# Hedonic Games with Graph-restricted Communication

Ayumi Igarashi
University of Oxford
Oxford, United Kingdom
ayumi.igarashi@cs.ox.ac.uk

Edith Elkind
University of Oxford
Oxford, United Kingdom
elkind@cs.ox.ac.uk

## ABSTRACT

We study hedonic coalition formation games in which cooperation among the players is restricted by a graph structure: a subset of players can form a coalition if and only if they are connected in the given graph. We investigate the complexity of finding stable outcomes in such games, for several notions of stability. In particular, we provide an efficient algorithm that finds an individually stable partition for an arbitrary hedonic game on an acyclic graph. We also introduce a new stability concept—in-neighbor stability—which is tailored for our setting. We show that the problem of finding an in-neighbor stable outcome admits a polynomial-time algorithm if the underlying graph is a path, but is NP-hard for arbitrary trees even for additively separable hedonic games; for symmetric additively separable games we obtain a PLS-hardness result.

## General Terms

Algorithms, Economics, Theory

## Keywords

Hedonic games, coalition formation, communication structure, trees

## 1. INTRODUCTION

In human and multiagent societies, agents often need to form coalitions in order to achieve their goals. The coalition formation process is guided by agents' beliefs about the performance of each potential coalition. Many important aspects of coalition formation can be studied using the formalism of *hedonic games* [3, 6]. In these games, each agent has preferences over all coalitions that she can be a part of, and an outcome is a partition of agents into coalitions. An important consideration in this context is *coalitional stability*: an outcome should be resistant to individual/group deviations, with different types of deviations giving rise to different notions of stability (such as core stability, individual stability, Nash stability, etc.; see the recent survey of Aziz and Savani [2] for an overview).

The standard model of hedonic games does not impose any restrictions on which coalitions may form. However, in reality we often encounter constraints on coalition formation. Consider, for instance, an international network of natural gas pipelines. It seems unlikely that two cities disconnected in the network would be able to coordinate a trading agreement without any help from intermediaries. Such restrictions on communication structure can be naturally described by undirected graphs, by identifying agents with nodes, communication links with edges, and feasible coalitions with connected subgraphs. In the context of cooperative transferable utility games this model was proposed in the seminal paper of Myerson [19], and has received a considerable amount of attention since then. In contrast, very little is known about hedonic games with graph-restricted communication, though some existing results for general non-transferable utility games have implications for this setting. In particular, the famous result of Demange [9] concerning stability in cooperative games on trees extends to non-transferable utility games, and implies that every hedonic game whose communication structure is acyclic admits a core stable partition (we discuss this result in more detail in Section 5). However, no attempt has been made to obtain similar results for other hedonic games solution concepts, or to explore algorithmic implications of constraints on the communication structure (such as acyclicity or having a small number of connected subgraphs) for computing the core and other solutions. The goal of this paper is to make the first step towards filling this gap.

**Our contribution** Inspired by Demange's work, we focus on hedonic games on acyclic graphs. We consider several well-studied notions of stability for hedonic games, such as individual stability, Nash stability, core stability and strict core stability (see Section 2 for definitions), and ask two questions: (1) does acyclicity of the communication structure guarantee the existence of a stable outcome? (2) does it lead to an efficient algorithm for computing a stable outcome, and if not, are there additional constraints on the communication structure that can be used to obtain such as algorithm? We remark that, in general, to represent the preferences of a player in an $n$-player hedonic game, we need to specify $2^{n-1}(2^{n-1}-1)/2$ values, which may be problematic if we are interested in algorithms whose running time is polynomial in $n$. We consider two approaches to circumvent this difficulty: (a) working in the oracle model, where an algorithm may submit a query of the form $(i, X, Y)$ where $X$ and $Y$ are two coalitions that both contain $i$, and learn in unit time whether $i$ prefers $X$ to $Y$, $Y$ to $X$ or is indifferent between them; (b) considering specific succinct representations of hedonic games, such as additively separable hedonic games [6], which can be described using $n(n-1)$ numbers.

We observe that Demange's algorithm for the core runs in time that is polynomial in the number of connected subtrees of the underlying graph $G$ (in the oracle model), and use similar ideas to obtain an algorithm for finding an outcome that is both core stable and individually stable as well as an algorithm for finding a Nash stable outcome (if it exists). The running time of these algorithms can be bounded in the same way; in particular, they run in polynomial time when $G$ is a path. However, we show that when $G$ is a star,

**Appears in:** *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

finding a core stable, strictly core stable or Nash stable outcome is NP-hard, even if we restrict ourselves to very simple subclasses of additively separable hedonic games. For symmetric additively separable hedonic games, we show that the PLS-hardness result for Nash stability [13] holds even if $G$ is a star.

In contrast, acyclicity turns out to be sufficient for individual stability: we show that every hedonic game on an acyclic graph admits an individually stable partition, and, moreover, such a partition can be computed in time polynomial in the number of players (in the oracle model). We believe that this result is remarkable, since in the absence of communication constraints finding an individually stable outcome is hard even for (symmetric) additively separable hedonic games [22, 13], and finding a Nash stable outcome in such games remains hard even for games on stars (Section 6).

Another contribution of our paper is a new stability concept that is tailored specifically to hedonic games on graphs, and captures the intuition that, to join a group, a player should be approved by the members of the group who know him. The resulting solution concept, which we call *in-neighbor stability*, lies between Nash stability and individual stability. However, we show that from the algorithmic perspective it behaves similarly to Nash stability; in particular, finding an in-neighbor stable outcome is NP-hard for additively separable hedonic games on stars and PLS-hard for symmetric additively separable hedonic games on stars. Our computational complexity results are summarized in Table 1.

**Related work** Sung and Dimitrov [22] were the first to consider complexity issues in additively separable hedonic games (ASHGs); they prove that it is NP-hard to determine if a game admits a core stable, strict core stable, individually stable, or Nash sable outcome (see also [20]). Aziz et al. [1] extend the first two or these results to symmetric additively separable hedonic games (SASHGs). While SASHGs always admit a Nash stable or individually stable partition [6, 7], finding one may still be difficult: Gairing and Savani [13] prove that finding such partitions is PLS-hard (PLS-hardness is a complexity class for total search problems, see [21]). The running time of some of our algorithms is polynomial in the number of connected coalitions; see the recent work of Elkind [12] for a characterization of graph families for which this quantity is polynomial in the number of nodes. Some papers [4, 5] use the phrase "hedonic game on a tree" to refer to a hedonic game where each player $i$ has a value $v_i(j)$ for every other player $j$, and pairs $\{i, j\}$ such that $v_i(j) \neq 0$ or $v_j(i) \neq 0$ form a tree; the preference relation of player $i$ is computed based on the values $v_i(j)$: the value of a coalition $S$, $i \in S$, could be $\sum_{j\in S\setminus\{i\}} v_i(j)$ (this corresponds to ASHGs) or $\frac{1}{|S|}\sum_{j\in S\setminus\{i\}} v_i(j)$ (such games are known as *fractional hedonic games*). This framework is different from ours: we allow preferences that are not derived from values assigned to individual players, and in the additively separable case we allow non-adjacent players to have a non-zero value for each other.

**Notes and Comments.** An extended version with full proofs is available on arXiv [16].

## 2. PRELIMINARIES

We start by introducing basic notation and definitions of hedonic games and graph theory.

**Hedonic games** A *hedonic game* is a pair $(N, (\succeq_i)_{i\in N})$ where $N$ is a finite set of players and each $\succeq_i$ is a complete and transitive preference relation over the nonempty subsets of $N$ including player $i$. The subsets of $N$ are referred to as *coalitions*. We let $\mathcal{N}_i$ denote the collection of all coalitions containing $i$. We call a coalition $X \subseteq N$ *individually rational* if $X \succeq_i \{i\}$ for all $i \in X$. Let $\succ_i$ denote the strict preference derived from $\succeq_i$, i.e., $X \succ_i Y$ if $X \succeq_i Y$, but $Y \not\succeq_i X$. Similarly, let $\sim_i$ denote the indifference relation induced by $\succeq_i$, i.e., $X \sim_i Y$ if $X \succeq_i Y$ and $Y \succeq_i X$.

