A Dialogue Game for Coalition Structure Generation with Self-Interested Agents

Luke RILEY [a,1], Katie ATKINSON [a] and Terry R. PAYNE [a]
[a] *Department of Computer Science, University of Liverpool, UK.*

Abstract. Since the seminal work of Dung, Argumentation Frameworks have been shown to find solutions to n-person cooperative games. In multi-agent systems, decentralised methods for multi-agent system coalition structure generation have been proposed. This paper offers the first dialogue game that utilises argumentation frameworks to find a coalition structure and a payoff vector in a decentralised manner. The payoff vector found is in the *core* set of stable solutions if the *core* is non-empty. This dialogue game also puts restrictions on the payoff vectors that can be proposed so that the most unfair ones are discarded. Lastly an algorithm is described that allows the agents to find out if the *core* is empty.

Keywords. Dialogue games, argumentation, cooperative game theory

1. Introduction

Coalition formation is the process of agents recognising that cooperation between other agents can occur in a mutually beneficial manner and then choosing an appropiate set of agents, named a *coalition*, to collaborate with to achieve some goal (or complete some task). Forming coalitions in multi-agent systems (MAS) has been shown to be an important topic; for example, [9] details that it has proved useful in areas such as e-commence (where coalitions can take advantage of price discounts) and distributed sensor networks (where coalitions can form to track targets). The cost of finding the best coalitions that satisfy all agents can be high, both computationally, and in terms of the necessary communication overhead due to the exponential number of possible coalitions and the possible self-interested behaviour of agents.

Coalition formation takes place in *n-person cooperative games* originally defined in [8]. The payoff for a coalition is traditionally measured numerically in *characteristic function games* where the value of a coalition is not influenced by the other coalitions in the system [3,6,8]. In *transferable utility characteristic function games (TU games)*, a *payoff vector* x is used to distribute the group's payoff. A *coalition structure* (CS) is a set of coalitions in a system and finding an *optimal coalition structure* (CS^*) that maximises social welfare is known as the *coalition structure generation* (CSG) problem [9].

As a general overview, the coalition formation process for a n-person cooperative game can be described as 3 stages [9]:

1. **Coalition value calculations** - This involves computing the expected payoff of each possible coalition (usually each subset of n agents - an exponential number).
2. **Coalition structure generation** - Agents are then organised into a coalition structure (preferably an optimal one that maximises social welfare).

[1] Corresponding author, e-mail: L.J.Riley@liverpool.ac.uk

3. **Determining the payoff distribution** - If the n-person cooperative game is a TU game then the payoff of each coalition in the coalition structure is divided between the agents of the system in a stable manner (see Section 2.2).

There has been a lack of focus in MAS on decentralized protocols for CSG with self-interested agents. The field of cooperative game theory (CGT) is used to find the stable payoff vectors when a system is full of self-interested agents, but many CGT solution concepts, make the assumption that the coalitions have already formed and offer no methods to form them from a MAS perspective [9].

The use of dialogue games has been shown to be a valid method to constrain the communication of self-interested agents [12]. Dialogue games are based on the theory of speech acts [11] and are rule-governed interactions where each player moves by making utterances (in the form of locutions) according to a defined set of rules but in a flexible manner [7]. In environments that contain self-interested agents a protocol is needed to set the rules as to what agents can or cannot do. In [12], dialogue games are identified as a satisfactory method for describing to agents what is forbidden and allowed in environments with self-interested agents.

The dialogue game proposed in this paper, named the *CSG dialogue game*, allows agents to build, in a decentralised manner, argumentation frameworks (AFs) [5] that have previously been shown to enable CSG (e.g. [1]). The advantage of using AFs for CSG is that the preferred extension of an AF always holds the best coalition structure for the agents given the dialogue history (as the for a dialogue example in Section 3.6 demonstrates). Argumentation schemes, which are patterns of reasoning that when instantiated provide presumptive justification for the particular conclusion of the scheme [2], are also used in the CSG dialogue game. The argumentation schemes allow the agents in the CSG dialogue game to assert arguments for different coalitions and payoff vectors.

