 *A* coalition *is represented as a labeled AND-graph which is a tuple $C = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is a finite set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times (2^{\mathcal{V}} \setminus \{\emptyset\}) \times Gl$ is a set of labeled AND-arcs.*

C satisfies two conditions: (1) for each node $ag_j \in \mathcal{V}$, there exists at least an AND-arc (ag_i, Q, g) such that $ag_j \in Q$ and (2) $\mathcal{E}$ does not contain an AND-arc in the form $(ag_i, \{ag_i\}, g)$.

With an abuse of notation we mean with $(Q, g) \in C$ that there exists a $(ag_i, Q, g) \in \mathcal{E}$. Following [8] we call (Q, g) a commitment of C. A sub-coalition C' is intended a subgraph of C where some commitments are suppressed, $(Q, g) \notin C'$, for some $(Q, g) \in C$.

We want to restrict the notion of coalitions assuring, first, reciprocity, and, second, that no sub-coalitions of a coalition can be formed independently. The former property is called do-ut-des [2, 8], and we refine it to match the second requirement called indecomposability.

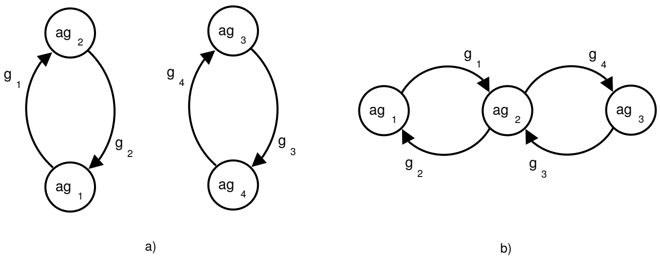

Figure 1. *Two coalitions that satisfy the do-ut-des property but that do not satisfy the i-dud property.*

The do-ut-des property assures that each chain of exchanges involved in a coalition returns something back to each agent involved in it. This property has been characterized in [2, 8] by means of a qualitative preference relation and a notion of dominance similar to those used in Game Theory, such as the notion of core. Here, instead, we propose a version of the do-ut-des property which is grounded on topological property of chains of exchanges; the equivalence with the dominance-based approach is shown in [8]. In [8] a goal is meant as a state of affairs that, once attained by a group of agents, benefits one or, in case, more agents. To simplify our formalism, we consider that there do not exist two agents which desire the same goal.

We introduce some preliminary notions. A finite sequence of AND-arcs $\mathcal{P} = (ag_{i_1}, Q_1, g_1), \ldots, (ag_{i_n}, Q_n, g_n)$ is a path if for all $2 \leq h \leq n$, $ag_{i_h} \in Q_{h-1}$. Paths formalize possible *chains of exchanges* among the agents. We denote with $out(ag_i)$ the set of labeled AND-arcs outgoing the agent ag_i, i.e. $out(ag_i) = \{(ag_j, Q, g)\}$. Given $\mathcal{O} \subseteq out(ag_i)$, we call $\mathcal{O}^*$ the propagation of $\mathcal{O}$, i.e. the set of AND-arcs containing all the paths $\mathcal{P} = (ag_{i_1}, Q_1, g_1), \ldots, (ag_{i_n}, Q_n, g_n)$ such that (1) $(ag_{i_1}, Q_1, g_1) \in \mathcal{O}$ and $\mathcal{P}$ does not contain an arc in $out(ag_i) \setminus \mathcal{O}$. We notice that $out^*(ag_i)$ contains all (and only) the paths starting from ag_i.

The do-ut-des property consists of two conditions. The first condition is an efficiency condition: there do not exist two distinct sets of agents in a coalition which are committed to the achievement of the same goal. The second condition expresses the notion of reciprocity: given an AND-arc $(ag_i, Q, g) \in \mathcal{E}$, each agent ag_j involved in Q agrees to provide the goal g to ag_i, only in the case this commitment returns the satisfaction of some of its goals by means of a path $\mathcal{P}$.

Definition 2 *A coalition $\langle \mathcal{V}, \mathcal{E} \rangle$ satisfies the* do-ut-des *property iff (1) there do not exist two commitments $(Q, g), (Q', g) \in C$ such that $Q \neq Q'$ and (2) for all $(ag_i, Q, g) \in \mathcal{E}$ and for all $ag_j \in Q$, $(ag_i, Q, g) \in out^*(ag_j)$.*

It can be seen that both the coalitions in Figure 1 (a) and (b) satisfy Definition 2.

