Overlapping Coalition Formation Games: Charting the Tractability Frontier

Yair Zick
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
yair0001@ntu.edu.sg

Georgios Chalkiadakis
Department of Electronic and Computer Engineering
Technical University of Crete, Greece
gehalk@ece.tuc.gr

Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
eelkind@ntu.edu.sg

ABSTRACT

Cooperative games with overlapping coalitions (OCF games) [3, 23] model scenarios where agents can distribute their resources among several tasks; each task generates a profit which may be freely divided among the agents participating in the task. The goal of this work is to initiate a systematic investigation of algorithmic aspects of OCF games. We propose a discretized model of overlapping coalition formation, where each agent $i \in N$ has a weight $w_i \in \mathbb{N}$ and may allocate an integer amount of weight to any task. Within this framework, we focus on the computation of outcomes that are socially optimal and/or stable. We discover that the algorithmic complexity of the associated problems crucially depends on the amount of resources that each agent possesses, the maximum coalition size, and the pattern of interaction among the agents. We identify several constraints that lead to tractable subclasses of OCF games, and provide efficient algorithms for games that belong to these subclasses. We supplement our tractability results by hardness proofs, which clarify the role of our constraints.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Theory

Keywords

Overlapping Coalition Formation, Complexity, Bounded Treewidth

1. INTRODUCTION

In many multiagent systems, agents split into teams in order to complete tasks or solve problems [14]. Typically, the collective efforts of a team are rewarded with a payoff, which then needs to be shared among the team members. When agents are selfish, i.e., aim to maximize their own payoff, such settings are modeled using the tools of *cooperative game theory*, which suggests a variety of approaches to team formation and payoff division [2].

The usual framework of cooperative games assumes that agents are divided into disjoint sets in order to perform tasks. However, sometimes the agents may benefit from splitting their resources among several jobs and forming *overlapping* coalitions. The study of such scenarios was initiated by Shehory and Kraus [21] (see also Dang et al. [6]), who assumed that agents are fully cooperative. More recently, Chalkiadakis et al. [3] proposed a formal model for cooperative games with overlapping coalition structures (OCF games), which, in contrast to the previous work, takes agent incentives into account. More specifically, Chalkiadakis et al. [3] define games in which agents can form *partial coalitions*, and an agent can participate in multiple (overlapping) coalitions. Such coalitions correspond to vectors in $[0,1]^n$: the i-th coordinate of the vector identifies the *fraction* of the i-th agent's resources devoted to this coalition.

The main focus of [3] is coalitional stability in OCF games. In such games, identifying outcomes that are stable, i.e., resistant to group deviations, is more difficult than in the standard model. This is because in OCF games one needs to take into account the non-deviators' reaction to deviation. Indeed, when no overlapping coalitions are allowed, the deviators do not care about the reaction of other agents to their actions: all of their resources are now devoted to maximizing their own welfare and they have no stake in what other agents do. In contrast, when agents are allowed to divide their resources among several tasks, group reaction to deviation must be taken into consideration: the deviating agents may remain involved in one or more partial coalitions with non-deviators, and they need to reason about the payoff they expect to get from such collaborations. These issues were raised in [3] and subsequently studied in detail by Zick and Elkind [23], who proposed a general framework for handling coalitional reaction to group deviations under a number of solution concepts.

While Chalkiadakis et al. [3] and Zick and Elkind [23] provide a rich framework for reasoning about OCF games, they give short shrift to computational aspects of such games. Indeed, Zick and Elkind [23] ignore the algorithmic efficiency issues altogether, and in [3] the algorithmic results are limited to a special class of OCF games known as *Threshold Task Games*; while these games supply a useful testing ground for comparing different stability concepts, they are clearly not expressive enough to capture all OCF games. We remark that, in general, designing efficient algorithms for coalitional games, even in the absence of overlapping coalitions, is a challenging task. Indeed, the sheer number of such games precludes the existence of a representation scheme that can describe any coalitional game in $\mathrm{poly}(n)$ bits (where n is the number of players). For this reason, a number of different representation languages for coalitional games have been proposed, with each language capturing a specific family of application scenarios and requiring purpose-built algorithms for computing various solution concepts (see, e.g., Chalkiadakis et al. [4] for a literature review).

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

For games with overlapping coalitions, the situation is even more dire: even identifying a finitary representation for OCF games is non-trivial, as we need to be able to specify the payoff of each partial coalition, and there are infinitely many such coalitions.

The aim of this paper is to initiate a systematic investigation of algorithmic aspects of OCF games. We propose a discretized model of overlapping coalition formation, where each agent $i \in N$ has a weight $w_i \in \mathbb{N}$ and may allocate an integer amount of resources to any partial coalition. This simplification ensures the existence of a finitary representation, and can be justified by observing that in practice, agents' resources have certain granularity and therefore cannot be divided with arbitrary precision. We then focus on the computation of outcomes that are socially optimal and/or stable. We discover that the complexity of the associated problems crucially depends on the amount of resources that each agent possesses, the maximum coalition size, the pattern of interaction among the agents, and the properties of the arbitration function. We identify the constraints that lead to tractable subclasses of OCF games, and provide efficient algorithms for games that belong to these subclasses. We supplement our tractability results by hardness proofs, showing that the constraints that we impose are, in a sense, necessary. Our results suggest a number of future research directions; we hope that they will serve as a starting point for a comprehensive algorithmic analysis of OCF games, which is a necessary precondition for the practical applicability of such games.

2. PRELIMINARIES

Throughout the paper, we use boldface lowercase letters to denote vectors and uppercase letters to denote sets. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we write $\mathbf{x} \le \mathbf{y}$ when $x_i \le y_i$ for all $i = 1, \dots, n$.

Cooperative Games with Overlapping Coalitions We briefly describe the model as presented in [3]. A *cooperative game with overlapping coalitions*, also referred to as an *overlapping coalition formation (OCF) game*, is given by a set of *agents* $N = \{1, \dots, n\}$ and a *characteristic function* $v : [0,1]^n \to \mathbb{R}_+$; we write $G = (N, v)$. Each agent has one unit of resource (time, money, etc.); using their resources, agents may form *partial coalitions*: a partial coalition is described by a vector $\mathbf{c} = (c_1, \dots, c_n)$, where c_i is the fraction of i's resource dedicated to this coalition. The *value* of the partial coalition $\mathbf{c}$ is given by $v(\mathbf{c})$, and its *support* is given by $supp(\mathbf{c}) = \{i \in N \mid c_i > 0\}$. A *coalition structure* over a subset S of N is a collection $CS = (\mathbf{c}^1, \dots, \mathbf{c}^m)$ of partial coalitions that satisfies $\sum_{j=1}^m \mathbf{c}^j \le \mathbf{e}^S$, where $\mathbf{e}^S \in \{0,1\}^n$ is the indicator vector of the set $S \subseteq N$. We denote by $\mathcal{CS}(S)$ the set of all coalition structures over S. Given a coalition structure $CS = (\mathbf{c}^1, \dots, \mathbf{c}^m)$, we overload notation and write $v(CS) = \sum_{j=1}^m v(\mathbf{c}^j)$; we refer to $v(CS)$ as the *value* of CS. The *superadditive cover* of a characteristic function v is the mapping

$$v^*(\mathbf{c}) = \sup_{CS \in \mathcal{CS}(N)} \{v(CS) \mid \sum_{\mathbf{c}' \in CS} \mathbf{c}' \le \mathbf{c}\},$$

which computes the most that the agents can earn if their resources are given by $\mathbf{c}$.