The main significance of this paper is to propose a dialogue game that finds a coalition structure and a payoff vector in a static TU game (a TU game where the coalition's payoffs do not change). The result of this dialogue will be an optimal coalition structure and a stable outcome for the participating dialogue agents when the *core* is non-empty (given enough time and utterences). To find a stable solution of a static TU game that has an empty *core* is outside the scope of this paper and left for future work.

The rest of the paper is structured as follows: Section 2 describes the relevent background of the CGT and argumentation fields. Section 3 details the CSG dialogue game including the AF, the new argumentation scheme, the restrictions placed on the *core*, a description of an algorithm to find when the *core* is empty and an example. Section 4 discusses related work and concludes the paper.

2. Background

2.1. Transferable Utility Games

An n-person cooperative game [8] is: $\mathscr{G} = \langle N, v\rangle$ where N is the set of agents and v is the *characteristic function* ($v(2^N) \rightarrow \mathbb{R}$), which assigns every possible coalition a real numeric payoff [8]. An **outcome** of a TU game is a CS and payoff vector pair, denoted: $\langle CS, x\rangle$ where CS is a set of coalitions, denoted $\{C^1,...,C^k\}$. Static TU games are TU games where a coalition's payoff does not change during the coalition formation process. In TU games an agent can only be a member of one coalition, so a TU game coalition structure takes the form [3] $CS = \{C^1,...,C^k\}$ such that: $\bigcup_{j=1}^{k} C^j = N$; and $C^i \cap C^j = \emptyset$ for any $i, j \in \{1,...,k\}$, $i \neq j$. The payoff vector is then fully denoted [3]: $x = (x_1,...x_n) \in \mathbb{R}^n$

where $x_i \geq 0$ for all $i \in N$ and x_i denotes the individual payoff for agent i. Throughout the paper $x\langle C\rangle$ is written to denote the part of x that has individual payoffs only for agents of C.

2.2. Cooperative Game Theory Solution Concepts

The most popular solution concept to find stability in the CGT field is known as the **core**, which corresponds to the set of feasible payoff vectors where no subset of agents of the system have an incentive to deviate from the current coalition game outcome [3,6]. The core for a set of agents N can be defined using the following [3,6]:

Definition 1: *The* **core**: *A payoff vector* $x = (x_1,...,x_n)$ *and a coalition Structure* $CS = (C^1,...,C^k)$ *is in the core iff:* $\forall C \subseteq N$, $\sum_{i\in C} x_i \geq v(C)$*; and for any* $C^j \in CS$ *then* $\sum_{i\in C^j} x_i = v(C^j)$.

If the first condition does not hold for some group of agents, C, then they have a reason to deviate and divide the payoff of $v(C)$ between themselves. The second condition restricts the agents so that no side payments between coalitions are allowed.

Two problems with the *core* are that it can sometimes be empty and some *core* outcomes can be classified as unfair [3,6]. The protocol detailed in this paper will find a solution in the *core* (if the *core* is non-empty). Also the *core* payoffs found will follow certain fairness principles (see Section 3.4.1) so that the most unfair payoff vectors in the *core* are not found. If the *core* is empty, the alternative CGT solution concept of the ε*-core* will have to be used to stabilise the game. This is left for future work.

2.3. Argumentation Frameworks

In the CSG dialogue game, an agent can propose arguments for a coalition and a payoff vector. To evaluate all the instantiated arguments in the CSG dialogue game to determine their *acceptability*, they will be organised into an Argumentation Framework (AF). AFs are a means to represent and reason with different, possibly conflicting, data. AFs use graphs of nodes and arcs, where the nodes represent abstract arguments, having no internal structure, and the arcs represent attacks between the arguments [5]. An AF is defined as:

Definition 2: *An Argumentation Framework is a tuple AF = (Args, R) where Args is a set of arguments and R is a binary attack relation* $R \subseteq Args \times Args$.

