Bounding the Cost of Stability in Games with Restricted Interaction

Reshef Meir, Yair Zick, Edith Elkind, and Jeffrey S. Rosenschein

Abstract

We study stability of cooperative games with restricted interaction, in the model that was introduced by Myerson [18]. We show that the cost of stabilityof such games (i.e., the subsidy required to stabilize the game) can be bounded in terms of natural parameters of their interaction graphs. Specifically, we prove that if the treewidth of the interaction graph is k, then the relative cost of stability is at most $k + 1$, and this bound is tight for all $k \geq 2$. Also, we show that if the pathwidth of the interaction graph is k, then the relative cost of stability is at most k.

1 Introduction

Coalitional game theory models scenarios where groups of agents can work together profitably: the agents form teams, or *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 so that each agent is satisfied with his share. In particular, it is desirable to allocate the payoffs in such a way that no group of agents can do better by deviating from their current 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: indeed, there are many games that 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*. Alternatively, one can assume that the deviators are non-myopic, and will not attempt a deviation if it may be followed by a counter-deviation that makes them worse off; this idea is captured by the notion of *bargaining set*. Yet another approach, which was pioneered by Myerson [18], is based on the idea that communication among agents may be limited, and agents cannot form a deviating coalition unless they can communicate with each other. In more detail, the communication network among the agents is described by an *interaction graph*, where agents are nodes, and an edge denotes the presence of a communication link; allowable coalitions correspond to connected subgraphs of the interaction graph. Myerson's model can be seen as a special case of a restriction scheme known as *partition systems* (see Chapter 5 in Bilbao [6] for an overview). Finally, coalitional stability may be achieved via *subsidies*: an external party may be willing to stabilize the game by offering a lump sum to the agents as long as they form some desired coalition structure. The minimal subsidy required in order to guarantee stability is known as the *cost of stability (CoS)* [4] (in what follows, it will be convenient to use a modified version of this notion known as *relative cost of stability ($RCoS$)* [17], which is defined as the ratio between the minimal total payoff needed to ensure stability and the total value of an optimal coalition structure).

In this paper, we study the interplay between the latter two concepts, namely, restricted interaction and the cost of stability. Our goal is to bound the (relative) cost of stability of a game in terms of natural parameters of its interaction graph. One such parameter is the *treewidth*: this is a combinatorial measure of graph structure that ranges from 1 (a tree or a forest) to $n-1$ (a complete graph on n vertices), and, intuitively, says whether the graph is close to being a tree. A closely related notion is that of *pathwidth*, which measures how close the graph is to being a path. We are motivated by the classic result of Demange [9], who showed that if the interaction graph is a tree then the core of the game is not empty. Given this result, it is natural to ask if games whose interaction graphs have small treewidth are close to having a non-empty core.

Our Contribution Our main contribution is a complete characterization of the relationship between the treewidth of the interaction graph and the cost of stability. We show that if the treewidth of the interaction graph of a game G is k, then the relative cost of stability of G is at most $k+1$. Moreover, we demonstrate that this bound is tight whenever $k \geq 2$. We also show that the bound on the relative cost of stability can be improved to k if the *pathwidth* of the interaction graph is k, and this is also tight.

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, [15, 11]). For profit-sharing games, Bachrach et al. [4] have recently introduced the notion of cost of stability (CoS), which is defined as the minimal 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 [19, 16, 2, 17, 12, 13]. In particular, Meir et al. [17] and Greco et al. [13] studied questions related to the CoS in games with restricted cooperation, 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., [8]). 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) has been bounded in terms of the treewidth of the corresponding graph [14, 3, 5, 12]. However, in the general case bounding the treewidth of the Myerson graph (except for the special case of width 1) *does not* guarantee a tractable solution for these computational questions [13]. Moreover, the notion of treewidth was mostly applied in the context of algorithmic analysis of cooperative games; to the best of our knowledge, our work is the first to employ treewidth to prove a game-theoretic result that is not computational in nature.

