An Extended Multi-agent Coalitions Mechanism with Constraints

Souhila Arib[1] and Samir Aknine[2]
[1]*Laboratoire Quartz, EISTI, France*
[2]*Laboratoire LIRIS, Université Claude Bernard, Lyon 1, France*

Keywords: Multi-agent Systems, Coalition Formation, Coordination, Negotiation.

Abstract: In multiple realistic scenarios, limited agent capabilities may negatively affect task performance. To overcome this, multi-agent cooperation may be required. Many studies have focused on cooperative task performance. To facilitate such cooperation, we develop and evaluate a coalition mechanism that enables agents to participate in concurrent tasks achievement in competitive situations, in which agents have several constraints. We consider a set of self-interested bounded-rational agents, each of which has a set of tasks, that leads an agent to achieving its goal. The agents have not a priori knowledge about the preferences of their opponents. All the agents have their specific constraints and this information is private. In this paper, we do not deal with the negotiation protocol but just introduce a new coalition formation mechanism ($\mathcal{CFM}$) that imposes minimal sharing of private information to ease negotiations. Specifically, we only require that agents share preferences over their constraints. The agents negotiate for coalition formation over these constraints, that may be relaxed during negotiations. They start by exchanging their constraints and making proposals, which represent their acceptable solutions, until either an agreement is reached, or the negotiation terminates. We explore two techniques that ease the search of suitable coalitions: we use a constraint-based model and a heuristic search method. We describe a procedure that transforms these constraints into a structured graph on which the agents rely during their negotiations to generate a graph of feasible coalitions. This graph is therefore explored by a Nested Monte-Carlo search algorithm to generate the best coalitions and to minimize the negotiation time.

1 INTRODUCTION

Forming coalitions of agents which are able to effectively perform tasks is a key issue for many practical applications. This paper mainly focuses on self-interested agents which aim to form coalitions with other agents as they cannot reach their objectives individually. Several methods have been developed to control the behaviors of the agents involved in such process (Shehory and Kraus, 1998). However few mechanisms cope with the dynamic of the constraints of the agents in such contexts. Indeed, these constraints can gradually be revealed, and relaxed by the agents at different moments of the negotiation in order to meet the requirements of their opponents and thus to ease the convergence. Some coalition methods have been developed to determine the optimal coalitions and take into account the constraints of the agents involved in the coalition process. These methods have addressed important issues such as computational complexity and heuristics approaches for the optimal coalition structure generation, (Rahwan et al., 2011), (Voice et al., 2012) , (Ramchurn et al., 2010). In this paper, we present an extended version of the mechanism presented in (Arib et al., 2015). We focus on contexts where agents neither have the same utility functions, nor they reveal these functions. Thus, it is infeasible to precisely estimate a priori the corresponding utility of each agent for each feasible proposal of solution with current optimal coalitions search algorithms. The issues with processing the constraints of the agents in the negotiation phase for the coalition formation deserve a particular attention and a deep study. Yet, only few works propose a mechanism to deal with the dynamic of such constraints while agents negotiate them. Note that, since we consider that the agents are self-interested and do not share their information and computations, our aim is not to identify the optimal solution of the coalitions, but to ease the convergence to an agreed common solution for these agents. Our main contribution is not to deal with the negotiation protocol itself but propose a new mechanism that enables agents to negotiate and form coalitions. This mechanism is

Arib, S. and Aknine, S.
An Extended Multi-agent Coalitions Mechanism with Constraints.
DOI: 10.5220/0008969001990207
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 1*, pages 199-207
ISBN: 978-989-758-395-7; ISSN: 2184-433X

based on three main abstractions: a constraint graph, a coalition graph and a Nested Monte-Carlo search method. First, we develop a constraints based graph which handles the revealed constraints of the agents. This graph of constraints can be used to specify different types of constraints relations, such as a constraints ordering over potential decision outcomes. Building upon this, we transform this representation into a flat representation of coalitions in the graph of coalitions. Each level of this graph allows generating a set of possible coalitions and in this set the agent selects the best coalitions that can be accepted. This graphical representation of constraints and coalitions specifies constraints relations in a relatively compact, intuitive, and structured manner. In the next sections, we first define the problem and link it to other existing problems, so that approximate solution techniques and anytime heuristics that provide increasingly better solutions if given more time can be reused. We advise new solutions that allow agents use a nested Monte-Carlo search algorithm (Cazenave, 2009) which finds the best coalitions that maximize the utility of each agent. Nested Monte-Carlo search methods address the problem of guiding the search towards better states when there is no available heuristic. These methods use nested levels of random games in order to guide the search of coalitions. These algorithms have been studied theoretically on simple abstract problems and applied successfully to several games (Gelly and Silver, 2007). Specifically, this paper advances the state of the art in the following ways. We advise new anytime heuristics to find approximate solutions fast, we empirically evaluate our algorithm and show that it computes (in less than 600 milliseconds) 689 proposals of solutions for non-trivial problems involving up to 30 agents and 50 tasks. Thus, our work encompasses essential aspects of the coalition formation, from the coalition model, negotiation, and an anytime heuristic. The reminder of the paper is organized as follows. Section 2 briefly describes the related works. Section 3 introduces some preliminaries and the case study. Section 4 presents the coalition formation mechanism, and a final section will conclude the work with a summary of the contributions.