In an AF, where $Args = \{a_1,...,a_n\}$, an attack $a_1 R a_2$ is said to **defeat** a_2 if a_1 has not been defeated by another argument in the AF. A set of arguments S is **acceptable** iff $\forall a_x \in Args$ if a_x attacks an argument a_y where $a_y \in S$ there $\exists a_z \in S$ where a_z defeats a_x. S is a **preferred extension** of the AF if S is the maximal acceptable set of arguments.

3. The CSG Dialogue Game

3.1. The CSG Dialogue Game Overview

The CSG dialogue game can be used by self-interested agents willing to partake in a static TU game. The most pertinent locutions available to the players of the game, inspired by [2], are *join*, *assert* and *close* (outlined in Table 1) and the pre/post conditions of the full set of dialogue moves are fully defined in [10]. The CSG dialogue game is designed to be turn based, so that an algorithm to check if the core is empty can be developed. This paper assumes that a suitable enforcement mechanism, in the form of another

Table 1. The informal meaning of the main dialogue moves

Move	Format	Meaning
join	$\langle i, join, a_n \rangle$	Agent i wants to be included in the dialogue and is willing to form the best coalition it can with any other agents in the dialogue. To join, the agent must assert its individual payoff.
assert	$\langle i, assert, a_n \rangle$	Agent i believes the coalition and payoff vector proposed in a_n will improve on i's current payoff.
close	$\langle i, close \rangle$	An agent closes the dialogue if it does not believe it has any more moves available. If every agent's last move was a close move, without another move in-between, then the dialogue is over.

argumentation scheme and dialogue move, is used to ensure the agent's assertions are truthful (the details of which are presented in [10]).

For an agent i to assert an argument for a coalition C, the condition $i \in C$ must hold. This condition is used as self-interested agents cannot be trusted to assert coalitions when only others would benefit[2]. So even though the outcome of this CSG dialogue consists of a CS and payoff vector pair, individual agents will only be able to argue over the subsets of both that involve it. As a consequence, each agent can assert 2^{n-1} coalitions; thus CSG dialogue games will only be feasible for small sets of agents or n-person cooperative games with restrictions.

The dialogue, based on [2], is denoted D_r^s where r is the timepoint of the first move of the dialogue and s the timepoint of the last, where $r, s \in \mathbb{N}$. Every time an agent makes an utterence of an argumentation scheme in the dialogue, this information is stored in a publicly readable commitment store (*CoSt*)[2]: An **individual commitment store** for a CSG dialogue (D_r^s), for every agent $i \in N$ a commitment store of agent i at time-point s is denoted $CoSt_i^s$. A **combined commitment store** for a CSG dialogue (D_r^s) with participants N and time-point s is denoted $CoSt_N^s$ where $CoSt_N^s = \bigcup_{\forall i \in N} CoSt_i^s$. If $s = 0$ then $CoSt_N^s = \emptyset$.

3.2. The Coalition Argumentation Schemes

Abstract arguments themselves are not always useful for representing instantiated arguments, that is, arguments with some internal structure or content. To reason on the best coalitions to form from a game theoretic perspective, argumentation schemes are used, instantiated and then the generated arguments are placed in an argumentation framework. Instantiations of argumentation schemes attack other instantiations of argumentation schemes under pre-defined conditions, which attack either the premises, inference rules or conclusions of the scheme. In the CSG dialogue game, all attacks focus on the conclusions of the argumentation schemes (that is, whether to form the coalition or not). The new argumentation scheme used in this paper is a $\mathscr{C}$-**Argument**, which is for the traditional static TU game and makes use of the concepts defined earlier, in Section 2.

A $\mathscr{C}$-argument should be instantiated when an agent wants a coalition C to form. At timepoint s a $\mathscr{C}$-Argument is informally described as:

$\mathscr{C}$-**Argument**: *Agent i asserts that coalition* C **should form**, *since given the current payoff of the agents of C, denoted* $x^{s-1}\langle C \rangle$, *and the coalition payoff of* $v^s(C)$, *then the payoff vector of* $x^s\langle C \rangle$ *should be implemented since* $x^s\langle C \rangle$ *offers an equal or better, fairer payoff for all of* C.