2 Preliminaries

We will now present the definitions that will be used in this paper. 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 $N = \{1, \dots, n\}$ is a finite set of *agents* and $v : 2^N \to \mathbb{R}$ is the *characteristic function* of the game. We assume that $v(\emptyset) = 0$. Also, unless explicitly stated otherwise, we restrict our attention to games where the characteristic function takes non-negative values only, i.e., $v(S) \geq 0$ 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, unlike, e.g., [20], we *do not* require simple games to be monotone; this allows us to use the inductive argument in Section 3.2. A coalition S in a simple game $G = \langle N, v \rangle$ is said to be *winning* if $v(S) = 1$ and *losing* if $v(S) = 0$.

Following [1], we assume that agents may form coalition structures. A *coalition structure* over N is a partition of N into disjoint subsets. The *value* of a coalition structure CS over N, denoted by $v(CS)$, is given by $v(CS) = \sum_{S \in CS} v(S)$. We denote the set of all coalition structures over N by $\mathcal{CS}(N)$, and write $OPT(G) = \max\{v(CS) \mid CS \in \mathcal{CS}(N)\}$. CS is said to be *optimal* if $v(CS) = OPT(G)$. Note that in superadditive games $v(N) = OPT(G)$.

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 $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n_+$, 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 let $\mathcal{S}(G)$ denote the set of all payoff vectors (not necessarily pre-imputations) that satisfy the stability constraints, i.e., we set

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

We refer to payoff vectors such that $x(N) \geq OPT(G)$ as *super-imputations*; note that $\mathcal{S}(G)$ consists of super-imputations only.

The *Relative Cost of Stability* of a game G is the minimal total payoff that stabilizes the game. Formally, we set

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

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

Interaction Graphs and Treewidth An *interaction network* 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 forms 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 S forms a connected subgraph of H.

A *tree decomposition* of H is a tree $\mathcal{T}$ with the set of nodes $V(\mathcal{T})$ that has the following properties:

1. Each node of $\mathcal{T}$ is a subset of N.
2. 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 $i \in Z$.
3. For every edge $e = \{i, j\}$ of E there exists a node $X \in V(\mathcal{T})$ such that $e \subseteq X$.[1]

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., [7].

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 $S \subseteq N$, we let $\mathcal{T}(S)$ denote the subgraph of $\mathcal{T}$ induced by the node set $\{X \in V(\mathcal{T}) \mid X \cap S \neq \emptyset\}$; 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 defined as $pw(H) = \min\{tw(\mathcal{T}) \mid \mathcal{T} \text{ is a path decomposition of } H\}$. It is known that for any graph H, $tw(H) \leq pw(H)$ and $pw(H) = tw(H) \cdot O(\log(n))$.

[1] We note that a tree decomposition of *hypergraphs* is defined in the same way, except that every *hyperedge* must be contained in some node.

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.

3.1 Simple Games

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

```
Fix an arbitrary R ∈ V(𝒯) to be the root;
t ← 0, N_1 ← N, x ← 0^n;
for A ∈ V(𝒯), traversed from the leaves upwards do
    t ← t + 1;
    if there is some S ⊆ N(𝒯_A) ∩ N_t such that v|_H(S) = 1 then
        for i ∈ A ∩ N_t do
            x_i ← 1
        N_{t+1} ← N_t \ N(𝒯_A);
        // remove all agents in N(𝒯_A) from the entire tree
    else
        N_{t+1} ← N_t;
return x = (x_1, ..., x_n);
```

We will now show that if G is a game with a set of agents N and H is an interaction network over N then $RCoS(G|_H) \leq tw(H)+1$. Our proof is constructive: we design an algorithm (Algorithm 1) that receives as its input a simple game $G = \langle N, v \rangle$, a network H, a parameter k, and a tree decomposition $\mathcal{T}$ of H of width of at most k, and outputs a stable super-imputation for $G|_H$. 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 $\mathcal{T}_A$ rooted in 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 very similar to the one proposed by Demange [9]; however, Algorithm 1 may pay $2 \cdot OPT(G|_H)$ if H is a tree.[2] Moreover, while Demange's algorithm runs in polynomial time, Algorithm 1 may require exponential time, since it is designed to work for non-monotone simple games. However, if the simple 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 $S \subseteq N(\mathcal{T}_A)$) will make it run in polynomial time.