The payoff $v(\mathbf{c}^j)$ needs to be divided among agents who contribute to $\mathbf{c}^j$, i.e., members of $supp(\mathbf{c}^j)$; a *division of payoffs* of a partial coalition $\mathbf{c}^j$ is a vector $\mathbf{x}^j \in \mathbb{R}^n_+$ that satisfies $\sum_{i=1}^n x_i^j = v(\mathbf{c}^j)$, $x_i^j = 0$ for any $i \notin supp(\mathbf{c}^j)$. A *pre-imputation* for a coalition structure $CS = (\mathbf{c}^1, \dots, \mathbf{c}^m)$ is a collection of vectors $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^m)$, where for each $j = 1, \dots, m$ the vector $\mathbf{x}^j$ is a division of payoffs of coalition $\mathbf{c}^j$. The set of all pre-imputations for a coalition structure CS is denoted by $\mathcal{I}(CS)$. The pair $(CS, \mathbf{x})$, where $\mathbf{x} \in \mathcal{I}(CS)$, is called a *feasible outcome*. The *total payoff* of a player i under a feasible outcome $(CS, \mathbf{x})$ with coalition structure $CS = (\mathbf{c}^1, \dots, \mathbf{c}^m)$ and a payoff vector $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^m)$ is defined as $p_i(CS, \mathbf{x}) = \sum_{j=1}^m x_i^j$; we extend this notation to sets of agents by setting $p_S(CS, \mathbf{x}) = \sum_{i \in S} p_i(CS, \mathbf{x})$ for $S \subseteq N$.

Stability and Arbitration Functions in Cooperative Games with Overlapping Coalitions Subsets of agents can *deviate* from an outcome by withdrawing some or all of their resources from some or all of the partial coalitions they participate in. The *arbitration function* [23] is a mapping $\mathcal{A}$ that receives as its input (a) a feasible outcome $(CS, \mathbf{x})$, (b) a deviating set $S \subseteq N$ and (c) S's proposed deviation, i.e., a list of resources that members of S intend to withdraw from each coalition in CS. Given this data, $\mathcal{A}$ returns a number for each coalition $\mathbf{c}$ with $supp(\mathbf{c}) \cap S \neq \emptyset$, $supp(\mathbf{c}) \cap (N \setminus S) \neq \emptyset$; this number represents how much of $\mathbf{c}$'s payoffs the agents in $S \cap supp(\mathbf{c})$ will be allowed to keep if they deviate. In general, this number may depend on S's behavior outside of $\mathbf{c}$: for instance, $\mathbf{c}$ may be unwilling to pay S if S hurts some players in $supp(\mathbf{c}) \cap (N \setminus S)$ in some other coalition, or it may withhold the payment completely if S deviates in any way whatsoever. We assume that $\mathcal{A}$ is *normalized*: if a set withdraws all of its resources from some partial coalition, it receives nothing from it. We also assume that $\mathcal{A}$ is *deviation-monotone*: the deviators cannot increase the payoff they receive from a partial coalition by withdrawing more resources from it.

We denote by $\mathcal{A}^*(CS, \mathbf{x}, S)$ the most that a set $S \subseteq N$ can get when deviating from $(CS, \mathbf{x})$ under the arbitration function $\mathcal{A}$ (including the payoff from the partial coalitions that the deviators form among themselves). An outcome $(CS, \mathbf{x})$ is said to be in the $\mathcal{A}$-*core*, or $\mathcal{A}$-*stable*, if no set S can deviate (and then share payoffs from the deviation) so that each $i \in S$ gets more than $p_i(CS, \mathbf{x})$, when payoffs to deviators from coalitions with non-deviators are given by $\mathcal{A}$. Zick and Elkind [23] show that an outcome is $\mathcal{A}$-stable if and only if $p_S(CS, \mathbf{x}) \ge \mathcal{A}^*(CS, \mathbf{x}, S)$ for all $S \subseteq N$.

Three types of reactions to set deviations are described in [3]; in the terminology of [23], these are arbitration functions. First, under the *conservative* arbitration function, any coalition $\mathbf{c}$ with $supp(\mathbf{c}) \cap (N \setminus S) \neq \emptyset$ pays nothing to S; this notion of deviation is the most restrictive, and allows a deviating set only the payoff from whatever coalitions it forms on its own. Second, under the *refined* arbitration function, $\mathbf{c}$ allows S to keep its payoff as long as no member of S changes his contribution to $\mathbf{c}$. Third, under the *optimistic* arbitration function, the deviators may keep some of $\mathbf{c}$'s payoff even if they withdraw some resources from $\mathbf{c}$; specifically, if they can ensure that each agent in $supp(\mathbf{c}) \cap (N \setminus S)$ receives as much from the reduced coalition as it did before the deviation, they can keep the remaining payoff. Under the assumptions on arbitration functions given in [23], the payoff given by the optimistic arbitration function is the most that any arbitration function may give. This means that if an outcome is stable w.r.t. the optimistic arbitration function, it is stable under any arbitration function.

3. OUR MODEL

In the model of [3, 23], each agent may divide his resources in any way he chooses. This leads to a variety of conceptual and algorithmic complications: for instance, there is no apriori bound on the size of the coalition structure, and it is not clear how to represent the characteristic function and the arbitration function.

To circumvent these difficulties, we assume that agents may only divide their resources in a discrete manner: each agent $i \in N$ has a positive integer weight w_i, and may allocate an integer part of it to a partial coalition. This is a reasonable assumption in most multiagent settings, where agents allocate hours, money or memory space to tasks. We will refer to such games as *discrete OCF games*. We set $\Psi = \max_{i\in N}\{w_i\}$; note that the case $\Psi = 1$ corresponds to the standard model of characteristic function games, i.e., one that does not admit overlapping coalitions. For our asymptotic bounds, we will assume $\Psi > 1$. Let $\mathbf{w} = (w_1, \dots, w_n)$ and $\mathcal{W} = [0, w_1]\times \dots \times [0, w_n]$.

We can interpret the characteristic function v of a discrete OCF game as a mapping from $\mathcal{W}$ to $\mathbb{R}$ and modify the definition of the superadditive cover and other notions introduced in Section 2 in a similar manner. Given a subset $S \subseteq N$ and some $\mathbf{q} \in \mathcal{W}$, we let $\mathbf{q}^S$ be the vector $\mathbf{q}$ with all coordinates $i \notin S$ set to 0. Also, we define $\mathcal{W}(S) = \{\mathbf{q} \in \mathcal{W} \mid \mathbf{q} \leq \mathbf{w}^S\}$. We will assume that the value of each partial coalition is a non-negative rational number that can be encoded using $\text{poly}(n, \log \Psi)$ bits. Under this assumption for any fixed value of Ψ there are finitely many discrete n-player OCF games where the weight of each player is bounded by Ψ, and any such game can be represented by a vector of length $(\Psi + 1)^n$.

The size of this vector representation is exponential in n, which is unacceptable for most applications. We will therefore limit our attention to games where there is an apriori bound on the admissible coalition size. Namely, we say that a game with overlapping coalitions is a *k-OCF game* if $v(\mathbf{q}) = 0$ for any $\mathbf{q} \in \mathcal{W}$ with $|supp(\mathbf{q})| > k$. Clearly, a discrete k-OCF game can be represented using $\binom{n}{k}(\Psi + 1)^k$ values; this number is polynomial in Ψ and n if k is bounded by a constant. In the rest of the paper, we will assume that a k-OCF game is represented by a list that consists of all partial coalitions $\mathbf{q}$ with $|supp(\mathbf{q})| \leq k$, together with their values. We will write $v_{a_1,\dots,a_k}(q_1, \dots, q_k)$ to denote the value of the partial coalition with support $\{a_1, \dots, a_k\}$ that receives q_i units of weight from agent a_i, $i = 1, \dots, k$.

An important advantage of this model is that it makes it relatively easy to deal with arbitration functions. Indeed, in a discrete OCF game each coalition structure consists of at most $n(\Psi + 1)$ partial coalitions. This means, in particular, that the input to the arbitration function can be represented using $\text{poly}(n, \Psi, ||\mathbf{x}||)$ bits, where $||\mathbf{x}||$ is the bitsize of the payoff vector $\mathbf{x}$. We will assume that our algorithms have oracle access to the arbitration function; the observation above means that in our model querying this oracle takes time polynomial in the game representation size.