2 RELATED WORK

In game-theoretic perspective, coalitional games with constraints have been addressed by a number of works. However, none of these mechanisms is able to model agents' negotiations for reaching joint agreements. The authors in (Yang et al., 2016) addressed the problem of coalition formation in small cell networks. Small cells can mitigate the co-tier interference within a coalition and thus increase the system capacity. (Bistaffa et al., 2017) adopted a cooperative game theoretic approach to deal with the problem of social ridesharing. Based on a social network representation of the set of commuters, they proposed an algorithm to form coalitions and arrange one-time rides at short notice which is based on two principales. First, the optimization problem of forming the coalitions that minimise the cost of the overall system, for which they restrict the feasible coalitions by means of a graph-constrained coalition formation model, allowing to specify both spatial and temporal preferences. Second, they address the payment allocation aspect of ridesharing. In (Demange, 2009), the authors proposed a game-theoretical study and focus on strategic, core-related issues rather than computational analysis of the coalition formation. This work is more close to (Rahwan et al., 2011) where authors proposed a constrained coalition formation model and an algorithm for optimal coalition structure generation. They developed a procedure that transforms the specified set of constraints, making it possible to identify all the feasible coalitions. Building upon this, they provide an algorithm for optimal coalition structure generation. (Skibski et al., 2016) have introduced the k-coalitional games. The authors have proposed an extension of the Shapley value for these games, and studied its axiomatic and computational properties. The authors in (Michalak et al., 2010) have considered only the issue of representing coalitional games in multi-agent systems with externalities in coalition formation. They have proposed a new representation which is based on Boolean expressions. Their aim was to construct much richer expressions that allow for capturing externalities induced upon coalitions. (Voice et al., 2012) addressed the problem of coalition formation with sparse synergies where the set of feasible coalitions is constrained by the edges of a graph. Their aim is to check whether the knowledge of the topology of the underlying social or organizational context graph could be used to speed up coalition enumeration and structure generation. (Ramchurn et al., 2010) defined the problem of allocating coalitions of agents to spatially distributed tasks with workloads and deadlines so as to maximize the total number of tasks completed over time. Nevertheless, these works have not deeply addressed the constraints of the agents in the proposed models or specify how agents negotiate over them to reach agreements. Constraints on coalition sizes have been considered for coalition structure value calculation (Sandholm et al., 1999), (Rahwan et al., 2007), (Rahwan et al., 2009). However, the semantics of these constraints has not been used on the same level as it is done in this paper. Work in (Wang et al., 2016) introduced a novel mathematical framework from cooperative games to model and solve cooperative scenarios where network devices have to become more autonomous and cooperate with one another, in two main classes of these games, namely, interference management and cooperative spectrum sensing. Such cooperative mechanisms involve the simultaneous sharing and distribution of resources among a number of overlapping cooperative groups or coalitions. (Ieong and Shoham, 2005), (Conitzer and Sandholm, 2006) developed succinct and expressive representations for coalitional games. In all the algorithms discussed so far, the main focus was on maximizing the social welfare, where agents consider every possible subset of agents as a potential coalition. Such formalism could be used to encode the constraints, but this is not the main concern of the constrained $\mathcal{CFM}$ mechanism considered in this paper. Negotiation is, mostly, about needed resources to perform tasks. In (Cao et al., 2013), the agents are assumed to have the same set of tasks and a unique common goal. The authors in (Arib and Aknine, 2011) addressed situations where agents plan their activities dynamically and use these plans to coordinate their actions and search for the coalitions to be formed. These studies have not considered the coalition formation with constraints changing requirement, which makes those approaches not suitable for our problem. In other hand, multi-agent negotiation has received much attention both in the context of coalition formation and in other contexts (An et al., 2011), (Rochlin and Sarne, 2014), (Sofer et al., 2016), (Kang, 2005). Indeed, negotiation is an important interaction mechanism that may allow a group of agents to reach a certain goal, particularly in situations where agents have several possibilities to explore. The authors in (An et al., 2011) considered interrelated negotiations, where selfish agents have to efficiently coordinate their negotiation with multiple resource providers to acquire multiple resources in order to accomplish a high level task. (Sofer et al., 2016) used negotiation as an exploration process, over a set of several possible alternatives, in situations where the negotiators have no information on the value of each alternative. Therefore, negotiation is assumed to be limited in time and a discount factor is used to make negotiation converge toward a solution. Another work in which agents have to explore different alternatives to find a solution is that of (Rochlin and Sarne, 2014). The agents explore several opportunities available to them, following a costly exploration process, without revealing the benefit with which they are associated. Thus, agents' goal is not to maximize the overall benefit, but rather to maximize each agent's own benefit. In (Jonge and Sierra, 2015), the authors have introduced a new heuristic-based negotiation algorithm where many self-interested agents have non-linear utility functions. In this work negotiation is, mostly, about needed resources to perform tasks. In our approach, negotiation is about groups of tasks and agents to perform them.