However, the do-ut-des property does not consider the possibility that a coalition can be decomposed in smaller sub-coalitions which can be formed independently. Forming smaller coalitions can be preferred by agents because, e.g., involving less agents they reduce the risk of defections, are easier to monitor, less expensive to form by means of agreements, less trust is required among all the agents, etc. The coalitions in Figure 1 (a) and (b) show two cases of this fact. Consider in Figure 1 (a) the sub-coalitions C_1, involving only agents ag_1 and ag_2 and the relative arcs, and C_2, involving only agents ag_3 and ag_4 and the relative arcs. As the agents involved in C_1 are not interested in the goals achieved in C_2 and vice versa, the two sub-coalitions can be formed independently.

Concerning Figure 1 (b), C denotes the whole coalition, C_1 the coalition consisting of the nodes ag_1 and ag_2 and the two arcs labelled with the goals g_1 and g_2. C_2 denotes the coalition consisting of the nodes ag_2 and ag_3 and the remaining arcs labelled with the goals g_3 and g_4. The difference is that in Figure 1 (a) both ag_1 and ag_2 are indifferent between C_1 and the whole coalition, while in Figure 1 (b) ag_2 is not indifferent between C_1 and C since in C it receives the goal g_4 which is not provided in C_1.

However, ag_1 is indifferent between C_1 and C and ag_3 is indifferent between C_2 and C as they receive and have to obtain the same goals in both coalitions. Thus, if they agree to C, then they would agree respectively to C_1 and C_2. When agent ag_2 wants to propose to ag_1 and ag_3 to form coalitions, it has to decide whether to propose C to both agents or C_1 and C_2 separately. It knows that if they agree to C, then they would also agree respectively to C_1 and C_2. Agent ag_2 chooses C_1 or C_2 since forming one of them does not affect the possibility for ag_2 to reach an agreement on the other sub-coalition, and C_1 and C_2 are individually more reliable to succeed with respect to the whole C - as they involve individually less agents.

The i-dud property consists of three conditions. Given a coalition C, the first condition is the condition (1) of the do-ut-des property. The second condition strengthens the condition (2) of do-ut-des property by imposing that for each agent ag_i in C, $out^*(ag_i) = \mathcal{E}$. Thus, a coalition cannot be decomposed in two subgraphs which are disconnected as in Figure 1 (a). Finally, the third condition takes into account the case shown in Figure 1 (b): there does not exist an agent ag_i and a bi-partition $\mathcal{O}_1, \mathcal{O}_2$ of $out(ag_i)$ - where we assume that bi-partitions are composed by nonempty sets - such that $\mathcal{O}_1^* \cap \mathcal{O}_2^*$ is empty. The idea underlying this third condition is that if $\mathcal{O}_1^* \cap \mathcal{O}_2^* = \emptyset$, then no agent $ag_j \neq ag_i$ involved in $\mathcal{O}_1^*$ would be interested in one of the goals achieved in $\mathcal{O}_2^*$ and vice versa. So, ag_i can deal separately with the formation of these two sub-coalitions.

Definition 3 *A coalition $\langle \mathcal{V}, \mathcal{E} \rangle$ satisfies the* i-dud *property iff for all the agents $ag_i \in \mathcal{V}$, (cond1) there do not exist two commitments $(Q, g), (Q', g) \in C$ such that $Q \neq Q'$, (cond2) $out^*(ag_i) = \mathcal{E}$ and (cond3) there does not exist a bi-partition $\mathcal{O}_1, \mathcal{O}_2$ of $out(ag_i)$ such that $\mathcal{O}_1^* \cap \mathcal{O}_2^* = \emptyset$.*

Considering again the coalitions in Figure 1 (a) and (b), as expected, they both do not satisfy the i-dud property. In Figure 1 (a), the first condition of Definition 3 is not satisfied as, for example, $out^*(ag_1) = \{(ag_1, \{ag_2\}, g_1), (ag_2, \{ag_1\}, g_2)\} \subset \mathcal{E}$. In Figure 1 (b), the third condition of Definition 3 is not satisfied. Indeed, considering the bi-partition of ag_2 composed by $\mathcal{O}_1 = \{(ag_2, \{ag_1\}, g_2)\}$ and $\mathcal{O}_2 = \{(ag_2, \{ag_3\}, g_4)\}$, we have that $\mathcal{O}_1^* = \{(ag_2, \{ag_1\}, g_2), (ag_1, \{ag_2\}, g_1)\}$ and $\mathcal{O}_2^* = \{(ag_2, \{ag_3\}, g_4), (ag_3, \{ag_2\}, g_3)\}$. Thus, $\mathcal{O}_1^* \cap \mathcal{O}_2^*$ is empty.