REMARK 3.1. If the characteristic function of a discrete OCF game is efficiently computable, it can be encoded more succinctly by a circuit that takes the vector of agents' resources as its input and outputs the value of the corresponding partial coalition. More formally, this circuit would have $n\lceil\log(\Psi + 1)\rceil$ inputs: the i-th $\lceil\log(\Psi + 1)\rceil$-bit block of inputs would encode the contribution of agent i. Any discrete OCF game with a polynomial-time computable characteristic function $v : \mathcal{W} \to \mathbb{Q}_+$ can be represented by a circuit of size $\text{poly}(n, \log \Psi)$. This representation is succinct even if we do not limit the coalition sizes. However, it has two important disadvantages. First computing v^*, which is the most basic computational problem associated with an OCF game, becomes NP-hard even for $n = 1$; this follows by a straightforward reduction from the UNBOUNDED KNAPSACK problem [15] (a similar reduction, albeit in a slightly different context, can be found in [3]). Second, since the agents may form a coalition structure of size $\Omega(n\Psi)$, querying the arbitration function may be exponentially expensive in this model. Therefore, in what follows, we will not consider this representation.

4. K-OCF GAMES: FIRST OBSERVATIONS

The representation of a k-OCF game explicitly provides the value of each partial coalition $\mathbf{q}$. However, if we are interested in computing the total profit that can be earned by agents whose resources are given by $\mathbf{w}$, we need to take into account that these agents may split their resources among several partial coalitions. Thus, we need to compute the value of the superadditive cover v^* on $\mathbf{w}$. This computational problem is formalized as follows.

Name: OPTVAL
Input: A discrete k-OCF game over n players with maximum weight Ψ, a coalition $\mathbf{q} \in \mathcal{W}$, and a value r.
Question: Is $v^*(\mathbf{q}) \geq r$?

It is not hard to show that this problem is tractable if we additionally require that $|supp(\mathbf{q})|$ is bounded by a constant.

PROPOSITION 4.1. *Given a discrete OCF game and a partial coalition $\mathbf{q}$ with $|supp(\mathbf{q})| \leq t$, one can compute $v^*(\mathbf{q})$ in time $\text{poly}(\Psi^t)$.*

PROOF. We have $v^*(\mathbf{q}) = \max\{v^*(\mathbf{q}-\mathbf{r}) + v(\mathbf{r}) \mid \mathbf{r} \leq \mathbf{q}\}$. Thus, if we have computed $v^*(\mathbf{q}')$ for all $\mathbf{q}' < \mathbf{q}$, we can compute $v^*(\mathbf{q})$ in $\mathcal{O}((\Psi + 1)^t)$ time. Hence, we can compute $v^*(\mathbf{q})$ in time $\mathcal{O}(t(\Psi + 1)^{t+1})$ by dynamic programming. □

However, in general OPTVAL is computationally difficult, even if the maximum coalition size and the maximum weight are bounded by small constants.

THEOREM 4.2. OPTVAL *is* NP-*complete even if* $k \leq 2$, $\Psi \leq 3$.

PROOF. To see that this problem is in NP, observe that is suffices to guess a coalition structure $CS = (\mathbf{q}^1, \dots, \mathbf{q}^m)$ with $\sum_{j=1}^m q_i^j \leq w_i$ and $v(CS) \geq r$; note that the size of this coalition structure is at most $n(\Psi + 1)$, which is polynomial in the input size.

For the hardness proof, we provide a reduction from EXACT COVER BY 3-SETS (X3C) [10]. Recall that an instance of X3C is given by a finite set A, $|A| = 3\ell$, and a collection of subsets $\mathcal{S} = \{S_1, \dots, S_t\} \subseteq 2^A$ such that $|S_j| = 3$ for all $j = 1, \dots, t$. It is a "yes"-instance if A can be covered by exactly ℓ sets from $\mathcal{S}$. Given an instance $(A, \mathcal{S})$ of X3C, we construct a discrete OCF game with $k = 2$, $\Psi = 3$ as follows. We have an agent a_i of weight 1 for every element $i \in A$ and an agent a_S with weight 3 for every $S \in \mathcal{S}$. The characteristic function is defined as follows: $v_{a_i,a_S}(1, 1) = 2$ if $i \in S$, $v_{a_S}(3) = 5$, and the value of every other partial coalition is 0.

Let $S = \{x, y, z\}$ and consider $G_S = \{a_S, a_x, a_y, a_z\}$. Collectively, the agents in G_S can earn 6 if a_S forms a partial coalition with each of a_x, a_y, and a_z, and contributes one unit of weight to each of these coalitions; in any other coalition structure G_S earns at most 5. Hence, $(A, \mathcal{S})$ admits an exact cover if and only if $v^*(\mathbf{q}) \geq 6\ell + 5(t - \ell) = 5t + \ell$. □

Thus, discrete OCF games present a challenge from the computational perspective even if we severely restrict the maximum weight and coalition size. This means that in order to find tractable classes of such games, we must place further constraints on the coalitions that the agents are allowed to form. To identify the appropriate constraints, let us first consider 2-OCF games. Any such game can be naturally identified with a graph: the vertex set of this graph is N; there is an edge between i and j if there exists a partial coalition $\mathbf{q}$ such that $supp(\mathbf{q}) = \{i, j\}$ and $v(\mathbf{q}) > 0$. We will refer to this graph as the *interaction graph* of the game (N, v). Given this perspective, one may wonder if placing constraints on the interaction graph leads to tractable OCF games. In Section 5, we show that this is indeed the case: many computational problems for discrete 2-OCF games become tractable if the interaction graph is a tree. In Section 6, we extend these results to k-OCF games: such games can be associated with hypergraphs, and we show that many—though not all—of the results of Section 5 hold for k-OCF games whose interaction hypergraph has bounded treewidth.

All hardness results for OCF games derived so far stem from the complexity of computing the superadditive cover; therefore, to circumvent them, we place constraints on the characteristic function v. We now show that if we are interested in stability-related questions, we have to place constraints on the arbitration function $\mathcal{A}$ as well. Specifically, recall that $\mathcal{A}^*(CS, \mathbf{x}, S)$ computes the most that a set $S \subseteq N$ can earn when deviating from $(CS, \mathbf{x})$ under the arbitration function $\mathcal{A}$. In other words, when a set of agents S decides whether to deviate from $(CS, \mathbf{x})$, it needs to compute $\mathcal{A}^*(CS, \mathbf{x}, S)$. In the non-overlapping case, a given coalition S can easily decide whether it should deviate: it suffices to compute $v(S)$ and compare it with the payoff that S receives under $(CS, \mathbf{x})$. In contrast, $\mathcal{A}^*$ can be hard to compute even if $n = 2$ and both v and $\mathcal{A}$ are poly-time computable.

THEOREM 4.3. *If there exists a poly-time algorithm that for any discrete OCF game (N, v) any $CS \in \mathcal{CS}(N)$, any $\mathbf{x} \in \mathcal{I}(CS)$ and any $S \subseteq N$ can compute $\mathcal{A}^*(CS, \mathbf{x}, S)$ given oracle access to $\mathcal{A}$, then* P=NP. *This remains true even if the algorithm is only required to work when $|N| = 2$ and both v and $\mathcal{A}$ are poly-time computable.*

PROOF. We will show that if such an algorithm exists, it can be used to solve instances of SET COVER [10]. Recall that an instance of SET COVER is given by a set of elements A, a collection of subsets $\mathcal{S} = \{S_1, \dots, S_t\} \subseteq 2^A$ and $\ell \in \mathbb{N}$; it is a "yes"-instance if A can be covered by at most ℓ sets from $\mathcal{S}$.

Given an instance of SET COVER, consider a 2-player discrete OCF game where $w_1 = w_2 = t + 2$. The valuation function v is defined as follows. Each player gets payoff 1 for each unit of effort he invests in working on his own, i.e., $v(0, x) = v(x, 0) = x$ for $x = 0, \dots, t + 2$. Further, we have $v(1, 1) = 2$, $v(2, 2) = 10(t + 2)$; the value of v on other partial coalitions can be defined arbitrarily. Let $CS = (\mathbf{q}^1, \dots, \mathbf{q}^t, \mathbf{q}^{t+1})$, where $\mathbf{q}^i = (1, 1)$ for $i = 1, \dots, t$ and $\mathbf{q}^{t+1} = (2, 2)$. Let $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^{t+1})$ be a payoff vector that allocates all payoff from $\mathbf{q}^1, \dots, \mathbf{q}^t$ to 2 and splits the payoff from $\mathbf{q}^{t+1}$ equally between 1 and 2.

