Graphical Hedonic Games of Bounded Treewidth

Dominik Peters
Department of Computer Science
University of Oxford, UK
dominik.peters@cs.ox.ac.uk

Abstract

Hedonic games are a well-studied model of coalition formation, in which selfish agents are partitioned into disjoint sets and agents care about the make-up of the coalition they end up in. The computational problems of finding stable, optimal, or fair outcomes tend to be computationally intractable in even severely restricted instances of hedonic games. We introduce the notion of a graphical hedonic game and show that, in contrast, on classes of graphical hedonic games whose underlying graphs are of bounded treewidth and degree, such problems become easy. In particular, problems that can be specified through quantification over agents, coalitions, and (connected) partitions can be decided in linear time. The proof is by reduction to monadic second order logic. We also provide faster algorithms in special cases, and show that the extra condition of the degree bound cannot be dropped. Finally, we note that the problem of allocating indivisible goods can be modelled as a hedonic game, so that our results imply tractability of finding fair and efficient allocations on appropriately restricted instances.

1 Introduction

Hedonic games, first studied by Banerjee, Konishi, and Sönmez (2001) and Bogomolnaia and Jackson (2002), provide a general framework for the study of *coalition formation*. Hedonic games subsume the well-studied *matching problems* (stable marriage, stable roommates, hospital-residents), but are able to express more general preference structures. They have been applied to problems in public good provision, voting, and clustering, and, as we show below, they also encapsulate a variety of allocation problems.

A *hedonic game* consists of a set N of agents, each of whom has a preference ordering over all coalitions $S \subseteq N$ containing her. The outcome of such a game is a *partition* of the agent set into disjoint coalitions, with agents preferring those partitions in which they are in a preferred coalition.

Unfortunately, it has turned out that many key questions about hedonic games are computationally hard to answer. For example, it is typically NP-complete to decide whether a hedonic game admits a *Nash stable* outcome; it is typically NP-hard to maximise social welfare; and it is often even Σ_2^p-complete to identify hedonic games with non-empty *core*. See Ballester (2004), Sung and Dimitrov (2010), Woeginger (2013), and Peters and Elkind (2015) for a selection of such results.

A standard criticism of hardness results such as these is that they apply only in the *worst case*. Instances arising in practice can be expected to show much more structure than the highly contrived instances produced in 3SAT-reductions. In non-cooperative game theory, *graphical games* (Kearns, Littman, and Singh 2001) are an influential way to allow formalisation of the notion of a 'structured' game. In a graphical game, agents form the vertices of an undirected graph, and each agent's utility function only depends on the actions taken by her neighbours. The underlying graph can guide algorithms in finding a stable outcome, and imposing restrictions on the graph topology can yield tractability (Gottlob, Greco, and Scarcello 2005).

A particularly successful restriction on graph topology is *bounded treewidth*. The treewidth of a graph (Robertson and Seymour 1986) measures how 'tree-like' a given graph is. Many NP-hard problems become fixed-parameter tractable with respect to the treewidth of some graph naturally associated with the problem instances. Indeed, dynamic programming over a given *tree decomposition* often yields algorithms that are exponential in the treewidth, but *linear* in the problem size. See Bodlaender's (1994) 'tourist guide through treewidth' for an introduction to this area.

The treewidth approach was first applied to the domain of hedonic games by Elkind and Wooldridge (2009), who propose a representation formalism for hedonic games called *hedonic coalition nets* that expresses agents' preferences by weighted boolean formulas. They also introduce a notion of treewidth for hedonic games—very close to ours—and show that (when numbers in the input are polynomially bounded) it is fixed-parameter tractable with respect to treewidth to decide whether a given partition is core-stable.

In this paper, we study *graphical hedonic games*, which we define in analogy to graphical games. Here, agents are again arranged in an underlying graph, and need to be partitioned into coalitions. Every agent only cares about which of her neighbours are in the same coalition as her. Every hedonic game can be made graphical by introducing edges whenever one agent's utility depends on the other's presence. We will then consider graphical hedonic games whose agent graphs have bounded treewidth and bounded degrees.

In the context of hedonic games, restricting treewidth and also degrees in the underlying social network seems particularly natural. Consider for example *Dunbar's number* (Dunbar 1992), a suggested limit on the number of individuals that a single human being can maintain stable social relationships with. This number has been suggested to lie between 100 and 250, which gives us a natural bound on the degree of any social network. Intuitively, it also seems sensible to suppose that social networks have relatively small treewidth, though see Adcock, Sullivan, and Mahoney (2013) who find mixed empirical support for this proposition.

We show that when restricted to a class of graphical hedonic games whose agent graphs have bounded treewidth and bounded degrees, many standard problems related to these hedonic games become linear-time solvable. More precisely, by a somewhat involved translation to monadic second-order logic and by appealing to Courcelle's theorem, it follows that we can decide in linear time whether a given such hedonic game satisfies any logical sentence of what we call *HG-logic*, which allows quantification over partitions, coalitions, and agents. Using this approach and on this restricted domain, we can efficiently find stable or fair outcomes of a hedonic game for all notions of stability that are commonly discussed in the literature.

We also show that HG-logic is expressive enough to capture problems that would at first appear to lie outside the domain of hedonic games, such as the problem of fair and efficient allocation of indivisible goods. This implies that questions regarding those problems can also be answered efficiently when we restrict treewidth and degree.

Our appeal to Courcelle's meta-algorithmic result, while powerful, comes at the cost of the hidden 'constant factor' growing dramatically as the treewidth and degree of the hedonic game increase (indeed, this growth cannot be bounded by an elementary function unless P = NP). To show that despite this the restriction to bounded treewidth is useful in practice, we present a variety of more specific problems that can be solved in the more manageable runtime $\tilde{O}(2^{kd^2}n)$, where k is the bound on treewidth, d is the bound on the degrees, and n is the number of agents. Such results suggest that finding good outcomes of hedonic games should be tractable on instances arising in practice.

