

# Dependency in Cooperative Boolean Games

Luigi Sauro<sup>1</sup> and Leendert van der Torre<sup>2</sup> and Serena Villata<sup>3</sup>

<sup>1</sup> Dipartimento di Scienze Fisiche, University of Naples, Italy. sauro@na.infn.it

<sup>2</sup> CSC, University of Luxembourg. leendert@vandertorre.com

<sup>3</sup> Dipartimento di Informatica, University of Turin, Italy. villata@di.unito.it

**Abstract.** Cooperative boolean games are coalitional games with both goals and costs associated to actions, and dependence networks for boolean games are a kind of social networks representing how the actions of other agents have an influence on the achievement of an agent's goal. In this paper, we introduce two new types of dependence networks, called the *abstract dependence network* and the *refined dependence network*. Moreover, we show that the notion of stability is complete with respect to the solution concept of the *core* in the case of cooperative boolean games with costly actions. We present a reduction, called  $\Delta$ -reduction, to pass from a cooperative boolean game  $G$  to game  $G'$  without losing solutions.

## 1 Introduction

Dunne *et al.* [5] recently introduced a kind of coalitional games called cooperative boolean games, and they characterize the complexity of the solution concepts of core and stable sets. Moreover, Bonzon [3, 2] defines stable sets and uses dependence networks to simplify the computation of the pure Nash equilibrium, showing that the notion of stability is complete with respect to the pure Nash equilibrium for actions without costs. In this paper, we propose a new step to make the computation of the core easier by means of the dependence networks associated with a cooperative boolean game. In particular, we study the following two questions:

1. Is the notion of stability complete also with respect of the solution concept of the core in the case of cooperative boolean games with costly actions?
2. How to pass from a cooperative boolean game  $G$  to game  $G'$  without losing solutions?

We define two kinds of dependence networks and we show how to use them to calculate the core. Abstract dependence networks (ADNs) represent a structure based on the model presented by Bonzon *et al.* [3] without labels on the edges and AND-arcs, and refined dependence networks (RDNs) with AND-arcs labeled by the boolean variables composing the goals. We use the dependence network to decompose games, such as separating a game into two smaller independent games (with disconnected interdependencies) or isolating an agent requiring the execution of a costly action useless for all the other agents. Moreover, we present a number of abstractions that allow to reduce the search space by means of a set of criteria principally based on graphs visit algorithms that are computationally tractable. Moreover, we underline a number of hidden

properties in the notion of core showing how, in certain cases, this notion is too strict and, thus, it can lead to counterintuitive results.

The remainder of the paper is as follows. Section 2 provides an introduction to the framework of cooperative boolean games and their solution concept, the core. Section 3 defines the dependency relations in the cooperative boolean games, defining the two types of dependence networks. Conclusions end the paper.

## 2 Cooperative Boolean games

A Cooperative Boolean Game (CBG) [5] consists of a set of agents  $1, \dots, n$ , a set of goals the agents desire to accomplish, the variables the agents control, and a cost function. Goals are represented by a propositional formula  $\gamma_i$  over some set of Boolean variables  $\Phi$ .

**Definition 1 ([5]).** *A cooperative boolean game  $G$  is a  $(2n+3)$ -tuple  $G = \langle A, \Phi, cost, \gamma_1, \dots, \gamma_n, \Phi_1, \dots, \Phi_n \rangle$  where  $A = \{1, \dots, n\}$  is a set of agents,  $\Phi = \{p, q, \dots\}$  is a finite set of boolean variables,  $cost$  is a cost function defined in  $\Phi \rightarrow \mathbb{R}_+$ ,  $\gamma_1, \dots, \gamma_n$  are the boolean formulas over  $\Phi$  representing the goals of the agents and  $\Phi_1, \dots, \Phi_n$  is a partition of  $\Phi$  over  $n$ , with the intended interpretation that  $\Phi_i$  is the set of Boolean variables under the control of agent  $i$ .*

The simple action theory works as follows, inspired by discrete event systems. For each variable  $p \in \Phi$ , there is one agent who can set its truth value to true or false. Setting a variable  $p \in \Phi$  to be  $\top$  is *performing the action  $p$* , and setting  $p \in \Phi$  to  $\perp$  is *doing nothing*; only in the former case, the costs of the action are taken into account. A set of variables  $\xi \subseteq \Phi$  (or a strategy) stands for a valuation, in the sense that the value of the variables belonging to  $\xi$  is true and the value of the other ones is false, and we write  $\xi \models \phi$  means that  $\phi$  is true under the valuation  $\xi$  under the standard propositional semantics.  $cost_i(\xi)$  denotes the cost to agent  $i$  of valuation  $\xi \subseteq \Phi$ , that is,  $cost_i(\xi) = \sum_{v \in (\xi \cap \Phi_i)} cost(v)$ . Agents minimize its costs only when they have achieved their goal, or cannot achieve their goal. In other words, if the only way an agent can achieve its goal is by making all its variables true, then the agent prefers to do this rather than not achieve its goal. However, if there are different ways to achieve its goal, then the agent prefers to minimize costs. This is represented by a utility function that is always positive if the valuation  $\xi$  satisfies the goal of the agent  $i$  and otherwise it is always negative. If  $\mu$  represents the total cost of all variables, the utility for agent  $i$  of a valuation  $\xi$ ,  $u_i(\xi)$  is defined as follows.