3 PRELIMINARIES AND CASE STUDY

To illustrate the coalition formation mechanism we propose, let us consider a carpooling example, where some travellers want to move from a city to another, and they want to share their means of transportation. Each traveller formulates to his agent the goals to be achieved. For example "I want to go from NY to Boston", his constraints as departure time, duration of the travel, and unit price of seat. To solve this problem, the agents have to deal with all the constraints and preferences over those of their associated travellers in order to enable them to share transportation. Agents negotiate for the coalitions to form to decrease the unit price of seat, increase the number of passengers, etc. They can step aside in favor of other agents, if an agreement can be found. More formally, consider a set of agents $\mathcal{N} = \{a_1, a_2, ..., a_n\}$, a set of actions $\mathcal{A} = \{b_1, b_2, ..., b_m\}$ and a set of constraints $\mathcal{C}_t = \{c_{t1}, c_{t2}, ..., c_{tk}\}$. The agents of $\mathcal{N}$ need to execute the actions of $\mathcal{A}$ by satisfying the constraints in $\mathcal{C}_t$.

The constraints are defined as intervals, for instance as an example: departure time: $D \in [10a.m., 12a.m.]$, travel duration in hours: $T \in [1H, 2H]$ and price: $P \in [20, 25]$. The agents' preferences are represented using a preference relation $\succ$ for those they want to share a car with, for instance $a_x \succ_i a_y$ (for agent a_i, a_x is preferred to a_y). We consider a coalition c as a nonempty subset of $\mathcal{N}$ ($c \subseteq \mathcal{N}$). We define $\mathcal{C}$ as the set of all possible coalitions. For a coalition c to be formed, each agent a_i in c should get a certain satisfaction. This satisfaction is defined by a utility function $u_i : \mathcal{C} \mapsto \mathcal{R}$. Note that a coalition is acceptable for agent a_i if it is preferred over, or equivalent to a reference coalition, $u_i(ref)$, which corresponds to the minimal guaranteed gain of the agent during the negotiation. A solution of the negotiation for each agent a_i introduces a coalition structure denoted CS_i which is defined on $\mathcal{N}$ with its associated utility $u_i(CS_i)$. CS_i contains a set of coalitions $\{c_1, c_2, .., c_q\}$ to be formed for the set of actions

$\mathcal{A}_i \subseteq \mathcal{A}$ where a_i is involved. Furthermore, for every $q' \in [1,q], c_{q'} \subseteq \mathcal{N}$ and $c_{q'}$ performs a set of actions $\mathcal{A}_{c_{q'}} \subseteq \mathcal{A}$ and $\forall (x,y) \in [1,q]^2, x \neq y, \mathcal{A}_{c_x} \cap \mathcal{A}_{c_y} = \varnothing$, $\bigcup_{q'=1,..,q} \mathcal{A}_{c_{q'}} \subseteq \mathcal{A}$ and $\bigcup_{q'=1,..,q} c_{q'} \subseteq \mathcal{N}$. The set of all coalition structures is denoted $\mathcal{S}$.

4 COALITION FORMATION MECHANISM ($\mathcal{CFM}$)

In order to satisfy the goals they have to achieve, the agents perform negotiations on the coalitions they want to form. So, the $\mathcal{CFM}$ requires an analysis step of constraints that agents exchange in order to guide the choice of the coalitions and a step of generating coalition structures from these constraints. Constraint analysis relies on constructing a graph of constraints and coalition generation is based on the mapping of the constraints to possible coalitions in a coalition graph. Exploring the search graph of coalitions toward better states is based on a Nested Monte-Carlo algorithm.