A labeled AND-graph can represent a potential coalition consisting of all, or a large part of, the opportunities of collaboration in a multiagent system. So we would like to establish not only if the whole coalition is admissible or not, but also which of its sub-coalitions are admissible to be formed. Figure 2 (a) shows a quite complex coalition that does not satisfy the i-dud property. Indeed, given the bi-partition of $out(ag_1)$ $\mathcal{O}_1 = \{(ag_1, \{ag_4, ag_5\}, g_1)\}$ and $\mathcal{O}_2 = \{(ag_1, \{ag_3\}, g_6)\}$, it can be verified that $\mathcal{O}_1^* \cap \mathcal{O}_2^* = \emptyset$. Figure 2 (b), (c), (d) and (e) show all the sub-coalitions of Figure 2 (a) which satisfy the i-dud property. These coalitions clearly satisfy also the do-ut-des property. However, since the do-ut-des property seeks only the reciprocity in a coalition, any composition of coalition (e) with one of the coalitions (b), (c) and (d) satisfies the do-ut-des

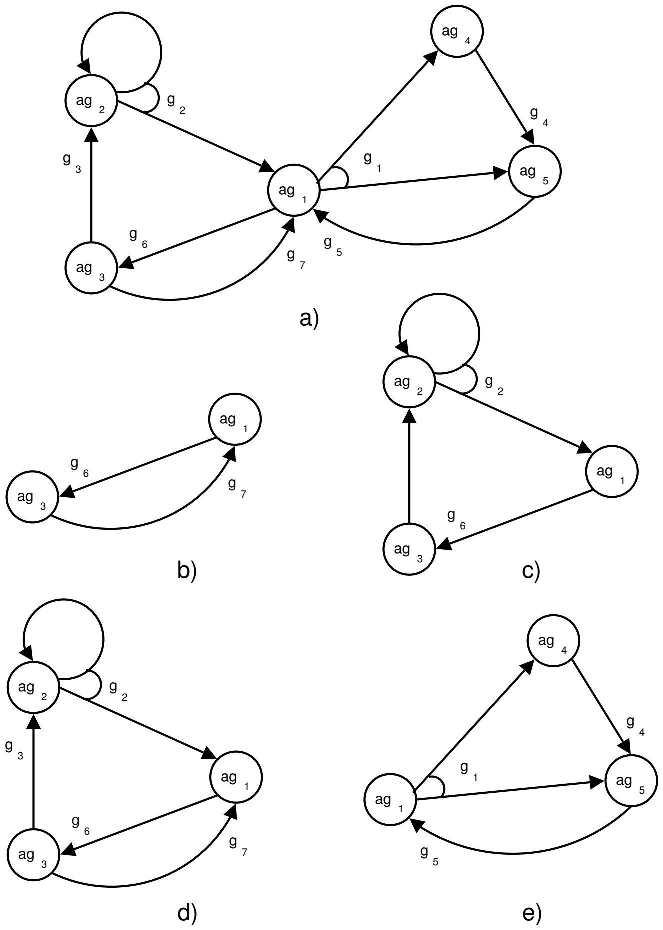

Figure 2. *A complex coalition (a) and the sub-coalitions which satisfy the i-dud property (b), (c), (d) and (e).*

property as well.

Given a coalition C, Definition 3 could be used *tout court* in order to design an algorithm which find all the sub-coalitions satisfying the i-dud property (C included). However, Definition 3 requires to verify, for each agent ag_i involved in C, a condition on the set of bi-partitions of $out(ag_i)$. The number of bi-partitions of a set A is equal to the Stirling number $S(n, 2) = 2^{n-1} - 1$, where n is the cardinality of A. Therefore, the problem to verify if just C satisfies the i-dud property would increase in complexity exponentially with the cardinality of $out(ag_i)$. For this reason we consider an alternative approach in order to make at least the verification of a single coalition tractable.