Now, recall that the input to the arbitration function $\mathcal{A}$ is an outcome $(CS, \mathbf{x})$ and a resource withdrawal pattern for the deviating coalition S. Let $(CS, \mathbf{x})$ be as constructed above, $S = \{1\}$, and suppose that player 1 withdraws from partial coalitions $\mathbf{q}^{i_1}, \dots, \mathbf{q}^{i_s}$, where $\{i_1, \dots, i_s\} \subseteq \{1, \dots, t\}$. We define $\mathcal{A}$ so that on this input player 1 receives nothing from $\mathbf{q}^1, \dots, \mathbf{q}^t$; moreover, player 1 keeps his payoff from $\mathbf{q}^{t+1}$ if and only if the collection $\{S_i \mid i \neq i_1, \dots, i_s\}$ is a cover for A; otherwise, player 1 gets nothing from $\mathbf{q}^{t+1}$. We can define $\mathcal{A}$ arbitrarily on other inputs; for concreteness, let us say that it coincides with the refined arbitrator. Note that $\mathcal{A}$ is normalized, deviation-monotone and polynomial-time computable.

If player 1 withdraws x units of resources from $\mathbf{q}^1, \dots, \mathbf{q}^t$, he can use them to earn x by working on his own. Thus, under $\mathcal{A}$ player 1 maximizes his payoff by withdrawing resources from as many coalitions among $\mathbf{q}^1, \dots, \mathbf{q}^t$ as possible, subject to the constraint that the coalitions he still contributes to correspond to a cover of A. Thus, $\mathcal{A}^*(CS, \mathbf{x}, \{1\}) \geq 5(t + 2) + (t - \ell)$ if and only if the input instance of SET COVER admits a cover of size at most ℓ. $\square$

Intuitively, the hardness of computing $\mathcal{A}^*$ stems from the fact that, when determining whether player 1 gets to keep his payoff from $\mathbf{q}^{t+1}$, the arbitration function bases its decision on player 1's global behavior. This motivates the following definition.

DEFINITION 4.4. *An arbitration function $\mathcal{A}$ for an OCF game (N, v) is said to be* local *if the payoff to a deviating set S from a coalition $\mathbf{q}$ is determined by the resources that S takes from $\mathbf{q}$.*

It is easy to see that the conservative, refined and optimistic arbitration functions that were defined in [3] are local. In contrast, the arbitration function used the proof of Theorem 4.3 is non-local. Another example of a non-local arbitration function is the *sensitive arbitrator* defined in [23]: under this arbitration function, the deviating set S keeps its payoff from a partial coalition $\mathbf{q}$ if none of the players in $\mathit{supp}(\mathbf{q}) \cap (N \setminus S)$ are hurt by the deviation.

Local arbitration functions are easier to work with, as they do not need to receive the entire coalition structure as their input; thus, a local arbitrator can be queried in polynomial time even assuming the circuit representation discussed in Remark 3.1. Even more importantly, they can be used to circumvent the hardness result of Theorem 4.3.

THEOREM 4.5. *For any discrete OCF game (N, v), given a local arbitration function $\mathcal{A}$, an outcome $(CS, \mathbf{x})$ and a set $S \subseteq N$, one can compute $\mathcal{A}^*(CS, \mathbf{x}, S)$ in time* $\mathrm{poly}(\Psi^{|S|})$.

PROOF. We first observe that a coalition structure CS has at most $(\Psi + 1)|S|$ coalitions that involve players in S. Given a coalition structure CS, let $\mathbf{q}^1, \dots, \mathbf{q}^m$ be the list of partial coalitions that receive contributions from both S and $N \setminus S$. Suppose that for $\ell = 1, \dots, m$, S's contribution to $\mathbf{q}^\ell$ is given by resource vector $\mathbf{r}^\ell \in \mathcal{W}(S)$, such that $\mathbf{r}^\ell \leq \mathbf{q}^\ell$; w.l.o.g., we assume that for each ℓ the vector $\mathbf{r}^\ell$ has at least one strictly positive coordinate. Now, suppose that players in S invest $\mathbf{s} \in \mathcal{W}(S)$ units of resources in partial coalitions among themselves and want to withdraw an additional $\mathbf{t} \in \mathcal{W}(S)$ from CS. They would get $v^*(\mathbf{s} + \mathbf{t})$ from working on their own, plus the most that S can get from the arbitration function, which depends on the coalitions affected by this deviation. Let us denote by $A(\mathbf{y}; \ell)$ the most that the arbitration function will give S if they withdraw $\mathbf{y}$ resources from the first ℓ coalitions, where $1 \leq \ell \leq m$. We also denote by $\mathcal{A}(\mathbf{z}; \ell)$ the payoff to S from coalition $\mathbf{q}^\ell$ if it withdraws $\mathbf{z} \leq \mathbf{r}^\ell$ from $\mathbf{q}^\ell$. We obtain

$$A(\mathbf{z}; \ell) = \max\{A(\mathbf{y}; \ell - 1) + \mathcal{A}(\mathbf{z} - \mathbf{y}; \ell) \mid 0 \leq \mathbf{y} \leq \mathbf{z}\}.$$

This shows that we can compute $A(\mathbf{z}; m)$ in $\mathcal{O}(m(\Psi + 1)^{|S|}) = \mathrm{poly}(\Psi|S|)$ steps. Now, the $\mathcal{A}^*(CS, \mathbf{x}, S)$ can be computed as $\max\{v^*(\mathbf{s} + \mathbf{t}) + A(\mathbf{t}; m) \mid 0 \leq \mathbf{t} \leq \mathbf{w}^S - \mathbf{s}\}$. $\square$

Theorem 4.5 implies that $\mathcal{A}^*$ can be computed in time polynomial in Ψ if $|S|$ is bounded by a constant. From this point onwards, we assume that $\mathcal{A}$ is local, unless explicitly stated otherwise.

5. 2-OCF GAMES ON TREES

In this section, we focus on discrete 2-OCF games where the interaction graph is a tree (all of our results generalize immediately to the case where this graph is a forest). We can root this tree at an arbitrary node; if $r \in N$ is chosen as the root, we denote by $C_i(r)$ the children of player i and by $T_i(r)$ the nodes of the subtree rooted at i. We omit r from the notation when it is clear from the context.

We can efficiently compute v^* for such games.

THEOREM 5.1. *If the interaction graph is a tree, $v^*(\mathbf{w})$ can be computed in time* $\mathrm{poly}(n, \Psi)$.

PROOF. We will show how to compute $v^*(\mathbf{w})$. To extend the algorithm to arbitrary $\mathbf{q}$, we can consider the game with the set of players $supp(\mathbf{q})$ and weights given by $\mathbf{q}$.

We arbitrarily choose some player $r \in N$ to be the root, and process the players starting from the leaves and moving towards the root. For each node i and each $w = 0, \dots, w_i$, let $u_i(w)$ denote the most that the players in T_i could earn if i had weight w. When processing i, we compute the quantities $u_i(0), \dots, u_i(w_i)$ based on the results of similar computations at each of i's children. $v^*(\mathbf{w}) = u_r(w_r)$, so once we reach the root, we output $u_r(w_r)$ and stop.

Consider first a leaf i. We have $u_i(w) = v_i^*(w)$, so, by Proposition 4.1, we can compute all $u_i(w)$, $0 \le w \le w_i$, in time $\mathcal{O}(\Psi^2)$.

Now consider an internal node i. Suppose that $C_i = \{i_1, \dots, i_\ell\}$ and for each $i_j \in C_i$ the quantities $u_{i_j}(w)$ for $w = 0, \dots, w_{i_j}$ have been computed. For $j = 0, \dots, \ell$, let $T_{i,j}$ be the tree obtained from T_i by removing subtrees rooted at $i_{j+1}, \dots, i_\ell$, and let $u_i(z;j)$ be the most that the players in $T_{i,j}$ could earn if i had weight z, $0 \le z \le w_i$. We have $u_i(z;0) = v_i^*(z)$. Further, having computed $u_i(z';j-1)$ for all $z' = 0, \dots, w_i$, we can compute $u_i(z;j)$ for all $z = 0, \dots, w_i$ as

$$u_i(z;j) = \max_{\substack{0 \le x \le z \\ 0 \le y \le w_{i_j}}} \{v^*_{i,i_j}(x,y) + u_i(z-x;j-1) + u_{i_j}(w_{i_j} - y)\}.$$

Indeed, players i and i_j need to decide how much weight to allocate to working together. Given this decision, they should optimally allocate their remaining weight to collaboration with, respectively, $i_1, \dots, i_{j-1}$ and T_{i_j}. The expression above optimizes over all choices available to i and i_j.