Many standard NP-complete problems defined on graphs become easy when bounding treewidth, without requiring a further restriction on the degrees of the graph. In the last section we show that this is not the case for graphical hedonic games: we give reductions showing that finding a core-stable partition and similar problems remain NP-hard even on games of treewidth 1 or 2 (but when degrees are unbounded).

2 Preliminaries

A *hedonic game* $\langle N, (\succcurlyeq_i)_{i\in N}\rangle$ is given by a finite set N of agents, and for each agent $i \in N$ a complete and transitive preference relation over $\mathcal{N}_i = \{S \subseteq N : i \in S\}$. We let $\succ_i$ and $\sim_i$ denote the strict and indifference parts of $\succcurlyeq_i$. The outcome of a hedonic game is a *partition* π of the agent set into disjoint coalitions. We write $\pi(i)$ for the coalition $S \in \pi$ that contains $i \in N$. We are interested in finding partitions that are *stable*, *optimal*, and/or *fair*. There are multiple ways of formalising these goals. For example, a partition π is *Nash stable* if no agent wants to join another (possibly empty) coalition of π, that is $\pi(i) \succcurlyeq_i S \cup \{i\}$ for all $S \in \pi \cup \{\varnothing\}$ and all $i \in N$. It is *individually stable* if there is no agent i and coalition $S \in \pi \cup \{\varnothing\}$ such that $S \cup \{i\} \succ_i \pi(i)$ and $S \cup \{i\} \succcurlyeq_j S$ for all $j \in S$. We say π is *core-stable* if there is no non-empty *blocking* coalition $S \subseteq N$ such that $S \succ_i \pi(i)$ for each $i \in S$. We say π is *strict*-core-stable if there is no non-empty coalition $S \subseteq N$ such that $S \succcurlyeq_i \pi(i)$ for each $i \in S$, with at least one preference strict. For fairness, we say that π is *envy-free* if no agent prefers taking another agent's place: $\pi(i) \succcurlyeq_i \pi(j) \setminus \{j\} \cup \{i\}$ for all $i, j \in N$ with $\pi(i) \neq \pi(j)$. We will only consider the concept of envy-freeness for hedonic games where every coalition is acceptably to every player, i.e., if $S \succcurlyeq_i \{i\}$ for all $S \in \mathcal{N}_i$, so that a player never envies a player being alone.

Note that the preference relations $\succcurlyeq_i$ have an exponentially sized domain of 2^{n-1} coalitions. For computational purposes, we need to use a language that represents such preferences *succinctly*, so that the representation preferably uses only $\text{poly}(n)$ symbols, where n is the number of agents. An attractive such representation is given by *additively separable hedonic games*, in which each agent specifies a *valuation function* $v_i : N \to \mathbb{R}$ assigning each agent a numeric value. We then say that $S \succcurlyeq_i T$ if and only if $\sum_{j\in S} v_i(j) \geqslant \sum_{j\in T} v_i(j)$. An additively separable game is thus given by n^2 numbers.

A more expressive representation is proposed by Elkind and Wooldridge (2009). They define *hedonic coalition nets* (or *HC-nets*) in which each agent specifies a set of weighted propositional formulas, called *rules*, with propositional atoms given by the agents. For example, the rule $i_2 \wedge i_3 \wedge \neg i_4 \mapsto_{i_1} 5$ means that agent i_1 derives utility 5 when i_1 is together with i_2 and i_3 but not together with i_4. If an agent specifies multiple rules, the agent obtains the sum of the weights of those formulas that are satisfied in the given coalition. By taking rules of form $j \mapsto_i v_i(j)$, we see that HC-nets can encode additively separable games. Elkind and Wooldridge (2009) show that many other standard representations can also be encoded in HC-nets.

In hedonic games with cardinal (numeric) utilities (such as those given by HC-nets), we can define the *social welfare* of a given partition π. Its *utilitarian* welfare is $\sum_{i\in N} u_i(\pi(i))$, its *egalitarian* welfare is $\min_{i\in N} u_i(\pi(i))$, and its *Nash product* is $\prod_{i\in N} u_i(\pi(i))$. A partition that maximises a chosen welfare notion is seen as *optimal*. Another optimality notion is that of a *perfect* (sometimes called *wonderfully stable*) partition in which every agent belongs to a most preferred coalition. Finally, a partition π is *Pareto optimal* if for no partition π' we have $\pi'(i) \succcurlyeq_i \pi(i)$ for all $i \in N$, with at least one preference strict.

Let us now define the treewidth of a graph. Given a graph G, we write $V(G)$ for its set of vertices, and $E(G)$ for its set of edges. A *tree decomposition* of an undirected graph G is given by a tree T and a map $\beta : V(T) \to 2^{V(G)}$ which assigns to each vertex $w \in V(T)$ of the tree T a *bag* $\beta(w) \subseteq V(G)$ of vertices of G, satisfying the following two conditions:

1. For each $v \in V(G)$, the set $\beta^{-1}(v)$ of bags containing v is non-empty and connected (in T).
2. For each edge $\{u, v\} \in E(G)$ in G, there is a bag containing both u and v, i.e., there is $w \in V(T)$ with $\{u, v\} \subseteq \beta(w)$.

The *width* of a tree decomposition is $\max_{w \in V(T)} |\beta(w)| - 1$, that is, one less than the maximum size of the bags. Then, the *treewidth* of G is the minimum width of a tree decomposition of G. Bodlaender (1994) gives more intuition and examples.

3 Graphical Hedonic Games

We now introduce the main notion of this paper:

Definition. *A* ***graphical hedonic game*** *is a pair of a hedonic game $\langle N, (\succcurlyeq_i)_{i \in N} \rangle$ and an undirected graph $G = (N, E)$ that jointly satisfy the following condition: for each agent $i \in N$ and all coalitions $S, T \in \mathcal{N}_i$, we have*

$$S \succcurlyeq_i T \text{ if and only if } S \cap \Gamma(i) \succcurlyeq_i T \cap \Gamma(i),$$

where $\Gamma(i) \subseteq N$ is the set of neighbours of i in G.