We reformulate cond2 as a property of strong connectivity of a directed graph. We define a direct graph $\mathbf{G}[C] = \langle \mathbf{V}, \mathbf{E} \rangle$ relative to the coalition $C = \langle \mathcal{V}, \mathcal{E} \rangle$ as follows: the set of nodes $\mathbf{V}$ is equal to the set $\mathcal{V}$ of agents involved in C and $(ag_i, ag_j) \in \mathbf{E}$ if and only if there exist a goal g and a group of agents Q such that $(ag_i, Q, g) \in \mathcal{E}$ and $ag_j \in Q$. It easy to see that if $(ag_j, Q, g) \in out^*(ag_i)$, then there exists a path in $\mathbf{G}[C]$ from ag_i to ag_j. So, since each agent is involved in the achievement of at least a goal, the condition that $out^*(ag_i) = \mathcal{E}$ is equivalent to say that $\mathbf{G}[C]$ is strongly connected, i.e. for each pair of nodes ag_i and ag_j there exists a path from ag_i to ag_j. Given a generic directed graph $\mathbf{G}$, we call the strongly connected components of $\mathbf{G}$ the maximal strongly connected sub-graphs which contains at least one arc.

Now we consider how to reformulate cond3. Under the assumption that $\mathbf{G}[C]$ is strongly connected, cond3 is closely related to the notion of biconnectivity for undirected graphs. An undirected graph G is biconnected if and only if it is connected and for all triples of distinct nodes ag_i, ag_j and ag_k, there exists a path p connecting ag_j and ag_k such that ag_i is not in p. In the contrary case, ag_i is called an articulation node [9]. As for strongly connected components of a directed graph, the biconnected components of an undirected graph G are the maximal biconnected subgraphs of G which contain at least one arc. It is easy to see that two distinct biconnected components share at most one node and, if so, this node is an articulation node.

Starting from the directed graph $\mathbf{G}[C] = \langle \mathbf{V}, \mathbf{E} \rangle$, we define an undirected graph $\mathsf{G}[C] = \langle \mathsf{V}, \mathsf{E} \rangle$ as follows: $\mathsf{V} = \mathbf{V}$ and, for $ag_i \neq ag_j$, $\{ag_i, ag_j\} \in \mathsf{E}$ if and only if (ag_i, ag_j) or (ag_j, ag_i) are in $\mathbf{E}$. The following theorem shows that the fact that cond3 is not satisfied is indicated by the presence of an articulation node ag_i.

Theorem 1 *Let $C = \langle \mathcal{V}, \mathcal{E} \rangle$ be coalition such that $\mathbf{G}[C]$ is strongly connected, if there exists an agent $ag_i \in \mathcal{V}$ and a bipartition $\mathcal{O}_1, \mathcal{O}_2$ of $out(ag_i)$ such that $\mathcal{O}_1^* \cap \mathcal{O}_2^* = \emptyset$, then ag_i is an articulation node of $\mathsf{G}[C]$.*

proof: Assume that there exists an agent $ag_i \in \mathcal{V}$ and a bipartition $\mathcal{O}_1, \mathcal{O}_2$ of $out(ag_i)$ such that $\mathcal{O}_1^* \cap \mathcal{O}_2^* = \emptyset$ and, *per absurdum*, ag_i is not an articulation node.

Strong connectivity of $\mathbf{G}[C]$ assures that (ass1) there exist two agents, say ag_1 and ag_2, such that ag_i, ag_1 and ag_2 are distinct and ag_1 is involved in $\mathcal{O}_1^*$ and ag_2 is involved in $\mathcal{O}_2^*$, and (ass2) $\mathcal{O}_1^* \cup \mathcal{O}_2^* = \mathcal{E}$. Since ag_i is not an articulation node, there exists an undirected path p connecting ag_1 and ag_2 such that ag_i is not a node of p. For (ass2) and condition (1) of Definition 1, each node in the path is an agent in $\mathcal{O}_1^*$ or $\mathcal{O}_2^*$. Thus, starting from ag_1 it is possible to walk through p until an agent ag_h is in $\mathcal{O}_1^*$ and the successor ag_k is in $\mathcal{O}_2^*$. The presence of an undirected arc connecting ag_h to ag_k means that one of them is involved in set out of the other one. Without loss of generality we assume that there exists a set of agents Q and a goal g such that $ag_h \in Q$ and $(ag_k, Q, g) \in \mathcal{E}$. This means that $out(ag_h)$ is contained in both $\mathcal{O}_1^*$ and $\mathcal{O}_2^*$. For strong connectivity of $\mathbf{G}[C]$, we also have that $out^*(ag_h) = \mathcal{E}$ and hence $out(ag_h)$ is not empty, then $\mathcal{O}_1^* \cap \mathcal{O}_2^* \neq \emptyset$ against the hypothesis. $\square$