An important subclass of hedonic games is *additively separable games*. These games model situations where each player has a specific value for every other player, and ranks coalitions according to the total value of their members [6]. Formally, a preference profile $(\succeq_i)_{i\in N}$ is said to be *additively separable* if there exists a utility matrix $U : N \times N \to \mathbb{R}$ such that for each $i \in N$ and each $X, Y \in \mathcal{N}_i$ we have $X \succeq_i Y$ if and only if $\sum_{j\in X} U(i, j) \geq \sum_{j\in Y} U(i, j)$ [6]. Without loss of generality, we will assume that $U(i, i) = 0$ for each $i \in N$. An additively separable preference is said to be *symmetric* if the utility matrix $U : N \times N \to \mathbb{R}$ is symmetric, i.e., $U(i, j) = U(j, i)$ for all $i, j \in N$. Dimitrov et al. [10] studied a subclass of additively separable preferences, which they called *enemy-oriented preferences*. Under these preferences each player considers every other player to be either a friend or an enemy, and has strong aversion towards her enemies: $U(i, j) \in \{1, -|N|\}$ for each $i, j \in N$ with $i \neq j$.

An *outcome* of a hedonic game is a partition of players into disjoint coalitions. Given a partition $\pi$ of $N$ and a player $i \in N$, let $\pi(i)$ denote the unique coalition in $\pi$ that contains $i$. The first stability concept we will introduce is *individual rationality*, which is often considered to be a minimum requirement that solutions should satisfy. A partition $\pi$ of $N$ is said to be *individually rational* if all players weakly prefer their own coalitions to staying alone, i.e., $\pi(i) \succeq_i \{i\}$ for all $i \in N$.

The *core* is one of the most studied solution concepts in hedonic games [11, 3, 6]. A coalition $X \subseteq N$ *strongly blocks* a partition $\pi$ of $N$ if $X \succ_i \pi(i)$ for all $i \in X$; it *weakly blocks* $\pi$ if $X \succeq_i \pi(i)$ for all $i \in X$ and $X \succ_j \pi(j)$ for some $j \in X$. A partition $\pi$ of $N$ is said to be *core stable* (CR) if no coalition $X \subseteq N$ strongly blocks $\pi$; it is said to be *strictly core stable* (SCR) if no coalition $X \subseteq N$ weakly blocks $\pi$.

We will also consider stability notions that capture resistance to deviations by individual players. Consider a player $i \in N$ and a pair of coalitions $X \notin \mathcal{N}_i$, $Y \in \mathcal{N}_i$. A player $i$ *wants to deviate* from $Y$ to $X$ if $X \cup \{i\} \succ_i Y$. A player $j \in X$ *accepts* a deviation of $i$ to $X$ if $X \cup \{i\} \succeq_j X$. A deviation of $i$ from $Y$ to $X$ is

- an *NS-deviation* if $i$ wants to deviate from $Y$ to $X$.
- an *IS-deviation* if it is an NS-deviation and all players in $X$ accept it.

A partition $\pi$ is called *Nash stable (NS)* (respectively, *individually stable (IS)*) if no player $i \in N$ has an NS-deviation (respectively, an IS-deviation) from $\pi(i)$ to another coalition $X \in \pi$ or to $\emptyset$.

We have the following containment relations among these classes of outcomes: SCR ⊆ CR, SCR ⊆ IS, NS ⊆ IS. However, a core stable outcome need not be individually stable, and an individually stable outcome may fail to be in the core.

**Graphs and digraphs** An *undirected graph*, or simply a *graph*, is a pair $(N, L)$, where $N$ is a finite set of *nodes* and $L \subseteq N \times N$ is a collection of *edges* between nodes. In this paper, we only consider graphs without self-loops and parallel edges. Given a set of nodes $X$, the *subgraph of* $(N, L)$ *induced by* $X$ is the graph $(X, L_X)$, where $L_X = \{\{i, j\} \in L \mid i, j \in X\}$.

For a graph $(N, L)$, a sequence of distinct nodes $(i_1, i_2, \ldots, i_k)$, $k \geq 2$, is called a *path* in $L$ if $\{i_h, i_{h+1}\} \in L$ for $h = 1, 2, \ldots, k-1$. A path $(i_1, i_2, \ldots, i_k)$, $k \geq 3$, is said to be a *cycle* in $L$ if $\{i_k, i_1\} \in L$. A graph $(N, L)$ is said to be a *forest* if it contains no cycles. A subset $X \subseteq N$ is said to be *connected in* $(N, L)$ if for every pair of distinct nodes $i, j \in X$ there is a path between $i$ and $j$ in $L_X$. The collection of all connected subsets of $N$ in $(N, L)$ is denoted by $\mathcal{F}_L$; also, we write $\mathcal{F}_L(i) = \mathcal{F}_L \cap \mathcal{N}_i$. By convention,

| | Acyclic graphs | | Stars | | | Paths |
|---|---|---|---|---|---|---|
| | Arbitrary | Additive | Additive | S-additive | S-enemy | Arbitrary |
| SCR | - | NP-h | NP-h | NP-h (Th.12) | NP-h$^*$ (Th.11) | ? |
| CR | - | NP-h$^*$ | NP-h$^*$ | NP-h$^*$ | NP-h$^*$ (Th.10) | P ([9]) |
| NS | - | NP-c | NP-c (Th.16) | PLS-c (Th.19) | P (Prop.20) | P (Th.13) |
| INS | - | NP-c | NP-c (Th.15) | PLS-c (Th.18) | P | P (Th.13) |
| IR-INS | - | NP-c ([16]) | P (Prop.17) | P | P | P (Th.13) |
| IS | P (Th.1) | P | P | P | P | P |

**Table 1: Complexity of computing stable outcomes for hedonic games on acyclic graphs. The top row corresponds to restrictions on graphs; the second row from the top indicates restrictions on preference profiles. The positive results for unrestricted preferences are in the oracle model. To avoid dealing with representation issues, when a problem is NP-hard for additively separable games on trees, we do not consider its complexity for unrestricted preferences (indicated by '-'). The hardness result marked with $^*$ holds with respect to Turing reductions. When no reference is given, the result follows trivially from other results in the table. The results for paths hold for all trees with $n$ nodes and $\mathrm{poly}(n)$ connected subtrees.**

we assume that $\emptyset \notin \mathcal{F}_L$. A forest $(N, L)$ is said to be a *tree* if $N$ is connected in $(N, L)$. A tree $(N, L)$ is called a *star* if there exists a central node $s \in N$ such that $L = \{\, \{s, j\} \mid j \in N \setminus \{s\} \,\}$. A subset $X \subseteq N$ of a graph $(N, L)$ is said to be a *clique* if for every pair of distinct nodes $i, j \in X$ we have $\{i, j\} \in L$.

A *directed graph*, or a *digraph*, is a pair $(N, A)$ where $N$ is a finite set of nodes and $A$ is a family of ordered pairs of nodes from $N$. The elements of $A$ are called the *arcs*. A sequence of distinct nodes $(i_1, i_2, \ldots, i_k)$, $k \geq 2$, is called a *directed path* in $A$ if $(i_h, i_{h+1}) \in A$ for $h = 1, 2, \ldots, k-1$. Given a digraph $(N, A)$, let $L(A) = \{\, \{i, j\} \mid (i, j) \in A \,\}$: the graph $(N, L(A))$ is the *undirected version* of $(N, A)$. A digraph $(N, A)$ is said to be a *rooted tree* if $(N, L(A))$ is a tree and each node has at most one arc entering it. A rooted tree has exactly one node that no arc enters, called the *root*, and there exists a unique directed path from the root to every node of $N$.