We will call any graph $G' = (N, E)$ satisfying this condition an (agent) dependency graph *for the hedonic game $\langle N, (\succcurlyeq_i)_{i \in N} \rangle$.*

Thus, in a graphical hedonic game, how much an agent i likes a coalition depends only on the presence of the neighbours of i in the dependency graph.

Examples. Pairing any hedonic game with the complete graph over the agent set yields a graphical hedonic game. The complete graph over the agent set is the unique agent dependency graph for *anonymous* hedonic games, where agents only care about the cardinality of their coalition. The same is true for *fractional hedonic games* (Aziz, Brandt, and Harrenstein 2014).

In additively separable hedonic games, the graph with

$$\{i, j\} \in E \iff v_i(j) \neq 0 \text{ or } v_j(i) \neq 0$$

is the edge-minimal dependency graph.

For a hedonic game given by an HC-net, we could take

$$\{i, j\} \in E \iff i \text{ appears in a rule of } j \text{ or vice versa},$$

though this might not be edge-minimal. □

We are usually interested in dependency graphs with as few edges as possible, so that the vertex-neighbourhoods and the treewidth of the graph are small.

Lemma 1. *For each hedonic game, there exists a unique edge-minimal agent dependency graph.*

Proof. We show that if edge sets $E_1, E_2 \subseteq N^{(2)}$ induce a dependency graph for a hedonic game $\langle N, (\succcurlyeq_i)_{i \in N} \rangle$, then so does the edge set $E_1 \cap E_2$. Let $i \in N$ be an agent, and let $S, T \ni i$ be coalitions. Then using the definition of a dependency graph once for E_1 and once for E_2, we see that

$$\begin{aligned} S \succcurlyeq_i T &\iff S \cap \Gamma_1(i) \succcurlyeq_i T \cap \Gamma_1(i) \\ &\iff S \cap \Gamma_1(i) \cap \Gamma_2(i) \succcurlyeq_i T \cap \Gamma_1(i) \cap \Gamma_2(i). \end{aligned}$$

The neighbourhoods $\Gamma_1(i) \cap \Gamma_2(i)$ are induced by $E_1 \cap E_2$. Hence $E_1 \cap E_2$ is the edge set of a dependency graph. □

It is often easy to find the minimal dependency graph, as in the case of additively separable hedonic games. On the other hand, finding the *minimal* graph of an HC-net is coNP-hard, since it is hard to decide whether some variable is redundant in a boolean formula.

Definition. *Let $(\langle N, (\succcurlyeq_i)_{i \in N} \rangle, G)$ be a graphical hedonic game. Its* treewidth *is the treewidth of G, and its* degree *is the maximum degree of G.*

We will show that many computational problems concerning hedonic games become easy when restricting attention to graphical hedonic games of small treewidth and small degree.

Example. The class of hedonic games in which agents can be placed in a cycle and only care about the presence of their immediate neighbours is of bounded treewidth and degree.

A crucial observation about graphical hedonic games is that, as far as the agents' preferences are concerned, we can often restrict our attention to *connected* coalitions.

Definition. *In a graphical hedonic game with dependency graph G, a coalition $S \subseteq N$ is* connected *if $G[S]$ is connected, that is if S induces a connected subgraph in G. A partition π of N is* connected *if each $S \in \pi$ is connected.*

Notice that given a non-connected partition π', we can split each coalition $S \in \pi'$ into its connected components, obtaining a connected partition π. Then, from the definition of dependency graphs, every agent is indifferent between π and π': so $\pi(i) \sim_i \pi'(i)$ for all $i \in N$.

Representation. In our computational study of graphical hedonic games, we will require games to be represented in some reasonably concise fashion. As we mostly deal with classes of graphical hedonic games that have bounded degree d, where we regard d to be small, it can be sensible to explicitly list every agent's preferences over all subsets of her neighbourhood, taking $O(2^{2d} \cdot n)$ space. If the preference relations $\succcurlyeq_i$ can be evaluated in time only depending on d, but not on n, this will be just as well for our FPT results. In other cases, we assume that we have oracle access to the $\succcurlyeq_i$.

4 A Logic for Hedonic Games

In this section, we define a logic that captures standard properties of hedonic games, for example the existence of stable partitions. The logic uses variables $i, j, k, \ldots$ ranging over agents, variables $S, T, \ldots$ ranging over coalitions, and variables $\pi, \pi', \ldots$ ranging over partitions of the agents.

Definition. *The formulas of* ***hedonic game logic (HG-logic)*** *are defined recursively as*

1. *atomic formulas: $i = j$, $i \in S$, $S = \pi(i)$, $S \succcurlyeq_i T$.*
2. *boolean combinations of formulas: $\neg\phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$.*
3. *quantification over agents: $\forall i\ \phi$, $\exists i\ \phi$.*
4. *quantification over coalitions: $\forall S\ \phi$, $\exists S\ \phi$.*
5. *quantification over partitions: $\forall \pi\ \phi$, $\exists \pi\ \phi$.*

We will use standard abbreviations in writing formulas of HG-logic. For example,

- $S \succcurlyeq_i \pi(i)$ means $\exists T\ (T = \pi(i) \wedge S \succcurlyeq_i T)$,

- $S \subseteq T$ means $\forall i\ (i \in S \rightarrow i \in T)$,
- $\exists i \in S\ \phi$ means $\exists i\ (i \in S \wedge \phi)$,
- $S \succ_i T$ means $(S \succcurlyeq_i T \wedge \neg T \succcurlyeq_i S)$.