By Proposition 4.1, $v^*_{i,i_j}(x,y)$ can be computed in time $\mathcal{O}(\Psi^3)$, and $u_i(z-x;j-1)$, $u_{i_j}(w_{i_j} - y)$ have been pre-computed. Thus, for any fixed z this computation takes $\mathcal{O}(\Psi^5)$ steps, and computing all $u_i(z;j)$, $z = 0, \dots, w_j$, for a fixed value of k takes $\mathcal{O}(\Psi^6)$ steps. We clearly have $u_i(w) = u_i(w;\ell)$; hence, i can compute $u_i(0), \dots, u_i(w_i)$ in $\mathcal{O}(|C_i|\Psi^6)$ steps. As this computation has to performed at every internal node, the overall running time of our algorithm is $\sum_{i=1}^n \mathcal{O}(|C_i|\Psi^6) = \mathcal{O}(n\Psi^6)$. □

We now move on to the study of stability-related questions. The first problem we consider is computing $\mathcal{A}^*$, i.e., deciding whether a given coalition can profitably deviate under $\mathcal{A}$. In general, this problem is NP-hard even for discrete 2-OCF games and local arbitrators: this follows from Theorem 4.2, combined with the observation that $\mathcal{A}^*(CS, \mathbf{x}, N) = v^*(\mathbf{w})$ for any outcome $(CS, \mathbf{x})$ and any arbitration function $\mathcal{A}$. However, computing $\mathcal{A}^*$ becomes easy when the interaction graph is a tree.

THEOREM 5.2. *Given a discrete 2-OCF n-player game and a local arbitration function $\mathcal{A}$, if the interaction graph is a tree, we can compute $\mathcal{A}^*(CS, \mathbf{x}, S)$ for any $S \subseteq N$ and any outcome $(CS, \mathbf{x})$ in time* $\text{poly}(n, \Psi)$.

PROOF SKETCH. We use the algorithm given in the proof of Theorem 5.1, with the modification that each of the deviators also has to decide how much weight to keep in his collaboration with non-deviators; this decision is not too difficult since the interactions are between pairs of agents, and the arbitration function is local. In more detail, given an outcome $(CS, \mathbf{x})$ and a deviating set S, we construct a new discrete 2-OCF game where the set of players is S, and the characteristic function $\bar{v}$ is defined so that $\bar{v}_{i,j} \equiv v_{i,j}$ for $i, j \in S$ and $\bar{v}_i^*(w)$ outputs the most that player i can make by allocating w units of weight to working on his own *and with his neighbors from* $N \setminus S$ (this quantity depends on $\mathcal{A}$ and $(CS, \mathbf{x})$). It can be shown that $\bar{v}^*(S)$ can be computed by dynamic programming in time $\text{poly}(n, \Psi)$ and $\mathcal{A}^*(CS, \mathbf{x}, S) = \bar{v}^*(S)$; we omit the full proof due to space constraints. □

We are now ready to present an algorithm for checking whether a given outcome is in the $\mathcal{A}$-core. This problem is closely related to that of computing $\mathcal{A}^*$: an outcome $(CS, \mathbf{x})$ is in the $\mathcal{A}$-core if and only if the *excess* $e(CS, \mathbf{x}, S) = \mathcal{A}^*(CS, \mathbf{x}, S) - p_S(CS, \mathbf{x})$ is non-positive for all coalitions $S \subseteq N$. Thus, we need to check whether there exists a subset $S \subseteq N$ with $e(CS, \mathbf{x}, S) > 0$. Note that it suffices to limit our attention to connected subsets of N: if $e(CS, \mathbf{x}, S) > 0$ and S is not connected, then some connected component S' of S also satisfies $e(CS, \mathbf{x}, S') > 0$.

THEOREM 5.3. *If the interaction graph is a tree, we can verify whether a given outcome $(CS, \mathbf{x})$ is $\mathcal{A}$-stable in time* $\text{poly}(n, \Psi)$.

PROOF. Fix an outcome $(CS, \mathbf{x})$ and set $p_i = p_i(CS, \mathbf{x})$ for all $i \in N$.

Again, we pick an arbitrary $r \in N$ as a root. We say that $S \subseteq N$ is *rooted* at $i \in N$ if $i \in S$ and the members of S form a subtree of T_i. We observe that every set $S \subseteq N$ is rooted at a unique $i \in N$. Given a vertex i, let E_i denote the maximum excess of a set rooted at i. Clearly, $(CS, \mathbf{x})$ is not $\mathcal{A}$-stable if and only if $E_i > 0$ for some $i \in N$. We will now show how to compute E_i for all $i \in N$. We proceed from the leaves to the root, and terminate (and report that $(CS, \mathbf{x})$ is not $\mathcal{A}$-stable) if we discover a vertex i with $E_i > 0$. If $E_i \le 0$ for all $i \in N$, we report that $(CS, \mathbf{x})$ is $\mathcal{A}$-stable.

Given two agents $i, j \in N$, let $w_{i,j}$ denote the weight that i assigns to interacting with j.We will now define two auxiliary values. First, given a neighbor j of i, we define $\alpha_{i,j}(w)$ to be the most that $\mathcal{A}$ will give i if he keeps a total weight of $w \le w_{i,j}$ in the coalitions that he formed with j in $(CS, \mathbf{x})$; by Theorem 4.5, $\alpha_{i,j}(w)$ is computable in time $\text{poly}(\Psi)$. Second, we define $D_i(w)$ to be the maximum excess of a subset rooted at i if i were to contribute w to T_i and nothing to his parent $p(i)$. In this notation,

$$E_i = \max\{D_i(w) + \alpha_{i,p(i)}(y) \mid w + y = w_i, w \ge w_i - w_{i,p(i)}\};$$

the condition $w \ge w_i - w_{i,p(i)}$ ensures that $p(i)$ is not among the deviators. It remains to show how to compute $D_i(w)$ in time $\text{poly}(n, \Psi)$ for all $i \in N$ and $w_i - w_{i,p(i)} \le w \le w_i$.

Consider an agent i with children $C_i = \{i_1, \dots, i_\ell\}$, and suppose that we have computed $D_{i_j}(z)$ for each $i_j \in C_i$ and each z, $w_{i_j} - w_{i_j,i} \le z \le w_{i_j}$ (this encompasses the possibility that i is a leaf, as $C_i = \emptyset$ in that case). For $j = 0, \dots, \ell$, let $T_{i,j}$ be the tree obtained from T_i by removing subtrees rooted at $i_{j+1}, \dots, i_\ell$. Let $D_i(w;j)$ be the maximum excess of a set rooted at i that is fully contained in $T_{i,j}$, assuming that i contributes w to $T_{i,j}$ and nothing to his parent or his children $i_{j+1}, \dots, i_\ell$; we have $D_i(w) = D_i(w;\ell)$. We will compute $D_i(w;j)$ by induction on j.

