Bounding the Cost of Stability in Games over Interaction Networks

Reshef Meir[1], **Yair Zick**[2], **Edith Elkind**[2], and **Jeffrey S. Rosenschein**[1]

[1]School of Engineering and Computer Science, The Hebrew University of Jerusalem, Israel
`{reshef24,jeff}@cs.huji.ac.il`
[2]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
`{yair0001,eelkind}@ntu.edu.sg`

Abstract

We study the stability of cooperative games played over an interaction network, in a model that was introduced by Myerson (1977). We show that the cost of stability of such games (i.e., the subsidy required to stabilize the game) can be bounded in terms of natural parameters of their underlying interaction networks. Specifically, we prove that if the treewidth of the interaction network H is k, then the relative cost of stability of *any* game played over H is at most $k + 1$, and if the pathwidth of H is k', then the relative cost of stability is at most k'. We show that these bounds are tight for all $k \geq 2$ and all $k' \geq 1$, respectively.

1. Introduction

Coalitional game theory models scenarios where groups of agents can work together profitably; the agents form *coalitions*, and each coalition generates a payoff, which then needs to be shared among the members of that coalition. The agents are assumed to be selfish, so the payoffs should be divided in such a way that each agent is satisfied with his share. In particular, it is desirable to allocate the payoffs so that no group of agents can do better by abandoning their coalitions and embarking on a project of their own; the set of all payoff division schemes that have this property is known as the *core* of the game. However, this requirement turns out to be very strong, as many games have an empty core.

There are several ways to capture the intuition behind the notion of the core, while relaxing the core constraints. For instance, one can assume that deviation comes at a cost, so players will not deviate unless the profit from doing so exceeds a certain threshold; formalizing this approach leads to the notions of ε-*core* and *least core*. Another approach, pioneered by Myerson (1977), assumes that communication among agents may be limited, and that agents cannot deviate unless they can communicate with one another. In more detail, the game has an underlying interaction network, called the *Myerson graph*; agents are nodes, and an edge indicates the presence of a communication link. Permissible coalitions correspond to connected subgraphs of the Myerson graph. Finally, stability may be achieved via *subsidies*: an external party may try to stabilize the game by offering a lump sum to the agents if they form some desired coalition structure. The minimum subsidy required to guarantee stability is known as the *cost of stability (CoS)* (Bachrach et al. 2009). In what follows, we use the *relative cost of stability (RCoS)* (Meir, Rosenschein, and Malizia 2011), which is defined as the ratio between the minimum total payoff needed to ensure stability and the total value of an optimal coalition structure.

In this paper, we study the interplay between restricted interaction and the cost of stability. Our goal is to bound the relative cost of stability in terms of structural properties of the interaction network. One such property is the *treewidth*: this is a combinatorial measure of graph structure that, intuitively, says how close a graph is to being a tree. A closely related notion is that of *pathwidth*, which measures how close a graph is to being a path. Breton, Owen and Weber (1992) have demonstrated a connection between structure and stability by showing that if the Myerson graph is a tree then the core of the game is non-empty. This result was later reproduced by Demange (2004), who also provided an efficient algorithm for constructing a core imputation. It is thus natural to ask if games whose Myerson graphs have small treewidth are close to having a non-empty core.

Related Work There is a significant body of work on subsidies in cooperative games. Many of the earlier papers focused on *cost-sharing games*, where agents share the *cost* of a project, rather than its profits (see, for example, (Jain and Vazirani 2001; Devanur, Mihail, and Vazirani 2005)). For profit-sharing games, Bachrach et al. (2009) have recently introduced the notion of cost of stability (CoS), which is defined as the minimum subsidy needed to stabilize such games. Bachrach et al. gave bounds on the cost of stability for several classes of coalitional games, and analyzed the complexity of computing the cost of stability in weighted voting games. Several groups of researchers have extended this analysis to other classes of coalitional games (Resnick et al. 2009; Meir, Bachrach, and Rosenschein 2010; Aziz, Brandt, and Harrenstein 2010; Meir, Rosenschein, and Malizia 2011; Greco et al. 2011a; 2011b). In particular, Meir et al. (2011) and Greco et al. (2011b) studied questions related to the CoS in games with restricted cooperation in the Myerson model, providing bounds on the CoS for some simple graphs.

It is well-known that many graph-related problems that are computationally hard in the general case become tractable once the treewidth of the underlying graph is bounded by a constant (see, e.g., (Courcelle 1990)). There are several graph-based representation languages for cooperative games, and for many of them the complexity of computational questions that arise in cooperative game theory (such as finding an outcome in the core or an optimal coalition structure) can be bounded in terms of the treewidth of the corresponding graph (Ieong and Shoham 2005; Aziz et al. 2009; Bachrach et al. 2010; Greco et al. 2011a; Voice, Polukarov, and Jennings 2012). However, in general bounding the treewidth of the Myerson graph (except for the special case of width 1) *does not* lead to a tractable solution for these computational questions, as shown by Greco et al. (2011b) and by Chalkiadakis et al. (2012).

Our Contribution We provide a complete characterization of the relationship between the treewidth of the interaction network and the worst-case cost of stability. We prove that for *any game* G played over a network of treewidth k, its relative cost of stability is at most $k + 1$, and this bound is tight whenever $k \geq 2$. A similar result with respect to the *pathwidth* of the interaction network is also given. These results stand in sharp contrast to the observation that bounding the treewidth of the Myerson graph does not lead to efficient algorithms (except on a tree). To the best of our knowledge, our work is the first to employ treewidth in order to prove a game-theoretic result that is not algorithmic in nature. We conclude by highlighting several implications of our results for some classes of games defined on graphs and hypergraphs. Some proofs have been deferred to the appendix.

2. Preliminaries

In what follows, we use boldface lowercase letters to denote vectors, and uppercase letters to denote sets of agents.

A *transferable utility (TU) game* is a tuple $G = \langle N, v \rangle$, where $N = \{1, \ldots, n\}$ is a finite set of *agents* and $v : 2^N \to \mathbb{R}$ is the *characteristic function* of the game. By convention $v(\emptyset) = 0$. Also, unless explicitly stated otherwise, we restrict our attention to games where $v(S) \geq 0$ for all $S \subseteq N$.

A TU game $G = \langle N, v \rangle$ is *superadditive* if $v(S \cup T) \geq v(S) + v(T)$ for every $S, T \subseteq N$ such that $S \cap T = \emptyset$; it is *monotone* if $v(S) \leq v(T)$ for every $S, T \subseteq N$ such that $S \subseteq T$. Further, G is said to be *simple* if for all $S \subseteq N$ it holds that $v(S) \in \{0, 1\}$. Note that we *do not* require simple games to be monotone; this allows us to use an inductive argument in Section 3. A coalition S in a simple game $G = \langle N, v \rangle$ is *winning* if $v(S) = 1$ and *losing* if $v(S) = 0$.

Following Aumann and Dréze (1974), we assume that agents may form coalition structures. A *coalition structure* over N is a partition of N into disjoint subsets. We denote the set of all coalition structures over N by $\mathcal{CS}(N)$. Given a $CS \in \mathcal{CS}(N)$ we define its *value* $v(CS)$ as $v(CS) = \sum_{S \in CS} v(S)$ and set $CS_+ = \{S \in CS \mid v(S) > 0\}$.