A *sentence* of HG-logic is a formula in which no variable occurs free. In general, every hedonic game can form a model of a sentence in HG-logic in the natural way. In our approach, however, the models of HG-logic are *graphical* hedonic games. A given sentence ϕ of HG-logic is *true* in a given graphical hedonic game $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G)$ if it is true when the formula is evaluated according to the obvious semantics using the universe N and relations $\succcurlyeq_i$ as specified by the hedonic game model, but where quantifications over partitions range only over *connected* partitions (according to the dependency graph G). Thus, according to our graphical semantics, the sentence $\forall i\ \forall j\ \exists \pi\ \pi(i) = \pi(j)$ is not valid, since i and j might not be together in any connected partition.

Let us give some examples of properties of hedonic games and partitions expressible in HG-logic.

- *a core-stable partition exists*: $\exists \pi\ \forall S\ \exists i \in S\ \pi(i) \succcurlyeq_i S$
- *π is Pareto-optimal*: $\neg \exists \pi'\ (\forall i\ \pi'(i) \succcurlyeq_i \pi(i) \wedge \exists j\ \pi'(j) \succ_j \pi(j))$
- *π is Nash stable:* $\forall i\ (\pi(i) \succcurlyeq_i \{i\} \wedge \forall S\ (\exists j\ S = \pi(j) \rightarrow \pi(i) \succcurlyeq_i S \cup \{i\}))$.
- *π is perfect*: $\forall i\ \forall S \ni i\ S \not\succ_i \pi(i)$.
- *π is envy-free:* $\forall i\ \forall j\ \pi(i) \succcurlyeq_i \pi(j) \setminus \{j\} \cup \{i\}$.
- *π' is reachable from π by actions of S:* $\forall i\ \forall j\ (i \notin S \wedge j \notin S \rightarrow (\pi(i) = \pi(j) \leftrightarrow \pi'(i) = \pi'(j)))$.

5 Main Result

An important computational problem we wish to solve is the model-checking problem of HG-logic.

ϕ-HEDONIC GAMES
Instance: a graphical hedonic game $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G)$ and a formula ϕ of HG-logic
Question: does $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G) \models \phi$, i.e. is the graphical hedonic game a model of the formula ϕ?

The perhaps most important special case of this problem is deciding the existence of stable partitions in a hedonic game.

Theorem 2. *The problem ϕ-HEDONIC GAMES is fixed-parameter tractable with respect to the length $|\phi|$ of the formula ϕ, and the treewidth k and degree d of the graph G. That is, the problem can be solved in time $O(f(|\phi|, k, d) \cdot n)$ where f is a computable function, and n is the number of agents. Here we assume that the relation "$S \succcurlyeq_i T$" can be decided in time only depending on d, but not on n.*

This means that for any formula ϕ and any class $\mathcal{C}$ of graphical hedonic games of bounded treewidth and bounded degree, we can decide in linear time whether ϕ is true in a given game $(\langle N, (\succcurlyeq_i)_{i\in N}\rangle, G) \in \mathcal{C}$. In case computing "$S \succcurlyeq_i T$" takes time depending on n, we will need $2^{2d} \cdot n$ calls to an oracle deciding this relation during a pre-processing step, after which the linear-time bound applies again.

Let us make explicit some special cases of Theorem 2.

Corollary 3. *For every class of graphical hedonic games of bounded treewidth and degree, there exist linear-time algorithms that can decide whether a given such game admits a partition that is (i) core-stable, (ii) strict-core-stable, (iii) Pareto-optimal, (iv) perfect (v) Nash-stable, (vi) individually stable, (vii) envy-free, or that satisfies any combination of these properties.*

Proof. This would follow immediately from Theorem 2 and the formulas in Section 4, except that we need to check that the fact that our semantics only quantify over connected partitions makes no difference. For (i)-(iv), this is immediate, since a partition π satisfies the relevant criterion if and only if the connected partition π' obtained from π by splitting its coalitions into their connected components satisfies the same property.

For (v)-(vii), we can achieve a similar behaviour by adding extra edges to the dependency graph G of the input game to obtain a new dependency graph G'. Precisely, whenever there are edges $\{u, v\} \in E(G)$ and $\{v, w\} \in E(G)$, then we add the edge $\{u, w\}$ to G' (note that this is different from taking a transitive closure since we do not apply this step repeatedly). It can then be seen that splitting a Nash- or individually stable partition into its G'-components preserves stability, and similarly for envy-freeness under the additional assumption that all coalitions are individually rational. Further, note that G' has treewidth at most kd and degree at most d^2, so that both parameters are still bounded. □

By calling such an algorithm repeatedly, we can adaptively build up a connected partition satisfying any of these properties (if it exists).

Theorem 2 is proved by reducing ϕ-HEDONIC GAMES to the model checking problem of monadic second-order logic, which by Courcelle's theorem is fixed-parameter tractable with parameter the treewidth of the underlying logical structure and the length of the input formula. We will explain this reduction in the following two sections.

6 MSO and Courcelle's Theorem

Here we give a statement of Courcelle's theorem (1990) and introduce monadic second-order logic (MSO).

First, we need some definitions. A *signature* σ is a finite collection of relation symbols $(R_1, \ldots, R_k)$, with each symbol $R_i \in \sigma$ being endowed with an *arity* $\mathrm{ar}(R_i) \geqslant 1$. A *$\sigma$-structure* $\mathcal{A} := \langle A, (R_1^{\mathcal{A}}, \ldots, R_k^{\mathcal{A}})\rangle$ is given by a finite set A, the *universe* of $\mathcal{A}$, as well as a realisation $R_i^{\mathcal{A}} \subseteq A^{\mathrm{ar}(R_i)}$ for each relation symbol R_i. The *size* of $\mathcal{A}$ is given by $\|\mathcal{A}\| = |\sigma| + |A| + \sum_{R_i \in \sigma} |R_i^{\mathcal{A}}| \cdot \mathrm{ar}(R_i)$.