$$u_i(\xi) = \begin{cases} 1 + \mu - cost_i(\xi) & \text{if } \xi \models \gamma_i \\ -cost_i(\xi) & \text{otherwise} \end{cases}$$

The meaning of the fact that coalition  $D$  blocks  $\xi_1$  through  $\xi_2$  is that  $\xi_2$  could do better than  $\xi_1$  only flipping the value of some of the variables under the control of  $D$ . Here a coalition  $D \subseteq A$  is a set of agents without represented relationship,  $\Phi_D$  denotes the set of variables under the control of some member of  $D$ ,  $\Phi_D = \bigcup_{i \in D} \Phi_i$ , and  $\xi_1 = \xi_2 \text{ mod } D$  means that valuation  $\xi_2$  is the same as a valuation  $\xi_1$  except at most in the value of variables controlled by  $D$ ,  $\xi_1 \succeq_i \xi_2$  iff  $u_i(\xi_1) \geq u_i(\xi_2)$ .

**Definition 2 (Blocked Valuation [5]).** A valuation  $\xi_1$  is blocked by a coalition  $D \subseteq A$  through a valuation  $\xi_2$  if and only if: (1)  $\xi_2$  is a feasible objection by coalition  $D$  which means that  $\xi_2 = \xi_1 \text{ mod } D$ ; (2)  $D$  strictly prefers  $\xi_2$  over  $\xi_1$ :  $\forall i \in D : \xi_2 \succ_i \xi_1$ .

The core is a fundamental concept in coalitional game theory. A valuation is in the core if and only if no coalition has an incentive to defect.

**Definition 3 (Core).** Given a CBG  $G$ ,  $\xi \in \text{core}(G)$  iff it is not blocked by any coalition.

The following example reproduces the classical schema of the Prisoner Dilemma, and illustrates that Definition 3 is a strengthening of the well-known Nash equilibrium in non-cooperative game theory [8], usually called Strong Nash Equilibrium (SNE). This form of solution satisfies at the same time some requirements of both the cooperative and non-cooperative game theory. From the solution criteria developed in cooperative game theory, it *borrow*s efficiency, a strategy cannot be a solution if the agents or a part of them can obtain better results. From non-cooperative game theory, it assumes that agents are suspicious and that agreements cannot be enforced, that is at any time agents may betray agreements.

*Example 1.* Let  $G$  be a CBG involving two agents.  $\Phi_1 = \{a\}$  and  $\Phi_2 = \{b\}$ ,  $\gamma_1 = \gamma_2 = a \vee b$ ,  $\text{cost}(a) = \text{cost}(b) = 1$ . The CBG can be represented by the following payoff matrix:

$a/b$	1	0
1	(1, 1)	(-1, 2)
0	(2, -1)	(0, 0)

(0, 0) is the only Nash equilibrium in this game, that is not efficient as the two agents collaborating can obtain better results, (1, 1), and hence it is not a SNE. On the other hand, as (1, 1) is not a Nash equilibrium, it is not a SNE too. Indeed, if (1, 1) is proposed as an agreement each agent would unilaterally betray it obtaining better results.

Note that whether SNE corresponds to the notion of core depends on how a strategic game is translated into a cooperative game (see [9]). For example, if we use for the strategic game in Example 1 the minimax representation of a corresponding cooperative game - as introduced by von Neumann and Morgensten - then the strategy  $\{1, 1\}$  is in the core. This kind of representation is called *offensive* because it is assumed that for a certain coalition  $D$ , the agents in  $A \setminus D$  works in order to minimize the outcomes of  $D$ . On the contrary if we adopt a *defensive* representation where the primary aim of the agents is to maximize their own utility, the SNE represents the core of the corresponding cooperative game. The following example illustrates a number of cases in which, in contrast to Example 1, the core is not empty.