Let $OPT(G) = \max\{v(CS) \mid CS \in \mathcal{CS}(N)\}$. A coalition structure CS is said to be *optimal* if $v(CS) = OPT(G)$. Note that if G is superadditive, $\{N\}$ is optimal.

Payoffs and Stability Having split into coalitions and generated profits, agents need to divide the gains among themselves. A *payoff vector* is simply a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n_+$, where the i-th coordinate is the payoff to agent $i \in N$. We denote the total payoff to a set $S \subseteq N$ by $x(S)$, i.e., we write $x(S) = \sum_{i \in S} x_i$. We say that a payoff vector $\mathbf{x}$ is a *pre-imputation* for a coalition structure CS if for all $S \in CS$ it holds that $x(S) = v(S)$. A pair of the form $(CS, \mathbf{x})$, where $CS \in \mathcal{CS}(N)$ and $\mathbf{x}$ is a pre-imputation for CS, is referred to as an *outcome* of the game $G = \langle N, v \rangle$; an outcome is *individually rational* if $x_i \geq v(\{i\})$ for every $i \in N$. If $\mathbf{x}$ is a pre-imputation for CS that is individually rational, it is called an *imputation* for CS. We say that an outcome $(CS, \mathbf{x})$ of a game $G = \langle N, v \rangle$ is *stable* if $x(S) \geq v(S)$ for all $S \subseteq N$. The set of all stable outcomes of G is called the *core* of G, and is denoted $Core(G)$. We denote by $\mathcal{S}(G)$ the set of all payoff vectors (not necessarily pre-imputations) that satisfy the stability constraints:

$$\mathcal{S}(G) = \{\mathbf{x} \in \mathbb{R}^n_+ \mid x(S) \geq v(S) \text{ for all } S \subseteq N\}.$$

We refer to elements of $\mathcal{S}(G)$ as stable payoff vectors.

The *Relative Cost of Stability* (RCoS) of a game G is the smallest total payoff that stabilizes the game:

$$RCoS(G) = \inf \left\{ \frac{x(N)}{OPT(G)} \mid \mathbf{x} \in \mathcal{S}(G) \right\}.$$

Note that $RCoS(G) \geq 1$ for every TU game G, and $RCoS(G) = 1$ implies $Core(G) \neq \emptyset$.

Interaction Networks and Treewidth An *interaction network* (also called a *Myerson graph*) over N is a graph $H = \langle N, E \rangle$. Given a game $G = \langle N, v \rangle$ and an interaction network over N, we define a game $G|_H = \langle N, v|_H \rangle$ by setting $v|_H(S) = v(S)$ if S is a connected subgraph of H, and $v|_H(S) = 0$ otherwise; that is, in $G|_H$ a coalition $S \subseteq N$ may form if and only if its members are connected.

A *tree decomposition* of H is a tree $\mathcal{T}$ over the nodes $V(\mathcal{T})$ such that: a) Each node of $\mathcal{T}$ is a subset of N. b) For every pair of nodes $X, Y \in V(\mathcal{T})$ and every $i \in N$, if $i \in X$ and $i \in Y$ then for any node Z on the (unique) path between X and Y in $\mathcal{T}$ we have $i \in Z$. c) For every edge $e = \{i, j\}$ of E there exists a node $X \in V(\mathcal{T})$ such that $e \subseteq X$.

The *width* of a tree decomposition $\mathcal{T}$ is $tw(\mathcal{T}) = \max_{X \in V(\mathcal{T})} |X| - 1$; the *treewidth* of H is defined as $tw(H) = \min\{tw(\mathcal{T}) \mid \mathcal{T} \text{ is a tree decomposition of } H\}$. Examples of graphs with low treewidth include trees (whose treewidth is 1) and series-parallel graphs (whose treewidth is at most 2); see, e.g., (Bodlaender 2005).

Given a subtree $\mathcal{T}'$ of a tree decomposition $\mathcal{T}$ (we use the term "subtree" to refer to any connected subgraph of $\mathcal{T}$), we denote the agents that appear in the nodes of $\mathcal{T}'$ by $N(\mathcal{T}')$. Conversely, given a set of agents $S \subseteq N$, let $\mathcal{T}(S)$ denote the subgraph of $\mathcal{T}$ induced by nodes $\{X \in V(\mathcal{T}) \mid X \cap S \neq \emptyset\}$; it is not hard to check that $\mathcal{T}(S)$ is a subtree of $\mathcal{T}$ for every $S \subseteq N$. Given a tree decomposition $\mathcal{T}$ of H and a node $R \in V(\mathcal{T})$, we can set R to be the root of $\mathcal{T}$. In this case, we denote the subtree rooted in a node $S \in V(\mathcal{T})$ by $\mathcal{T}_S$.

A tree decomposition of a graph H such that $\mathcal{T}$ is a path is called a *path decomposition* of H. The *pathwidth* of H is

$pw(H) = \min\{tw(\mathcal{T}) \mid \mathcal{T} \text{ is a path decomposition of } H\}$. For any graph H, $tw(H) \leq pw(H) \leq O(tw(H)\log(n))$.

3. Treewidth and the Cost of Stability

Our goal in this section is to provide a general upper bound on the cost of stability for TU games whose interaction networks have bounded treewidth. We start by proving a bound for simple games; we then show how to extend it to the general case. However, prior to proving our main result, we refute an alternative suggestion by Meir et al. (2011).

RCoS and the degree of H Meir et al. conjectured that $RCoS(G|_H) \leq d(H)$, where $d(H)$ is the maximum degree of a node in H. They also claimed that this bound is tight (if true), using the projective plane as an example.

Our next proposition shows that this conjecture is false. Moreover, the "tight" example given by Meir et al. is incorrect: the game G_q that corresponds to the projective plane of dimension q satisfies $q \leq RCoS(G_q) \leq q + 1$ (see (Bachrach et al. 2009)), but it can be shown that for any interaction network H such that $G_q|_H = G_q$ it holds that the degree of H is at least $2q$.

Proposition 1. *There exists an interaction network H with $d(H) = 6$ such that for any $k \in \mathbb{N}$ there exists a simple superadditive game G with $RCoS(G_H) \geq k$.*

Simple Games

We now show that for any simple game $G = \langle N, v \rangle$ and an interaction network H over N, $RCoS(G|_H) \leq tw(H) + 1$. Our proof is constructive: we show that Algorithm 1, whose input is a simple game $G = \langle N, v \rangle$, a network H, a parameter k, and a tree decomposition $\mathcal{T}$ of H of width at most k, outputs a stable payoff vector $\mathbf{x}$ for $G|_H$ such that $x(N) \leq (tw(H) + 1) \cdot OPT(G|_H)$. Briefly, Algorithm 1 picks an arbitrary node $R \in V(\mathcal{T})$ to be the root of $\mathcal{T}$ and traverses the nodes of $\mathcal{T}$ from the leaves towards the root. Upon arriving at a node A, it checks whether the subtree $\mathcal{T}_A$ contains a coalition that is winning in $G|_H$ (note that we have to check every subset of $N(\mathcal{T}_A) \cap N_t$, since $G|_H$ is not necessarily monotone). If this is the case, it pays 1 to all agents in A and removes all agents in $\mathcal{T}_A$ from every node of $\mathcal{T}$. Note that every winning coalition in $\mathcal{T}_A$ has to be connected, so either it is fully contained in a proper subtree of $\mathcal{T}_A$ or it contains agents in A. The reason for deleting the agents in $\mathcal{T}_A$ is simple: every winning coalition that contains members of $\mathcal{T}_A$ is already stable (one of its members is getting a payoff of 1). The algorithm then continues up the tree in the same manner until it reaches the root. Note that Algorithm 1 is similar to the one proposed by Demange (2004); however, Algorithm 1 may pay $2 \cdot OPT(G|_H)$ if H is a tree.[1] Moreover, unlike Demange's algorithm, Algorithm 1 may require exponential time, since it is designed to work for non-monotone simple games. However, if the simple

---

[1] This is because Algorithm 1 operates on $\mathcal{T}$, which has nodes of size 2. In this special case we can modify our algorithm by only paying one of the agents in A—the one that does not appear above A in the tree. The resulting payoff vector would then coincide with the one constructed by Demange's algorithm.