4.1 Constraint Graph

An effective technique for solving a coalition formation problem is a heuristic search through abstract problem spaces. The first problem space can be represented by a directed connected graph, where nodes correspond to constraint sets and edges correspond to actions (cf. Figure 1). The constraint graph may include many paths from the start to any node. Since the agents are self-interested, to search among the constraints to deal with in the coalitions, every agent constructs its own graph of constraints based on its own constraints and those revealed by other agents during this negotiation. Given a set of constraints that must be satisfied by an agent to execute a set of actions and starting from the source node labeled with $\{b\text{: } \varnothing,\ c_t\text{: } \varnothing\}$, initially there are not constraints and actions associated with this source node, let us define a graph denoted $G(c_t, b)$ as follows.

Definition 1. *Given a node labeled $\{b\text{: } \varnothing,\ c_t\text{: } \varnothing\}$ the constraint graph $G(c_t, b)$ is a directed connected graph, containing all possible nodes of constraints represented by intervals, labeled $\{X_1, ..., X'_k\}$ for each action $b_i, 1 \leq i \leq m$, that has to be executed by the agent. Each node has a utility labeled u, and directed edges from this node are labeled $\{b_j, ..., b_k\}$ where $1 \leq j \,..\, k \leq m$.*

A constraint graph gathers, the most preferred constraints' intervals in its nodes. At the root node, no action and constraint are added. Each node generates a finite set of child nodes which correspond to the accepted sets of constraints, where the first node of the graph is an outgoing node and the last nodes are incoming nodes. This constraint graph is built following a preference rate on the intervals of constraints.

Let us consider two agents, a_i which has its own interval X and receives from a_j an interval Y. The agent a_i wants to create a new interval Z that meets its constraints and those of a_j, $\{Z \models a_i\}$, by merging its interval and the one received from a_j.

We will adopt the convention of the left and right endpoints of an interval X by $\underline{X}$ and $\overline{X}$, respectively (Moore et al., 2009).

First, a_i tests if $X \subseteq Y$. Thus, if $\underline{Y} \leqslant \underline{X}$ and $\overline{X} \leqslant \overline{Y}$, it will get $X \subseteq Y$ and $Z = X$; else the agent tests if $X \cap Y$ and calculates the new interval Z. If $X \cap Y = \varnothing$ there are no points in common with a_j. Otherwise, $Z = \{max\{\underline{X}, \underline{Y}\},\ min\{\overline{X}, \overline{Y}\}\}$ and the agents tests if Z complies with its actions. If a_i does not choose this interval, it calculates $Z = X \underline{\cup} Y$ which means the union between X and Y and tests if it complies with its actions. For more details about the operations over the intervals see (Moore et al., 2009). Based on this graph, constraint analysis consists for an agent of comparing and grouping its constraints and those received from others. A natural constraint graph analysis involves constructing and linking optimal nodes. Constraints are gathered based on their relations into sets represented in the nodes of this graph. Each level of the graph of constraints refers to an action to be performed by a coalition. The advantage of the suggested method consists in directing the search of the solutions of coalitions towards primary constraints, i.e., important constraints to satisfy, thus, reducing search complexity. To move from one node of this graph to another, an action is added to the graph. The utility of a move, which labels the corresponding edge in the search space, is the utility of the action when it is added and performed by the coalition. A solution path represents a particular succession order of the added actions, and the width of that order is the sum of the edge utilities on the solution path.

To construct the constraint graph and to search for feasible proposals agents use Algorithm 1. Let us consider constructing the constraint graph by the agent a_1 on our previous example using these algorithms. First, assume that agent a_1 started a negotiation with agents a_2 to a_5 and in which each of these five agents revealed certain of its constraints. The actions that have to be executed are: b_1, b_2, b_3 which correspond respectively to: go from NY to Amherst, find a hotel room in Amherst, and go from Amherst to Boston. The constraints identified

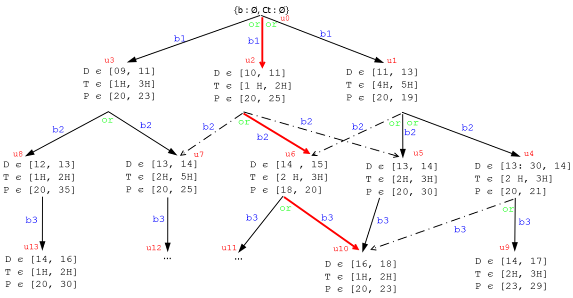

Figure 1: An example of a graph of constraints against different actions of the agent a_1. The child nodes are created based on Algorithm 1 and each node is labeled with its associated utility.