So, if $\mathsf{G}[C]$ is biconnected (i.e. it does not have articulation points), then it satisfies cond3. However, the inverse implication of Theorem 1 does not hold and cond3 can be satisfied even if $\mathsf{G}[C]$ has an articulation point ag_i. This is due to the fact that the undirected graph $\mathsf{G}[C]$ *breaks* an AND-arc in several undirected arcs, so the biconnected components sharing ag_i may not correspond to any bi-partition of $out(ag_i)$. Figure 3 considers this fact. Figure 3 (b) represents the undirected graph $\mathsf{G}[C]$ of the coalition C in Figure 3 (a). There exist two biconnected components of $\mathsf{G}[C]$, one for each arc, sharing ag_1 as articulation node. However, both arcs $\{ag_1, ag_2\}$ and $\{ag_1, ag_3\}$ correspond to the AND-arc $(ag_1, \{ag_2, ag_3\}, g_1)$, thus they have to be considered as a single component because $\{(ag_1, \{ag_2, ag_3\}, g_1)\}^*$ contains both of them. Being $out(ag_1)$ equal to $\{(ag_1, \{ag_2, ag_3\}, g_1)\}$, there does not exist a bi-partition $\mathcal{O}_1, \mathcal{O}_2$ of $out(ag_1)$ such that $\mathcal{O}_1^* \cap \mathcal{O}_2^* = \emptyset$. Thus, C satisfies cond3.

This property holds in general, when some biconnected components of $\mathsf{G}[C]$ contain some arcs which correspond to the same AND-arc of C, they are considered as a single component. If this grouping process ends with a single component consisting of the whole undirected graph $\mathsf{G}[C]$, then no bi-partitions of $out(ag_1)$ falsify cond3.

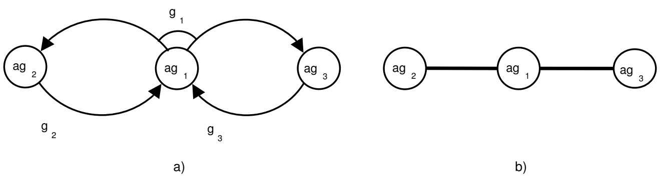

Figure 3. *A labeled AND-graph and the corresponding undirected graph.*

Algorithm 1: FIND-I-DUD

Data: $C = \langle \mathcal{V}, \mathcal{E} \rangle$.
Result: I_DUD, the set of sub-coalitions of C that satisfy the i-dud property

1 $I\_DUD \leftarrow \emptyset$;
2 $NDC =$ NO-DUPL-COMMITMENTS (C);
3 **forall** $C' \in NDC$ **do**
4 $\quad I\_DUD \leftarrow I\_DUD \cup$ FIND-2-3 (C');

3 The algorithm for finding coalitions

In this section we design a procedure FIND-I-DUD (see Algorithm 1) which finds all the sub-coalitions of a coalition C satisfying the i-dud property (C included). We use the reformulation of cond2 and cond3 in Definition 3 in terms of strong connectivity of directed graphs and biconnectivity of undirected graphs. By doing so, we also decompose our problem as much as possible in well known problems in graph theory.

The variable I_DUD in line 1 stores the set of sub-coalitions of C which satisfy the i-dud property. In line 2 NO-DUPL-COMMITMENTS checks in $\mathcal{E}$ the presence of commitments with the same goal but assigned to different sets of agents (cond1) and it returns the set NDC of all combinations C' obtained from C by deleting all the duplicated commitments except one. This way, the sub-coalitions in NDC are the maximal sub-coalitions which satisfy cond1. Since all their sub-coalitions satisfy this condition as well, we do not need to check recursively this condition on them, i.e. NO-DUPL-COMMITMENTS can be run only once.

For each coalition in NDC the procedure FIND-2-3 is run (lines 3-4). FIND-2-3 (Algorithm 2) takes as input a coalition C - which satisfies cond1 - and it returns the set of sub-coalitions of C, C included, that satisfy cond2 and cond3. As we already have checked cond1, we can add the results of FIND-2-3 to the set I_DUD.