Algorithm 1: STABLE-TW($G = \langle N, v \rangle, H, k, \mathcal{T}$)

> Fix an arbitrary $R \in V(\mathcal{T})$ to be the root;
> $t \leftarrow 0$, $N_1 \leftarrow N$, $\mathbf{x} \leftarrow 0^n$;
> **for** $A \in V(\mathcal{T})$, *traversed from the leaves upwards* **do**
> > $t \leftarrow t + 1$;
> > **if** $\exists S \subseteq N(\mathcal{T}_A) \cap N_t$ *s.t.* $v|_H(S) = 1$ **then**
> > > **for** $i \in A \cap N_t$ **do**
> > > > $x_i \leftarrow 1$
> > >
> > > $N_{t+1} \leftarrow N_t \setminus N(\mathcal{T}_A)$;
> >
> > **else**
> > > $N_{t+1} \leftarrow N_t$;
>
> **return** $\mathbf{x} = (x_1, \ldots, x_n)$;

---

game given as input is monotone, a straightforward modification (check whether $v|_H(S) = 1$ only for $S = N(\mathcal{T}_A)$ rather than for every $S \subseteq N(\mathcal{T}_A)$) makes it run in polynomial time.

Theorem 2. *For every simple game $G = \langle N, v \rangle$ and every interaction network H over N, $RCoS(G|_H) \leq tw(H) + 1$.*

Proof. Let $\mathcal{T}$ be a tree decomposition of H such that $tw(\mathcal{T}) = k$. Let $\mathbf{x}$ be the output of Algorithm 1. We claim that $\mathbf{x}$ is stable and $x(N) \leq (k + 1)OPT(G|_H)$.

To prove stability, consider a coalition S with $v|_H(S) = 1$; we need to show that $x(S) > 0$. Suppose for the sake of contradiction that $x(S) = 0$; this means that each agent in S is deleted before he is allocated any payoff. Consider the first time-step when an agent in S is deleted; suppose that this happens at step t when a node $A \in V(\mathcal{T})$ is processed. Clearly for an agent in S to be deleted at this step it has to be the case that $\mathcal{T}(S) \cap \mathcal{T}_A \neq \emptyset$. Further, it cannot be the case that $S \cap (A \cap N_t) \neq \emptyset$, since each agent in $A \cap N_t$ is assigned a payoff of 1 at step t, and we have assumed that $x(S) = 0$. Therefore, $\mathcal{T}(S)$ must be a proper subtree of $\mathcal{T}_A$. Let B be the root of $\mathcal{T}(S)$, and consider the time-step $t' < t$ when B is processed. At time t', all agents in S are still present in $\mathcal{T}$, so the node B meets the **if** condition in Algorithm 1, and therefore each agent in B gets assigned a payoff of 1. This is a contradiction, since B is the root of $\mathcal{T}(S)$, and therefore $B \cap S \neq \emptyset$, which implies $x(S) > 0$.

It remains to show that $x(N) \leq (k + 1)OPT(G|_H)$. To this end, we will construct a specific coalition structure CS^* and argue that $x(N) \leq (k + 1)v|_H(CS^*)$. The coalition structure CS^* is constructed as follows. Let A_t be the node of the tree considered by Algorithm 1 at time t, and let $S_t = N(\mathcal{T}_{A_t}) \cap N_t$, i.e., S_t is the set of all agents that appear in $\mathcal{T}_{A_t}$ at time t. Let T^* be the set of all values of t such that A_t meets the **if** condition in Algorithm 1. For each $t \in T^*$ the set S_t contains a winning coalition; let W_t be an arbitrary winning coalition contained in S_t. Finally, let $L = N \setminus (\cup_{t \in T^*} W_t)$, and set $CS^* = \{L\} \cup \{W_t \mid t \in T^*\}$.

Observe that CS^* is a coalition structure, i.e., a partition of N. Indeed, $L \cap W_t = \emptyset$ for all $t \in T^*$, and, moreover, if $i \in W_t$ for some $t > 0$, then i was removed from $\mathcal{T}$ at time t, and cannot be a member of coalition $W_{t'}$ for $t' > t$. Further, we have $v_H(CS^*) \geq |T^*|$.

To bound the total payment, we observe that no agent is assigned any payoff at time $t \notin T^*$, and each agent that is assigned a payoff of 1 at time $t \in T^*$ is a member of A_t. Hence we have

$$x(N) = \sum_{t \in T^*} x(A_t) \leq \sum_{t \in T^*} |A_t| \leq (k+1)|T^*|$$
$$\leq (k+1)v|_H(CS^*) \leq (k+1)OPT(G),$$

which proves that $RCoS(G) \leq k+1$. $\square$

We remark that under the payment scheme constructed by Algorithm 1 the payoff of every agent is either 1 or 0. Note also that the proof of Theorem 2 goes through as long as $G|_H$ is simple, even if G itself is not simple.

The General Case

Using Theorem 2, we are now ready to prove our main result.

Theorem 3. *For every game $G = \langle N, v \rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq tw(H) + 1$.*

Proof. Given a game $G' = \langle N, v' \rangle$, let $\#(G') = |\{S \subseteq N \mid v'(S) > 0\}|$. We prove the theorem by induction on $\#(G|_H)$. If $\#(G|_H) = 1$ then $RCoS(G|_H) = 1$: any outcome of this game where the positive-value coalition forms is stable. Now suppose that our claim is true whenever $\#(G|_H) < m$; we will show that it holds for $\#(G|_H) = m$. To simplify notation, we identify v with $v|_H$, i.e., we write v in place of $v|_H$ throughout the proof.

We define a simple game $G' = \langle N, v' \rangle$ by setting $v'(S) = 1$ if $v(S) > 0$ and $v'(S) = 0$ otherwise. By Theorem 2, there exists a payoff vector $\mathbf{x}'$ such that $x'(S) \geq v'(S)$ for all $S \subseteq N$ and $x'(N) \leq (tw(H) + 1)v(CS')$, where CS' is an optimal coalition structure for G'. Moreover, we can assume that $\mathbf{x}' \in \{0,1\}^n$, as Algorithm 1 outputs such a payoff vector.

We set $\varepsilon = \min\{v(S) \mid v(S) > 0\}$ and define a game $G'' = \langle N, v'' \rangle$ by setting $v''(S) = \max\{0, v(S) - \varepsilon x'(S)\}$. Intuitively, we "split" G to a simple game $\varepsilon G'$ and a remainder G'', and stabilize each one independently.