Let $(N, A)$ be a rooted tree. We say that a node $j \in N$ is a *parent* of $i$ in $A$ if $(j, i) \in A$. We denote by $\mathrm{pr}(i, A)$ the unique parent of $i$ in $A$. A node $j \in N$ is called a *successor* of $i$ in $A$ if there exists a directed path from $i$ to $j$ in $A$. We write

$$\mathrm{succ}(i, A) = \{i\} \cup \{\, j \in N \mid j \text{ is a successor of } i \text{ in } A \,\}.$$

A node $i \in N$ is called a *child* of $X \subseteq N$ in $A$ if $i \notin X$ and $\mathrm{pr}(i, A) \in X$. We write

$$\mathrm{ch}(X, A) = \{\, i \in N \mid i \notin X \text{ and } \mathrm{pr}(i, A) \in X \,\}.$$

The *height* of a node $i \in N$ of $(N, A)$ is defined inductively: $\mathrm{height}(i, A) = 0$ if $\mathrm{succ}(i, A) = \{i\}$ and $\mathrm{height}(i, A) = 1 + \max_{j \in \mathrm{ch}(\{i\}, A)} \mathrm{height}(j, A)$ otherwise.

## 3. OUR MODEL

The goal of this paper is to study hedonic games where agent communication is constrained by a graph.

DEFINITION 1. *A* hedonic game with graph structure, *or a* hedonic graph game, *is a triple $(N, (\succeq_i)_{i \in N}, L)$ where $(N, (\succeq_i)_{i \in N})$ is a hedonic game, and $L \subseteq \{\, \{i, j\} \mid i \neq j, i, j \in N \,\}$ is the set of communication links between players. A coalition $X \subseteq N$ is said to be* feasible *if it is connected in $(N, L)$.*

If $(N, L)$ is a clique, a hedonic graph game $(N, (\succeq_i)_{i \in N}, L)$ is equivalent to the ordinary hedonic game $(N, (\succeq_i)_{i \in N})$.

A partition $\pi$ of $N$ is said to be *feasible* if $\pi \subseteq \mathcal{F}_L$. An *outcome* of a hedonic graph game is a feasible partition. The standard definitions of stability concepts (see Section 2) can be adapted to hedonic graph games in a straightforward manner. Specifically, we say that a coalitional deviation is *feasible* if the deviating coalition itself is feasible; an individual deviation where player $i$ joins a coalition $X$ is *feasible* if $X \cup \{i\}$ is feasible. Now, we modify the definitions in Section 2 by only requiring stability against feasible deviations.

We use the notation $(N, U, L)$ to denote an additively separable graph game with utility matrix $U : N \times N \to \mathbb{R}$.

EXAMPLE 1. Consider the coalition formation problem in a parliament consisting of three parties: left-wing ($\ell$), centrist ($\mathsf{c}$), and right-wing ($\mathsf{r}$). Then $\ell$ and $\mathsf{r}$ cannot form a coalition without $\mathsf{c}$. We describe this scenario as an additively separable graph game $(N, U, L)$ where $N = \{\ell, \mathsf{c}, \mathsf{r}\}$, $L = \{\{\ell, \mathsf{c}\}, \{\mathsf{c}, \mathsf{r}\}\}$, and the utility matrix $U$ is given by

$$U(\ell, \mathsf{c}) = 1, U(\ell, \mathsf{r}) = -2, U(\mathsf{c}, \ell) = 2,$$
$$U(\mathsf{c}, \mathsf{r}) = 0, U(\mathsf{r}, \mathsf{c}) = 2, U(\mathsf{r}, \ell) = 0.$$

The resulting preference profile is as follows:

$$\ell \;:\; \{\ell, \mathsf{c}\} \succ_\ell \{\ell\} \succ_\ell \{\ell, \mathsf{c}, \mathsf{r}\} \succ_\ell \{\ell, \mathsf{r}\}$$
$$\mathsf{c} \;:\; \{\ell, \mathsf{c}, \mathsf{r}\} \sim_\mathsf{c} \{\ell, \mathsf{c}\} \succ_\mathsf{c} \{\mathsf{c}, \mathsf{r}\} \sim_\mathsf{c} \{\mathsf{c}\}$$
$$\mathsf{r} \;:\; \{\ell, \mathsf{c}, \mathsf{r}\} \sim_\mathsf{r} \{\mathsf{c}, \mathsf{r}\} \succ_\mathsf{r} \{\ell, \mathsf{r}\} \sim_\mathsf{r} \{\mathsf{r}\}$$

The individually rational feasible partitions of this game are $\pi_1 = \{\{\ell, \mathsf{c}\}, \{\mathsf{r}\}\}$, $\pi_2 = \{\{\ell\}, \{\mathsf{c}, \mathsf{r}\}\}$, and $\pi_3 = \{\{\ell\}, \{\mathsf{c}\}, \{\mathsf{r}\}\}$. The partition $\pi_1$ is both core stable and individually stable. However, there is no Nash stable partition in this game: in $\pi_1$, player $\mathsf{r}$ wants to join $\{\ell, \mathsf{c}\}$, while in $\pi_2$ and $\pi_3$, player $\mathsf{c}$ wants to join $\{\ell\}$.

## 4. INDIVIDUAL STABILITY

The main contributions of this section are (1) an efficient algorithm for finding an individually stable feasible partition in a hedonic graph game whose underlying graph is a forest; (2) a proof that in the presence of cycles the existence of an IS feasible partition is not guaranteed.

THEOREM 1. *Suppose that we are given oracle access to the preference relations $\succeq_i$ of all players in a hedonic graph game $\mathcal{G} = (N, (\succeq_i)_{i \in N}, L)$, where $(N, L)$ is a forest. Then we can find an individually stable feasible outcome of $\mathcal{G}$ in time polynomial in $|N|$.*

PROOF. We first give an informal description of our algorithm, followed by pseudocode. If the input graph $(N, L)$ is a forest, we can process each of its connected components separately, so we can assume that $(N, L)$ is a tree. We choose an arbitrary node $r$ to be the root; this transforms $(N, L)$ into a rooted tree $(N, A^r)$

with root $r$ and determines a hierarchy of players. For each player $i$, from the bottom player to the top of the hierarchy, we compute a tentative partitioning of the subtree rooted at $i$. To this end, among all coalitions that $i$'s children belong to, we identify those whose members would be willing to let $i$ join them. Then we let $i$ choose between his most preferred option among all such coalitions and the singleton $\{i\}$. We then check if any of the successors of $i$ who are adjacent to $i$'s coalition want to join it; we let them do so if they are approved by the current coalition members.

For a family of subsets $\mathcal{P} \subseteq \mathcal{N}_i$, we set

$$\max_i \mathcal{P} = \{\, X \in \mathcal{P} \mid X \succeq_i Y \text{ for all } Y \in \mathcal{P} \,\}.$$

Given a pair of nonempty subsets $X, Y \subseteq N$, we write $X \overset{m}{\succeq} Y$ if $X \cap Y \neq \emptyset$ and $X \succeq_i Y$ for all $i \in X \cap Y$.

---

**Algorithm 1** Finding IS partitions

---

**Input:** tree $(N, L)$, $r \in N$, oracles for $\succeq_i$, $i \in N$.
**Output:** $\pi^{(r)}$.

1: make a rooted tree $(N, A^r)$ with root $r$ by orienting all the edges in $L$.
2: initialize $\mathrm{B}(i) \leftarrow \emptyset$ and $\pi^{(i)} \leftarrow \emptyset$ for each $i \in N$.
3: **for** $t = 0, \ldots, \mathrm{height}(r, A^r)$ **do**
4:     **for** $i \in N$ with $\mathrm{height}(i, A^r) = t$ **do**
5:         $C(i) = \{\, k \in \mathrm{ch}(\{i\}, A^r) \mid \mathrm{B}(k) \cup \{i\} \overset{m}{\succeq} \mathrm{B}(k) \,\}$.
6:         choose $\mathrm{B}(i) \in \max_i(\{\{i\}\} \cup \{\, \mathrm{B}(k) \cup \{i\} \mid k \in C(i) \,\})$.
7:         **while** there exists $j \in \mathrm{ch}(\mathrm{B}(i), A^r)$ such that $\mathrm{B}(i) \cup \{j\} \succ_j \mathrm{B}(j)$ and $\mathrm{B}(i) \cup \{j\} \overset{m}{\succeq} \mathrm{B}(i)$ **do**
8:             $\mathrm{B}(i) \leftarrow \mathrm{B}(i) \cup \{j\}$
9:         **end while**
10:        $\pi^{(i)} \leftarrow \{\mathrm{B}(i)\} \cup \{\, \pi^{(k)} \mid k \in \mathrm{ch}(\mathrm{B}(i), A^r) \,\}$
11:    **end for**
12: **end for**

---

We will now argue that Algorithm 1 correctly identifies an individually stable partition. Our argument is based on two lemmas.