*Example 2.* Consider a game where we have four agents  $A = \{1, 2, 3, 4\}$  who want to go in holidays to the seaside or to the mountains. We represent with the boolean variable  $a$  to go to the seaside and with  $b$  to go to the mountains. Agent  $i$  going to the seaside is represented by setting  $a_i$  to true whereas the holiday to the mountains is represented by  $b_i$ . For each agent  $i$ ,  $\text{cost}(a_i) = 2$  and  $\text{cost}(b_i) = 1$ . Agent 1 is in love with agent 2 and he wants to go everywhere with her, thus its goal is represented by

$\gamma_1 = (a_1 \wedge a_2) \vee (b_1 \wedge b_2)$ . Agent 2 is in love with agent 1 but she cannot tolerate the change of temperature of the mountains thus her goal is  $\gamma_2 = a_1 \wedge a_2$ . Agent 3 is in love with 2 and, as he is jealous of agent 1, he would like to stay with 2 without the presence of 1  $\gamma_3 = (a_2 \wedge a_3 \wedge \neg a_1) \vee (b_2 \wedge b_3 \wedge \neg b_1)$ . Agent 4 is in love with agent 3, who does not like him, but she is not able to swim thus its goal is  $\gamma_4 = b_3 \wedge b_4$ . Let us say that agent  $i$  is satisfied given a valuation  $\xi$  if  $\xi \models \gamma_i$ , i.e., if  $i$ 's goal is satisfied. It can be verified that  $\{a_1, a_2\}$  is in the core.

### 3 Dependency Relations in CBG

Dependence networks have been developed by Conte and Sichman [12] as a kind of social network representing how each agent depends on other agents to achieve the goals he cannot achieve himself. The notion of agent dependence used to define dependence networks is related to the concept of social power, introduced by Castelfranchi [4]. Sauro [11, 1] shows how to use dependence networks to discriminate among different potential coalitions during the coalition formation process. The authors develop a criterion of admissibility called *do-ut-des property* describing a condition of reciprocity. Moreover, they define another criterion, called the indecomposable do-ut-des property, establishing which coalitions cannot be formed under the assumption that agents are self-interested. These two criteria have only a qualitative connotation and thus, they cannot be directly applied to the solutions developed in game theory. Moreover, goals are not structured and they do not represent explicitly costs of the actions.

The first attempts of use of the dependence networks to represent and simplify the computation of the solution concepts for boolean games are given by Bonzon [2] and Bonzon et al. [3]. Representing these dependencies on a graph, they show how to compute pure-strategy Nash equilibria without enumerating all combinations of strategies. This work does not consider costly actions and dependence networks are simple graph without labeled edges.

In this section we present two types of dependence networks defined starting from a CBG. These networks, on the one hand, explicitly represent the inter-dependencies among agents according to their goals, on the other hand, they abstract from the quantitative aspects of a game associated to the cost function. Abstract Dependence Networks describe only which agents can play a role in the satisfaction of an agent's goal, therefore they abstract from how they can contribute and in particular if they have to execute a costly action. As this information may help in the study of the core and reduces the search space, we define another type of dependence network, the Refined Dependence Networks, which, as the name suggests, refine Abstract Dependence Networks by representing how agents contribute to the satisfaction of a goal and whether this involves a positive cost (without quantifying it). Starting from the Refined Dependence Networks, we define a method called  $\Delta$  - *reduction* to reduce the admissible strategies and we prove the completeness of the  $\Delta$  - *reduction* with respect to the computation of the core.

### 3.1 Abstract Dependence Networks

As in Bonzon et al. [3], in order to correctly establish the dependencies among agents we need to define which variables are relevant for the satisfaction of the agents' goal. A variable  $p$  is said irrelevant for a formula  $\phi$  in case there exists an equivalent formula  $\phi'$  where  $p$  does not occur. With  $RV_G(i)$  we represent the set of all variables  $p \in \Phi$  that are relevant for  $\gamma_i$ , whereas  $RA_G(i)$  is the set of agents  $j \in A$  such that  $j$  controls at least one relevant variable of  $i$ . Using the notion of relevant agents, we define a dependence network where nodes represent agents and an edge from  $i$  to  $j$  represents the dependence of  $i$  from  $j$  ( $j \in RA_G(i)$ ).

**Definition 4 (Abstract Dependence Network).**

Given the CBG  $G = \langle A, \Phi, cost, \gamma_1, \dots, \gamma_n, \Phi_1, \dots, \Phi_n \rangle$ , the abstract dependence network of  $G$  is the directed graph  $ADN(G) = \langle N, R \rangle$  such that the set of nodes  $N$  corresponds to the agents in  $G$ ,  $N = A$ , and  $(i, j) \in R$  iff  $j \in RA_G(i)$ .

As in [3] we say that a set of agents in  $ADN(G)$  is stable in case it is closed under the relation  $R$ .  $R(C)$  is the set of players from which  $C$  may need some action in order to be satisfied.