Theorem 3.1. *For every simple game $G = \langle N, v \rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq tw(H) + 1$.*

Proof. Let $\mathcal{T}$ be a tree decomposition of H such that $tw(\mathcal{T}) = k$. Suppose first that $G|_H$ is superadditive. This means that any two winning coalitions in $G|_H$ intersect. Hence, for every pair

[2] This is because Algorithm 1 operates on the tree decomposition $\mathcal{T}$ of H, 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.

of winning coalitions $S_1, S_2 \subseteq N$ the subtrees $\mathcal{T}(S_1)$ and $\mathcal{T}(S_2)$ intersect. This implies that there exists a node $A \in V(\mathcal{T})$ that belongs to the intersection of all subtrees that correspond to winning coalitions in $\mathcal{T}$ (this fact is known as Helly's Theorem for Trees), and hence intersects every winning coalition. Therefore we can stabilize the game by paying 1 to every agent in A. Thus, our total payment is $|A| \leq tw(\mathcal{T}) + 1 \leq k + 1$.

We now turn to the more general case of arbitrary simple games. Let $\mathbf{x}$ be the output of Algorithm 1. We claim that $\mathbf{x}$ is stable (i.e., $\mathbf{x} \in \mathcal{S}(G|_H)$) and $x(N) \leq k + 1$.

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 $x(N) \leq (k + 1)OPT(G)$. To this end, we will construct a specific coalition structure CS^* and argue that $x(N) \leq (k + 1)v(CS^*)$.

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 $v(CS^*) = |T^*|$.

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

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

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

We note 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 3.1 goes through as long as $G|_H$ is simple, even if G itself is not simple.

3.2 The General Case

Using Theorem 3.1, we are now ready to prove our main result.

Theorem 3.2. *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. We first prove the claim for all integer-valued games. We use an inductive argument on $OPT(G|_H) = m$. If $OPT(G|_H) = 1$ then in particular $G|_H$ is simple, so we are done by Theorem 3.1. Now suppose that our claim is true for all $m' < m$; we will show that it holds for 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 the following simple game $G' = \langle N, v' \rangle$:

$$v'(S) = \begin{cases} 1 & \text{if } v(S) > 0 \\ 0 & \text{otherwise} \end{cases}$$

By Theorem 3.1, there exists a super-imputation $\mathbf{x}'$ 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 over G'. Moreover, we can assume that $\mathbf{x}' \in \{0, 1\}^n$, as Algorithm 1 outputs such a super-imputation. We define a game $G'' = \langle N, v'' \rangle$ by setting

$$v''(S) = \max\{0, v(S) - x'(S)\}.$$

Note that $v''(S) \in \mathbb{Z}^+$ for all $S \subseteq N$, since $\mathbf{x}' \in \{0, 1\}^n$ and G is integer-valued. Moreover, let CS'' be an optimal coalition structure for G'', and let $CS''_+ = \{S \in CS'' \mid v''(S) > 0\}$. We have

$$\sum_{S \in CS''} v''(S) = \sum_{S \in CS''_+} v''(S) = \sum_{S \in CS''_+} v(S) - x'(S).$$

Moreover, for every $S \in CS''_+$ we have $v(S) - x'(S) > 0$; in particular this means that $v(S) > 0$, which implies that $v'(S) = 1 \leq x'(S)$. Therefore for any $S \in CS''_+$ we have

$$v''(S) = v(S) - x'(S) \leq v(S) - 1 < v(S).$$

We conclude that

$$\sum_{S \in CS''} v''(S) = \sum_{S \in CS''_+} v''(S) < \sum_{S \in CS''_+} v(S) \leq m.$$

Thus, the value of an optimal coalition structure over G'' is strictly less than m, i.e., we can apply the induction hypothesis to G''. This means that there is a super-imputation $\mathbf{x}''$ such that $x''(N) \leq (tw(H) + 1)v''(CS'')$ and $x''(S) \geq v''(S)$ for all $S \subseteq N$. We set $\mathbf{x} = \mathbf{x}' + \mathbf{x}''$. We will now show that $x(N) \leq (tw(H) + 1)OPT(G)$ and $x(S) \geq v(S)$ for all $S \subseteq N$.