We have $D_i(w;0) = v_i^*(w) - p_i$ for all $w = w_i - w_{i,p(i)}, \dots, w_i$. Now, consider $j > 0$. Agent i can either include i_j in the deviating set or deviate (partially or fully) from the coalitions that it forms with i_j in $(CS, \mathbf{x})$. Thus, $D_i(w;j) = \max\{D_1, D_2\}$, where

$$D_1 = \max_{\substack{y=0,\dots,w \\ z=0,\dots,w_{i_j}}} \{D_i(y;j-1) + v^*_{i,i_j}(w-y,z) + D_j(w_{i_j} - z)\}.$$

and

$$D_2 = \max_{z=0,\dots,w_{i,i_j}} \{D_i(w-z;j-1) + \alpha_{i,i_j}(z)\}.$$

Thus, we can efficiently compute $D_i(w;j)$, and hence also $D_i(w)$ and E_i. □

We have shown how to check whether a specific outcome is in the $\mathcal{A}$-core. We can use this algorithm as a subroutine to check whether the $\mathcal{A}$-core is non-empty. Specifically, we can go over all coalition structures in $\mathcal{CS}$, and, for each $CS \in \mathcal{CS}$, check if there exists a payoff vector $\mathbf{x} \in \mathcal{I}(CS)$ such that $(CS, \mathbf{x})$ is $\mathcal{A}$-stable: the conditions on $\mathbf{x}$ can be encoded by a linear program that, despite being exponential in size, admits a polynomial-time separation oracle, namely, the one constructed in Theorem 5.3. This implies that this linear program can be solved in polynomial time [19]; we omit the details of this argument due to space constraints. However, enumerating all candidate coalition structures is prohibitively expensive. We will now argue that, at least for the conservative core, this is not necessary: we will show how to explicitly construct an outcome in the conservative core of a discrete 2-OCF game on a tree.

Consider a coalition structure CS such that $v(CS) = v^*(\mathbf{w})$. For any $i, j \in N$ that are connected by an edge, we denote by $w_{i,j}$ the amount of weight that i devotes to interacting with j under CS; note that $w_{j,i}$ need not be equal to $w_{i,j}$. If we remove the edge (i, j), our tree splits into two trees: we will denote the vertex sets of these trees by $V_{i,j}$ and $V_{j,i}$, respectively (where $i \in V_{i,j}$, $j \in V_{j,i}$). We have

$$\begin{aligned} v^*(\mathbf{w}) &= v^*_{i,j}(w_{i,j}, w_{j,i}) + v^*(\mathbf{w}^{V_{i,j}} - w_{i,j}\mathbf{e}^{\{i\}}) \\ &+ v^*(\mathbf{w}^{V_{j,i}} - w_{j,i}\mathbf{e}^{\{j\}}). \end{aligned} \tag{1}$$

We observe that i and j can divide the value $v^*_{i,j}(w_{i,j}, w_{j,i})$ between themselves in any way they wish. Indeed, there is a coalition structure $CS_{i,j} \in \mathcal{CS}(\{i,j\})$ such that $v^*_{i,j}(w_{i,j}, w_{j,i}) = v(CS_{i,j})$, and every coalition in $CS_{i,j}$ receives contributions from both i and j.

We will now derive some constraints on the outcomes in the conservative core.

PROPOSITION 5.4. *If $(CS, \mathbf{x})$ is in the conservative core, then the total payoff to i from interacting with j is at least $v^*(\mathbf{w}^{V_{i,j}}) - v^*(\mathbf{w}^{V_{i,j}} - w_{i,j}\mathbf{e}^{\{i\}})$ and at most $v^*_{i,j}(w_{i,j}, w_{j,i}) - v^*(\mathbf{w}^{V_{i,j}}) + v^*(\mathbf{w}^{V_{i,j}} - w_{i,j}\mathbf{e}^{\{i\}})$.*

PROOF. Since $v^*(\mathbf{w}^{V_{i,j}}) + v^*(\mathbf{w}^{V_{j,i}}) \le v^*(\mathbf{w})$, from (1) we obtain $v^*(\mathbf{w}^{V_{i,j}}) - v^*(\mathbf{w}^{V_{i,j}} - w_{i,j}\mathbf{e}^{\{i\}}) \le v^*_{i,j}(w_{i,j}, w_{j,i}) - v^*(\mathbf{w}^{V_{j,i}}) + v^*(\mathbf{w}^{V_{j,i}} - w_{j,i}\mathbf{e}^{\{j\}})$.

If i gets less than $v^*(\mathbf{w}^{V_{i,j}}) - v^*(\mathbf{w}^{V_{i,j}} - w_{i,j}\mathbf{e}^{\{i\}})$ from interacting with j, then the total payoff to $V_{i,j}$ is less than $v^*(\mathbf{w}^{V_{i,j}})$, a contradiction with $(CS, \mathbf{x})$ being in the conservative core.

Similarly, if i gets more than $v^*_{i,j}(w_{i,j}, w_{j,i}) - v^*(\mathbf{w}^{V_{j,i}}) + v^*(\mathbf{w}^{V_{j,i}} - w_{j,i}\mathbf{e}^{\{j\}})$ from interacting with j, then j gets less than $v^*(\mathbf{w}^{V_{j,i}}) - v^*(\mathbf{w}^{V_{j,i}} - w_{j,i}\mathbf{e}^{\{j\}})$ from their interaction, which implies that the total payoff to $V_{j,i}$ is less than $v^*(\mathbf{w}^{V_{j,i}})$, a contradiction. □

Now, consider a coalition structure CS with $v^*(\mathbf{w}) = v(CS)$. We now show how to assign payoffs to agents so that the resulting outcome is in the conservative core; in doing so, we are guided by Proposition 5.4. Note that we do not construct a pre-imputation $\mathbf{x} \in \mathcal{I}(CS)$ explicitly. Rather, we simply indicate the cumulative payments to the agents: this is sufficient as long as we are only interested in the conservative core.

THEOREM 5.5. *For any discrete 2-OCF game $G = (N, v)$ whose interaction graph is a tree and any coalition structure $CS \in \mathcal{CS}(N)$ such that $v^*(\mathbf{w}) = v(CS)$, there is a payoff vector $\mathbf{x} \in \mathcal{I}(CS)$ such that $(CS, \mathbf{x})$ is in the conservative core.*

PROOF. Let CS be a coalition structure such that $v^*(\mathbf{w}) = v(CS)$. Let $r \in N$ be the root. Recall that T_i is the set of vertices of the tree rooted in i, i.e., if p is the parent of i then $T_i = V_{i,p}$. We allocate to agent $i \in N$:

- all payoff from coalitions he forms on his own;
- $v^*(\mathbf{w}^{T_i}) - v^*(\mathbf{w}^{T_i} - w_{i,p}\mathbf{e}^{\{i\}})$ from the interaction with his parent p (assuming $i \neq r$);
- $v^*_{i,j}(w_{i,j}, w_{j,i}) - v^*(\mathbf{w}^{T_i}) + v^*(\mathbf{w}^{T_i} - w_{i,j}\mathbf{e}^{\{i\}})$ from the interaction with each of his children $j \in C_i$.

This payoff division is feasible and efficient: the payoff from every edge (i, j) is split between i and j. Thus, there exists a pre-imputation $\mathbf{x} \in \mathcal{I}(CS)$ supporting these payoffs. It remains to show that the resulting outcome $(CS, \mathbf{x})$ is stable with respect to the conservative arbitrator. We will require the following lemma.

LEMMA 5.6. *Under the outcome $(CS, \mathbf{x})$, the total payoff to agents in $V_{i,j}$ is exactly $v^*(\mathbf{w}^{V_{i,j}})$, for any $i \in N$.*

PROOF. The lemma is clearly true if $i = r$. If $i \neq r$, let p be the parent of i. Agent i contributes weight $w_i - w_{i,p}$ to T_i. Since CS is an optimal coalition structure, agents in T_i earn $v^*(\mathbf{w}^{T_i} - w_{i,p}\mathbf{e}^{\{i\}})$, which they share among themselves. Further, i also receives $v^*(\mathbf{w}^{T_i}) - v^*(\mathbf{w}^{T_i} - w_{i,p}\mathbf{e}^{\{p\}})$ from his parent. Together, this adds up to $v^*(\mathbf{w}^{T_i})$. □

Now, suppose that a subset of agents S can profitably deviate from $(CS, \mathbf{x})$ by forming some $CS' \in \mathcal{CS}(S)$; assume that S is rooted at i. Let R consist of all vertices in $T_i \setminus S$ whose parents belong to S. Note that we have $T_i = S \cup (\cup_{j \in R} T_j)$. Now, consider a coalition structure over T_i where for each $j \in R$ the agents in T_j form the optimal coalition structure among themselves, and agents in S form CS'. By Lemma 5.6, in this new coalition structure the value of each T_j, $j \in R$, is the same as its payoff in $(CS, \mathbf{x})$. On the other hand, since S can profitably deviate using CS', $v(CS') > p_S(CS, \mathbf{x})$. We conclude that in this coalition structure the agents in T_i earn more than in $(CS, \mathbf{x})$. However, by Lemma 5.6 their total payoff in $(CS, \mathbf{x})$ is exactly $v^*(\mathbf{w}^{T_i})$, which is a contradiction. □

Theorem 5.5 does not hold for other arbitration functions, as the following example shows.