**Definition 5 (Stable set).** Given a directed graph  $\langle N, R \rangle$ ,  $C \subseteq N$  is stable iff  $R(C) \subseteq C$ , i.e. for all  $i \in C$ , for all  $j$  such that  $(i, j) \in R$   $j \in C$ .

Note that the notion of stable set is not related to the strategic criterion of the same name originally introduced by von Neumann and Morgenstern [10].

**Definition 6 (ADN Projection).**

Let  $G = \langle A, \Phi, cost, \gamma_1, \dots, \gamma_n, \Phi_1, \dots, \Phi_n \rangle$  be a CBG,  $ADN(G) = \langle N, R \rangle$  the corresponding abstract dependence graph and  $C = \{i_1, \dots, i_m\} \subseteq N$  a stable set, the projection of  $G$  on  $C$  is defined by  $G_C = \langle C, \Phi_C, cost_C, \gamma_{i_1}, \dots, \gamma_{i_m}, \Phi_{i_1}, \dots, \Phi_{i_m} \rangle$ , where  $cost_C : \Phi_C \rightarrow \mathbb{R}^+$  is the restriction of  $cost$  on  $\Phi_C$ .

As shown in Bonzon et al. [3] the projection of a CBG on a stable set is itself a CBG.

**Proposition 1.** Given a CBG  $G$ , if  $C$  is a stable set,  $G_C$  is a cooperative boolean game.

In Bonzon et al. [3] the authors show that by restricting a boolean game  $G$  to the projection  $G_C$  of a stable set  $C$ , if a strategy profile  $\xi_C$  in  $G_C$  is not a Nash equilibrium, then all of its extensions in  $G$  are not a Nash equilibrium. Here we extend this result to the case of the core in CBG with costly actions.

**Proposition 2.** Given a stable set  $C$ , if  $\xi$  is in  $core(G)$ , then  $\xi_C$  is in  $core(G_C)$ , where  $\xi_C$  is the projection of  $\xi$  on the variables controlled by  $C$ ,  $\xi_C = \xi \cap \Phi_C$ .

*Proof.* Let  $\bar{\xi}$  a generic valuation in  $G$  and  $\bar{\xi}_C$  the projection of  $\bar{\xi}$  on the variables controlled by  $C$ . Clearly, for all  $i \in C$ ,  $cost_i(\bar{\xi}) = cost_i(\bar{\xi}_C)$ . Furthermore, as  $C$  is stable, for all  $i \in C$ ,  $RV_G(i) = RV_{G_C}(i)$  and hence  $\bar{\xi} \models \gamma_i$  iff  $\bar{\xi}_C \models \gamma_i$ . This entails that for two generic valuations  $\bar{\xi}$  and  $\hat{\xi}$  and for all  $i \in C$ ,  $u_i(\bar{\xi}) \leq u_i(\hat{\xi})$  iff  $u_i(\bar{\xi}_C) \leq u_i(\hat{\xi}_C)$ .

Assume that there exists a valuation  $\xi'_C$  that blocks  $\xi_C$ , i.e. there exists a  $C' \subseteq C$  such that  $\xi'_C = \xi_C \text{ mod } C'$  and for all  $i \in C'$   $\xi'_C \succ_i \xi_C$ . Now let  $\xi_{-C} = \xi \cap \Phi_{A \setminus C}$  and  $\xi' = \xi'_C \cup \xi_{-C}$ . Clearly,  $\xi' = \xi \text{ mod } C'$  and, since for all  $i \in C$   $u_i(\xi) \leq u_i(\xi')$  iff  $u_i(\xi_C) \leq u_i(\xi'_C)$ ,  $\xi$  is blocked by  $C'$  through  $\xi'$ .

Finally, we define *consistency* such that given two coalitions  $D_1$  and  $D_2$ , we say that two relative strategies  $\xi_{D_1}$  and  $\xi_{D_2}$  are *consistent* if and only if for each agent  $i \in D_1 \cap D_2$ ,  $\Phi_i \cap \xi_{D_1} = \Phi_i \cap \xi_{D_2}$ . The following two propositions hold - the proof is straightforward and it is left to the reader.