by a_1 for the action b_1 are: $D \in [10a.m., 01p.m.]$, $T \in [1H, 2H]$ and $P \in [20, 25]$ and for b_2 are: $D \in [01p.m., 02p.m.]$, $T \in [1H, 2H]$ and $P \in [20, 25]$. The nodes in the first level of the graph assemble possible sets of constraints concerning the action b_1. a_1 compares its own constraints and those received from these agents and creates new intervals of constraints X_{kc_t} (cf. Algorithm 1). Let us consider again the agent a_1 who received these intervals of constraints from the agent a_2 concerning the action b_1: $D \in [10a.m., 11a.m.]$, $T \in [1H, 3H]$ and $P \in [20, 35]$. Applying the algorithm 1, $D \in [10a.m., 11a.m.] \subseteq [10a.m., 01p.m.]$ so $Z = [10a.m., 11a.m.] \models (a_1, a_2)$. Thus, a_1 selects the interval Z. For $T \in [1H, 2H] \subseteq [1H, 3H]$, $Z = [1H, 2H] \models (a_1, a_2)$, so a_1 chooses $T \in [1H, 2H]$. For $P \in [20, 25] \subseteq [20, 35]$, $Z = [20, 25] \models (a_1, a_2)$. These results are resumed in the Figure 1. On the left of this figure, agent a_1 represents the first node created for the action b_1 by the ordered set of intervals $D \in [09a.m., 11a.m.]$, $T \in [1H, 3H]$ and $P \in [20, 23]$. We notice in this node that a_1 chooses the interval $[09a.m., 11a.m.]$ even if its departure time is not completely included in $D \in [10a.m., 01p.m.]$ because it has a good proposal of the seat price. $D \in [10a.m., 11a.m.]$, $T \in [1H, 2H]$ and $P \in [20, 25]$ are associated with the second one and $D \in [11a.m., 01p.m.]$, $T \in [4H, 5H]$ and $P \in [20, 19]$ with the last child. So, for each action, a_1 generates the different possible intervals of constraints that satisfy its action b_1. We observe that the constraints in each child node are created taking into account the end of execution of the antecedent action. So, to generate intervals of constraints that satisfy the action b_{i+1}, the agent takes into account the end time of b_i. This allows the agent to manage the relations between the actions that have to be executed. In this example, in the second level of the graph the agent a_1 identified for the action b_2 these intervals: $D \in [01p.m., 02p.m.]$, $T \in [2H, 5H]$ and $P \in [20, 25]$. The beginning of b_2 is in $[01p.m., 02p.m.]$ because b_1

begin
  Require: Node(b_i, X_{kc_t});
  Loop;
  if $X_{kc_t} \models b_i \wedge X_{kc_t} \not\models b_{i+1}$ **then**
    Generate $(X'_{1c_t}, ..., X'_{k'c_t}) \models b_{i+1}$;
    Split(b_i, X_{kc_t}) into $(b_{i+1}, X'_{yc_t} / y : 1..k')$;
    for *b_{i+1} will be executed after b_i* **do**
      if $(X'_{1c_t}, ..., X'_{k'c_t}) \cap (X_{kc_t}) = \phi$ **then**
        Create child nodes $(b_{i+1} : (X'_{1c_t}, ..., X'_{k'c_t}))$;
      end
    end
    else
      break
    end
  end
  else
    break
  end
  End Loop;
end

Algorithm 1: Constraints linking algorithm.

ends at the latest at $01p.m.$ The nodes are generated following the Algorithm 1. The dashed arcs show that nodes can share the same child nodes and the red and bold ones show the most preferred path from the root node to the last one, they result from the Monte-Carlo exploration (detailed below). From the Algorithm 1, every agent a_i which has to negotiate to execute an action b_i while satisfying its constraints c_t, chooses intervals of constraints: $X_{kc_t}(b_i) \models a_i$. It then creates child nodes for the feasible intervals that satisfy b_i. Each node created can be split under appropriate restrictions to other child nodes. The agent a_i starts with a node, labeled X_{kc_t}, for the action b_i. For each action b_{i+1} that must be executed after b_i and needs negotiation, a_i creates the new intervals for $b_{i+1} : (X'_{1c_t}, ..., X'_{k'c_t})$, and splits the node b_i, X_{kc_t} to the child nodes $b_{i+1} : (X'_{1c_t}, ..., X'_{k'c_t})$. The agent a_i uses this procedure until no action needs negotiation.

A notable detail of the constraints search space construction is that a solution is measured by its maximum path utility. We use an additive utility function, where a path is evaluated by summing its edge utilities. For each iteration, feasible solutions are only explored if their utility is not under a certain reference situation $u_i(ref)$. If an iteration is completed without finding a new possible solution, then all solutions provide less utility than that of the reference situation, $u_i(ref)$, thus, $u_i(ref)$ may be decreased and the search is repeated. To optimize the search time for the new coalitions to propose or to accept, agents use the Nested Monte-Carlo (NMC) algorithm (Cazenave, 2009).