First, observe that for all $S \subseteq N$ we have $x(S) = x'(S) + x''(S) \geq x'(S) + v''(S) \geq x'(S) + v(S) - x'(S) = v(S)$, so $\mathbf{x}$ is a stable super-imputation for G. Now, let CS'' be an optimal coalition structure over G'', and consider $CS'' \setminus CS''_+$, i.e., the set of all coalitions of value 0 in CS''. We can assume without loss of generality that $CS'' \setminus CS''_+$ is a singleton, i.e., there is only one coalition of value 0 in CS''; we denote this coalition by S_0. Let CS' be an optimal coalition structure over G', and let $CS'_+ = \{S \in CS' \mid v'(S) = 1\}$. Set $N^* = N \setminus S_0$; then we have

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

Let $t^* = \{S \in CS^*_+ \mid S \cap N^* \neq \emptyset\}$ and let $t_0 = \{S \in CS^*_+ \mid S \subseteq S_0\}$. We denote the number of coalitions in CS^*_+ that intersect N^* by t^*, and the number of those who are contained in S_0 by t_0. The total value of CS' is thus $|CS'_+| = t^* + t_0$.

We are now ready to bound $x(N)$. Indeed, we have

$$\begin{aligned} x(N) &= x'(N) + x''(N) \leq (tw(H) + 1)v''(CS'') + (tw(H) + 1)v'(CS') \\ &= (tw(H) + 1) \left(\sum_{S \in CS''_+} (v(S) - x'(S)) + |CS'_+| \right) \\ &= (tw(H) + 1) \left(\sum_{S \in CS''_+} v(S) - x'(N^*) + |CS'_+| \right) \\ &\leq (tw(H) + 1) \left(v(CS''_+) - t^* + |CS'_+| \right) = (tw(H) + 1) \left(v(CS''_+) + t_0 \right) \end{aligned} \quad (1)$$

Further, we have $t_0 = \sum_{S \in CS'_+ : S \subseteq S_0} v'(S) \leq \sum_{S \in CS'_+ | S \subseteq S_0} v(S)$, so the final term in (1) is at most $(tw(H) + 1)\left(v(CS''_+) + \sum_{S \in CS'_+ : S \subseteq S_0} v(S)\right)$. This is a sum over a partition of (a subset of) N, so its total value is at most that of $OPT(G|_H)$, which concludes the proof for integer-valued games.

To see why the result still holds for non-integer-valued games, we make the following observation. Given a game $G = \langle N, v\rangle$, we can consider the game $\varepsilon G = \langle N, v_\varepsilon\rangle$ given by $v_\varepsilon(S) = \varepsilon v(S)$ for every $S \subseteq N$; we note that if G is simple, then for any $\varepsilon > 0$ Algorithm 1 can be applied to the game εG and hence Theorem 3.1 remains true for εG. Moreover, in εG every agent receives a payoff of either ε or 0. Further, when defining the modified characteristic function v', we can set $\varepsilon = \min_{S \subseteq N}\{v(S) \mid v(S) > 0\}$ and let $v'(S) = \varepsilon$ whenever $v(S) > 0$ (instead of setting $v'(S) = 1$). The rest of the proof can be modified appropriately (with a different ε chosen at each iteration); in particular, instead of using induction on $OPT(G|_H)$, we use induction on the number of coalitions with non-zero value. □

The $RCoS$ of any cooperative game, even with unrestricted cooperation, is at most $\sqrt{n}$ (see [4, 16]). Thus, we obtain $RCoS(G|_H) \leq \min\{tw(H) + 1, \sqrt{n}\}$, assuming that coalition structures are allowed. Moreover, when applied to superadditive games, Theorem 3.2 implies that there is some stable super-imputation $\mathbf{x}$ such that $x(N) \leq (tw(H) + 1)v(N)$.[3]

Finally, since a simple superadditive game can be viewed as a collection of intersecting sets, we obtain the following corollary, which may be of independent interest.