LEMMA 2. *For every $i \in N$ and every $k \in ch(\{i\}, A^r)$, if $\mathrm{B}(k) \cup \{i\} \overset{m}{\succeq} \mathrm{B}(k)$, then $\mathrm{B}(i) \succeq_i \mathrm{B}(k) \cup \{i\}$.*

Lemma 2 follows immediately from the choice of $\mathrm{B}(i)$ in Line 6 and the stopping criterion of the **while** loop in lines 7–9.

LEMMA 3. *For each $i \in N$, $j \in succ(i, A^r)$ and all $X \in \pi^{(i)} \cup \{\emptyset\}$ there is no IS feasible deviation of $j$ from $\pi^{(i)}(j)$ to $X$.*

PROOF. We use induction on $\mathrm{height}(i, A^r)$. For $\mathrm{height}(i, A^r) = 0$ our assertion is trivial. Suppose that it holds for all $j \in N$ with $\mathrm{height}(j, A^r) \leq t - 1$, and consider $i \in N$ with $\mathrm{height}(i, A^r) = t$. Assume towards a contradiction that there exists an IS feasible deviation of $j \in \mathrm{succ}(i, A^r)$ from $\pi^{(i)}(j)$ to $X \in \pi^{(i)} \cup \{\emptyset\}$, i.e., $X \cup \{j\}$ is connected, and

$$X \cup \{j\} \succ_j \pi^{(i)}(j), \text{ and} \tag{1}$$

$$X \cup \{j\} \overset{m}{\succeq} X. \tag{2}$$

By construction, $\pi^{(i)}$ is individually rational, so $X \neq \emptyset$. If $j \notin \mathrm{B}(i)$ and $X \neq \mathrm{B}(i)$, this would contradict the induction hypothesis. Moreover, by the stopping criteria in Line 7, it cannot be the case that $j \notin \mathrm{B}(i)$ and $X = \mathrm{B}(i)$. Thus, $j \in \mathrm{B}(i)$ and $X = \mathrm{B}(k)$ for some $k \in \mathrm{ch}(\{j\}, A^r)$. If $j = i$, Lemma 2 and (2) imply that $\pi^{(i)}(i) = \mathrm{B}(i) \succeq_i \mathrm{B}(k) \cup \{i\}$, contradicting (1). Thus, $j \neq i$.

Suppose that player $j$ joins the coalition $\mathrm{B}(i)$ when $\mathrm{B}(i)$ is initialized in Line 6. Then, $j \in \mathrm{B}(k^*)$ for some $k^* \in \mathrm{ch}(\{i\}, A^r)$. It follows that $\mathrm{B}(k) \in \pi^{(k^*)}$, since $k \notin \mathrm{B}(k^*)$ and $k \in \mathrm{ch}(\{j\}, A^r)$. Further, the second stopping criterion of the **while** loop in Line 7 ensures that $j$'s utility does not decrease during the execution of Algorithm 1. Thus, $\mathrm{B}(i) \succeq_j \mathrm{B}(k^*)$. Combining this with (1) and (2) yields

$$\mathrm{B}(k) \cup \{j\} \succ_j \pi^{(i)}(j) = \mathrm{B}(i) \succeq_j \mathrm{B}(k^*) = \pi^{(k^*)}(j),$$

and $\mathrm{B}(k) \cup \{j\} \overset{m}{\succeq} \mathrm{B}(k)$. It follows that $\pi^{(k^*)}$ admits an IS feasible deviation of $j$ from $\pi^{(k^*)}(j)$ to $\mathrm{B}(k) \in \pi^{(k^*)}$. This contradicts the induction hypothesis.

On the other hand, suppose that player $j$ joins $\mathrm{B}(i)$ during the **while** loop in Lines 7–9. Then at that point player $j$ is made better off by leaving $\mathrm{B}(j)$ and joining $\mathrm{B}(i)$. From then on, she vetoes all candidates whose presence would make her worse off. Hence, $\mathrm{B}(i) \succ_j \mathrm{B}(j)$. However, (2) and Lemma 2 imply that $\mathrm{B}(j) \succeq_j \mathrm{B}(k) \cup \{j\}$. Thus, $\mathrm{B}(i) \succ_j \mathrm{B}(k) \cup \{j\}$, contradicting (1). □

The partition $\pi^{(r)}$ is feasible by construction, so applying Lemma 3 with $i = r$ implies that $\pi^{(r)}$ is an individually stable feasible partition of $N$.

It remains to analyze the running time of Algorithm 1. Consider the execution of the algorithm for a fixed player $i$. Let $c = |\mathrm{ch}(\{i\}, A^r)|$, $s = |\mathrm{succ}(i, A^r)|$. Line 5 requires at most $s$ oracle queries: no successor of $i$ is queried more than once. Line 6 requires $c$ oracle queries. Moreover, at each iteration of the **while** loop in lines 7–9 at least one player joins $\mathrm{B}(i)$, so there are at most $s$ iterations, in each iteration we consider at most $s$ candidates, and for each candidate we perform at most $s$ queries. Summing over all players, we conclude that the number or oracle queries is bounded by $O(|N|^4)$. This completes the proof of the theorem. □

Theorem 1 provides a constructive proof that every hedonic graph game whose underlying graph $(N, L)$ is a forest admits an individually stable feasible partition. In contrast, if $(N, L)$ contains a cycle, the players' preferences can always be chosen so that no individually stable feasible partition exists.

PROPOSITION 4. *Suppose that the graph $(N, L)$ contains a cycle $C = \{i_1, i_2, \ldots, i_k\}$ with $k \geq 3$, $\{i_h, i_{h+1}\} \in L$ for $h = 1, 2, \ldots, k$, where $i_{k+1} := i_1$. Then, we can choose preference relations $(\succeq_i)_{i \in N}$ so that the set of IS feasible partitions of the game $(N, (\succeq_i)_{i \in N}, L)$ is empty.*

PROOF. Let $d$ be the smallest natural number that does not divide $k$. For each $i_h \in C$, define

$$\mathcal{C}(i_h) = \{\, X \in \mathcal{F}_L(i_h) \mid X \subseteq C, |X| \leq d \,\}.$$

Notice that $d \geq 2$, and hence $\{i_h\}, \{i_h, i_{h+1}\} \in \mathcal{C}(i_h)$. Define a graph game $(N, (\succeq_i)_{i \in N}, L)$ so that for each $i_h \in C$ we have

1. $X \succ_{i_h} Y$ for all $X, Y \in \mathcal{C}(i_h)$ such that $i_{h+1} \in X \setminus Y$,
2. $X \sim_{i_h} Y$ for all $X, Y \in \mathcal{C}(i_h)$ such that $i_{h+1} \in X \cap Y$,
3. $X \sim_{i_h} Y$ for all $X, Y \in \mathcal{C}(i_h)$ such that $i_{h+1} \notin X \cup Y$,
4. $X \succ_{i_h} Y$ for all $X \in \mathcal{C}(i_h), Y \notin \mathcal{C}(i_h)$.

Let $\pi$ be an arbitrary individually rational feasible partition of $N$. By individual rationality, $\pi(i) \in \mathcal{C}(i)$ for every $i \in C$. Since $d$ does not divide $k$, there exists $Y \in \pi$ such that $|Y| \leq d - 1$ and $Y \subseteq C$. Let $Y = \{i_{h+1}, i_{h+2}, \ldots, i_m\}$. Then we have $\{i_h\} \cup Y \succ_{i_h} \pi(i_h)$, and $\{i_h\} \cup Y \sim_{i_j} Y$ for all $i_j \in Y$. Thus, $\pi$ is not individually stable. □

We summarize our results for individually stable feasible partitions in the following corollary.