EXAMPLE 5.7. Consider a 3 player game where $w_1 = 2, w_2 = 2, w_3 = 1$. Also, $v_1(1) = 5, v_{1,2}(1,1) = 10, v_{2,3}(1,1) = 9$. The rest of the valuations are set to 0. One can verify that the refined core of this game is empty.

Note also that the payoff division proposed in Theorem 5.5 depends of the choice of the root, with nodes that are closer to the root reaping the benefits from the collaboration with their children. As a result, this payoff division scheme is not particularly "fair", as players who contribute equally to an interaction may not be paid equally. Consider for example a two-agent setting where both agents have a weight 1 and the value of their interaction is 1. While, intuitively, both players have equal claim to the profit, only one of them will get the payoff, while the other receives nothing.

6. INTERACTION HYPERGRAPHS WITH BOUNDED TREEWIDTH

While 2-OCF games correspond to graphs, k-OCF games with $k > 2$ can be modeled as *hypergraphs*, whose hyperedges are of size at most k: the vertex set of this hypergraph in N and there is an edge $E \subseteq N$ if and only if $v(\mathbf{q}) > 0$ for some $\mathbf{q} \in \mathcal{W}$ with $supp(\mathbf{q}) = E$. The resulting hypergraph is called the *interaction hypergraph* of the corresponding k-OCF game.

Several NP-hard combinatorial optimization problems on hypergraphs become tractable for hypergraphs whose treewidth is known to be bounded by a constant [18]. Problems in cooperative game theory are no exception: Ieong and Shoham [13] and Greco et. al [12] show that when a certain graphical representation of a cooperative game has bounded treewidth, several computational problems (e.g. deciding if the core is not empty) become tractable. Thus, it is only natural to ask whether our previous tractability results for 2-OCF games on trees can be extended to cases where the treewidth of the interaction hypergraphs in question is bounded by a constant. The answer appears to be mostly positive.

We begin by formally introducing the notion of *treewidth* of a hypergraph $H = (V, \mathcal{E})$ as given by Gottlob et al. [11]; this definition is based on the original notion of treewidth given in [18]. Given a hypergraph H, a *tree decomposition* of H is a tree $\mathcal{T}$ with node set $V(\mathcal{T})$ and edge set $E(\mathcal{T})$ such that each node $X \in V(\mathcal{T})$ is a non-empty subset $X \subseteq V$ of vertices of H. We require that if $E \in \mathcal{E}$, then there is some $X \in V(\mathcal{T})$ such that $E \subseteq X$. Moreover, the nodes are required to have the *running intersection property*: if $z \in X \cap Y$, then all nodes Z that are on the path between X and Y contain z as well. The *width* of a tree decomposition $\mathcal{T}$, denoted $\omega(\mathcal{T})$, equals $\max_{X \in V(\mathcal{T})}\{|X| - 1\}$. The *treewidth* of H, $tw(H)$, is the minimum of $\omega(\mathcal{T})$ over all tree decompositions of H. If $tw(H) = d$, then a tree decomposition of H with width d can be found in $\mathcal{O}(|H|^d)$ time.

Given a tree decomposition $\mathcal{T}$ of H with node set $V(\mathcal{T})$ and two nodes $X, Y \in V(\mathcal{T})$, we can associate the edge between X and Y with the set $X \cap Y$. Note that $X \cap Y$ is not empty if $(X, Y) \in E(\mathcal{T})$. Given a subtree $\mathcal{T}'$ of $\mathcal{T}$, we define $N(\mathcal{T}')$ to be $\bigcup_{X \in V(\mathcal{T}'} X$. We will now show how to adapt the proofs of Theorems 5.1, 5.2, and 5.3 for hypergraphs with bounded treewidth.

THEOREM 6.1. *Given a k-OCF game (N, v) whose interaction hypergraph admits a tree decomposition $\mathcal{T}$ of width d and a partial coalition $\mathbf{q} \in \mathcal{W}$, we can compute $v^*(\mathbf{q})$ in time* $\mathrm{poly}(n, \Psi^{d+1})$.

PROOF. We will give the proof for the case $\mathbf{q} = \mathbf{w}$; the general case can be handled similarly (see the proof of Theorem 5.1). We pick one of the sets $R \in V(\mathcal{T})$ to be the root of $\mathcal{T}$. Given a node $X \in V(\mathcal{T})$, we denote by $\mathcal{T}_X$ the subtree of $\mathcal{T}$ that is rooted at X and by $p(X)$ the parent of X in $\mathcal{T}$ (if $X = R$, we assume $p(X) = X$). For every vector $\mathbf{q} \in \mathcal{W}(X \cap p(X))$, we denote by $\mathcal{T}_X(\mathbf{r})$ the tree $\mathcal{T}_X$ with $X \cap p(X)$ devoting $\mathbf{r}$ to interacting with $\mathcal{T}_X$; note that $|X \cap p(X)| \leq d$. Let $u_X(\mathbf{r})$ denote the most that $N(\mathcal{T}_X(\mathbf{r}))$ can make; clearly, we have $v^*(\mathbf{w}) = u_R(\mathbf{w})$.

We will now show how to compute $u_X(\mathbf{q})$ for each node X and each $\mathbf{q} \in \mathcal{W}(X \cap p(X))$. As in the proof of Theorem 5.1, we proceed by dynamic programming, starting from the leaves and terminating at R. Fix a node X, and let $\mathcal{C}_X = \{C_1, \dots, C_\ell\}$ be the set of X's children; we denote by $u_X(\mathbf{q}; j)$ the most that $N(\mathcal{T}_X)$ can make if X devotes $\mathbf{q}$ to interacting with $C_1, \dots, C_j$ and none to the rest of its children. We have $u_X(\mathbf{q}; 0) = v^*(\mathbf{q})$; this quantity can be computed in time $\mathcal{O}((\Psi + 1)^d)$ by Theorem 4.1. Further, $u_X(\mathbf{q}; j)$ is given by

$$\max\{u_X(\mathbf{q} - \mathbf{y}; j - 1) + u_{C_j}(\mathbf{y}) \mid \mathbf{y} \in \mathcal{W}(C_j \cap X); \mathbf{y} \leq \mathbf{q}\}.$$

The requirement that $\mathbf{y} \leq \mathbf{q}$ is necessary, as $C_j \cap p(X)$ may be non-empty, in which case the amount that $C_j \cap X$ can give to $\mathcal{T}_{C_j}$ is limited by its previous commitment to the parent of X. Hence, we can compute $u_X(\mathbf{q}) = u_X(\mathbf{q}; \ell)$ in time linear in $|\mathcal{C}_X|$ and polynomial in Ψ^{d+1}; summing over all nodes of $\mathcal{T}$, we obtain the desired bound on the running time. □

We can use similar techniques to compute the most that a set can get by $\mathcal{A}$-deviating from some outcome $(CS, \mathbf{x})$.

THEOREM 6.2. *Given a k-OCF game, an outcome $(CS, \mathbf{x})$ and a set $S \subseteq N$ such that the interaction graph induced by S has treewidth d, we can compute $\mathcal{A}^*(CS, \mathbf{x}, S)$ in* $\mathrm{poly}(n, \Psi^{d+1})$ *time.*

PROOF SKETCH. We denote by $\alpha_L(\mathbf{w})$ the most that $\mathcal{A}$ will give a subset $L \subseteq S$ of size at most $d + 1$ if it decides to leave $\mathbf{q} \in \mathcal{W}(L)$ of its weight allocated to non-S members. Since the support of any coalition contains at most $d + 1$ players, computing $\alpha_L(\mathbf{w})$ can be done in a similar manner to Theorem 5.2. Now, given a tree decomposition of S with width d, we again replace the most that any subset $L \subseteq S$ with $|L| \leq d + 1$ can make with the value $\bar{v}^*(\mathbf{q}) = \max\{v^*(\mathbf{x} + \mathbf{y}) + \alpha_i(\mathbf{q} - \mathbf{y}) \mid 0 \leq \mathbf{y} \leq \mathbf{q}\}$ and repeat the computation described in Theorem 6.1. Correctness holds for similar reasons to those described in Theorem 5.2. □

We can also provide an analogue to Theorem 5.3; we omit the proof due to space constraints.