Consider a coalition S with $v(S) = \varepsilon$. We have $v'(S) = 1$ and hence $x'(S) = 1$. Therefore, $v''(S) = 0$ and hence $\#(G'') < m$, so the induction hypothesis applies to G''. Therefore, there is a stable payoff vector $\mathbf{x}''$ such that $x''(N) \leq (tw(H) + 1)OPT(G'')$. We set $\mathbf{x} = \varepsilon \mathbf{x}' + \mathbf{x}''$. We will now show that $x(N) \leq (tw(H) + 1)OPT(G)$ and $x(S) \geq v(S)$ for all $S \subseteq N$.

We have $x(S) = \varepsilon x'(S) + x''(S) \geq \varepsilon x'(S) + v''(S) \geq \varepsilon x'(S) + v(S) - \varepsilon x'(S) = v(S)$ for all $S \subseteq N$, so $\mathbf{x}$ is a stable payoff vector for G.

Let CS'' be an optimal coalition structure for G''. We can assume without loss of generality that there is only one coalition of value 0 in CS''; we denote this coalition by S_0. Set $N^* = N \setminus S_0$; we have

$$\sum_{S \in CS''_+} x'(S) = x'(N^*) \geq \sum_{S \in CS'_+} x'(S \cap N^*) \qquad (1)$$
$$\geq \sum_{S \in CS'_+} v'(S \cap N^*) \geq |\{S \in CS'_+ \mid S \cap N^* \neq \emptyset\}|.$$

Let $t^* = |\{S \in CS'_+ \mid S \cap N^* \neq \emptyset\}|$, $t_0 = |\{S \in CS'_+ \mid S \subseteq S_0\}|$. We have $v'(CS') = |CS'_+| = t^* + t_0$.

We are now ready to bound $x(N)$. Using (1), we obtain

$$x(N) = \varepsilon x'(N) + x''(N) \qquad (2)$$
$$\leq \varepsilon(tw(H) + 1)v'(CS') + (tw(H) + 1)v''(CS'')$$
$$= (tw(H) + 1)\left(\varepsilon|CS'_+| + \sum_{S \in CS''_+} (v(S) - \varepsilon x'(S))\right)$$
$$\leq (tw(H) + 1)\left(\varepsilon|CS'_+| + v(CS''_+) - \varepsilon t^*\right)$$
$$= (tw(H) + 1)\left(v(CS''_+) + \varepsilon t_0\right).$$

Further,

$$t_0 = \sum_{S \in CS'_+ : S \subseteq S_0} v'(S) \leq \sum_{S \in CS'_+ : S \subseteq S_0} \frac{1}{\varepsilon}v(S),$$

so $x(N) \leq (tw(H) + 1)\left(v(CS''_+) + \sum_{S \in CS'_+ : S \subseteq S_0} v(S)\right)$. The coalitions in the right-hand side of this expression form a partition of (a subset of) N, so their total value under v does not exceed $OPT(G|_H)$. This concludes the proof. $\square$

The relative cost of stability of any TU game, even with unrestricted cooperation, is at most $\sqrt{n}$ (see (Bachrach et al. 2009; Meir, Bachrach, and Rosenschein 2010)). Thus, we obtain $RCoS(G|_H) \leq \min\{tw(H)+1, \sqrt{n}\}$, assuming that coalition structures are allowed. For superadditive games Theorem 3 implies that there is some stable payoff vector $\mathbf{x}$ such that $x(N) \leq (tw(H) + 1)v(N)$.[2]

Tightness

Demange (2004) showed that if $tw(H) = 1$, i.e., H is a tree, then the game $G|_H$ admits a stable outcome, i.e., $RCoS(G|_H) = 1$. We will now show that if the treewidth of the interaction network is at least 2, i.e., H is *not* a tree, then the upper bound of $tw(H) + 1$ proved in Theorem 3 is tight.

Theorem 4. *For every $k \geq 2$ there is a simple superadditive game $G = \langle N, v \rangle$ and an interaction network H over N such that $tw(H) = k$ and $RCoS(G|_H) = k + 1$.*

Proof sketch. Instead of defining H directly, we will describe its tree decomposition $\mathcal{T}$. There is one central node $A = \{z_1, \ldots, z_{k+1}\}$. For every unordered pair $I = \{i, j\}$, where $i, j \in \{1, \ldots, k+1\}$ and $i \neq j$, we define a set D_I that consists of 7 agents and set $N = A \cup \bigcup_{i \neq j \in \{1, \ldots, k+1\}} D_{\{i,j\}}$.

The tree $\mathcal{T}$ is a star, where leaves are all sets of the form $\{z_i, z_j, d\}$, where $d \in D_{\{i,j\}}$. That is, there are $7 \cdot \binom{k+1}{2}$ leaves, each of size 3. Since the central node of $\mathcal{T}$ is of size $k + 1$, it corresponds to a network of treewidth at most k. We set $\mathcal{D}_i = \bigcup_{j \neq i} D_{\{i,j\}}$; observe that for any two agents $z_i, z_j \in A$ we have $\mathcal{D}_i \cap \mathcal{D}_j = D_{\{i,j\}}$. Given $\mathcal{T}$,

---

[2] Note that, while the proof for *simple* superadditive games is straightforward, we cannot use the inductive argument made in Theorem 3 directly, as superadditivity may not be preserved.

it is now easy to construct the underlying interaction network H: there is an edge between z_i and every $d \in D_{\{i,j\}}$ for every $j \neq i$; see Figure 1 for more details.

For every unordered pair $I = \{i, j\} \subseteq \{1, \ldots, k+1\}$, let $\mathcal{Q}_I$ denote the projective plane of dimension 3 (a.k.a. the Fano plane, marked by dotted lines in Fig. 1) over D_I. That is, $\mathcal{Q}_I$ contains seven triplets of elements from D_I, so that every two triplets intersect, and every element $d \in D_I$ is contained in exactly 3 triplets in $\mathcal{Q}_I$. Winning sets are defined as follows. For every $i = 1, \ldots, k+1$ the set $\{z_i\} \cup \bigcup_{j \neq i} Q_{\{i,j\}}$ is winning. Thus for every z_i there are 7^k winning coalitions containing z_i, each of size $1 + 3k$.

We can observe that all winning coalitions intersect, which implies that the simple game induced by these winning coalitions is indeed superadditive and has an optimal value of 1. It remains to verify that every stable payoff vector must pay at least $k+1$ to the agents; we omit the details of this step due to space constraints. $\square$

The proof of Theorem 4 is not applicable when $k = 1$, since the width of our construction is at least 2 (each leaf is of size 3). Indeed, if Theorem 4 were to hold for $k = 1$, we would obtain a contradiction with Demange's result.

4. Pathwidth and the Cost of Stability

For some graphs we can bound not just their treewidth, but also their pathwidth. For example, for a simple cycle graph both the treewidth and the pathwidth are equal to 2. For games over networks with bounded pathwidth, the bound of $tw(H) + 1$ shown in Section 3 can be tightened.