COROLLARY 5. *For the class of hedonic graph games, the following statements are equivalent.*

($i$) $(N, L)$ *is a forest.*

($ii$) *For every hedonic graph game* $(N, (\succeq_i)_{i\in N}, L)$ *there exists an individually stable feasible partition of* $N$.

# 5. CORE STABILITY

As mentioned in Section 1, classic results by Le Breton et al. [18] and Demange [8, 9] for non-transferable utility games imply an analogue of Corollary 5 for core stable partitions.

THEOREM 6 ([18, 8, 9]). *For the class of hedonic games with graph structure, the following statements are equivalent.*

($i$) $(N, L)$ *is a forest.*

($ii$) *For every hedonic graph game* $(N, (\succeq_i)_{i\in N}, L)$ *there exists a core stable feasible partition of* $N$.

We will now show that these two results can be combined, in the following sense: if $(N, L)$ is acyclic, then every hedonic game on $(N, L)$ admits a feasible partition that belongs to the core and is individually stable; moreover, the converse is also true.

THEOREM 7. *For the class of hedonic games with graph structure, the following statements are equivalent.*

(i) $(N, L)$ *is a forest.*

(ii) *For every hedonic graph game* $(N, (\succeq_i)_{i\in N}, L)$*, there exists a feasible partition of* $N$ *that belongs to the core and is individually stable.*

PROOF. The direction (ii) $\Rightarrow$ (i) immediately follows from Theorem 6. We will prove (i) $\Rightarrow$ (ii). Recall that a preference relation $\succeq_1$ on a set $X$ is a *refinement* of a preference relation $\succeq_2$ on $X$ if for every $a, b \in X$ it holds that $a \succ_2 b$ implies $a \succ_1 b$.

Suppose that $(N, L)$ is a forest. For each $i \in N$, let $>_i$ be a refinement of $\succeq_i$ that satisfies $X >_i Y$ whenever $X \sim_i Y$ and $Y \subsetneq X$. By Theorem 6, the game $(N, (>_i)_{i\in N}, L)$ admits a core stable feasible partition $\pi$. By construction, $\pi$ is core stable in the original game $(N, (\succeq_i)_{i\in N}, L)$ as well. We will now argue that it is also individually stable. Assume towards a contradiction that there exists an IS feasible deviation of some player $i \in N$ from $\pi(i)$ to $X \in \pi \cup \{\emptyset\}$. That is, $X \cup \{i\} \in \mathcal{F}_L$, $X \cup \{i\} \succ_i \pi(i)$, and $X \cup \{i\} \succeq_j X$ for each $j \in X$. By construction of $(>_i)_{i\in N}$, this implies that $X \cup \{i\} >_i \pi(i)$ and $X \cup \{i\} >_j X$ for each $j \in X$. This contradicts the fact that $\pi$ is a core stable partition of the game $(N, (>_i)_{i\in N}, L)$. $\square$

## 5.1 Computational complexity of CR

Demange's proof that every hedonic graph game on an acyclic graph admits a core stable outcome is constructive: her paper [9] provides an algorithm to find a core stable partition. This algorithm is similar in flavor to Algorithm 1: it processes the players starting from the leaves and moving towards the root, calculates the "guarantee level" of each player, and then partitions the players into disjoint groups in such a way that the final outcome satisfies their "guarantee levels"; this is shown to ensure core stability. While Demange does not analyze the running time of her algorithm, it can be verified that it runs in time polynomial in the number of connected subsets of the underlying graph. Thus, in particular, Demange's algorithm runs in polynomial time if this graph is a path.

THEOREM 8 (IMPLICIT IN [9]). *Suppose that we are given oracle access to the preference relations* $\succeq_i$ *of all players in a hedonic graph game* $\mathcal{G} = (N, (\succeq_i)_{i\in N}, L)$*, where* $(N, L)$ *is a forest. Then we can find a core stable feasible outcome of* $\mathcal{G}$ *in time polynomial in the number of connected subsets of* $(N, L)$.

Combining Demange's algorithm with the construction in the proof of Theorem 7, we obtain an algorithm that has the same worst-case running time as Demange's algorithm and outputs a feasible partition that belongs to the core and is individually stable.

COROLLARY 9. *Suppose that we are given oracle access to the preference relations* $\succeq_i$ *of all players in a hedonic graph game* $\mathcal{G} = (N, (\succeq_i)_{i\in N}, L)$*, where* $(N, L)$ *is a forest. Then we can find a feasible outcome of* $\mathcal{G}$ *that belongs to the core and is individually stable in time polynomial in the number of connected subsets of* $(N, L)$.

However, if the number of connected subsets of $(N, L)$ is superpolynomial in $|N|$, so is the running time of Demange's algorithm, because for each player $i$ this algorithm considers all feasible coalitions containing $i$. Now, for many $n$-node trees the number of connected subtrees is superpolynomial in $n$: for instance, this is the case for every tree with $\omega(\log n)$ leaves, simply because we can delete any subset of leaves and still obtain a connected graph. Therefore, it is natural to ask if checking all feasible coalitions is indeed necessary. Note that when the goal is to find an individually stable partition, the answer to this question is "no": Algorithm 1 only considers some of the feasible coalitions, yet is capable of finding an individually stable feasible outcome. In contrast, for the core it seems unlikely that one can obtain a substantial improvement over the running time of Demange's algorithm: our next theorem shows that finding a core stable feasible partition of an additively separable hedonic graph game is NP-hard even if the underlying graph is a star and even if the input game is a symmetric enemy-oriented game.

THEOREM 10. *If one can find a core stable feasible partition in a symmetric enemy-oriented graph game whose underlying graph is a star in time polynomial in the number of players then* P = NP.

PROOF. We provide a reduction from the NP-complete CLIQUE problem [15]. An instance of CLIQUE is a pair $(G, t)$, where $G$ is an undirected graph and $t$ is a positive integer. It is a "yes"-instance if $G$ contains a clique of size at least $t$ and a "no"-instance otherwise. We will show how a polynomial-time algorithm for our problem can be used to decide CLIQUE in polynomial time.

Given an instance $(G, t)$ of CLIQUE, where $G = (V, E)$, we construct a symmetric enemy-oriented graph game as follows. We let $N = V \cup \{s\}$ and $L = \{\,\{s, v\} \mid v \in V\,\}$. We will now describe the (symmetric) matrix $U$. Briefly, player $s$ likes all other players and two players in $V$ like each other if and only if they are connected by an edge of $G$. Formally, we set $U(s, v) = 1$ for each $v \in V$ and for each $u, v \in V$ we set $U(u, v) = 1$ if $\{u, v\} \in E$ and $U(u, v) = -|V| - 1$ otherwise.

Let $\pi$ be an individually rational feasible partition of this game. Note that all players in $N \setminus \pi(s)$ form singleton coalitions in $\pi$, and $\pi(s) \setminus \{s\}$ is a clique in $G$. We will now argue that $\pi$ is core stable if and only if $\pi(s) \setminus \{s\}$ is a maximum-size clique in $G$.

Indeed, if $C$ is a maximum-size clique in $G$ and $|\pi(s) \setminus \{s\}| < |C|$, every player in $C \cup \{s\}$ strictly prefers $C \cup \{s\}$ to its current coalition. Conversely, suppose that $\pi(s) \setminus \{s\}$ is a maximum-size clique, yet coalition $X$ strictly blocks $\pi$. Then it has to be the case that $s \in X$, and hence $|X| > |\pi(s)|$; but this means that $X \setminus \{s\}$

is not a clique, and therefore players in $X \setminus \{s\}$ prefer $\pi$ to $X$, a contradiction.

It follows that, by looking at a core stable feasible outcome $\pi$, we can decide whether $G$ contains a clique of size at least $t$. $\square$

Dimitrov et al. [10] show that finding a core outcome in symmetric enemy-oriented games is NP-hard; however, in their model there is no constraint on communication among the players, i.e., their result is for the case where $(N, L)$ is a clique, whereas our result holds even if $(N, L)$ is a star.