The variable S stores the subsets of C that satisfy cond2 and cond3, in line 1 this variable is initialized to the empty set.

In line 2 SC-COMPONENTS calculates the strongly connected components SCC of $\mathbf{G}[C]$ - algorithms for this procedure are well known [5, 9]. Three cases are distinguished.

Case 1: $\mathbf{G}[C]$ has no strongly connected components. Since strong connectivity is a necessary condition for the satisfaction of cond2, no sub-coalitions of C satisfy the i-dud property. Therefore, S is empty.

Case 2: $\mathbf{G}[C]$ is not strongly connected, but there exist some strongly connected components. In this case only the sub-coalitions of C such that the relative directed graphs are subgraphs of a strongly connected component can satisfy cond2. Therefore, in lines 7-10, for each strongly connected component, the maximal labeled AND-graph $\langle \mathcal{V}', \mathcal{E}' \rangle$, *included* in the component, is constructed. The function FIND-2-3 is recursively called on $\langle \mathcal{V}', \mathcal{E}' \rangle$ and its output is added to S.

Algorithm 2: FIND-2-3

Data: $C = \langle \mathcal{V}, \mathcal{E} \rangle$.
Result: S, the set of sub-coalitions of C that satisfy cond2 and cond3.

1 $S \leftarrow \emptyset$;
2 $SCC \leftarrow$ SC-COMPONENTS $(\mathbf{G}[C])$;
3 **switch do**
4 $\quad$ **case** $\mathbf{G}[C]$ *has no strongly connected components*
5 $\quad\quad S \leftarrow \emptyset$;
6 $\quad$ **case** $\mathbf{G}[C]$ *is not strongly connected, but it has some strongly connected components*
7 $\quad\quad$ **forall** $\langle \mathbf{V}, \mathbf{E} \rangle \in SCC$ **do**
8 $\quad\quad\quad \mathcal{V}' \leftarrow \mathbf{V}$;
9 $\quad\quad\quad \mathcal{E}' \leftarrow \{(ag_i, Q, g) \in \mathcal{E} \mid ag_i \in \mathbf{V} \wedge Q \subseteq \mathbf{V}\}$;
10 $\quad\quad\quad S \leftarrow S \cup$ FIND-2-3 $(\langle \mathcal{V}', \mathcal{E}' \rangle)$;
11 $\quad$ **case** $\mathbf{G}[C]$ *is strongly connected*
12 $\quad\quad (BC, A\_NODES) \leftarrow$ BC-COMPONENTS $(\mathsf{G}[C])$;
13 $\quad\quad$ **forall** $(ag_i, Q, g) \in \mathcal{E}$ *s.t.* $ag_i \in A\_NODES$ **do**
14 $\quad\quad\quad BC' \leftarrow \{\langle \mathsf{V}', \mathsf{E}' \rangle \in BC \mid \{ag_i, ag_j\} \in \mathsf{E}'$ with $ag_j \in Q\}$;
15 $\quad\quad\quad BC \leftarrow [BC \setminus BC'] \cup \{\bigcup BC'\}$;
16 $\quad\quad$ **if** $|BC| = 1$ **then**
17 $\quad\quad\quad S \leftarrow \{C\}$;
18 $\quad\quad\quad$ **forall** $(Q, g) \in C$ **do**
19 $\quad\quad\quad\quad C' \leftarrow C \setminus \{(Q, g)\}$;
20 $\quad\quad\quad\quad S \leftarrow S \cup$ FIND-2-3 (C');
21 $\quad\quad$ **else**
22 $\quad\quad\quad$ **forall** $\langle \mathsf{V}, \mathsf{E} \rangle \in BC$ **do**
23 $\quad\quad\quad\quad \mathcal{E}' \leftarrow \{(ag_i, Q, g) \in \mathcal{E} \mid ag_i \in \mathsf{V} \wedge Q \subseteq \mathsf{V}\}$;
24 $\quad\quad\quad\quad \mathcal{V}' \leftarrow \mathsf{V}$;
25 $\quad\quad\quad\quad S \leftarrow S \cup$ FIND-2-3 $(\langle \mathcal{V}', \mathcal{E}' \rangle)$;
26
27 **return** S;

Case 3: $\mathbf{G}[C]$ is strongly connected, therefore, cond2 is satisfied. It remains to check cond3 and for complexity reasons we use the characterization by means of the biconnected components of $\mathsf{G}[C]$ (see Section 2).