Algorithm 2: STABLE-PW($G = \langle N, v \rangle, H, k, \mathcal{T}$)

Set $\mathcal{T} = (A_1, \ldots, A_m)$;
$\mathbf{x} \leftarrow 0^n$;
$I \leftarrow \{i \in N \mid v(\{i\}) = 1\}$;
for $i \in I$ **do**
  $x_i \leftarrow 1$;
$N_1 \leftarrow N \setminus I$;
$t \leftarrow 1$;
for $j = 1$ *to* m **do**
  if *there is some* $S \subseteq N(\mathcal{T}_{A_j}) \cap N_j$ *such that* $v(S) = 1$ **then**
    for $i \in A_j \cap N_j$ **do**
      if $i \in N(\mathcal{T}_{A_j}) \setminus A_j$ **then**
        $x_i \leftarrow 1$
    $N_{j+1} \leftarrow N_j \setminus N(\mathcal{T}_{A_j})$;
  else
    $N_{j+1} \leftarrow N_j$;
return $\mathbf{x} = (x_1, \ldots, x_n)$;

Theorem 5. *For every TU game $G = \langle v, N \rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq pw(H)$, and this bound is tight.*

Proof Sketch. We argue that, given a simple game G and a network H, Algorithm 2 outputs a stable payoff vector $\mathbf{x}$
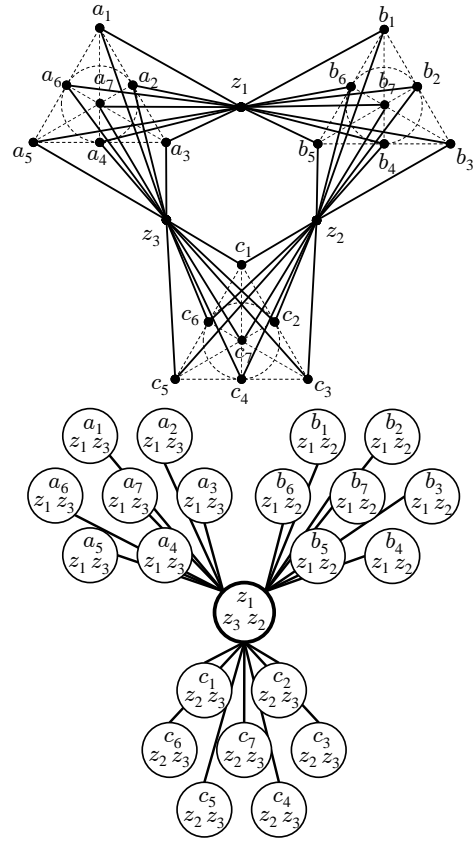

Figure 1: The interaction network H (top) when $k = 2$ in Theorem 4, and its tree decomposition (bottom). Here, $D_{1,3} = \{a_1, \ldots, a_7\}$, $D_{1,2} = \{b_1, \ldots, b_7\}$ and $D_{2,3} = \{c_1, \ldots, c_7\}$. An edge connects z_1 to all agents in $D_{1,3}$ and $D_{1,2}$, z_2 to $D_{1,2}$ and $D_{2,3}$, and z_3 to $D_{1,3}$ and $D_{2,3}$. Agent z_1 forms winning coalitions with triplets of agents from $D_{1,2}$ and $D_{1,3}$ that are on a dotted line; z_2 and z_3 form winning coalitions with their respective sets as well.

such that $x(N) \leq pw(H) \cdot OPT(G|_H)$. First, Algorithm 2 pays 1 to all winning singletons and removes them from the game; it can be shown that this step does not increase the cost of stability. Next, we proceed in a manner similar to Algorithm 1; however, when processing a node A_j such that $N(\mathcal{T}_{A_j})$ contains a winning coalition, we do not pay any agent $i \in A_j$ such that $i \notin N(\mathcal{T}_{A_j}) \setminus A_j$. Paying such agents is not necessary, as any winning coalition that contains them must contain some other agent in A_j that is paid 1 by the algorithm. It can be shown that such agents are guaranteed to exist, thus not all agents in A_j are paid. We then employ an inductive argument similar to the one in Theorem 3. To show tightness, we use a slight modification of the construction from Section 3. $\square$

5. Implications for Games on Graphs

Our results apply to several well-studied classes of cooperative games. The following definition, which appears in (Potters and Reijnierse 1995), becomes useful in showing this.

Let $H = \langle N, E \rangle$ be an interaction network. We say that two coalitions $S, T \subseteq N$ are *connected* in H if there ex-

ists an edge $(i, j) \in E$ such that $i \in S$, $j \in T$; otherwise S and T are said to be *disconnected*. A TU game $G = \langle N, v \rangle$ is said to be *H-component additive* if for every pair of coalitions S, T that are disconnected in H, it holds that $v(S \cup T) = v(S) + v(T)$. If G is H-component additive then G is essentially equivalent to $G|_H$: these games can only differ in values of infeasible coalitions.

There are many classes of combinatorial TU games defined over graphs, where every game in the class is component-additive with respect to the graph on which it is defined; our results hold for all of these classes. Some examples include induced subgraph games (Deng and Papadimitriou 1994); matching games, edge cover games, coloring games and vertex connectivity games (Deng, Ibaraki, and Nagamochi 1997); and social distance games (Brânzei and Larson 2011).[3] While some of these games are known to have a non-empty core, our results hold for unstable variants of them as long as they maintain component-additivity.

Games over hypergraphs Another two classes of games—Synergy Coalition Groups (Conitzer and Sandholm 2006) and Marginal Contribution Nets (Ieong and Shoham 2005)—are defined over collections of subsets, i.e., hypergraphs. Now, the notion of an interaction network can be naturally extended to that of an interaction hypergraph, an idea suggested by Myerson himself as well as by others (see (Bilbao 2000), p. 112): a coalition can form only if for any two coalition members i and j there is a sequence of overlapping hyperedges that connect them.

The concepts of treewidth and tree decomposition of a hypergraph coincide with the corresponding definitions applied to its *primal graph* (Gottlob, Leone, and Scarcello 2001). Therefore, all of our proofs work for games whose interaction networks are hypergraphs with bounded treewidth. The notion of a component-additive game can be extended to games on hypergraphs, and it is not hard to show that both Synergy Coalition Groups and Marginal Contribution Nets are component-additive with respect to their underlying hypergraphs. Hence, our results hold for these models as well.

6. Conclusions, Discussion, and Future Work

There is a strong connection between treewidth and the minimum subsidy required to stabilize a game: simply put, as the interaction becomes "simpler", the game becomes easier to stabilize. To the best of our knowledge, this is the first time that the notion of treewidth has been used to obtain results that are purely game-theoretic rather than algorithmic in nature.

While we provide a stronger bound with respect to pathwidth, the bound on the treewidth is more significant; indeed, Theorem 5 improves upon Theorem 3 only when the treewidth equals the pathwidth, which is uncommon.

Our results imply a separation between games whose interaction networks are acyclic, which have been shown to be stable (Demange 2004), and other games. That is, treewidth of 1 implies RCoS of 1, but for any higher value of treewidth,

the RCoS is somewhat higher than the treewidth. In particular, the result of Demange *is not* a special case of our theorem, although similar techniques to ours can be used to provide an alternative proof for Demange's theorem.

Treewidth and complexity Many NP-hard algorithmic problems over graphs can be solved in polynomial time assuming bounded treewidth; unfortunately, this is not the case for TU games over Myerson graphs. Indeed, common problems in TU games—and computing the RCoS in particular—remain computationally hard even when the treewidth of the interaction network is 2 (Greco et al. 2011b). We find it quite remarkable that, contrary to the common wisdom, the treewidth of the Myerson graph plays no role from an algorithmic perspective (except for the special case of a tree), but does have significant game-theoretic implications.