## 5.2 Computational complexity of SCR

Unlike core stable outcomes, strictly core stable outcomes need not exist even in symmetric enemy-oriented games on stars. Consider, for instance, a variant of our parliamentary coalition formation example (Example 1) where the centrist party (c) is equally happy to collaborate with the left-wing party ($\ell$) or the right-wing party (r), but the left-wing party and the right-wing party hate each other. This setting can be captured by a symmetric enemy-oriented graph game whose underlying graph is a path, and whose core stable feasible partitions are $\pi_1 = \{\{\ell, \mathsf{c}\}, \{\mathsf{r}\}\}$ and $\pi_2 = \{\{\ell\}, \{\mathsf{c}, \mathsf{r}\}\}$. However, neither $\pi_1$ nor $\pi_2$ is in the strict core: $\pi_1$ is weakly blocked by $\{\mathsf{c}, \mathsf{r}\}$ and $\pi_2$ is weakly blocked by $\{\ell, \mathsf{c}\}$.

Our next theorem shows that checking whether a given symmetric enemy-oriented hedonic graph game admits a feasible outcome in the strict core is NP-hard with respect to Turing reductions, even if the underlying graph is a star.

THEOREM 11. *If there exists a polynomial-time algorithm that, given a symmetric enemy-oriented graph game whose underlying graph is a star, decides whether this game has a strictly core stable feasible partition then* P = NP.

PROOF SKETCH. We define UNIQUE CLIQUE as the decision problem of determining whether a graph has a unique maximum-size clique. That is, let $s$ be the size of a maximum clique in $G$; $G$ is a "yes"-instance of UNIQUE CLIQUE if it contains exactly one clique of size $s$ and a "no"-instance otherwise. CLIQUE admits a Turing reduction to UNIQUE CLIQUE. Specifically, given an instance $(G, t)$ of CLIQUE where $G = (V, E)$, we construct $|V|$ instances of UNIQUE CLIQUE as follows. For each $s = 1, \dots, |V|$, let $C_s$ be a set of size $s$ with $V \cap C_s = \emptyset$, and let $H_s = (V \cup C_s, E_s)$, where $\{u, v\} \in E_s$ if and only if $u, v \in C_s$, or $u, v \in V$ and $\{u, v\} \in E$. Note that the maximum clique size in $G$ is $r$ if and only if $H_r$ is a "no"-instance of UNIQUE CLIQUE, but $H_s$ is a "yes"-instance of UNIQUE CLIQUE for $r = s+1, \dots, |V|$. Hence, a polynomial-time algorithm for UNIQUE CLIQUE can be used to decide CLIQUE in polynomial time.

We will now argue that UNIQUE CLIQUE can be reduced to our problem. Given an undirected graph $G = (V, E)$, we construct the same symmetric enemy-oriented graph game $(N, U, L)$ as in the proof of Theorem 10. Then, one can readily see that the $G$ is a "yes"-instance of UNIQUE CLIQUE if and only if $(N, U, L)$ has a feasible outcome that is strictly core stable. $\square$

For the broader class of symmetric additively separable hedonic graph games on stars, we obtain an NP-hardness result under the more standard notion of a many-one reduction (for games on cliques, this follows from the results of Aziz et al. [1]).

THEOREM 12. *Given a symmetric additively separable hedonic graph game whose underlying graph is a star, it is* NP*-hard to determine whether it has a strictly core stable feasible partition.*

PROOF SKETCH. Again, we provide a reduction from CLIQUE. Given an undirected graph $G = (V, E)$ and a positive integer

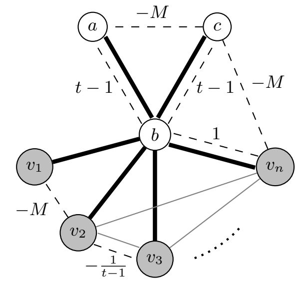


**Figure 1: Graph used in the proof of Theorem 12. Thick black lines represent communication links between players, whereas gray lines stand for edges of the given instance $G$. Values on dashed lines are utilities of players.**

$t \geq 2$, we construct a symmetric additively separable graph game $(N, U, L)$ where $N = \{a, b, c\} \cup V$, $L = \{\{a, b\}, \{b, c\}\} \cup \{\{b, v\} \mid v \in V\}$. Let $M = |N| + 1$. The utility matrix $U : N \times N \to \mathbb{R}$ is given as follows (see Figure 1):

$$U(a,b) = U(c,b) = t-1, U(a,c) = -M,$$
$$U(a,v) = U(c,v) = -M, U(b,v) = 1 \text{ for each } v \in V,$$
$$U(u,v) = -1/(t-1) \text{ if } \{u,v\} \in E$$
$$\text{and } U(u,v) = -M \text{ otherwise, for all } u, v \in V.$$

Suppose that $G$ contains a clique $C$ of size $t$. Then, the partition $\pi = \{\{a\}, \{c\}, C \cup \{b\}\} \cup \{\{v\} \mid v \in V \setminus C\}$ can be shown to be strictly core stable. Conversely, if there exists a strictly core stable feasible partition $\pi$ of $N$, $\pi(b) \setminus \{b\}$ forms a clique of size at least $t$ in $G$. $\square$

# 6. IN-NEIGHBOR STABILITY

In many real-life situations, when people move from one group to another, they need approvals from their contacts in the new group. Suppose, for instance, that Alice is an early-career researcher applying for academic positions in universities: her application is unlikely to be accepted if it is rejected by her prospective mentors (even if Alice expects to collaborate with several other faculty members as well).

Motivated by these considerations, we will now describe a new notion of stability, which is specific to hedonic graph games.

DEFINITION 2. *Given a hedonic graph game $(N, (\succeq_i)_{i \in N}, L)$, we say that $j$ is a* neighbor *of $i$ if $\{i, j\} \in L$. A feasible deviation of a player $i$ to $X \notin \mathcal{N}_i$ is called*

- in-neighbor feasible *if it is NS feasible and accepted by all of $i$'s neighbors in $X$.*
- IR-in-neighbor feasible *if it is in-neighbor feasible and for all $j \in X$ it holds that $X \cup \{i\} \succeq_j \{j\}$.*

*A feasible partition $\pi$ is called* in-neighbor stable (INS) *(respectively,* IR in-neighbor stable (IR-INS)*) if no player $i$ has an in-neighbor feasible deviation (respectively, an IR-in-neighbor feasible deviation) from $\pi(i)$ to a coalition $X \in \pi \cup \{\emptyset\}$.*

Note that every INS partition is IR-INS, and each IR-INS partition is individually stable. However, the converse may not be true: partition $\pi_1$ in Example 1 is individually stable, but admits an in-neighbor feasible deviation. Indeed, all IR partitions in that example are not in-neighbor stable, so the existence of in-neighbor stable outcomes is not guaranteed, even in additively separable games on paths. Note however, that $\pi_1$ is IR-in-neighbor stable.

## 6.1 Computational complexity of INS

We will now present an algorithm to determine the existence of NS, INS and IR-INS outcomes for games on arbitrary acyclic graphs.

THEOREM 13. *Suppose that we are given oracle access to the preference relations $\succeq_i$ of all players in a hedonic graph game $\mathcal{G} = (N, (\succeq_i)_{i \in N}, L)$, where $(N, L)$ is a forest. Then we can decide whether $\mathcal{G}$ admits a Nash stable, in-neighbor stable or IR-in-neighbor stable feasible outcome (and find one if it exists) in time polynomial in the number of connected subsets of $(N, L)$.*

PROOF. Our algorithm is similar to Demange's algorithm for the core [9]. Again, we assume that $(N, L)$ is a tree. Given a hedonic graph game $(N, (\succeq_i)_{i \in N}, L)$, we make a rooted tree $(N, A^r)$ by orienting edges in $L$. Then, for each player $i \in N$ and each $X \subseteq \mathrm{succ}(i, A^r)$ with $X \in \mathcal{F}_L(i)$, we determine whether there exists a stable partition $\pi$ of $\mathrm{succ}(i, A^r)$ with $X \in \pi$. We set $f(X) = 1$ if such a partition exists, and set $f(X) = 0$ otherwise. To this end, for each $j \in \mathrm{ch}(X, A^r)$ we try to find a coalition $X_j \in \mathcal{F}_L(j)$, $X_j \subseteq \mathrm{succ}(j, A^r)$ such that no player wants to move across the "border" between $X$ and $\mathrm{ch}(X, A^r)$ under the given stability requirement. A stable solution exists if and only if $f(X) = 1$ for some coalition $X \in \mathcal{F}_L(r)$. Algorithm 2 describes in detail how $f(X)$ is computed.