[2]Unless a sufficient enforcement mechanism is used, but this is outside the scope of this paper.

The attack rule for the instantiated argumentation scheme is: A newly asserted argument a_s about a coalition C, attacks any other argument a_p previously asserted in the dialogue, that shares a member of C. This attack is used so that agents can only be part of one coalition. The following functions operate over any a_n; $Coal(a_n)$ returns the coalition C proposed in a_n; $Vect(a_n)$ returns the section of the new payoff vector $x^n\langle C\rangle$ proposed in a_n; $Val(a_n)$ returns the value $v^n(C)$ of the coalition proposed in a_n.

3.3. *Coalition Argumentation Framework*

Now to find the *outcome* of the CSG dialogue game ($\langle CS, x\rangle$), firstly the preferred extension (PE) of the AF is found. Then the agents systematically consider the internal structure of the abstract arguments in the PE by looking at the instantiated versions of these abstract arguments. The CS is then the collection of all the coalitions proposed by the instantiated versions of the arguments in the PE and x is the conjunction of all the payoff vectors that are proposed by the same $\mathscr{C}$-arguments.

The following functions are used to find the *outcome* of the CSG dialogue game (additional minor functions are detailed in [10]). While only the *PayVect* and *CoalStruct* functions directly find the *outcome*, all other functions are used to help the agents work towards finding a *core* stable outcome, if one exists.

- $PE(CoSt) = \Phi$. Where Φ is the preferred extension of $CoSt$ given the attacks taken from the instantiated argumentation schemes of $CoSt$.
- $CoalStruct(CoSt) = \lambda$. $\forall a_n \in PE(CoSt)$, $Coal(a_n) \in \lambda$. This function returns the set of all the coalitions in the instantiated argumentation schemes of the preferred extension of the given CoSt.
- $PayVect(CoSt) = x$, where $x = (x_p, ..., x_q)$. $\forall a_n \in PE(CoSt), \forall x'_j \in Vect(a_n)$ then $x_j = x'_j$. This function returns as one tuple the payoffs of all the individual agents in the instantiated schemes of the preferred extension of the given CoSt.
- $CoalStructVal(CoSt) = \sum_{a_n \in PE(CoSt)} Val(a_n)$. This function finds the value of the coalition structure of the given CoSt.
- $BestCoal(CoSt, i) = C$. $\forall C' \in CoalStruct(CoSt)$, if i $\in C'$ then $C = C'$. This function returns the best coalition for the given agent i.
- $FairPayDist(CoSt, a_n, i)$. This function restricts the most unfair payoffs from being asserted. If a payoff that is deemed unfair is found in the given a_n then $\perp$ is returned, else $\top$ is returned. Further details are given in Section 3.4.1.

In the CSG dialogue game only one PE can exist, as no attack cycles can be made. This is because of the argumentation scheme attack rule, which states that no argument can ever attack another argument asserted after it. This rule is in place as newer arguments should have fairer payoff vectors and so will attack older arguments that have less fair payoff vectors (according to Section 3.4.1). Therefore computing the preferred extension of this CSG dialogue game takes time linear in the number of arguments [5].

3.4. *Formalising the Argumentation Schemes*

As the functions of Section 3.3 are now defined, the formal definition can now be presented of the new argumentation scheme (informally described in Section 3.2) at time-point s in dialogue D^s_r:

Definition 3: *A* $\mathscr{C}$**-argument** $\mathscr{C} = \langle i,\ C,\ x^{s-1}\langle C\rangle,\ v^s(C),\ x^s\langle C\rangle\rangle$ *s.t:* $i \in C$, $C \subseteq N$; $x^{s-1}\langle C\rangle \in \mathrm{PayVect}(CoSt^{s-1}_N)$; *and* $\mathrm{FairPayDist}(CoSt^{s-1}_N, \mathscr{C}) = \top$.