Hypertreewidth We have argued in Section 5 that our results can be extended to hypergraphs, giving a bound on the RCoS in terms of the treewidth of the interaction hypergraph. Gottlob et al. (2001) describe a stronger notion of width for hypergraphs, called *hypertreewidth*. This definition can result in a much lower width for general hypergraphs, and it is an open question whether it can provide us with a better bound on the RCoS.

The least core The cost of stability is closely related to another important notion of stability in cooperative games, namely, the least core (Maschler, Peleg, and Shapley 1979); specifically, Meir et al. (2011) show that the value of both the strong least core and the weak least core of a cooperative game can be bounded in terms of its additive cost of stability. Our results, combined with those of Meir et al., imply that any bound on the treewidth or pathwidth of the interaction graph translates into a bound on this other well-known measure of inherent instability.

Future Work

While our bound on the cost of stability is tight in the worst case, it may be further improved by considering finer restrictions on the structure of the interaction network and/or the value function itself. Other notions of graph cyclicity (such as hypertreewidth) may also be useful for providing bounds on the cost of stability.

More generally, we believe that the unexpected connection between a well-studied graph parameter such as the treewidth and the stability properties of a related game is fascinating. We look forward to studying how such parameters can be used to unearth other hidden connections in both cooperative and non-cooperative game theory.

---

[3]Brânzei and Larson (2011) define an NTU version of social distance games; however a TU version can be naturally defined.

References

Aumann, R., and Drèze, J. H. 1974. Cooperative games with coalition structures. *International Journal of Game Theory* 3:217–237.

Aziz, H.; Lachish, O.; Paterson, M.; and Savani, R. 2009. Power indices in spanning connectivity games. In *AAIM'09*, 55–67.

Aziz, H.; Brandt, F.; and Harrenstein, P. 2010. Monotone cooperative games and their threshold versions. In *AAMAS'10*, 1017–1024.

Bachrach, Y.; Elkind, E.; Meir, R.; Pasechnik, D.; Zuckerman, M.; Rothe, J.; and Rosenschein, J. S. 2009. The cost of stability in coalitional games. In *SAGT'09*, 122–134.

Bachrach, Y.; Meir, R.; Jung, K.; and Kohli, P. 2010. Coalitional structure generation in skill games. In *AAAI'10*, 703–708.

Bilbao, J. M. 2000. *Cooperative Games on Combinatorial Structures*. Kluwer Publishers.

Bodlaender, H. L. 2005. Discovering treewidth. In *SOFSEM'05*, 1–16.

Brânzei, S., and Larson, K. 2011. Social distance games. In *IJCAI'11*, 91–96.

Breton, M.; Owen, G.; and Weber, S. 1992. Strongly balanced cooperative games. *International Journal of Game Theory* 20:419–427.

Chalkiadakis, G.; Markakis, E.; and Jennings, N. R. 2012. Coalitional stability in structured environments. In *AAMAS'12*, 779–786.

Cohen, R.; Eades, P.; Lin, T.; and Ruskey, F. 1995. Three-dimensional graph drawing. In *Graph Drawing*, volume 894 of *Lecture Notes in Computer Science*.

Conitzer, V., and Sandholm, T. 2006. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence* 170(6–7):607–619.

Courcelle, B. 1990. Graph rewriting: An algebraic and logic approach. In *Handbook of Theoretical Computer Science*, volume B. 193–242.

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

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

Deng, X.; Ibaraki, T.; and Nagamochi, H. 1997. Algorithms and complexity in combinatorial optimization games. In *SODA'97*, 720–729.

Devanur, N. R.; Mihail, M.; and Vazirani, V. V. 2005. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems* 39:11–22.

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

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

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

Greco, G.; Malizia, E.; Palopoli, L.; and Scarcello, F. 2011b. On the complexity of the core over coalition structures. In *IJCAI'11*, 216–221.

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

Jain, K., and Vazirani, V. 2001. Applications of approximation algorithms to cooperative games. In *STOC'01*, 364–372.

Maschler, M.; Peleg, B.; and Shapley, L. S. 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research* 4(4):303–338.

Meir, R.; Bachrach, Y.; and Rosenschein, J. S. 2010. Minimal subsidies in expense sharing games. In *SAGT'10*, 347–358.

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

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

Potters, J., and Reijnierse, H. 1995. γ-component additive games. *International Journal of Game Theory* 24(1):49–56.

Resnick, E.; Bachrach, Y.; Meir, R.; and Rosenschein, J. S. 2009. The cost of stability in network flow games. In *MFCS'09*, 636–650.

Voice, T.; Polukarov, M.; and Jennings, N. R. 2012. Coalition structure generation over graphs. *Journal of Artificial Intelligence Research* 45:165–196.

A   Proofs

Proposition 1. *For any $k \in \mathbb{N}$, there is a simple superadditive game with $RCoS(G) \geq k$ over an interaction network H with $d(H) = 6$.*

Proof. We show that *any* superadditive simple game can be embedded in a 3-dimensional grid network $H = \langle N', E \rangle$, if N' is sufficiently large.

For this, consider first a 3-dimensional grid drawing W of the complete graph K_n. This is an embedding of n vertices in a grid, s.t. every edge (i, j) is replaced by a path, and pathes—if drawn as straight lines—do not intersect. Such a drawing always exists using a grid of $O(n) \times O(n) \times O(n)$ (see e.g., (Cohen et al. 1995)). However, W itself is not a grid graph, but just another representation of K_n.

The graph $H' = \langle N', E' \rangle$ that we will use is a 3-dimensional grid that is attained by replacing every vertex in the grid underlying W, with a grid of $n \times n \times n$ (thus $|N'| = O(n^6)$). In particular, every original vertex $i \in N$ is replaced with a cube $A_i \subseteq N'$ of n^3 vertices. Next, for every $(i, j) \in E$ (assume $i < j$), we identify a path $P(i, j) \subseteq N'$, s.t. $P(i, j)$ connects A_i and A_j; and no two paths intersect. Since the projection of W on H' is extremely sparse, it is very easy to refrain from path intersections.

We next use G to define the embedded game $G' = \langle N', v' \rangle$, with the following winning coalitions. For every winning coalition $S \subseteq N$ of G, we set $v'(S') = 1$, where $S' = \bigcup_{i \in S} A_i \cup \bigcup_{i,j \in S} P(i, j)$. Since S is connected in K_n, then S' is connected in H'. Moreover, since G is superadditive, every two winning coalitions S_1, S_2 intersect at some $i \in N$. Thus S'_1, S'_2 also intersect (in all vertices of A_i), which entails that G' is also superadditive.

Finally, we argue that $RCoS(G'|_{H'}) = RCoS(G') \geq RCoS(G)$. Indeed, since every winning S' is connected, the first equality applies. Then, assume that there is some payoff vector $\mathbf{x}' \in \mathcal{S}(G')$ that stabilizes G'. We define a payoff vector $\mathbf{x}$ for G, where $x_i = x'(A_i) + \sum_{j \in N} x'(P(i, j))$. Clearly $x(N) \leq x'(N') = RCoS(G')$. Moreover, for every winning $S \subseteq N$, $x(S) = x'(S') \geq v'(S') = 1$, thus $\mathbf{x}$ stabilizes G.