**Algorithm 2** Determining the existence of $\alpha$ feasible partitions, where $\alpha \in \{$NS, INS, IR-INS$\}$

**Input:** tree $(N, L)$, $r \in N$, oracles for $\succeq_i$, $i \in N$.
**Output:** $f : \mathcal{F}_L \to \{0, 1\}$.

```
1: make a rooted tree (N, A^r) with root r by orienting all the
   edges in L.
2: initialize f(X) ← 1 for X ∈ F_L.
3: for t = 0, ..., height(r, A^r) do
4:   for i ∈ N with height(i, A^r) = t do
5:     for X ∈ F_L(i) such that X ⊆ succ(i, A^r) do
6:       if X is not individually rational then
7:         f(X) ← 0
8:       else
9:         for j ∈ ch(X, A^r) do
10:          if for each X_j ∈ F_L(j) such that X_j ⊆
             succ(j, A^r) and f(X_j) = 1 the deviation of j
             from X_j to X or the deviation of pr(j, A^r) from
             X to X_j is α feasible then
11:            f(X) ← 0
12:          end if
13:        end for
14:      end if
15:    end for
16:  end for
17: end for
```

LEMMA 14. *For each $\alpha \in \{$NS, INS, IR-INS$\}$, each $i \in N$ and each $X \in \mathcal{F}_L(i)$ such that $X \subseteq succ(i, A^r)$ we have $f(X) = 1$ if and only if there exists an $\alpha$ feasible partition $\pi$ of $succ(i, A^r)$ such that $X \in \pi$.*

PROOF. The proof is by induction on $\mathrm{height}(i, A^r)$. The claim is immediate when $\mathrm{height}(i, A^r) = 0$. Suppose that it holds for all $j \in N$ with $\mathrm{height}(j, A^r) \le t - 1$, and consider a player $i$ with $\mathrm{height}(i, A^r) = t$. Consider an arbitrary $X \in \mathcal{F}_L(i)$ such that $X \subseteq \mathrm{succ}(i, A^r)$.

Suppose first that $f(X) = 1$. Line 6 ensures that $X$ is individually rational. Hence, if $X = \mathrm{succ}(i, A^r)$, then $X$ is an $\alpha$ feasible partition of $\mathrm{succ}(i, A^r)$. Now, suppose that $X \neq \mathrm{succ}(i, A^r)$, i.e., $\mathrm{ch}(X, A^r) \neq \emptyset$. Since $f(X) = 1$, Line 10 ensures that for each $j \in \mathrm{ch}(X, A^r)$ there exists a coalition $X_j \in \mathcal{F}_L(j)$ such that $X_j \subseteq \mathrm{succ}(j, A^r)$, $f(X_j) = 1$ and neither the deviation of $j$ from $X_j$ to $X$ nor the deviation of $\mathrm{pr}(j, A^r)$ from $X$ to $X_j$ is $\alpha$ feasible. By the induction hypothesis, for each $j \in \mathrm{ch}(X, A^r)$ there exists an $\alpha$ feasible partition of $\mathrm{succ}(j, A^r)$ that contains $X_j$; combining these partitions with $X$, we obtain an $\alpha$ feasible partition of $\mathrm{succ}(i, A^r)$ that contains $X$.

Conversely, if $f(X) = 0$, then $X$ is not individually rational, or the condition of the **if** statement in Line 10 is satisfied. In either case, there is no $\alpha$ feasible partition of $\mathrm{succ}(i, A^r)$ containing $X$. □

Lemma 14 immediately implies that the input game admits an $\alpha$ feasible partition for $\alpha \in \{$NS, INS, IR-INS$\}$ if and only if $f(X) = 1$ for some $X \in \mathcal{F}_L(r)$. If this is the case, an $\alpha$ feasible partition can be found using standard dynamic programming techniques.

It remains to analyze the running time of our algorithm. Let $n = |N|$, $s = |\mathcal{F}_L|$. Algorithm 2 considers each coalition $X \in \mathcal{F}_L$ exactly once. To check that it is individually rational, it makes at most $n$ oracle calls. Further, $X$ has at most $n$ children. For each child $j$, the algorithm considers at most $s$ candidate coalitions $X_j$. To check the conditions in Line 10 for a given $X_j$, we need at most two oracle calls in case of Nash stability and in-neighbor stability and at most $n$ calls in case of IR-in-neighbor stability. We conclude that our algorithm performs at most $O(n^2 s^2)$ oracle calls. □

The following result shows that we should not hope to obtain a polynomial-time algorithm for finding in-neighbor stable outcomes, even for additively separable hedonic graph games on stars.

THEOREM 15. *Given an additively separable hedonic graph game whose underlying graph is a star, it is* NP*-complete to determine whether it has an in-neighbor stable feasible partition.*

PROOF SKETCH. In-neighbor stability can be verified in polynomial time, so our problem is is NP. To prove NP-hardness, we reduce from CLIQUE. Given an undirected graph $G = (V, E)$ and a positive integer $t$, we construct an additively separable hedonic graph game $(N, U, L)$ as follows. We set $N = V \cup \{a, b, c\}$, and use the same graph $(N, L)$ as in the proof of Theorem 12 (see Figure 1).

Let $M = |N| + 1$. We define the utility matrix $U : N \times N \to \mathbb{R}$ as follows.

$$U(a, b) = 1, U(a, c) = -2, U(b, a) = t,$$
$$U(b, c) = 0, U(c, a) = 0, U(c, b) = 2,$$
$$U(a, v) = U(c, v) = U(v, a) = U(v, c) = -M \text{ for each } v \in V,$$
$$U(b, v) = 1, U(v, b) = 0 \text{ for each } v \in V,$$
$$U(u, v) = 0 \text{ if } \{u, v\} \in E$$
and $U(u, v) = -M$ otherwise, for all $u, v \in V$.

As in the proof of Theorem 12, one can verify that $G$ contains a clique of size $t$ if and only if the game admits an in-neighbor stable feasible partition. We omit the details due to space constraints. □

A similar reduction shows that it is hard to find a Nash stable outcome.

THEOREM 16. *Given an additively separable hedonic graph game whose underlying graph is a star, it is* NP*-complete to determine whether it has a Nash stable feasible partition.*

In contrast, for any hedonic game on a star, we can construct an IR-in-neighbor stable partition efficiently.

PROPOSITION 17. *Every hedonic graph game $(N, (\succeq_i)_{i\in N}, L)$ where $(N, L)$ is a star has an IR-in-neighbor stable partition, and given oracle access to the players' preference relations, such a partition can be found using $O(|N|^3)$ oracle calls.*

PROOF. If the central node strictly prefers being on her own, rather than being in any coalition of size two, a partition with all singletons is IR-in-neighbor stable. Otherwise, choose a favorite two-player coalition of the center, and keep adding players to this coalition one by one if this deviation is IR-in-neighbor feasible. When no player can be added, the resulting partition is IR-in-neighbor stable, since the utility of the central node does not decrease during the execution, and there is no player who can IR-in-neighbor deviate to the coalition of the center. The bound on the running time is immediate. □

## 6.2 Computational complexity of INS: Symmetric additively separable games

The proof of Theorem 16 is based on the the construction in the proof of Theorem 15, which uses an additively separable hedonic game that is not symmetric. Indeed, Bogomolnaia and Jackson [6] observe that in symmetric additively separable hedonic games, any NS deviation strictly increases the sum of all players' utilities $\sum_{i\in N}\sum_{j\in\pi(i)} U(i,j)$. This implies that for this class of games a sequence of NS deviations converges to a Nash stable outcome. Thereby, the set of Nash stable outcomes (and hence also INS, IR-INS, and IS outcomes) is always non-empty. However, the number of deviations needed to reach a Nash stable outcome may be exponential in the number of players, so it remains unclear if a Nash stable outcome can be computed efficiently.