In line 12 the set of biconnected components BC and the set of articulation points A_NODES are calculated. In lines 13-15 FIND-2-3 checks, for each articulation node ag_i, if there exists an AND-arc (ag_i, Q, g) such that the other agents in Q are involved in two, or more, biconnected components, then these biconnected components replaced with their union (see Figure 3). In the case we end with a single component, $|BC| = 1$, C satisfies also cond3 and it is added to S. Then, in lines 18-20, the sub-coalitions C' obtained removing a single commitment (Q, g) from C are constructed and FIND-2-3 is recursively called on them. If $|BC| > 1$, then C does not satisfy cond3. Also all the subsets C' of C such that $\mathsf{G}[C']$ is not included in a component of BC cannot satisfy cond3, therefore for each component $\langle \mathsf{V}, \mathsf{E} \rangle$ in BC, the maximal subgraph of C *included* in $\langle \mathsf{V}, \mathsf{E} \rangle$ is selected. FIND-2-3 is recursively called on C' and the output is added to S (lines 22-25). Finally, S is returned, line 27.

4 Complexity of the algorithm

In this section we discuss the complexity of FIND-I-DUD. First of all, we show that the problem of checking if a coalition satisfies the i-dud property is tractable. Algorithms FIND-I-DUD and FIND-2-3 can be easily modified to simply check if a given coalition satisfies the i-dud property. First, in FIND-I-DUD we replace the FOR statement with an IF-THEN-ELSE statement which returns false if C does not satisfy cond1, it calls FIND-2-3 on C, otherwise. In FIND-2-3 we replace lines 5,7-10, 22-25 with an instruction returning false, and lines 17-20 with an instruction returning true. We denote with m the number of agents involved in C and with l the number of arcs in $\mathbf{G}[C]$. The procedure SC-COMPONENTS takes a time proportional to l [5]. In the case $\mathbf{G}[C]$ is not strongly connected, C does not satisfies the i-dud property and FIND-2-3 returns false. In the contrary case, the procedure BC-COMPONENTS is called on the undirected graph $\mathsf{G}[C]$.

Also BC-COMPONENTS can be executed in a time that is proportional to $|\mathsf{E}|$ and, since $|\mathsf{E}| \leq l$, so far Algorithm 2 has a complexity that is proportional to l. We have to consider now the complexity of the cycle corresponding to the lines 13-15. The number of iterations of the cycle 13-15 is less than l. The instruction in line 14 has as upper bound m, assuming that, during the execution of BC-COMPONENTS, a data structure is stored associating each arc with the biconnected component in which it is included. Since the sets of arcs of two distinct biconnected components are disjoint, also the instruction in line 15 can be performed in time proportional to the set of distinct biconnected components found in line 12, which has an upper bound in m. Therefore, the cycle 13-15 has an upper bound in $O(l \cdot m)$. Since $O(l \cdot m)$ is an upper bound also to check cond1, it is an upper bound for the problem to verify if a coalition satisfies the i-dud property.

With respect to the original problem to find all the sub-coalitions of C that satisfy the i-dud property, consider that C satisfies cond1 and it contains only AND-arcs as $(ag_i, \{ag_j\}, g)$. In this case we can represent C as a directed graph $\mathbf{G}[C]$, where each arc (u, v) univocally corresponds to a goal. We show that a single run of FIND-2-3 is not computationally tractable. FIND-2-3 finds all the sub-coalitions of a coalition $\mathbf{G}[C]$ that satisfy cond2 and cond3. This requires to find in particular all the subgraphs of $\mathbf{G}[C]$ that are the strongly connected subgraphs and such that the relative undirected graph, $\mathsf{G}[C]$, is biconnected. Since an hamiltonian cycle in $\mathbf{G}[C]$ - if any exists - satisfies the previous two conditions, FIND-2-3 has to find a set of subgraphs which contains all the hamiltonian cycles of $\mathbf{G}[C]$. Thus, this problem is exponential with respect to the number of arcs l. In contrast, since checking if a subgraph of $\mathbf{G}[C]$ is an hamiltonian cycle is linear with the cardinality of the nodes V, also the problem of finding an hamiltonian cycle would be polynomial with respect to number of arcs.