For any k, there is a simple superadditive game G_k whose RCoS is at least k (e.g., the game defined by the projective plane of order k. See (Bachrach et al. 2009)). As shown above, G_k can be embedded (like any other game) in a grid H' of degree 6. $\qquad\square$

Theorem 4. *For every $k \geq 2$ there is a simple superadditive game $G = \langle N, v \rangle$ and an interaction network H over N such that $tw(H) = k$ and $RCoS(G|_H) = k + 1$.*

Proof. Instead of defining H directly, we will describe its tree decomposition $\mathcal{T}$. There is one central node $A = \{z_1, \ldots, z_{k+1}\}$. Further, for every unordered pair $I = \{i, j\}$, where $i, j \in \{1, \ldots, k+1\}$ and $i \neq j$, we define a set D_I that consists of 7 agents and set $N = A \cup \bigcup_{i \neq j \in \{1, \ldots, k+1\}} D_{\{i,j\}}$.

The tree $\mathcal{T}$ is a star, where leaves are all sets of the form $\{z_i, z_j, d\}$, where $d \in D_{\{i,j\}}$. That is, there are $7 \cdot \binom{k+1}{2}$ leaves, each of size 3. Since the maximal node of $\mathcal{T}$ is of size $k + 1$, it corresponds to some network whose treewidth is at most k. We set $\mathcal{D}_i = \bigcup_{j \neq i} D_{\{i,j\}}$; observe that for any two agents $z_i, z_j \in A$ we have $\mathcal{D}_i \cap \mathcal{D}_j = D_{\{i,j\}}$. Given $\mathcal{T}$, it is now easy to construct the underlying interaction network H: there is an edge between z_i and every $d \in D_{\{i,j\}}$ for every $j \neq i$; see Figure 1 for more details.

For every unordered pair $I = \{i, j\} \subseteq \{1, \ldots, k+1\}$, let $\mathcal{Q}_I$ denote the projective plane of dimension 3 (a.k.a. the Fano plane) over D_I. That is, $\mathcal{Q}_I$ contains seven triplets of elements from D_I, so that every two triplets intersect, and every element $d \in D_I$ is contained in exactly 3 triplets in $\mathcal{Q}_I$. Winning sets are defined as follows. For every $i = 1, \ldots, k+1$ and every selection $\{Q_{\{i,j\}} \in \mathcal{Q}_{\{i,j\}}\}_{j \neq i}$ the set $\{z_i\} \cup \bigcup_{j \neq i} Q_{\{i,j\}}$ is winning. Thus for every z_i there are 7^k winning coalitions containing z_i, each of size $1 + 3k$. Let us denote by $\mathcal{W}_i$ the set of winning coalitions that contain z_i; observe that for every $d \notin A$, d appears in exactly $3 \cdot 7^{k-1}$ winning coalitions in $\mathcal{W}_i$: d belongs to some $D_{\{i,j\}}$, and is selected to be in a winning coalition with z_i if a triplet $Q_{\{i,j\}}$ containing d is joined to z_i. There are 3 triplets in $\mathcal{Q}_{\{i,j\}}$ that contain d, and there are 7^{k-1} ways to choose the other triplets (seven choices from every one of the other $k - 1$ sets).

We first argue that all winning coalitions intersect. Indeed, let C_i, C_j be winning coalitions such that $z_i \in C_i, z_j \in C_j$. Then both C_i and C_j contain some triplet from $\mathcal{Q}_{\{i,j\}}$. Suppose $Q_{\{i,j\}} \subseteq C_i, Q'_{\{i,j\}} \subseteq C_j$. Since $Q_{\{i,j\}}, Q'_{\{i,j\}} \in \mathcal{Q}_{\{i,j\}}$, they must intersect, and thus C_i and C_j must also intersect. This implies that the simple game induced by these winning coalitions is indeed superadditive and has an optimal value of 1. Note that if we pay 1 to each $z_i \in A$, then the resulting super-imputation is stable, since every winning coalition intersects A. To conclude the proof, we must show that any stable super-imputation must pay at least $k + 1$ to the agents.

Given a stable super-imputation $\mathbf{x}$, we know that $x(C_i) \geq 1$ for every $C_i \in \mathcal{W}_i$. Thus, $\sum_{C_i \in \mathcal{W}_i} x(C_i) \geq 7^k$. We can write $\sum_{C_i \in \mathcal{W}_i} x(C_i)$ as

$$
\begin{aligned}
\sum_{C_i \in \mathcal{W}_i} x(C_i) &= \sum_{C_i \in \mathcal{W}_i} \left(x_{z_i} + \sum_{d \neq z_i | d \in C_i} x_d \right) = 7^k x_{z_i} + \sum_{C_i \in \mathcal{W}_i} \sum_{d \neq z_i | d \in C_i} x_d \\
&= 7^k x_{z_i} + \sum_{d \in \mathcal{D}_i} 1 \sum_{C_i \in \mathcal{W}_i | d \in C_i} x_d = 7^k x_{z_i} + \sum_{d \in \mathcal{D}_i} 3 \cdot 7^{k-1} x_d \\
&= 7^k x_{z_i} + 3 \cdot 7^{k-1} x(\mathcal{D}_i).
\end{aligned}
$$

This immediately implies that $x_{z_i} \geq 1 - \frac{3}{7} x(\mathcal{D}_i)$. Observe that $\sum_{z_i \in A} x(\mathcal{D}_i) = 2 \sum_{i<j} x(D_{\{i,j\}})$, as each $D_{\{i,j\}}$ appears exactly twice in the summation: once in $\mathcal{D}_i$ and once in $\mathcal{D}_j$. Also, observe that $\sum_{i<j} x(D_{\{i,j\}}) = x(N \setminus A)$, so $\sum_{i=1}^{k+1} x(\mathcal{D}_i) = 2x(N \setminus A)$. Finally,

$$
\begin{aligned}
x(N) &= x(A) + x(N \setminus A) = \sum_{i=1}^{k+1} x_{z_i} + x(N \setminus A) \\
&\geq \sum_{i=1}^{k+1} \left(1 - \frac{3}{7} x(\mathcal{D}_i) \right) + x(N \setminus A) = \sum_{i=1}^{k+1} 1 - \frac{3}{7} 2x(N \setminus A) + x(N \setminus A) \\
&= k + 1 + (1 - \frac{6}{7}) x(N \setminus A) \geq k + 1
\end{aligned}
$$

Thus, the relative cost of stability in our game is at least $k + 1$. $\qquad \square$

Theorem 5. *For every TU game $G = \langle v, N \rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq pw(H)$, and this bound is tight.*

Proof. Note first that it suffices to show that our bound holds for simple games; we can then use the reduction described in the proof of Theorem 3. For simple games, our proof is very similar to the proof of Theorem 2; however, here we will show that in every node A_j that satisfies the **if** condition of Algorithm 2 we can identify an agent that we do not need to pay.

Our algorithm first deals with winning coalitions of size 1. This step can be justified as follows. Suppose we remove all agents in $I = \{i \in N \mid v(\{i\}) = 1\}$ and construct a stable super-imputation $\mathbf{x}'$ for the game $G'|_H$, where $G' = \langle N', v' \rangle$, $N' = N \setminus I$, and $v'(S) = v(S)$ for each $S \subseteq N \setminus I$, so that $x'(N') \leq pw(H)$. Now, consider a super-imputation $\mathbf{x}$ for G given by $x_i = 1$ for $i \in I$, $x_i = x'_i$ for $i \in N'$. We have $x(N) = x'(N') + |I|$, and, furthermore, $x(S) \geq v|_H(S)$ for every $S \subseteq N$, i.e., $\mathbf{x}$ is a stable super-imputation for $G|_H$. On the other hand, it is not hard to check that $OPT(G|_H) = OPT(G'|_H) + |I|$. Hence, we obtain

$$
\frac{x(N)}{OPT(G|_H)} = \frac{x'(N') + |I|}{OPT(G'|_H) + |I|} < \frac{x'(N')}{OPT(G'|_H)} \leq pw(H),
$$

i.e., $\mathbf{x}$ witnesses that $RCoS(G|_H) \leq pw(H)$. Thus, we begin Algorithm 2 by paying all winning singletons 1 and ignoring them (and any winning coalitions that contain them) for the rest of the execution; note, however, that we *do not* remove the winning singletons from H, i.e., we do not modify our path decomposition or its width.