Now the formalised **attack rule** (described in Section 3.2) for an argument a_s is: a_s attacks every argument a_p in $CoSt^{s-1}_N$ **if** $Coal(a_s) \cap Coal(a_p) \neq \emptyset$.

3.4.1. Ensuring Fair Payoff Distributions

The CSG dialogue game finds *core* payoffs (compared to other CGT solution concepts) for communication and computation cost reasons; see [10] for a detailed explanation.

During the dialogue, an agent i will only be able to deviate from the current CS by asserting a new coalition C if $i \in C$ and all agents of C (including i) receive an incentive. The actual additional *incentive amount* agents must receive above their current payoff to defect to another coalition is set using Defn. 4, which ensures that the CS* is found when no more moves are possible, as increasing the social welfare does not incur a cost.

Definition 4: $inc(CoSt, a_n, i) = 0$ *iff*

$$\mathrm{CoalStructVal}(CoSt \cup \{a_n\}) > \mathrm{CoalStructVal}(CoSt) \textit{ or}$$
$$\mathrm{CoalStructVal}(CoSt \cup \{a_n\}) == \mathrm{CoalStructVal}(CoSt) \textit{ and}$$
$$|\mathrm{Coal}(a_n)| > |\mathrm{BestCoal}(CoSt, i)|$$

else $inc(CoSt, a_n, i) > 0$

In the CSG dialogue game if an agent wants to assert a new argument a_n, then a_n needs to pass the following test (defined in Defn. 5), which checks if a_n results in the payoff vector converging towards *core* stability. The first bullet point of Defn. 5 ensures individual rationality and that the new payoff for all agents of C is greater than or equal to their previous payoff. The second bullet point ensures that no utility is lost out of the game. The third bullet point negates the unfair payoff criticism of the *core* by ensuring all agents get either an equal split of the coalition's payoff or each agent j that can get a greater payoff elsewhere is given at least that payoff in this coalition. The additional pay to agent j (above the equal split of the coalition's payoff) is taken equally from the remaining agents (unless this would motivate others to deviate).

Definition 5: *The* **fairPayDist**$(CoSt, a_n, i)$ *function, where* $C = \mathrm{Coal}(a_n)$*,* $CS = $ CoalStruct $(CoSt \cup \{a_n\})$*,* $x = \mathrm{PayVect}(CoSt \cup \{a_n\})$ *and* $x' = \mathrm{PayVect}(CoSt)$*, returns* $\perp$ *unless the following are satisfied, if so the function returns* $\top$*:*

- $\forall j \in C, x_j \geq v(\{j\})$ *and* $x_j \geq x'_j + \mathrm{inc}(CoSt, a_n, i)$
- $\sum_{j \in C} x_j = \mathrm{Val}(a_n)$
- double *split* ← *split'* ← *0*

  $C'' \leftarrow C' \leftarrow C$

  while $C'' \neq \emptyset$ **do**

      boolean *equal* ← *true*

      $split \leftarrow \dfrac{split'}{|C''|}$

      for all $j \in C''$ *where* $x'_j + \mathrm{inc}(CoSt, a_n, i) \geq \dfrac{v(C)}{|C|} - split$ **do**

          $x_j \leftarrow x'_j + \mathrm{inc}(CoSt, a_n, i)$

          $split' \leftarrow split' - \dfrac{v(C)}{|C|} + x_j$

          $C' \leftarrow C' \backslash \{j\}$

          equal ← *false*

      if *equal* == *true* **then**

          for all $k \in C''$ **do**

              $x_k \leftarrow \dfrac{v(C)}{|C|}$ - *split*

              $C' \leftarrow C' \backslash \{k\}$

      $C'' \leftarrow C'$

3.5. Checking if the Core is Empty

This dialogue game can only find solutions to a coalition game if the *core* is **not** empty and so a valid question is: how do the agents know if the *core* is empty? To find the answer to the question the agents should communicate in stages: Firstly they should join the dialogue; secondly they should compute the coalition's values; and thirdly each agent i should assert either a $\mathscr{C}$-argument where the *inc* function returns 0 (if one can be found, as this coalition will improve social welfare) or assert a $\mathscr{C}$-argument where the *inc* function returns > 0 (as this coalition will improve the payoff for agent i but not the social welfare), else the agent should assert a close move.