In the case a coalition C can be represented by the corresponding directed graph $\mathbf{G}[C]$, the set of subgraphs to check is equal to 2^l. However, if C does not satisfy the i-dud property, then, either $\mathbf{G}[C]$ is not strongly connected or $\mathsf{G}[C]$ has more than one component as calculated in the lines 12-15. In both cases FIND-2-3 is called directly on the subgraphs calculated respectively in lines 8-9 and 23-24. So, if there are k of these subgraphs, each of them with l_i arcs, we have that the number of the graph which remain to be verified is $2^{l_1} + \cdots + 2^{l_k}$ instead of (approximately from below) $2^{l_1+\cdots+l_k} - 1$. In the worst case $\mathbf{G}[C]$ is not strongly connected and it has one strongly component with $l-1$ arcs. In this case 2^{l-1} graphs remain to be verified instead of $2^l - 1$.

We note that this fact occurs not only once, but every time a sub-coalition of C does not satisfy the i-dud property. Moreover, if C is a proper AND-graph this phenomenon can be amplified by the fact that when an AND-arc is removed in line 8, it may disconnect a strongly connected component of $\mathbf{G}[C]$.

Returning to the coalition C in Figure 2 (a), the number of AND-arcs is 7, so *a priori* $2^7 = 128$ sub-coalitions should be checked by the algorithm FIND-I-DUD. However, C does not satisfy cond3 and, after BC results to be greater than 1 in line 16 of FIND-2-3, FIND-2-3 is called on the sub-coalitions in Figure 2 (d) and (e). The first sub-coalition has 4 AND-arcs and the second one has just 3 arcs. So after a single call of FIND-2-3, it remains $2^4 + 2^3 = 24$ sub-coalitions instead of 127. It can be shown that the total number of sub-coalitions checked is equal to 16, i.e. only the 12,5% of the number of all sub-coalitions of C.

5 Conclusion

In this work we define a criterion of admissibility for coalition formation which is based on the representation of a coalition as a net of exchanges [4]: the i-dud property. This property refines the do-ut-des property [2] by taking into account the fact that two distinct coalitions cannot be considered a whole coalition if they can be formed independently. This condition arises from the fact that agents prefer to form small coalitions because, as coalitions spring from unanimously agreements, the more are the agents involved in a coalition the more is the risk that one of them gives up joining it.

The i-dud property inherits from the do-ut-des property the fact that it uses only the internal topology of exchanges to check the admissibility of a coalition. Approaches based on Cooperative Game Theory, as [7, 10], abstract from this internal structure, and hence they need to compare a coalition with the other possible coalitions in order to establish its admissibility. This way, also the problem to see if a coalition is admissible, applying for example the notion of core, in intractable.

REFERENCES

[1] R. Alur, T.A. Henzinger, and O. Kupferman, 'Alternating-time temporal logic', *Journal of ACM*, **49**(5), 672–713, (2002).

[2] G. Boella, L. Sauro, and L. van der Torre, 'Admissible agreements among goal-directed agents', in *Procs. of 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)*, Paris, France, (2005).

[3] C. Castelfranchi, 'Founding agents' "autonomy" on dependence theory.', in *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*, pp. 353–357, Berlin, Germany, (2000).

[4] R. Conte and J. S. Sichman, 'Dependence graphs: Dependence within and between groups', *Computational & Mathematical Organization Theory*, **8**(2), 87–112, (2002).

[5] T. Cormen and C. Leiserson R. Rivest, *Introduction to Algorithms*, MIT Press, 1990.

[6] S. Kraus and T. Plotkin, 'Algorithms of distributed task allocation for cooperative agents', *Theoretical Computer Science*, **242**(1-2), 1–27, (2000).

[7] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohm, 'Coalition structure generation with worst case guarantees', *Artificial Intelligence*, **111**(1-2), 209–238, (1999).

[8] L. Sauro, *Formalizing Admissibility Criteria in Coalition Formation Among Goal-directed Agents*, Ph.D. dissertation, 2006.

[9] R. Tarjan, 'Depth-first search and linear graph algorithms', *SIAM Journal of Computation*, **1**(2), 146–160, (1972).

[10] M. Wooldridge and P.E. Dunne, 'On the computational complexity of qualitative coalitional games', *Artificial Intelligence*, **158**(1), 27–73, (2004).