In the first step of the mechanism, the NMC explores each level of the graph of constraints and stores the best path of constraints that satisfies the agent a_i.

The idea is to explore a graph randomly from a given position in the graph, agents use a function which plays the move in the position and returns the resulting position.

4.2 Coalitions Processing

Based on its graph of constraints and the resulting path solution, each agent a_i constructs a search graph of coalitions for all the actions that have to be executed. Each node of this graph gathers a set of coalitions that comply with the corresponding node in the graph of constraints. Each level of this graph concerns an action. To decide on the coalition to attach to each node, a_i uses the information gathered from other agents, i.e. on the coalitions they propose during their negotiations. a_i chooses from it the coalitions to propose and negotiate with others.

As during their negotiations the agents exchange their proposals of coalitions, each agent updates its sets of coalitions in its own graph. Agents perform a graph search to achieve these goals and the output of this search is a list of proposals of coalitions, one for each action. At any step in the search, a path corresponds to a set of actions that have to be executed. Once the preferred path is identified in the constraint graph, the agent a_i generates partially the graph of coalitions that satisfies at best the path of constraints and completes this graph during the negotiation phase. Hence, to complete the graph, the agent a_i adds the coalition proposal received in the corresponding level or tries to generate a new feasible proposal of coalition that takes into account revealed preferences of other agents. Therefore, a_i generates a new arc in the graph and checks possible connections with the other nodes of the graph. In case that the agents changed their constraints associated to an action, then a_i tries to generate other intervals in the graph of constraints. Then, a_i re-explores this graph with the NMC algorithm. a_i chooses a new path and completes the graph of coalitions with new coalitions if the old ones do not meet the constraints of the new path. Let us consider again the agent a_1 that constructs its coalition graph to satisfy the path which is identified from its constraint graph. Figure 2 shows this graph.

5 SOME PROPERTIES OF THE MECHANISM

Let U be the set of actions that are already used on a path p:

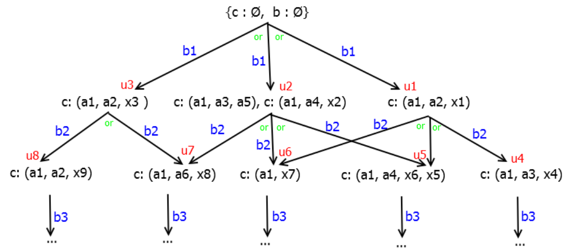

Figure 2: An example of a search graph against different actions of the agent a_1. Each node gathers possible coalitions that satisfy its path of constraints, where each coalition c may contain a set of variables of agents x_i, $\{i \geqslant 0$ and $x_i \in \mathcal{N}\}$. These variables are instantiated as the negotiation progresses.

$$U = \bigcup_{m/b_m \in p} b_m \tag{1}$$

A path ends when no action can be added. This occurs when for every proposal, some of its actions have already been used on the path ($\forall m', b_{m'} \in U$). A utility that an agent a_i gets from one path p is:

$$u_i(p) = \sum_{c_j \in p} u_i(c_j) \tag{2}$$

Proposition 2. *Every feasible coalition structure $CS \in \mathcal{S}$ is represented in the search graph by exactly one path from the root to a node. If the children of a node are the proposals such that: (1) they include the actions that have not been used on the path yet, (2) they do not include actions that have already been used on the path; formally, for any node, Θ, of the search graph, $children(\Theta) = \{b \in \{b_1, \ldots, b_m\}, b \cap U = \emptyset\}$.*

Proof. We first prove that each relevant structure $CS \in \mathcal{S}$ is represented by at most one path from the root to a node.

The first condition of the proposition leads to the fact that a coalition structure can only be generated in one order of actions on the path. Thus, there can not exist more than one path for a given coalition structure.

What still has to be shown is that each relevant coalition structure is represented by some path from the root to a node in the graph. Assume for contradiction that some relevant coalition structure $CS \in \mathcal{S}$ is not. Then, at some point, there has to be a coalition in that structure such that this coalition has an action which is not on the path, but that coalition does not belong to the path. $\square$

To summarize, in the search graph, a path from the root to a leaf corresponds to a relevant structure of coalitions. Each relevant structure CS is represented by exactly one path in the graph of coalitions.