As communication happens in these stages and agents assert social welfare improving arguments first, the core can be checked to be non-empty under a certain condition: for all arguments asserted in the dialogue for the same coalition, if there does not exist an agent in these arguments that has a strictly increasing or decreasing payoff over time, then the *core* is empty. The reasons for this are discussed in [10].

3.6. A Dialogue Example

A full example can now be developed in Table 2 and Figure 1. As can be seen in Table 2, communication happens in stages: firstly the agents join the dialogue and assert their individual payoff; secondly the agents assert new coalitions or modify previously asserted coalitions' payoff vectors to gain fairer payoffs; finally when each agent can find no other variables that satisfy the argumentation scheme restrictions they utter a close move.

The AF generated from the example in Table 2 can be seen in Figure 1 where the argument number corresponds to the move number. Finding the outcome ($\langle CS, x\rangle$) of the CSG dialogue after move n requires computing the preferred extension of the AF of Figure 1 for the arguments: A1 to An. The preferred extension, after the CSG dialogue game of Table 2 has completed, is: $\{A1, A8\}$. When looking at the instantiated argumentation schemes of A1 and A8 we can see that the CS = $\{\{1\},\{2,3\}\}$ and the payoff vector is x(4,11,13). As all agents only uttered a close move when no more moves could be asserted, we find that this CS is the optimal CS of the game and x is inside the *core*.

Table 2. This table details a CSG dialogue for a $\mathscr{G} = \langle N, v\rangle$ where $N = \{1,2,3\}$, $v(\{1\}) = v(\{2\}) = 4$, $v(\{3\}) = 5$, $v(\{1,2\}) = 8$, $v(\{1,3\}) = 18$, $v(\{2,3\}) = 24$, $v(\{1,2,3\}) = 12$. A move at timepoint s is of the following form: the agent's identifier; the move type; and (if not a close move) an argumentation scheme instantiation i.e a $\mathscr{C}$-argument of the form $\langle i, C, x^{s-1}\langle C\rangle, v^s(C), x^s\langle C\rangle\rangle$. The attack rule is $a_p R a_q$ iff $Coal(a_p) \cap Coal(a_q) \neq \emptyset$ and argument a_p is asserted after a_q. For the dialogue to finish, n close moves need to be asserted consecutively.

Move No.	**Move**	**CS**	**x**
1	$\langle 1, join, \langle 1, \{1\}, (-), 4, (4)\rangle\rangle$	$\{\{1\}\}$	$x(4)$
2	$\langle 2, join, \langle 2, \{2\}, (-), 4, (-,4)\rangle\rangle$	$\{\{1\},\{2\}\}$	$x(4,4)$
3	$\langle 3, join, \langle 3, \{3\}, (-,-), 5, (-,-,5)\rangle\rangle$	$\{\{1\},\{2\},\{3\}\}$	$x(4,4,5)$
4	$\langle 1, assert, \langle 1, \{1,3\}, (4,-,5), 18, (9,-,9)\rangle\rangle$	$\{\{2\},\{1,3\}\}$	$x(9,4,9)$
5	$\langle 2, assert, \langle 2, \{2,3\}, (-,4,9), 24, (-,12,12)\rangle\rangle$	$\{\{1\},\{2,3\}\}$	$x(4,12,12)$
6	$\langle 3, assert, \langle 3, \{1,3\}, (4,-,12), 18, (5,-,13)\rangle\rangle$	$\{\{2\},\{1,3\}\}$	$x(5,4,13)$
7	$\langle 1, close\rangle$	$\{\{2\},\{1,3\}\}$	$x(5,4,13)$
8	$\langle 2, assert, \langle 2, \{2,3\}, (-,4,13), 24, (-,11,13)\rangle\rangle$	$\{\{1\},\{2,3\}\}$	$x(4,11,13)$
9	$\langle 3, close\rangle$	$\{\{1\},\{2,3\}\}$	$x(4,11,13)$
10	$\langle 1, close\rangle$	$\{\{1\},\{2,3\}\}$	$x(4,11,13)$
11	$\langle 2, close\rangle$	$\{\{1\},\{2,3\}\}$	$x(4,11,13)$