THEOREM 6.3. *Given a k-OCF game whose interaction hypergraph H has treewidth at most d, we can check if an outcome $(CS, \mathbf{x})$ is in the $\mathcal{A}$-core in time* $\mathrm{poly}(n, \Psi^{d+1})$.

Using Theorem 6.3, we obtain the following corollary.

COROLLARY 6.4. *Suppose that the treewidth of the interaction hypergraph of a discrete k-OCF game is at most d. Then, given a coalition structure CS and an arbitration function $\mathcal{A}$, we can check in time* $\mathrm{poly}(n, \Psi^{d+1})$ *if there exists an imputation $\mathbf{x}$ such that $(CS, \mathbf{x})$ is $\mathcal{A}$-stable (and output $\mathbf{x}$ if it exists).*

Briefly, this problem can be encoded as a linear program. Even though this program has exponentially many constraints, it can be solved in time $\mathrm{poly}(n, \Psi^{d+1})$ using the algorithm described in the proof of Theorem 6.3 as a separation oracle.

Finally, we remark that even the conservative core of OCF games with bounded treewidth may be empty. Thus, an analogue of Theorem 5.5 does not hold.

EXAMPLE 6.5. Consider a 2-OCF game with $N = \{1, 2, 3\}$ and $w_i = 1$ for all $i \in N$. Set $v_{i,j}(1, 1) = 1$ for any $i \neq j \in N$, and suppose that $v \equiv 0$ for all other partial coalitions; this is essentially the classic *3-player majority game* [2], which is known to have an empty core. The argument for the classic case can be adapted to show that the conservative core of our game is empty.

7. RELATED WORK

Our work builds directly on the overlapping coalition formation framework of [3, 23]. The main difference between our model and that of [3, 23] is the assumption that the agents' resources are discrete; however, all theoretical results proven in these papers can be shown to hold for the discretized setting.

Restricting the size of admissible coalitions to ensure tractability is a fairly standard approach, see, e.g., the classic work of Shehory and Kraus [20]. More recently, Shrot et al. [22] and Chitnis et al. [5] investigated the parameterized complexity of (non-overlapping) coalitional games, with the maximum coalition size as a parameter. They show that, in the absence of additional constraints on the characteristic function, restricting the coalition size is insufficient for tractability; this is consistent with our results (Theorem 4.2).

Many of our tractability results rely on restricting the interaction between the agents to trees and tree-like structures. This bears close similarity to models in classic cooperative game theory that limit agent interaction. The first such model, proposed by Myerson [17], describes cooperative games where agent interaction is limited by an underlying graph structure; in this model, a coalition may form only if it corresponds to a connected subgraph. Cooperative games on graphs have been subsequently studied by a number of authors; see, e.g., [7, 9, 16, 17]. In particular, Demange [7] describes cooperative games where agents form a hierarchical tree structure and proposes an algorithm that computes a core allocation in this setting; our analysis of 2-OCF games on trees is somewhat similar to this work. However, deriving results in the OCF model is significantly more complicated than in the non-overlapping setting, and the algorithm of [7] cannot be applied directly to our model. Section 6 is inspired by the work of Ieong and Shoham [13] and Greco et al. [12], who analyze the complexity of core-related solution concepts for interaction graphs with bounded treewidth in the non-overlapping setting.

Recently, Anshelevitz and Hoefer [1] introduced *network contribution games*: in these games, each agent has a weight that he may divide among his neighbors, and the value of an interaction depends on the weight each agent devotes to the edge. Their analysis differs from ours in that they assume that the payoff from an edge is divided equally between the agents and study the resulting *non-cooperative* game.

Finally, we remark that the study of computational aspects of coalitional games is a well-established research topic, which received a significant amount of attention in recent years; see, e.g., [4]. Our work makes the first step towards extending this analysis to OCF games.

8. CONCLUSIONS AND FUTURE WORK

Finding optimal coalition structures and stable outcomes are key issues in the analysis of OCF games; we show that these problems are hard in general, but formulate several conditions that make them tractable. We mostly focus on 2-OCF games and acyclic agent interaction graphs; however, we show that our results extend to k-OCF games with constant k and interaction (hyper-)graphs with bounded treewidth.

While our work focuses on achieving computational efficiency by restricting agent interaction, one can also obtain tractability results for OCF games by other means. A natural way of doing so is to extend existing representation languages for non-overlapping coalitional games, such as, e.g., MC-nets [13]—and the algorithms for them—to the OCF setting; the analysis of threshold task games in [3] can be viewed as an example of this approach.

Another way of dealing with hardness results is by designing *approximation* algorithms, i.e., procedures that output a coalition structure that is almost optimal and/or stable. Designing such algorithms (or proving hardness of approximation results) is a fruitful direction for future research.

We have focused mostly on one solution concept: the arbitrated core. Other solution concepts, such as the arbitrated nucleolus, have been proposed and analyzed in [23]. It would be interesting to analyze the computational complexity of finding a nucleolus outcome, or the Shapley value of an agent in cooperative games with overlapping coalitions.

Acknowledgements

This research was supported by the National Research Foundation (Singapore) under grant 2009-08, by NTU SUG (Edith Elkind), and by SINGA graduate fellowship (Yair Zick).

9. REFERENCES

[1] E. Anshelevich and M. Hoefer. Contribution games in social networks. In *ESA'10*, pages 158–169, 2010.

[2] R. Brânzei, D. Dimitrov, and S. Tijs. *Models in cooperative game theory*. Springer, 2005.

[3] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. Jennings. Cooperative games with overlapping coalitions. *Journal of AI Research*, 39:179–216, 2010.

[4] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan and Claypool, 2011.

[5] R. H. Chitnis, M. T. Hajiaghayi, and V. Liaghat. Parameterized complexity of problems in coalitional resource games. In *AAAI'11*, pages 620–626, 2011.

[6] V. D. Dang, R. K. Dash, A. Rogers, and N. R. Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *AAAI'06*, pages 635–640, 2006.

[7] G. Demange. On group stability in hierarchies and networks. *Journal of Political Economy*, 112(4):754–778, 2004.

[8] X. Deng and C. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[9] U. Faigle. Cores of games with restricted cooperation. *Mathematical Methods of Operations Research*, 33(6):405–422, 1989.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractibility*. W. H. Freeman and Company, 1979.

[11] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions: A survey. In *MFCS'01*, pages 37–57, 2001.

[12] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of core, kernel, and bargaining set. *Artificial Intelligence*, 175(12–13):1877–1910, 2011.

[13] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *ACM EC'05*, pages 193–202. ACM, 2005.

[14] V. Lesser. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Trans. on Knowl. and Data Eng.*, 11:133–142, 1999.

[15] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, 1990.

[16] R. Meir, J. S. Rosenschein, and E. Malizia. Subsidies, stability, and restricted cooperation in coalitional games. In *IJCAI'11*, pages 301–306, 2011.

[17] R. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3):225–229, 1977.

[18] N. Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *J Comb Theory, B*, 36(1):49–64, 1984.

[19] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[20] O. Shehory and S. Kraus. Task allocation via coalition formation among autonomous agents. In *IJCAI'95*, pages 655–661, 1995.

[21] O. Shehory and S. Kraus. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In *ICMAS'96*, pages 330–337, 1996.

[22] T. Shrot, Y. Aumann, and S. Kraus. Easy and hard coalition resource game formation problems: a parameterized complexity analysis. In *AAMAS'09*, pages 433–440, 2009.

[23] Y. Zick and E. Elkind. Arbitrators in overlapping coalition formation games. In *AAMAS'11*, pages 55–62, 2011.