Given a signature σ, the *language* $\mathrm{MSO}[\sigma]$ *of monadic second order logic* is given by the grammar

$$\phi ::= x = y \mid R_i x_1 \ldots x_{\mathrm{ar}(R_i)} \mid Xx \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid \neg\phi \mid \exists x\, \phi \mid \forall x\, \phi \mid \exists X\, \phi \mid \forall X\, \phi,$$

where $x, y, x_1, x_2, \ldots$ are first-order variables, and X denotes set variables. Notice that MSO allows quantification only over *unary* relations, i.e. over subsets of the universe A. For a formula ϕ of $\mathrm{MSO}[\sigma]$ and a σ-structure $\mathcal{A}$, we define the semantic notion of $\mathcal{A} \models \phi$ in the obvious way.

Next, let us define the notion of treewidth for a σ-structure. A *tree decomposition* of a σ-structure $\mathcal{A}$ is given by a tree T, each vertex v of T being associated with a subset $\beta(v) \subseteq A$ of the universe, called a *bag*, satisfying the following two conditions: (1) each $a \in A$ is contained in some bag, and the set $\beta^{-1}(a)$ of bags containing a forms a connected subtree of T, and (2) for each $R_i \in \sigma$ and all $a_1, \dots, a_{\mathrm{ar}(R_i)} \in A$ such that $(a_1, \dots, a_{\mathrm{ar}(R_i)}) \in R_i^{\mathcal{A}}$, we have that $\{a_1, \dots, a_{\mathrm{ar}(R_i)}\} \subseteq \beta(v)$ for some vertex v of T. The *width* of such a tree decomposition is the maximum cardinality of the bags minus 1, and the *treewidth* $\mathrm{tw}(\mathcal{A})$ of $\mathcal{A}$ is the minimum width among tree decompositions of $\mathcal{A}$. The (usual) treewidth of a graph is a special case of this definition: just take σ to consist of a single binary relation, namely the adjacency relation of the graph.

We are now ready to state Courcelle's (1990) theorem.

Theorem 4 (Courcelle). *Given a formula ϕ of $\mathrm{MSO}[\sigma]$ and a σ-structure $\mathcal{A}$, we can in time $g(|\phi|, \mathrm{tw}(\mathcal{A})) \cdot |A| + O(\|\mathcal{A}\|)$ decide whether $\mathcal{A} \models \phi$, where g is a computable function.*

Courcelle's theorem is often stated for fixed formulas ϕ, but the more general statement as we are stating it here is true: the model checking problem for MSO is fixed-parameter tractable with respect to the joint parameter consisting of the formula length $|\phi|$ and the treewidth of $\mathcal{A}$. We will use this stronger result in what follows.

7 Encoding of HG-logic into MSO

In this section we prove Theorem 2. Suppose we are given a formula ϕ of HG-logic and a graphical hedonic game $(\langle N, (\succcurlyeq_i)_{i \in N}\rangle, G)$, where G has treewidth k and max-degree d. In the following, we will generate a relatively large (in d, not n) σ-structure containing all information about the game, and then rewrite the formula ϕ into a formula of MSO.

Step 1: $\boldsymbol{\sigma}$-structure. Our signature σ will have four relation symbols: unary symbols VERT and EDGE, a binary symbol INCI, and a $(2d+1)$-ary symbol PREF.

We build a σ-structure $\mathcal{G}$ with universe $N \cup E \cup \{*\}$, and relations VERT $= N$, EDGE $= E$, the vertex-edge incidence relation INCI $= \{(i, e) : i \in N, e \in E, i \in e\}$, and for each $i \in N$, let $(i, i_1, \dots, i_d, i_{d+1}, \dots, i_{2d}) \in$ PREF if and only if $i_s \in N \cup \{*\}$ for all $1 \leqslant s \leqslant 2d$, and each $i_s \in N$ is a neighbour of i in G, and

$$\{i_1, \dots, i_d\} \setminus \{*\} \succcurlyeq_i \{i_{d+1}, \dots, i_{2d}\} \setminus \{*\}.$$

That is, we use the defining property of dependency graphs to encode every agent's essential preferences in the relation PREF. The structure $\mathcal{G}$ can be computed in $O(2^{2d} \cdot n)$ calls to an oracle deciding $\succcurlyeq_i$. For convenience, let us define within $\mathrm{MSO}[\sigma]$ the adjacency relation between vertices as

$$\mathrm{adj}(u, v) \;\equiv\; u \neq v \wedge \exists e\, (\text{INCI}\, ue \wedge \text{INCI}\, ve).$$

Step 2: bounding treewidth. By assumption, the agent dependency graph G has treewidth at most k. We show that the σ-structure $\mathcal{G}$ constructed above has treewidth at most $k \cdot (d+1) + 1$, which is still bounded for bounded d and k. First find in time linear-in-n a tree-decomposition of G of width at most k using Bodlaender's (1993) algorithm. For each of the $O(n)$ edges $e = \{u, v\}$ of G, find a bag $\beta(w)$ that contains both u and v, and introduce a new bag $\{e, u, v\}$ that gets attached as a leaf to w in the tree underlying the tree decomposition, not increasing its width. For each vertex v and every bag $\beta(w)$ of the original tree decomposition containing v, add to $\beta(w)$ the set $\Gamma_G(v)$ of the at most d neighbours of v in G. This operation increases the width by at most $k \cdot d$. Finally, add $*$ to every bag, increasing the width by 1, for a total width of at most $k + kd + 1$. As is easy to see, this is a tree decomposition of $\mathcal{G}$.

Step 3: encoding partitions. In our encoding, a partition π of the agent set will be a 'transitive' subset $E' \subseteq E$ of the edge set of the dependency graph. (The set E' represents the equivalence relation associated with the partition π, in the sense that the two endpoints of an edge $e \in E'$ are in the same coalition of π.) Formally, a set variable X represents a partition if $X \subseteq E$ and whenever $e_1 = \{x, y\}, e_2 = \{y, z\}, e_3 = \{x, z\}$ are edges in E with $e_1, e_2 \in X$, then also $e_3 \in X$. This condition can clearly be expressed in $\mathrm{MSO}[\sigma]$. We can also express the relation "two agents have an edge between them, and this edge is part of X" by a formula. It is well-known that MSO can express the transitive closure of every binary relation it can express (Courcelle and Engelfriet 2012, p. 42). Hence we can express the transitive closure of the preceding relation, which is "two agents are connected by a path of edges that are in X", which is equivalent in our understanding to "two agents are in the same coalition in partition X". This we can use to express "$S = \pi(i)$".