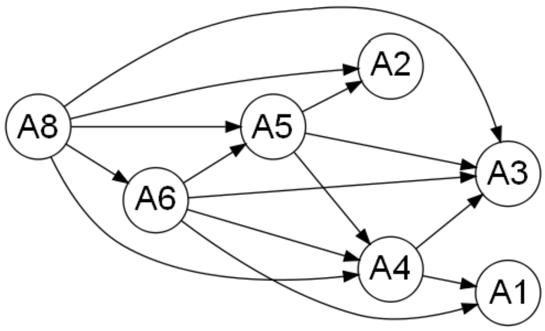

Figure 1. The AF generated given the CSG dialogue of Table 2

4. Related Work and Conclusion

So far in the argumentation and dialogue game literature, a few attempts have been made to detail coalition formation techniques, such as [1,5] but only [5] links argumentation frameworks to agent payoff vectors (which is the norm in the cooperative game theory literature). However [5] did *not* investigate how these solutions can be collaboratively built by multi-agent systems in a decentralised manner. This issue is investigated here.

Using dialogue games for coalition formation can be traced back to [4], which uses a dialogue game to form coalitions. However [4] does not show how a coalition structure can be generated, which can be found with the dialogue game in this paper.

In this paper, an argumentation-based dialogue to find coalition structures has been proposed, with the novel contribution being: the definition of an *n*-person dialogue game for coalition formation that can find the optimal coalition structure combined with a payoff vector in the *core* if the *core* is non-empty. Additionally the paper discusses: the restrictions placed on the *core* so that the most unfair *core* payoffs can never be suggested; and an algorithm is described that shows how the agents can find out if the *core* is empty. This paper outlines the framework that will be built on by future research.

References

[1] L. Amgoud. An argumentation-based model for reasoning about coalition structures. *The second International Workshop on Argumentation in Multi-Agent Systems, ArgMAS'2005*, pages 217–228, 2005.

[2] E. Black and K. Atkinson. Dialogues that account for different perspectives in collaborative argumentation. In *Proc of the 8th Int. AAMAS Conf.*, pages 867–874, 2009.

[3] G. Chalkiadakis, E. Elkind, and M. Wooldrdidge. *Computational Aspects of Cooperative Game Theory*. Morgan & Claypool Publishers, 2011.

[4] F. Dignum, B. Dunin-Keplicz, and R. Verbrugge. Agent theory for team formation by dialogue. In *Proc. of the 7th Int. Workshop on Agent Theories, Architectures and Languages*, pages 141 –156, 2000.

[5] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artificial Intelligence*, 77:321–357, 1995.

[6] D. Gillies. *Some theorems on n-person games*. PhD thesis, Princeton University, 1953.

[7] P. McBurney and S. Parsons. Dialogue games for agent argumentation. *Argumentation in Artificial Intelligence*, pages 261–280, 2009.

[8] J. Neumann and O. Morgenstern. *The Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[9] T. Rahwan. *Algorithms for Coalition Formation in Multi-Agent Systems*. PhD thesis, University of Southampton, 2007.

[10] L. Riley, K. Atkinson, and T. Payne. Coalition structure generation for self interested agents in a dialogue game. Technical report, ULCS-12-004, University of Livepool, 2012.

[11] J. Searle and D. Vanderveken. *Foundations of Illocutionary Logic*. Cambridge University Press, 1985.

[12] C. Sierra, J. A. Rodriguez-Aguilar, P. Noriega, M. Esteva, and J. L. Arcos. Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional*, V(4):33–39.