**Proposition 3.** *Let  $\xi$  be a strategy blocked by a coalition  $D$  through  $\xi'$  and  $C$  be a stable set such that  $C' = C \cap D \neq \emptyset$ , then  $\xi$  is blocked by  $C'$  through  $\xi'_C$ .*

**Proposition 4.** *Given the stable sets  $C_1, \dots, C_n$  and relative strategies  $\xi_{C_1}, \dots, \xi_{C_n}$ , if*

1. *for all  $1 \leq i \leq n$ ,  $\xi_i \in \text{core}(G_{C_i})$ ;*
2.  *$\xi_{C_1}, \dots, \xi_{C_n}$  are consistent;*
3.  *$A = \bigcup_{i=1}^n C_i$ ,*

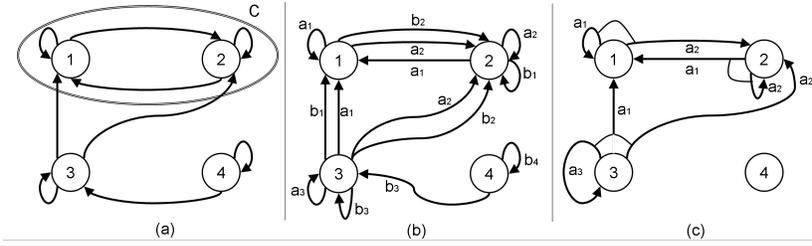
*then  $\bigcup_{i=1}^n \xi_i \in \text{core}(G)$ .*

Propositions 2 and 4 provide a way to decompose the problem of determining the core of a CBG  $G$  into the subgames  $G_{C_i}$ , where  $C_1, \dots, C_n$  are stable sets that involve all the agents in  $G$ . Then, once each  $\text{core}(G_{C_i})$  is determined, it remains to gather any union  $\xi$  of consistent strategies  $\xi_{C_i} \in \text{core}(G_{C_i})$ , with  $1 \leq i \leq n$ . Due to Proposition 4  $\xi \in \text{core}(G)$ , whereas, Proposition 2 ensures that in this way we find all the elements in  $\text{core}(G)$ , indeed if  $\xi' \in \text{core}(G)$ , then  $\xi'_{C_i} \in \text{core}(G_{C_i})$ , with  $1 \leq i \leq n$  and hence  $\xi'$  is the union of consistent solutions in each  $G_{C_i}$ .

*Example 3 (Continued).* The abstract dependence network of the game in Example 2 is  $ADN = \langle N, E \rangle$ , where  $N = \{1, 2, 3, 4\}$  and  $R = \{(1, 2), (2, 1), (3, 1), (3, 2), (4, 3)\}$ . This  $ADN$  is depicted in Figure 1-(a) where the edges represent the dependence of the first agent on a second one for a boolean variable composing its goal and the circle represents the stable set  $C$ . Let us consider agents 1 and 2 (see Figure 1-(a)), following Definition 5 they represent a stable set (all the edges that go out from agent 1 enter in agent 2 and converse), so we can consider first the projection of the game on them (Definition 6). We can represent valuations as  $uvyz \in \{0, 1\}^4$ , where  $u$  represents the value of  $a_1$ ,  $v$  that of  $a_2$ ,  $y$  the value of  $b_1$  and  $z$  that of  $b_2$ . Following Definition 6, it can be found that 1100 is the only strategy in the core. Due to Propositions 2, also in the complete game all the strategies containing either  $b_1$  or  $b_2$  and not containing one of  $a_1$  and  $a_2$  are blocked.

### 3.2 Refined Dependence Networks

As seen before, by using stable sets Abstract Dependence Networks can be safely used to split the original problem in subproblems without losing solutions. However, Abstract Dependence Networks may hide some useful information that can also be used to prune some strategies that cannot belong to the core - and hence to reduce the search space. For this reason we define another notion of dependence network at a lower level of abstraction and we call it Refined Dependence Network (RDN). These networks



**Fig. 1.** a)- ADN of Example 3 with the stable set C, b)- RDN of Example 6, c)- RDN of Example 6 after the  $\Delta$ -reduction

may seem actually equivalent to the boolean game itself, except for the cost of variables. These costs, however, are not a minor point since two boolean games that result in the same RDN can have different solutions.

A Refined Dependence Network represents how the goals can be satisfied by means of AND-arcs among the agents whose single edges are labeled with literals. Furthermore, costly actions are *marked* in a set  $\Delta$ .

**Definition 7 (Refined Dependence Network).**

A Refined Dependence Network is an AND-graph  $\langle N, \Phi, \Delta, E, \Phi_1, \dots, \Phi_n \rangle$  where  $N$  is the set of nodes,  $\Phi$  is the set of boolean variables,  $\Delta \subseteq \Phi$  is the subset of costly variables<sup>1</sup>,  $E \subseteq N \times 2^{(N \times \text{Litt}(\Phi))}$  where  $\text{Litt}(\Phi) = \Phi \cup \{\neg p \mid p \in \Phi\}$  and  $\Phi_i \subseteq \Phi$  where  $n$  is the cardinality of  $N$ .