Next we show stability. Given a node A_j, we must make sure that each winning coalition in $N(\mathcal{T}_{A_j})$ is paid at least 1. By the proof of Theorem 2, paying all agents in A_j is sufficient. Note, however, that there is no need to pay an agent i that is not in $N(\mathcal{T}_{A_j}) \setminus A_j$: since we removed all winning singletons, every winning coalition in $N(\mathcal{T}_{A_j})$ that contains i (and that is not yet stabilized) must also contain another agent from A_j.

Finally, we must show that in every paid node A_j, $j \geq 2$, there is at least one agent that is not paid. Note that A_j has a unique child A_{j-1}. If $A_j \subseteq A_{j-1}$, then no agent in A_j is being paid (as they had already been paid when processing A_{j-1}). Otherwise, there is some agent $i \in A_j \setminus A_{j-1}$. Since $\mathcal{T}$ is a path and all nodes containing i must be connected, we have $i \notin N(A_j) \setminus A_j$. Thus i is not paid. Note that in Algorithm 2 the agents in A_1 are not paid in the first iteration of the algorithm.

To show tightness, we use a slight modification of the construction from Section 3. For any $k \geq 3$:

- Take the tree-width example for $k - 1$, remove all edges from the (star) tree.
- Add the central node (of size k) to all leaf nodes. Thus we get $O(k^2)$ nodes of size $k + 1$.
- Connect all nodes by an arbitrary path.

Then the path-width is $(k + 1) - 1 = k$, whereas the CoS is exactly as before (k) since we have the same set of winning coalitions. For $k = 2$, we can use the cycle example from (Meir, Rosenschein, and Malizia 2011), taking the number of agents to infinity. $\qquad \square$

B Computational Complexity

We define the decision problem OPTCS as follows: it receives as input a game $G = \langle N, v \rangle$, an interaction network H and some value $\alpha \in \mathbb{R}$; it outputs yes if and only if there is some partition $S_1, \ldots, S_k$ of N such that $\sum_{j=1}^{k} v|_H(S_j) \geq \alpha$. We assume oracle access to v.

It is known that if H is a tree and G is a simple monotone game then there is a simple polynomial algorithm for OPTCS . This is by selecting an arbitrary root and iteratively isolate winning coalitions from the leafs upwards (similarly to the procedure of Algorithm 1). However if we relax either of these three requirements, and the tree structre in particular, the problem becomes computationaly hard.

Proposition 6. OPTCS *(G, H) is NP-hard even if G is simple and $tw(H) = pw(H) = 2$.*

Proof. Our reduction is from an instance of the SET-COVER (Garey and Johnson 1979) problem. Recall that an instance of SET-COVER is given by a finite set C, list of sets $\mathcal{S} = (S_1, \ldots, S_n)$ and an integer M; it is a "yes" instance if and only if there is a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\mathcal{S}'$ *covers* C, i.e. $\bigcup_{S \in \mathcal{S}'} S = C$, and $|\mathcal{S}'| \leq M$. Given an instance of SET-COVER $(C, \mathcal{S}, M)$, as described above, we define the player set to be $\{1, \ldots, n, x, y\}$. We define the characteristic function as follows: for any $S \subseteq \{1, \ldots, n\}$, $v(S \cup \{x\}) = 1$ if and only if the set $\{S_i\}_{i \in S}$ covers C; $v(S \cup \{y\}) = 1$ if and only if $|S| \geq n - M$. Our interaction network H over the player set is defined as follows: there are edges (i, x) and (i, y) for all $1 \leq i \leq n$; observe that $tw(H) = 2$. One can easily verify that an optimal coalition structure over $G|_H$ has a value of 2 if and only if $(C, \mathcal{S}, M)$ is a "yes" instance of SET-COVER . $\square$

A similar reduction from the PARTITION (Garey and Johnson 1979) shows that OPTCS is still hard under the conditions of Proposition 6, even if we limit G to be a weighted voting game.

Limiting our attention to monotone simple games seems to be somewhat restrictive. However, both monotonicity and bi-values are required for tractability. Note that in both cases we show that it is hard even to distinct between the cases where $v(CS^*(G|_T)) = 1$ and $v(CS^*(G|_T)) = 0$. Thus there is no efficient approximation algorithm either.

Proposition 7. OPTCS *(G, T) is NP-complete if we allow inputs with a non-monotone G, even if we assume that the interaction network T is a tree and G is simple.*

Proof. Our reduction is from SUBSET-SUM (Garey and Johnson 1979); recall that an instance of SUBSET-SUM is given by a list of integer weights $w_1, \ldots, w_n$ and some quota q. It is a "yes" instance if and only if there is some subset of weights whose total weight is exactly q. Given an instance of SUBSET-SUM $\langle w_1, \ldots, w_n; q \rangle$, we construct the following game on $n+1$ players: player i is assigned a weight w_i, while player $n + 1$ has a weight of 0. The value of $v(S)$ is 1 if and only if $\sum_{i \in S} w_i = q$ (and otherwise 0). The communication network H is a star centered in player $n + 1$, with the other n players as leaves. Observe that in this game, at most one coalition containing more than one member of $\{1, \ldots, n\}$ can form. To conclude, assuming that $w_i < q$ for all i, the optimal coalition structure in $G|_H$ has value of at most 1, and is 1 if and only we have a "yes" instance of SUBSET-SUM . $\square$

Finally, OPTCS is NP-complete for monotone non-simple games as well.

Proposition 8. OPTCS *(G, T) is NP-complete if we allow inputs with a non-simple G, even if the interaction network T is a tree, and that v is allowed only three different values.*

Proof. Our reduction is from the SET-COVER (Garey and Johnson 1979) problem. Recall that an instance of SET-COVER is given by a finite set of elements M, a set $\mathcal{F} = \{S_1, \ldots, S_m\} \subseteq 2^M$ and a parameter k. It is a "yes" instance if and only there is some $\mathcal{F}' \subseteq \mathcal{F}$ of size $\leq k$ such that $\bigcup_{S \in \mathcal{F}'} S = M$. We define the characteristic function as follows: there is an agent i_j corresponding to each $S_j \in \mathcal{F}$, plus one dummy agent d. The value of a coalition $C \subset N$ is 0 if it is empty, n if $\{S_j\}_{i_j \in C}$ cover M, and 1 otherwise. Our interaction network H is a star with d in the center, and with all i_j as leaves. Thus, only one coalition that covers M may form. Clearly, in an optimal coalition structure a coalition C^* that covers M will form, with the addition of as many singletons as possible. The value of the optimal coalition structure is more than $n + (n + 1 - k) = 2n + 1 - k$ if and only if $|C^*| \leq k$, which concludes the proof. $\square$