The complexity class that appears to be useful for capturing the complexity of this problem is PLS (Polynomial Local Search) [17]. A problem in PLS consists of a finite set of candidate solutions, each of which has associated neighborhood and cost. It is specified by three polynomial-time algorithms. The first algorithm computes an initial candidate solution (e.g. the all-singleton partition). The second algorithm returns the cost of each candidate solution (e.g. the social welfare of a partition). Finally, the third algorithm tests whether a given candidate solution is optimal in its neighborhood, and if not, finds a solution with better cost (e.g. an improved partition after a profitable deviation). Given two PLS problems $A$ and $B$, we say that $A$ is *PLS-reducible* to $B$ if there exist polynomial time computable functions $f$ and $g$ such that $f$ maps instances of $A$ to $B$ and $g$ maps the local optima of $B$ to local optima of $A$.

Gairing and Savani show that search problems related to NS and IS for symmetric additively separable games are PLS-complete [13, 14]. However, if one were to interpret the hedonic game in their reduction as a graph game, the underlying graph would necessarily contain cycles. In what follows, we will show that computing in-neighbor stable outcomes for symmetric additively separable games is PLS-complete even when the graph $(N, L)$ is a star.

THEOREM 18. *Given a symmetric additively separable hedonic graph game whose underlying graph is a star, it is* PLS-*complete to find an in-neighbor stable feasible partition.*

PROOF SKETCH. With symmetric additively separable preferences, we are able to test local optimality of a partition and, if it is not optimal, find an improving move in polynomial time. Hence, our problem is clearly in PLS. To prove PLS-hardness, we provide a reduction from LOCAL MAX-CUT, which is known to be PLS-complete [21]. Recall that an instance of LOCAL MAX-CUT is given by a weighted graph $G = (V, E, w)$, where $w : V \times V \to \mathbb{N}$ is the weight function with the convention that $w(u, v) = 0$ for each $\{u, v\} \notin E$. A *cut* is a partition of $V$ into two parts $S$ and $V \setminus S$; its *weight* is given by $\sum_{u\in S, v\in V\setminus S} w(u, v)$. The neighborhood of a cut $(S, V \setminus S)$ is defined as the set of all cuts that can be obtained by moving one node from $S$ to $V \setminus S$ or vice versa; the goal is to find a cut that has the maximum weight in its neighborhood.

Given an instance $(V, E, w)$ of LOCAL MAX-CUT, we set $N = V \cup \{s\}$, $L = \{ \{s, v\} \mid v \in V \}$ and construct a symmetric additively separable graph game $(N, U, L)$ with the utility matrix $U : N \times N \to \mathbb{R}$ defined as follows. We set $U(u, s) = \sum_{v\in V} w(u, v)$ for each $u \in V$. For every pair of distinct nodes $u, v \in V$, we define $U(u, v) = -2w(u, v)$ if $\{u, v\} \in E$ and $U(u, v) = 0$ otherwise. Note that every $u \in V$ will be in a coalition with $s$ or on her own in any feasible partition. Moreover, player $s$ will accept all in-neighbor feasible deviations since her utility for every other player is non-negative.

Let $\pi$ be an in-neighbor stable feasible partition of $N$. Let $V_1 = \{ u \in V \mid u \in \pi(s) \}$ and $V_2 = V \setminus V_1$. Each player $u \in V_1$ has non-negative utility $U(u, s) + \sum_{v\in V_1} U(u, v)$ by individual rationality. Hence, $\sum_{v\in V_1} w(u, v) \le \sum_{v\in V_2} w(u, v)$ for every $u \in V_1$. On the other hand, no player $u \in V_2$ wants to deviate to $\pi(s)$: by a similar calculation, $\sum_{v\in V_1} w(u, v) \ge \sum_{v\in V_2} w(u, v)$ for every $u \in V_2$. Thus, $(V_1, V_2)$ is a local max-cut of $G$. □

The construction in the proof of Theorem 18 can also be used to show PLS-completeness of finding a Nash stable partition in this class of games.

THEOREM 19. *Given a symmetric additively separable hedonic graph game whose underlying graph is a star, it is* PLS-*complete to find a Nash stable feasible partition.*

However, we cannot extend Theorem 19 to enemy-oriented games.

PROPOSITION 20. *A Nash stable feasible outcome of a symmetric enemy-oriented game on a star can be computed in polynomial time.*

PROOF. We initialize coalition $X$ to the center of the star; as long as there is a player that likes (and is liked by) all current members of $X$, we add him to $X$. Eventually, no player can be added to $X$. At this point, $\{X\} \cup \{\{i\} \mid i \in N \setminus X\}$ is a Nash stable feasible partition: no player outside of $X$ wants to deviate to $X$, and no player in $X$ wants to leave. □

# 7. CONCLUSIONS AND FUTURE WORK

We have explored the existence and computational complexity of stable partitions in hedonic games on acyclic graphs. We obtained a number of algorithmic results in the general oracle-based framework, thereby showing that acyclicity of the communication network has important implications for stability. It remains unknown whether a strictly core stable partition for a hedonic game on a tree can be computed in time polynomial in the number of connected coalitions; we leave this problem as a direction for future work. It would also be interesting to see if our algorithms can be extended to graphs that are "almost" acyclic, and, more broadly, if there are constraints on the communication structure other than acyclicity that lead to existence/tractability results for common hedonic games stability concepts.


## Acknowledgements

We would like to thank the reviewers for their very useful comments. The first author acknowledges financial support from Funai Foundation for Information Technology.

# REFERENCES


[1] H. Aziz, F. Brandt, and H. G. Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316–334, feb 2013.

[2] H. Aziz and R. Savani. Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*. Cambridge University Press, 2015.

[3] S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

[4] V. Bilò, A. Fanelli, M. Flammini, G. Monaco, and L. Moscardelli. Nash stability in fractional hedonic games. In *WINE'14*, pages 486–491, 2014.

[5] V. Bilò, A. Fanelli, M. Flammini, G. Monaco, and L. Moscardelli. On the price of stability of fractional hedonic games. In *AAMAS'15*, pages 1239–1247, 2015.

[6] A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[7] N. Burani and W. Zwicker. Coalition formation games with separable preferences. *Mathematical Social Sciences*, 45(1):27–52, 2003.

[8] G. Demange. Intermediate preferences and stable coalition structures. *Journal of Mathematical Economics*, 23(1):45–58, 1994.

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

[10] D. Dimitrov, P. Borm, R. Hendrickx, and S. C. Sung. Simple priorities and core stability in hedonic games. *Social Choice and Welfare*, 26(2):421–433, 2006.

[11] J. H. Drèze and J. Greenberg. Hedonic Coalitions: Optimality and Stability. *Econometrica*, 48(4):987–1003, 1980.

[12] E. Elkind. Coalitional games on sparse social networks. In T. Liu, Q. Qi, and Y. Ye, editors, *WINE'14*, volume 8877 of *Lecture Notes in Computer Science*, pages 308–321. Springer, 2014.

[13] M. Gairing and R. Savani. Computing stable outcomes in hedonic games. In S. Kontogiannis, E. Koutsoupias, and P. Spirakis, editors, *SAGT'10*, volume 6386 of *Lecture Notes in Computer Science*, pages 174–185. Springer Berlin Heidelberg, 2010.

[14] M. Gairing and R. Savani. Computing stable outcomes in hedonic games with voting-based deviations. In *AAMAS'11*, pages 559–566, 2011.

[15] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[16] A. Igarashi and E. Elkind. Hedonic games with graph-restricted communication. arxiv:1602.05342 [cs.GT], 2016.

[17] D. S. Johnson and C. H. Papadimitriou. How easy is local search? *Journal of Computer and System Sciences*, 37:79–100, 1988.

[18] M. Le Breton, G. Owen, and S. Weber. Strongly balanced cooperative games. *International Journal of Game Theory*, 20(4):419–427, 1992.

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

[20] D. Peters and E. Elkind. Simple causes of complexity in hedonic games. In *IJCAI'15*, pages 617–623, 2015.

[21] A. A. Schäffer and M. Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.

[22] S.-C. Sung and D. Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, jun 2010.