Step 4: encoding preference. We now encode the relation $S \succcurlyeq_i T$. This depends crucially on the definition of the agent dependency graph, so that we actually encode the equivalent relation $S \cap \Gamma(i) \succcurlyeq_i T \cap \Gamma(i)$. To do this, we use d variables for S and d variables for T which will represent the agents from $\Gamma(i)$ that are present in S and T respectively. If there are fewer than d such agents, we assign $*$ as a placeholder. Note that the relation $x \in N \cup \{*\}$ is expressible in $\mathrm{MSO}[\sigma]$ as $\neg$EDGE x. With this, we can express $S \succcurlyeq_i T$ as

$$\begin{aligned}
&\exists x_1, \dots, x_d, y_1, \dots, y_d \in N \cup \{*\} \\
&\qquad \forall x \in S\, (\mathrm{adj}(i, x) \to x = x_1 \vee \dots \vee x = x_d) \\
&\quad \wedge \forall y \in T\, (\mathrm{adj}(i, y) \to y = y_1 \vee \dots \vee y = y_d) \\
&\quad \wedge \text{PREF}\, i\, x_1 \dots x_d\, y_1 \dots y_d.
\end{aligned}$$

Step 5: encoding HG-syntax. Using steps 3 and 4, we can translate ϕ (a formula of HG-logic) into a formula ϕ' of $\mathrm{MSO}[\sigma]$. Here, we replace quantifications over partitions by quantifications over edge sets, as indicated in step 3.

This finishes our translation of HG-logic into $\mathrm{MSO}[\sigma]$. Using Courcelle's algorithm, we can now check whether $\mathcal{G} \models \phi'$. We achieve the claimed time bound by noting that any blow-ups in formula size and treewidth are still bounded whenever k and d are bounded.

8 Faster Algorithms in Special Cases

In the algorithms arising through our use of Courcelle's theorem, the dependence on k and d in their runtime is quite bad; indeed, we cannot bound the function $f(|\phi|, k, d)$ by an elementary function unless P = NP (Frick and Grohe 2004). This phenomenon is especially bad in our case as we are using multiple quantifier alternations in our MSO-encoding. Clearly, we cannot just ignore this as a merely 'constant factor'. In this sense, Theorem 2 should be seen as an *existence result*, but not as providing an actually usable algorithm.

In this section, we use an alternative approach due to Bodlaender (1988) that produces algorithms with more manageable dependence on k and d for some important stability and optimality problems. Bodlaender (1988) defines the very general class of *local condition composition problems* (short LCC or 1-LCC) and shows that they are linear-time solvable on classes of graphs of bounded treewidth and degree. In the following, we will give a definition of an LCC-problem, suitably specialised for our purposes.

Let $G = (V, E)$ be a graph with max-degree d for which a tree decomposition of width k is given. Let $E_2(v)$ be the set of edges from E whose endpoints are both within distance 2 of v. For maps $f : E \to \{0, 1\}$, let $P(v, f|_{E_2(v)})$ be a 0/1-property, and let $W(v, f|_{E_2(v)})$ be an integer-valued function, both computable in time polynomial in their input length. Finally, let $\oplus$ denote the binary operation of a totally ordered commutative monoid over the integers. Examples include taking $\oplus$ to be the sum, product, or minimum of the values. Now consider the following computational problem:

Instance: Graph $G = (V, E)$, additional data about G, target value K.
Question: Does there exist a map $f : E \to \{0, 1\}$ such that for each $v \in V$ the property $P(v, f|_{E_2(v)})$ is true, and $\bigoplus_{v \in V} W(v, f|_{E_2(v)}) \geqslant K$?

Bodlaender (1988) shows that any problem of this form can be solved in $\tilde{O}(2^{kd^2} n)$ time. Here, the soft-$\tilde{O}$ hides factors polynomial in k and d which will depend on the runtime of evaluating P and W.

Using this apparatus, we can encode hedonic games problems in a similar fashion as before. Again, a connected partition π of N will be represented by a set $E' \subseteq E$ of edges, where $E' = f^{-1}(1) = \{e \in E : f(e) = 1\}$. Using the property P, we can enforce transitivity of E'. We can also calculate in P the utility of a given agent in the partition described by E' (since $f|_{E_2(v)}$ tells us which relevant players are in the same coalition as v), and we know who is in the coalition of every agent w adjacent to v. For example, we can thus let $P(v, f|_{E_2(v)})$ express that (i) E' is transitive at v and (ii) v does not want to Nash deviate under the partition specified by E'. Hence deciding the existence of a connected Nash-stable partition is an LCC problem. Since we can calculate players' utilities in E', maximising utilitarian or egalitarian or Nash social welfare is also an LCC problem. Using this general technique, we find the following.

Theorem 5. *There is an $\tilde{O}(2^{kd^2} n)$ algorithm that, given a graphical hedonic game and a tree decomposition, decides whether there exists a connected partition π of the agent set that satisfies (a combination of) (i) individual rationality, (ii) Nash stability, (iii) individual stability, (iv) envy-freeness. Subject to any combination (or none) of the preceding conditions, we can also maximise utilitarian, egalitarian, or Nash social welfare under π.*

A perfect partition can be found in slightly worse time. We can always modify the dynamic programming implementation to actually return a partition π (if it exists) in the same time bound. By using the technique of Corollary 3, we can drop the condition that π is connected in exchange for a worse time bound of $\tilde{O}(2^{kd^5} n)$.