Corollary 3.3. *Let $H = \langle N, E\rangle$ be a graph, and let $R_k = \langle N, \mathcal{F}, k\rangle$ be an instance of* HITTING SET, *where $\mathcal{F} = \{S_j\}_{j=1}^m$ is a collection of pairwise intersecting subsets of N, and every S_j is connected in H (i.e., $\langle S_j, E|_{S_j}\rangle$ is connected). Then for all $k \leq tw(H) - 1$ it holds that R_k is a "yes"-instance of* HITTING SET *and a hitting set of size (at most) k can be found efficiently.*

3.3 Tightness

Demange [9] showed that if $tw(H) = 1$, then the game $G|_H$ admits a stable outcome, i.e., $RCoS(G|_H) = 1$. This result is limited to games whose interaction networks are trees. However, we will now show that if the treewidth of the interaction network is at least 2, then the upper bound of $tw(H) + 1$ proved in Theorem 3.2 is tight.

Theorem 3.4. *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 $A = \{z_1, \dots, z_{k+1}\}$. Further, for every unordered pair $I = \{i, j\}$, where $i, j \in \{1, \dots, k+1\}$ and $i \neq j$, we define a set D_I that consists of 7 agents and set $N = A \cup \bigcup_{i \neq j \in \{1,\dots,k+1\}} D_{\{i,j\}}$.

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 $I = \{i, j\} \subseteq \{1, \dots, k + 1\}$, 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 $i = 1, \dots, k + 1$ and every selection

[3] Note that, while the proof for *simple* superadditive games is straightforward, we cannot use the inductive argument made in Theorem 3.2 directly, as superadditivity may not be preserved; therefore, we must go through all steps of the proof.

$\left\{Q_{\{i,j\}} \in \mathcal{Q}_{\{i,j\}}\right\}_{j \neq i}$ 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 the 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 cost of stability in our game is at least $k+1$. □

We observe that Theorem 3.4 does not hold when $k = 1$, since this would stand in contradiction to Demange's theorem. Note that the width of our construction is at least 2 (each leaf is of size 3). For ease of exposition, we provide a somewhat simpler construction for the case of $k = 2$ in full detail.

Example 3.5. Consider the following game: there is a set $Z = \{z_1, z_2, z_3\}$ and three sets containing four players each: $A = \{a_1, \ldots, a_4\}$, $B = \{b_1, \ldots, b_4\}$ and $C = \{c_1, \ldots, c_4\}$; we set $N = A \cup B \cup C \cup Z$. The interaction network $H = \langle N, E \rangle$ is defined as follows: z_1 is connected by an edge to all agents in A and B, z_2 to all agents in B and C, and z_3 to all agents in C and A.

There are six winning coalitions:

$$W_{1,1} = \{z_1, a_1, a_4, b_3, b_1\}; \quad W_{1,2} = \{z_1, a_2, a_3, b_2, b_4\}$$
$$W_{2,1} = \{z_2, b_1, b_4, c_3, c_1\}; \quad W_{2,2} = \{z_2, b_2, b_3, c_2, c_4\}$$
$$W_{3,1} = \{z_3, c_1, c_4, a_3, a_1\}; \quad W_{3,2} = \{z_3, c_2, c_3, a_2, a_4\}$$