In the following, given a literal  $l$ , we denote by  $|l|$  the corresponding boolean variable, that is if  $l \in \Phi$ , then  $|l| = l$  whereas if  $l = \neg p$ ,  $|l| = p$ . Furthermore, to simplify the formalism, we represent an AND-arc  $(i, \{(j_1, l_1), \dots, (j_m, l_m)\})$  as the set of triples  $\{(i, j_1, l_1), \dots, (i, j_m, l_m)\}$ . Of course, a set as  $\{(1, 3, p), (3, 4, q)\}$  has no meaning in our context.

As already done for Abstract Dependence Networks, we use Refined Dependence Networks to reveal the structure of interdependencies among agents. First, we assume that the goals of the agents do not contain irrelevant variables and are given in disjunctive normal form, i.e.  $\gamma_i = \gamma_{i_1} \vee \dots \vee \gamma_{i_m}$  where each  $\gamma_{i_j}$  is a conjunction of literals. Note that the deletion of these irrelevant variables is coNP-complete. We expect that a goal does not contain irrelevant variables, alternatively it is possible to consider the whole set of variables occurring in a formula, avoiding in this way this computational cost. To simplify again the formalism we describe respectively  $\gamma_i$  as a set of  $\gamma_{i_j}$  and each  $\gamma_{i_j}$  as a set of literals - the empty set has the usual meaning respectively of  $\perp$  referred to  $\gamma_i$  and  $\top$  referred to the  $\gamma_{i_j}$ . Roughly, each AND-arc  $e$  outgoing the agent  $i$  corresponds to a  $\gamma_{i_j} \in \gamma_i$ , where each single edge that composes  $e$  is labeled with a literal occurring in  $\gamma_{i_j}$  and reaches the agent that controls the corresponding variable. The set  $L$  consists of the actions that have a strictly positive cost.

<sup>1</sup> The set of variables with an associated cost.

**Definition 8 (From CBGs to RDNs).**

Given the CBG  $G = \langle A, \Phi, cost, \gamma_1, \dots, \gamma_n, \Phi_1, \dots, \Phi_n \rangle$ , the corresponding Refined Dependence Network  $RDN(G) = \langle N, \Phi, \Delta, E, \Phi_1, \dots, \Phi_n \rangle$  is such that  $N = A$ ,  $\Delta = \{p \in \phi \mid cost(p) > 0\}$  and  $\{(i, j_1, l_1), \dots, (i, j_m, l_m)\} \in E$  iff  $\{l_1, \dots, l_m\} \in \gamma_i$  and for all  $1 \leq h \leq m$ ,  $l_h \in \Phi_{j_h}$ .

*Example 4.* Let  $G$  be a cooperative boolean game defined by  $A = \{1, 2, 3, 4\}$ ,  $\Phi = \{a, b, c, d, e\}$ ,  $cost(a) = cost(b) = cost(c) = cost(d) = cost(e) = 1$ ,  $\gamma_1 = a$ ,  $\gamma_2 = c \wedge e$ ,  $\gamma_3 = b \wedge c$ ,  $\gamma_4 = d$ ,  $\Phi_1 = \{b, e\}$ ,  $\Phi_2 = \{d\}$ ,  $\Phi_3 = \{a\}$ ,  $\Phi_4 = \{c\}$ . The associated refined dependence network  $RDN_G = \langle A, \Phi, E, \Phi_1, \dots, \Phi_n \rangle$ , where  $E$  is composed by the following dependencies:  $\{(1, 3, a)\}$ ,  $\{(3, 1, b)$ ,  $(3, 4, c)\}$ ,  $\{(2, 1, e), (2, 4, c)\}$ ,  $\{(4, 2, d)\}$ .

Given a Refined Dependence Network  $RDN(G) = \langle N, \Phi, \Delta, E \rangle$ , we mean with  $R_E \subseteq N \times N$  the binary relation such that  $(i, j) \in R_E$  just in the case there exists an AND-arc  $e \in E$  that starts from  $i$  and reaches  $j$ , i.e. for some literal  $l$ ,  $(i, j, l) \in e$ . It is easy to see that  $ADN(G) = \langle N, R_E \rangle$  and hence  $RDN(G)$  describes  $G$  at a lower level of abstraction with respect to  $ADN(G)$ .

We want to use Redefined Dependence Networks to impose some constraints to the set  $core(G)$ . To this scope some preliminary results are needed. A boolean variable  $a \in \Phi_i$  is said to be *unfavourable* if and only if  $a \in \Delta$ , i.e.  $cost(a) > 0$ , and for each  $\{l_1, \dots, l_m\} \in \gamma_i$ ,  $a \notin \{l_1, \dots, l_m\}$ . In the following we denote by  $[i]^-$  the set of unfavourable variables of the agent  $i$ .