Lemma 3. *Every feasible coalition structure CS contains exactly one coalition from level 1 and at most one coalition from every level of the graph.*

Proof. From the proposition 2, in the search graph, a path from the root to a leaf corresponds to a relevant structure of coalitions. Each relevant structure CS is represented by exactly one such path in the coalition graph, and each path contains exactly one coalition from each level. □

Proposition 4. *The number of leaves in the graph search is no greater than L^m with $L = 2^{n-1}$. Furthermore, the number of levels in the graph search (excluding the root) is not greater than m. The number of nodes in the graph search is not greater than $L^{m+1} - 1$.*

Proof. The depth of the graph is at most m since every node on a path uses up at least one action. Let $C(b_m)$ be the set of coalitions that is created to execute the action b_m. Let $L = |C(b_m)|$ the number of possible coalitions created for the action b_m in one level.

There are at most 2^{n-1} combinations of agents i.e. coalitions, so L is not greater than 2^{n-1}. Therefore, the number of leaves in the graph is not greater than m times the number of coalitions in each level ($L * L * L * \ldots * L$, m times), so not greater than L^m.

Next we prove that the number of nodes is not greater than $L^{m+1} - 1$. The number of children of each node is not greater than L because we have not greater than 2^{n-1} possibilities of coalitions. Therefore, the number of nodes is: $\sum_{i=1}^{m} L^i$, the maximization of this sum gives: $(2^{n-1})^1 + \ldots + (2^{n-1})^m = (2^{n-1})^{m+1} - 1 = L^{m+1} - 1$. □

6 EXPERIMENTAL EVALUATION

We have carried out an experimental evaluation of the CFM in a testbed implemented in JAVA. In these experiments, we have tested different sets of agents. Each agent is randomly attributed between 30 and 50 actions (i.e., $30 \leq |b_m| \leq 50$) and the constraints for each action are chosen randomly between 10 to 100 (i.e., $10 \leq |C_{tk}| \leq 100$). At each step of the negotiation, agents propose, accept, refuse proposals until agreements are reached or the deadline for the process is met. To evaluate the performance of our mechanism, we recorded the number of solutions reached, the time taken to reach such solutions, and the number of proposals sent to reach these solutions. Our experiments are repeated 100 times. We do not compare our method to some methods seeking optimal coalitions since we do not make a priori computation of coalitions before the negotiation but coalitions are computed gradually during the negotiation. We also consider that the agents do not have the same utility functions which are not known by others. Thus, agents do not necessarily look for an optimal solution. They try to converge to an acceptable solution in their negotiations since convergence is difficult to address with self-interested agents. Figure 3 shows that when NMC is not used, the time taken to reach a solution is greater, compared to that based on NMC using 100 constraints and 50 actions. This is because our algorithm allows agents to speed up their search and thus, to explore relevant solutions.

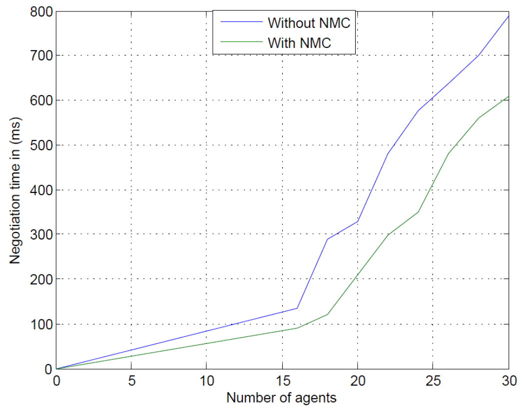

Figure 3: Taken time to reach a solution with and without NMC.

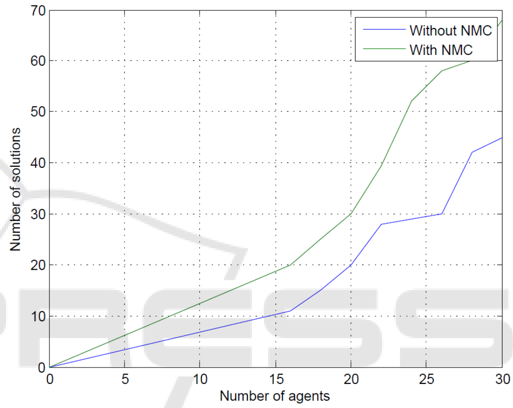

Figure 4: Number of solutions.

In what follows, we evaluate the influence of the NMC on the number of the solutions during the negotiation process while we use 100 constraints and 50 actions. We notice from Figure 4 that the number of agreements reached after negotiation is more significant in the cases where the NMC is used. These results clearly show that using the NMC enables agents to get more possible coalitions, and thus, more feasible alternatives in a reasonable time limit.