We can also use the LCC approach to get the following result about verifying whether a *given* partition satisfies a stability or optimality criterion.

Theorem 6. *There is an $\tilde{O}(2^{kd^2} n)$ algorithm that given a hedonic game, an associated dependency graph, a tree decomposition, and a partition π of N, decides whether π is (i) Pareto optimal, (ii) core-stable, (iii) strict-core-stable.*

While the method of reduction to LCC problems is evidently quite powerful, it does not seem to capture Σ_2^p-questions like whether a core-stable outcome exists.

9 Allocation of Indivisible Goods

In the problem of allocating indivisible goods, we are given a set $\mathcal{O} = \{o_1, \dots, o_m\}$ of objects that need to be allocated to agents N who have preferences over bundles $B \subseteq \mathcal{O}$ of objects (see Bouveret, Chevaleyre, and Maudet (2016) for a survey). Throughout, we will assume that no bundle is unacceptable to any agent (a weak free-disposal assumption). This setting can quite naturally be captured as a hedonic game with agent set $N \cup \mathcal{O}$, where no coalition containing 2 different agents from N is allowed, and $i \in N$ likes a coalition $S \in \mathcal{N}_i$ just as much as i likes the bundle $S \setminus \{i\}$. The objects, on the other hand, are indifferent between all outcomes. With this implementation, the hedonic-game- and allocation-notions of envy-freeness, Pareto-optimality, and of maximising social welfare coincide perfectly.

This hedonic game is also a graphical hedonic game whose dependency graph is bipartite with N on one side and $\mathcal{O}$ on the other, with an edge from i to o whenever i cares about whether o is part of i's bundle. Note that this graphical hedonic game does not capture the requirement that coalitions may only contain a single agent from N, but we will enforce this condition later.

Let us now take a class of allocation problems whose associated bipartite graphs have bounded treewidth and bounded degree. The latter condition implies that every agent desires a bounded number of objects, and every object is desired by a bounded number of agents. The results developed over the preceding sections will imply that on such a restricted class, we can efficiently find allocations that are fair and/or efficient, in contrast to many hardness results in the unrestricted case.

Using HG-logic, we can identify agents that belong to $\mathcal{O}$ as those agents that are indifferent between all coalitions containing them. Hence HG-logic is expressive enough to require that a given coalition contains at most 1 non-object agent. Hence, using HG-logic, we have an algorithm that decides the existence of a Pareto-optimal and envy-free allocation, which is Σ_2^p-complete for general preferences represented in a logic representation (Bouveret and Lang 2008) and even for additive utilities (de Keijzer et al. 2009). Similarly, there is an algorithm that decides the existence of an envy-free and *complete* allocation (where every object must be allocated); this problem is NP-hard for general additive utilities (Lipton et al. 2004). We can also use an LCC-based algorithm to find an allocation that maximises social welfare among envy-free ones, or an algorithm that finds a complete allocation of minimum envy. Looking at the allocation problem from a hedonic game perspective, we can also readily define intriguing notions of *stable* allocations in which agents don't want to swap items (possibly even in larger swap cycles). Many such properties can be described in HG-logic.

In the context of combinatorial auctions, Conitzer, Derryberry, and Sandholm (2004) provide a different way of exploiting a tree decomposition to efficiently allocate objects. Here, the objects are arranged in an *item graph* of bounded treewidth, and agents are assumed to only demand bundles inducing a connected subgraph of the item graph. With this restriction, winner determination becomes feasible in polynomial time. Note, however, that constructing a suitable item graph of small treewidth is computationally hard (Gottlob and Greco 2007).

10 Necessity of the Degree Bound

In this section, we show that a variety of problems of type ϕ-HEDONIC GAMES are NP-hard for games of bounded treewidth but unbounded degree. This establishes that unless P = NP it is necessary for our fixed-parameter tractability result that we bound the degree of the hedonic games. The bound on the treewidth is also necessary; this follows from slight modifications of standard hardness reductions (e.g., those of Sung and Dimitrov (2010)).

Theorem 7. CORE-EXISTENCE *is NP-hard even for graphical hedonic games of treewidth 2 that are given by an HC-net.*

Proof. By reduction from 3SAT. Given a formula ϕ which we may assume not to be satisfied by setting all variables false, introduce one agent $x_1, \dots, x_n$ for each variable occurring in ϕ, and add 3 other agents a, b, c. The variable agents are indifferent between all outcomes (no associated rules). The preferences of agents a, b, c are cyclic and given by the net

$$\begin{aligned} \phi \mapsto_a 3, \quad b \mapsto_a 2, \quad c \mapsto_a 1, \quad b \wedge c \mapsto_a -10; \\ c \mapsto_b 2, \quad a \mapsto_b 1, \quad c \wedge a \mapsto_b -10; \\ a \mapsto_c 2, \quad b \mapsto_c 1, \quad a \wedge b \mapsto_c -10. \end{aligned}$$

An agent dependency graph of this game is given by a triangle on $\{a, b, c\}$ plus n leaves attached to a; this is easily seen to have treewidth 2 (and actually even pathwidth 2).

We show the game admits a core-stable outcome if and only if ϕ is satisfiable. Suppose ϕ is satisfied by some assignment. Then take the partition π where a is together with all true variable agents, with $\{b, c\} \in \pi$, and with all false variable agents together. Then π is core-stable, because the variable agents (being indifferent) are not part of any blocking coalition, so that a does not obtain utility larger than 3 in any blocking coalition.

Conversely, suppose that the game admits a core-stable partition π. We show that a obtains utility 3 in π, which implies that a is together with variables that satisfy ϕ so that ϕ is satisfiable, as required. Suppose not. By individual rationality, a, b, c are not all together, so one of them is not together with either of them, say b. But then $\{a, b\}$ blocks. Such a blocking coalition exists for any choice of lonely agent as a obtains utility $\leqslant 2$, so π is not stable, contradiction. ☐

This result cannot be improved to apply to HC-nets of treewidth 1, which follows from results of Igarashi and Elkind (2016). By adapting the reduction of Peters (2015), this problem should be Σ_2^p-complete, at least for treewidth 4.