**Proposition 5.** *Given a cooperative boolean game  $G$  and an agent  $i \in A$ , for each  $a \in [i]^-$ ,  $\xi \in core(G)$  implies  $a \notin \xi$ .*

*Proof.*  $a \in [i]^-$  means that  $cost(a) > 0$  and  $a$  does not occur (positive) in  $\gamma_i$ . Assume that  $\xi \models \gamma_i$ , then, for some  $\{l_1, \dots, l_m\} \in \gamma_i$ ,  $\xi \models \{l_1, \dots, l_m\}$ . Since  $a \notin \{l_1, \dots, l_m\}$ , this means that also  $\xi \setminus \{a\} \models \{l_1, \dots, l_m\}$ .

Now it remains to show that if for each  $\xi$ ,  $\xi \models \gamma_i$  implies  $\xi \setminus \{a\} \models \gamma_i$ , then  $\xi \in core(G)$  implies  $a \notin \xi$ . Assume that  $a \in \xi$ , clearly  $\xi \setminus \{a\} = \xi \text{ mod } \{i\}$  as  $a \in \Phi_i$ . Furthermore, as  $c_i(\xi \setminus \{a\}) < cost_i(\xi)$  and by hypothesis  $\xi \models \gamma_i$  implies  $\xi \setminus \{a\} \models \gamma_i$ , then  $\xi \prec_i \xi \setminus \{a\}$ . But this means that  $\xi$  is blocked by  $i$  through  $\xi \setminus \{a\}$ .

Note that according to Proposition 5 it can be easily seen in Example 4 that, as all the variables are unfavourable, the core can contain only the empty strategy. We also prove that a goal depending on an unfavourable variable  $a$  can be reduced into one that do not depend on  $a$  without affecting the possible solutions. More precisely, we define the notion of reduction as follows.

**Definition 9.** *Given a cooperative boolean game  $G$  and an unfavourable variable  $a \in [i]^-$ , we say that the cooperative boolean game  $G'$  is a  $\Delta$ -reduction of  $G$  just in the case it is obtained from  $G$  applying the following steps:*

1. remove  $a$  from  $\Phi_i$ ;
2. remove from each  $\gamma_i$  any conjunction of type  $\{l_1, \dots, a, \dots, l_n\}$ ;
3. replace in each  $\gamma_i$  any conjunction of type  $\{l_1, \dots, \neg a, \dots, l_n\}$  with  $\{l_1, \dots, l_n\}$ .

**Proposition 6.** Let  $G'$  be a  $\Delta$ -reduction of a CBG  $G$ ,  $core(G) \subseteq core(G')$ .

*Proof.* For each valuation  $\xi$  that does not contain the unfavourable variable  $a$ , it clearly holds for each agent  $i$  that  $u_i(\xi)$  in  $G$  is equal to  $u_i(\xi)$  in  $G'$ . Therefore, if in  $G'$ ,  $\xi'_1$  is blocked by  $C$  through  $\xi'_2$ , then the same holds in  $G$  and hence  $core(G) \subseteq core(G')$ .

Note that the converse does not hold. Consider Example 5:

*Example 5.* Let  $G$  be a cooperative boolean game composed by 3 agents and such that  $\Phi_1 = \{a\}$ ,  $\Phi_2 = \{b, c\}$  and  $\Phi_3 = \{d\}$ ,  $\gamma_1 = b$ ,  $\gamma_2 = a \vee (c \wedge d)$  and  $\gamma_3 = c \wedge b$ ,  $cost(a) = cost(b) = 1$  and  $cost(c) = cost(d) = 2$ . The boolean variable  $a$  is an unfavourable variable then the  $\Delta$ -reduced game  $G'$  is such that  $\Phi_1 = \emptyset$ ,  $\Phi_2 = \{b, c\}$  and  $\Phi_3 = \{d\}$ ,  $\gamma_1 = b$ ,  $\gamma_2 = c \wedge d$  and  $\gamma_3 = c \wedge b$ . The function  $cost$  is the same as in  $G$ . It is easy to see that  $\{c, d\} \in core(G')$  whereas in  $G$  it is blocked by  $\{1, 2\}$  through  $\{a, b\}$ .

Note that the previous results do not use *quantitative* values of the cost function but only the fact that an action has a strictly positive value, therefore they reside in the level of abstraction of RDNs. We can now define a procedure on  $RDN(G)$  which uses unfavourable variables and  $\Delta$ -reductions to reduce the search space in finding the core.

**Definition 10 (Reduction rule).** Let  $RDN(G)$  be a Refined Dependence Network and let  $\omega$  denoting a boolean formula initially set to  $\top$ , the reduction rule  $RDN(G)$  is given by applying exhaustively the following rule:

**Condition:** for some  $a \in \Phi_i \cap \Delta$ , there does not exist an AND-arc  $e$  outgoing from  $i$  such that  $(i, i, a) \in e$  (i.e.  $a$  is unfavourable).