Any two winning coalitions intersect, so the optimal coalition structure has a value of 1. Given a stable super-imputation $\mathbf{x}$, it must be that $x(W_{1,1}) \geq 1$, thus $x_{z_1} \geq 1 - x_{a_1} + x_{a_4} + x_{b_3} + x_{b_1}$. A similar condition on $W_{1,2}$ gives us $x_{z_1} \geq 1 - x_{a_2} + x_{a_3} + x_{b_2} + x_{b_4}$. This means that $x_{z_1} \geq 1 - \frac{x(A)+x(B)}{2}$; a similar computation gives us that $x_{z_2} \geq 1 - \frac{x(B)+x(C)}{2}$, and $x_{z_3} \geq 1 - \frac{x(A)+x(C)}{2}$. Finally,

$$\begin{aligned} x(N) &= x(Z) + x(A) + x(B) + x(C) \\ &\geq 1 - \frac{x(A)+x(B)}{2} + 1 - \frac{x(B)+x(C)}{2} + 1 - \frac{x(A)+x(C)}{2} + x(A) + x(B) + x(C) \\ &= 3; \end{aligned}$$

since paying 1 to z_1, z_2 and z_3 is a stable super-imputation, we conclude that $RCoS(G) = 3$.

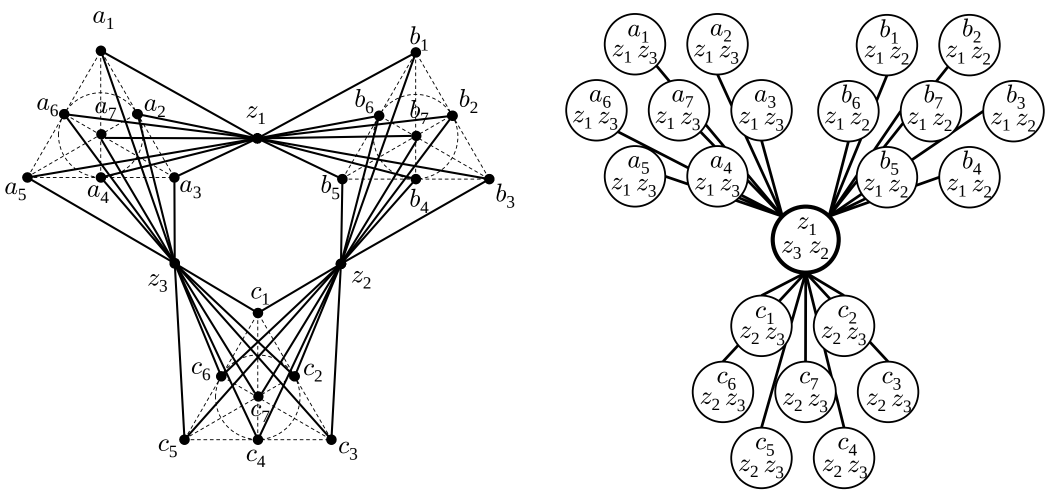

Figure 1: The interaction network H when $k = 2$ in Theorem 3.4. On the right there is the tree decomposition $\mathcal{T}$. There are three sets: $A = D_{1,3} = \{a_1, \ldots, a_7\}$, $B = D_{1,2} = \{b_1, \ldots, b_7\}$ and $C = D_{2,3} = \{c_1, \ldots, c_7\}$. An edge connects z_1 to all agents in A and B, z_2 to B and C, and z_3 to C and A. Agent z_1 forms winning coalitions with triplets of agents from A and B that are on a dotted line, Similarly, z_2 and z_3 form winning coalitions with their respective sets.

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 interaction networks with bounded pathwidth, the bound of $tw(H) + 1$ shown in Section 3 can be tightened.