7 CONCLUSION

In this paper, we addressed the problem of coalition formation with constraints, where self-interested agents have individual alternative sets to reach different goals.We introduced a new coalition formation mechanism enriched with several principles to deal with the constraints of the agents and a Nested Monte-Carlo based search algorithm. Thus, each agent may have several possible solutions represented in the form of sequential interdependent coalitions. Our mechanism aims to allow each agent to take into account the dependencies among its tasks, which lead to inter-dependencies among possible coalitions, and to keep an overall view of all of its possible solutions throughout the coalition formation process. We have detailed how the constraints are modeled as a graph and how this graph is explored using the Nested Monte-Carlo search. From the graph of constraints, each agent gets its most preferred path of constraints and constructs a coalition graph that is used to generate the coalitions to negotiate. We have detailed some proprieties that the graphs satisfy. Then we have presented an empirical evaluation of the proposed mechanism.

REFERENCES

An, B., Lesser, V., and Sim, K. (2011). Strategic agents for multi-resource negotiation. *Autonomous Agents and Multi-Agent Systems*, 23:114–153.

Arib, S. and Aknine, S. (2011). A plan based coalition formation model for multi-agent systems. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT '11, pages 365–368.

Arib, S., Aknine, S., and Cazenave, T. (2015). Nested monte-carlo search of multi-agent coalitions mechanism with constraints. In *MIWAI*.

Bistaffa, F., Farinelli, A., Chalkiadakis, G., and Ramchurn, S. (2017). A cooperative game-theoretic approach to the social ridesharing problem. *Artificial Intelligence*, 246:86–117.

Cao, J., Wang, H., and Wang, X. (2013). A distributed algorithm for agent coalition formation with complex tasks. pages 127–132.

Cazenave, T. (2009). Nested monte-carlo search. In *IJCAI*, pages 456–461.

Conitzer, V. and Sandholm, T. (2006). Complexity of constructing solutions in the core based on synergies among coalitions. *Artif. Intell.*, 170.

Demange, G. (2009). The strategy structure of some coalition formation games. *Games and Economic Behavior*, pages 83–104.

Gelly, S. and Silver, D. (2007). Combining online and offline knowledge in UCT. In *ICML '07*, pages 273–280.

Ieong, S. and Shoham, Y. (2005). Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 193–202. ACM.

Jonge, D. and Sierra, C. (2015). Nb3: a multilateral negotiation algorithm for large, non-linear agreement spaces with limited time. *Autonomous Agents and Multi-Agent Systems*, 29.

Kang, B.-K. (2005). Optimal stopping problem with double reservation value property. *European Journal of Operational Research*, page 765–785.

Michalak, T., Marciniak, D., Szamotulski, M., Rahwan, T., Wooldridge, M., McBurney, P., and Jennings, N. R. (2010). A logic-based representation for coalitional games with externalities. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 125–132.

Moore, R. E., Kearfott, R. B., and Cloud, M. J. (2009). *Introduction to Interval Analysis*.

Rahwan, T., Michalak, T., Elkind, E., Faliszewski, P., J.Sroka, Wooldridge, M., and Jennings, N. R. (2011). Constrained coalition formation. In *AAAI*.

Rahwan, T., Ramchurn, S. D., Dang, V. D., and Jennings, N. R. (2007). Near-optimal anytime coalition structure generation. In *IJCAI*, pages 2365 – 2371.

Rahwan, T., Ramchurn, S. D., Dang, V. D., Jennings, N. R., and Giovannucci, A. (2009). An anytime algorithm for optimal coalition structure generation. *J. Artif. Int. Res.*, 34(1):521–567.

Ramchurn, S. D., Polukarov, M., Farinelli, A., Truong, C., and Jennings, N. R. (2010). Coalition formation with spatial and temporal constraints. In *AAMAS*, pages 1181–1188.

Rochlin, I. and Sarne, D. (2014). Utilizing costly coordination in multiagent. *Multiagent and Grid Systems,*, 10:23–49.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohme, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111:209–238.

Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artif. Intell.*, pages 165–200.

Skibski, O., Matejczyk, S., Michalak, T. P., Wooldridge, M., and Yokoo, M. (2016). k-coalitional cooperative games. In *Proceedings of the 2016 International Conference on Autonomous Agents*, AAMAS '16, pages 177–185. International Foundation for Autonomous Agents and Multiagent Systems.

Sofer, I., Sarne, D., and Hassidim, A. (2016). Negotiation in exploration-based environment. *Autonomous Agents and Multi-Agent Systems*, 30:724–764.

Voice, T., Ramchurn, S., and Jennings, N. (2012). On coalition formation with sparse synergies. In *AAMAS*, pages 223–230.

Wang, T., Song, L., Han, Z., and Saad, W. (2016). Overlapping coalition formation games for emerging communication networks. *IEEE Network*, 30:46–53.

Yang, G., Esmailpour, A., Cao, Y., and Nasser, N. (2016). A novel coalitional structure generation algorithm for interference mitigation in small cell networks. pages 1–4.