**Action:** remove any and-arc  $e'$  such that  $(j, i, a) \in e'$ ,  $a$  from  $\Phi_i$  and update  $\omega$  with  $\omega \wedge \neg a$ .

Due to Propositions 5 and 6, the boolean formula  $\omega$  we obtain from definition 10 constraints the strategies that can be in the core of  $G$ , that is  $\xi \in core(G)$  implies  $\xi \models \omega$ .

*Example 6 (Continue).* Let us consider again the cooperative boolean game of Example 2. In the corresponding RDN all the actions involve some positive cost, thus  $\Delta = \Phi$ . The AND-arcs are shown in Figure 1-(b) (without connections for simplicity of the figure) and all the edges are labeled with the boolean variable they represent. By exhaustively applying the rule in the Definition 10,  $b_2$  satisfies the given condition, therefore we remove the AND-arc  $\{(1, 2, b_2), (1, 1, b_1)\}$  and the AND-arc  $\{(3, 3, b_3), (3, 2, b_2), (3, 1, \neg b_1)\}$ . After these deletions, also  $b_1$  and  $b_3$  satisfy the condition in Definition 10, and hence also the AND-arc  $\{(4, 3, b_3), (4, 4, b_4)\}$  has to be removed. Finally, also  $b_4$  satisfies the condition, therefore as output  $\omega$  is equal to  $\neg b_1 \wedge \neg b_2 \wedge \neg b_3 \wedge \neg b_4 \wedge \neg a_4$ , therefore the only strategies that can be in the core are the subsets of  $\{a_1, a_2\}$ . The RDN of Example 6 after the application of the  $\Delta$ -reduction is depicted in Figure 1-(c).

## 4 Conclusion

In this paper we present a new approach to cooperative boolean games [5] based on dependence networks [12]. Differently from Bonzon et al. [3], we use dependence networks to reduce the search space thanks to the application of graphs' visit algorithms and to argue on the notion of core, showing a number of hidden properties of this solution concept. Moreover, we define two different kinds of dependence networks, abstract and refined dependence networks in which, differently from [3], we introduce costly actions, labeled edges and ADN-arcs. Finally, we present the  $\Delta$ -reduction that allows to reduce the search space to find the strategies in the core without losing solutions.

Concerning future work, we can address our methodology and results to the other solution concepts, for example instead of Strong Nash equilibrium we could represent the core with a less restrictive notion of stability such as the Coalitional-proof Nash equilibrium. In particular, it has been shown that if a game has only one Nash Equilibrium, then it is also a Coalitional-proof Nash equilibrium. Moreover, we are working on an algorithm FIND\_CORE to search the strategies in the core using the  $\Delta$  - *reduction* without losing solutions.

Finally, we aim at evaluate the implemented algorithm in concrete domains - such as the Grid - which can be represented as exchange networks [11]. Several game theoretical approaches used to model such domains require numerical information about private utilities or the degrees of collaboration, however such information is generally not available. Conversely, Cooperative Boolean Games can be set up on the base of known information: the goals that the agents request when they join an exchange market and the costs of the resources they offer.

## References

1. G. Boella, L. Sauro, and L. van der Torre. Strengthening admissible coalitions. In *ECAI*, pages 195–199, 2006.
2. E. Bonzon. *Modelisation des interactions entre agents rationnels: les jeux booléens*. PhD thesis, Université Paul Sabatier, Toulouse, 2007.
3. E. Bonzon, M.-C. Lagasque-Schiek, and J. Lang. Dependencies between players in boolean games. In *ECSQARU*, pages 743–754, 2007.
4. C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.
5. P. E. Dunne, W. van der Hoek, S. Kraus, and M. Wooldridge. Cooperative boolean games. In *AAMAS*, pages 1015–1022, 2008.
6. P. Harrenstein. *Logic in Conflict*. PhD thesis, Utrecht University, 2004.
7. P. Harrenstein, W. van der Hoek, J.-J. Meyer, and C. Witteveen. Boolean games. In *TARK*, pages 287–298, 2001.
8. H. Keiding and B. Peleg. Representation of effectivity functions in coalition proof nash equilibrium: A complete characterization. Technical report, University of Copenhagen, 1999.
9. R. B. Myerson. *Game Theory*. Harvard University Press, 1997.
10. J. Neumann and O. Morgenstern. *Theory of games and economic behaviour*. NJ, 1944.
11. L. Sauro. *Formalizing admissibility criteria in coalition formation among goal directed agents*. PhD thesis, University of Turin, 2005.
12. J. S. Sichman and R. Conte. Multi-agent dependence by dependence graphs. In *AAMAS*, pages 483–490, 2002.