Theorem 4.1. *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.*

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

```
Set 𝒯 = (A_1, ..., A_m);
x ← 0^n;
I ← {i ∈ N | v({i}) = 1};
for i ∈ I do
    x_i ← 1;
N_1 ← N \ I;
// Remove all singletons
t ← 1;
for j = 1 to m do
    if there is some S ⊆ N(𝒯_{A_j}) ∩ N_j such that v(S) = 1 then
        for i ∈ A_j ∩ N_j do
            if i ∈ N(𝒯_{A_j}) \ A_j then
                // Pay agents unless it is the first node they
                   appear in
                x_i ← 1
        N_{j+1} ← N_j \ N(𝒯_{A_j});
        // Remove all agents in N(𝒯_{A_j}) from the entire path
    else
        N_{j+1} ← N_j;
return x = (x_1, ..., x_n);
```

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 3.2. For simple games, our proof is very similar to the proof of Theorem 3.1; 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 3.1, 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 slightly modify the construction from Section 3.3 (we omit the full details). □

5 Discussion

Our main result shows a tight connection between the treewidth of an interaction network and the maximal subsidy required to stabilize a game played by the interacting agents. Simply put, as the interaction becomes "simpler", the game becomes easier to stabilize.

5.1 Implications

Our results have broad implications regarding the stability of cooperative games. While interaction networks have been introduced as an external restriction independent of the value function, in some families of cooperative games this restriction is implicit in the game description. A prominent example is *induced subgraph games* (ISG) by Deng and Papadimitriou [10], where agents correspond to vertices of a graph, and the value of a coalition is the sum of weights of the edges between coalition members. Imposing the very same graph as an interaction network will preserve the value of any coalition structure in the game. Therefore, we can deduce a bound on the $RCoS$ of a given ISG directly from its description, by measuring the treewidth of its underlying graph. Other families that implicitly induce a Myerson graph are matching games and some variations of network-flow games.

Hypergraphs Myerson's model can be generalized to use *hypergraphs* rather than graphs [21]. Since our methods work with tree decompositions rather than the interaction networks themselves, they apply equally well to this case. Interestingly, the underlying hypergraph of games defined via *marginal contribution nets* [**?**] also induces a Myerson (hyper)graph, which can in turn be used to bound the required subsidy.

Our result implies a separation between games whose interaction networks are acyclic, which have been shown to be stable [9], and other games. That is, treewidth of 1 implies $RCoS$ of 1, but for any higher value of treewidth, the $RCoS$ is somewhat higher. In particular, the result of Demange *is not* a special case of our theorem, although it can be proved using a very similar technique (i.e. by breaking to game to multiple simple games). Yet, a stronger bound is guaranteed if we can also bound the pathwidth of the interaction network—which is, however, quite restrictive.

The least-core While the CoS is a stability measure that reflects required *subsidies*, it is strongly related to other stability measures, such as the *least core* (An approximation of the core that allows coalitions to be "almost satisfied"). In particular, the CoS equals the minimal value of the weak-ε-core (multiplied by n), and provides an upper bound on the minimal value of the strong-ε-core [17]. It follows any bound on the treewidth or pathwidth of the underlying graph immediately supplies us with a bound on the level of dissatisfaction of coalitions in the least core.

5.2 Future work

While the bound on the subsidy 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.

More generally, we believe that this new 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 reveal other hidden connections in both cooperative and non-cooperative game theory.

Acknowledgements

This work was partially supported by Israel Ministry of Science and Technology grant #3-6797, and the Google Inter-University Center for Electronic Markets and Auctions.

References

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

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

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

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

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

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

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

[8] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science: Volume B: Formal Models and Semantics*, pages 193–242. Elsevier, Amsterdam, 1990.

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

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

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

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

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

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

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

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

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

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

[20] A. D. Taylor and W. S. Zwicker. *Simple games: desirability relations, trading, pseudoweightings*. Princeton University Press, 1999.

[21] A. van den Nouweland, P. Borm, and S. Tijs. Allocation rules for hypergraph communication situations. *International Journal of Game Theory*, 20:255–268, 1992.

Reshef Meir, Jeffrey S. Rosenschein
Hebrew University of Jerusalem, Israel
Email: reshef24@gmail.com, jeff@cs.huji.ac.il

Yair Zick, Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: yair0001@e.ntu.edu.sg, eelkind@ntu.edu.sg