Theorem 8. NASH-STABLE-EXISTENCE *is NP-hard even for graphical hedonic games of treewidth 1 that are given by an HC-net.*

Proof. By reduction from X3C. Given an instance with elements $x_1, \dots, x_n$ and sets $s_1, \dots, s_m$, we construct an HC-net where the agents are given by the elements x_i and an extra stalker agent. Every element hates the stalker, but the stalker loves every coalition of elements—except coalitions of form s_j. Note that the dependency graph of this game is a star with the stalker in the center. If the X3C-instance has a solution, then partitioning the x_i as in the solution and putting the stalker in a singleton is Nash stable. Conversely, in every Nash stable partition, the stalker needs to be alone by individual rationality for the elements. By Nash stability, the stalker does not want to join any coalition, and so every coalition must be of form s_j; thus the partition of the elements gives an X3C-solution. ☐

A similar construction works for individual stability for treewidth 2, see also Peters and Elkind (2015, Thm. 2). A reduction from 3SAT gives hardness for PERFECT-EXISTENCE on trees (agents for literals, complementary ones hate each other, a formula agent is satisfied iff the formula is satisfied).

11 Conclusions and Future Work

We have shown that restricting treewidth and degree of hedonic games is a potent avenue to obtaining tractability results for a broad array of important computational problems concerning hedonic games. Our application to the problem of allocating indivisible goods shows how useful tractability results for hedonic games can be: because hedonic games are a very general model encompassing far more than just 'how to find friends', a possibility result for hedonic games translates to easiness for any problem that involves partitions with some elements having preferences over who gets what. It will be interesting to see whether this idea can be further applied elsewhere.

The notion of a graphical hedonic game suggests a wide variety of interesting questions for future work: Are there alternative conditions on graph topology that yield tractability? Examples could be bipartiteness, planarity, or H-minor freeness. Can we say anything about the structure of stable outcomes in dependence on the structure of the graphical hedonic game?

An open problem more closely related to this paper is the problem of finding faster algorithms than those provided through HG-logic for Σ_2^p-type questions like the existence of a core-stable partition or of finding a Pareto-optimal partition.

We should also note that the hardness results in the preceding section are not easily generalised to, say, additively separable hedonic games. It would be interesting to know if we can dispense with the degree bound on this more restricted class. Some results for welfare-maximisation can be found in Bachrach et al. (2013).

Acknowledgements Thanks to four anonymous reviewers whose feedback improved this paper considerably, and to Edith Elkind for advice and guidance. I am supported by EPSRC.

References

Adcock, A. B.; Sullivan, B. D.; and Mahoney, M. W. 2013. Tree-like structure in large social and information networks. In *International Conference on Data Mining (ICDM) 2013*, 1–10. IEEE.

Aziz, H.; Brandt, F.; and Harrenstein, P. 2014. Fractional hedonic games. AAMAS '14, 5–12.

Bachrach, Y.; Kohli, P.; Kolmogorov, V.; and Zadimoghaddam, M. 2013. Optimal coalition structure generation in cooperative graph games. AAAI '13.

Ballester, C. 2004. NP-completeness in hedonic games. *Games and Economic Behavior* 49(1):1–30.

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

Bodlaender, H. 1988. Dynamic programming on graphs with bounded treewidth. *Automata, Languages and Programming* 105–118.

Bodlaender, H. L. 1993. A linear time algorithm for finding tree-decompositions of small treewidth. STOC 1993, 226–234. ACM.

Bodlaender, H. L. 1994. A tourist guide through treewidth. *Acta cybernetica* 11(1-2):1.

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

Bouveret, S., and Lang, J. 2008. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research* 525–564.

Bouveret, S.; Chevaleyre, Y.; and Maudet, N. 2016. Fair allocation of indivisible goods. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 15.

Conitzer, V.; Derryberry, J.; and Sandholm, T. 2004. Combinatorial auctions with structured item graphs. AAAI '04, 212–218.

Courcelle, B., and Engelfriet, J. 2012. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138. Cambridge University Press.

Courcelle, B. 1990. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and computation* 85(1):12–75.

de Keijzer, B.; Bouveret, S.; Klos, T.; and Zhang, Y. 2009. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *Algorithmic Decision Theory*. Springer. 98–110.

Dunbar, R. I. 1992. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution* 22(6):469–493.

Elkind, E., and Wooldridge, M. 2009. Hedonic coalition nets. AAMAS '09, 417–424.

Frick, M., and Grohe, M. 2004. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic* 130(1):3–31.

Gottlob, G., and Greco, G. 2007. On the complexity of combinatorial auctions: structured item graphs and hypertree decomposition. EC '07, 152–161. ACM.

Gottlob, G.; Greco, G.; and Scarcello, F. 2005. Pure nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research* 357–406.

Igarashi, A., and Elkind, E. 2016. Hedonic games with graph-restricted communication. (in preparation).

Kearns, M.; Littman, M. L.; and Singh, S. 2001. Graphical models for game theory. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, 253–260. Morgan Kaufmann Publishers Inc.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. EC '04, 125–131. ACM.

Peters, D., and Elkind, E. 2015. Simple causes of complexity in hedonic games. IJCAI '15, 617–623, arXiv:1507.03474 [cs.GT].

Peters, D. 2015. Σ_2^p-complete problems on hedonic games. arXiv:1509.02333 [cs.GT].

Robertson, N., and Seymour, P. D. 1986. Graph minors. II. Algorithmic aspects of tree-width. *Journal of algorithms* 7(3):309–322.

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

Woeginger, G. J. 2013. Core stability in hedonic coalition formation. In *SOFSEM 2013: Theory and Practice of Computer Science*. Springer. 33–50.