Core Stability in Additively Separable Hedonic Games of Low Treewidth

Tesshu Hanaka
Kyushu University, Fukuoka, Japan

Noleen Köhler
University of Leeds, UK

Michael Lampis
Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Abstract

Additively Separable Hedonic Games (ASHGs) are coalition-formation games where we are given a directed graph whose vertices represent n selfish agents and the weight of each arc uv denotes the preferences from u to v. We revisit the computational complexity of the well-known notion of core stability of symmetric ASHGs, where the goal is to construct a partition of the agents into coalitions such that no group of agents would prefer to diverge from the given partition and form a new coalition. For Core Stability Verification (CSV), we first show the following hardness results: CSV remains coNP-complete on graphs of vertex cover 2; CSV is coW[1]-hard parameterized by vertex integrity when edge weights are polynomially bounded; and CSV is coW[1]-hard parameterized by tree-depth even if all weights are from $\{-1, 1\}$. We complement these results with essentially matching algorithms and an FPT algorithm parameterized by the treewidth $\mathtt{tw}$ plus the maximum degree Δ (improving a previous algorithm's dependence from $2^{O(\mathtt{tw}\Delta^2)}$ to $2^{O(\mathtt{tw}\Delta)}$). We then move on to study Core Stability (CS), which one would naturally expect to be even harder than CSV. We confirm this intuition by showing that CS is Σ_2^p-complete even on graphs of bounded vertex cover number. On the positive side, we present a $2^{2^{O(\Delta\mathtt{tw})}} n^{O(1)}$-time algorithm parameterized by $\mathtt{tw} + \Delta$, which is essentially optimal assuming Exponential Time Hypothesis (ETH). Finally, we consider the notion of k-core stability: k denotes the maximum size of the allowed blocking (diverging) coalitions. We show that k-CSV is coW[1]-hard parameterized by k (even on unweighted graphs), while k-CS is NP-complete for all $k \geq 3$ (even on graphs of bounded degree with bounded edge weights).

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Hedonic games, Treewidth, Core stability

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2024.39

Related Version *Full Version*: `https://arxiv.org/abs/2402.10815`

Funding *Tesshu Hanaka*: Partially supported by JSPS KAKENHI Grant Numbers JP21H05852, JP21K17707, JP22H00513, JP23H04388.
Noleen Köhler: Partially supported by ANR project ANR-18-CE40-0025-01 (ASSK).
Michael Lampis: Supported by ANR project ANR-21-CE48-0022 (S-EX-AP-PE-AL).

1 Introduction

Coalition-formation games model situations where a subset of selfish agents need to be partitioned into teams (coalitions) in such a way that takes into account their preferences. Because such games capture a vast array of interesting situations in the real world [40], they have been a subject of intense study in computational social choice and the social sciences at large. One particularly interesting and natural special case of such games is when the preferences of each agent only depend on the other agents that she is placed together with in the same coalition (and not on the placement of agents on other coalitions). Such games are referred to in the literature as *hedonic games* and have also attracted much interest from the computer science perspective [1, 2, 6, 7, 10, 12, 13, 20, 30, 35, 42], thanks in part to their numerous applications in, for example, social network analysis [36], scheduling group activities [18], and allocating tasks to wireless agents [41]. For more information we refer the reader to [14] and the relevant chapters of standard computational social choice texbooks [4].

Hedonic games are extremely general. Unfortunately, this generality renders them hard to study from the computer science perspective – indeed, even listing the preferences of all n agents takes space exponential in n as the naïve approach would require listing an ordering of all coalitions for each agent. This motivates the study of natural restrictions of hedonic games. In this paper we focus on one of the most natural such restrictions: *Additively Separable Hedonic Games* (ASHGs) introduced in [11], where the input is an edge-weighted directed graph, vertices represent the agents, and the weight of the arc uv denotes the preference of u for v, that is, the utility that agent u derives from being in the same coalition as v. The utility of an agent u in a coalition C can then be succinctly encoded as the sum of the weights of edges incident on u with their other endpoint in C. If the weights of uv and vu are the same for every pair of agents u, v, ASHGs are called *symmetric*. For symmetric ASHGs, the input graph becomes undirected.

In any situation where agents behave selfishly, it becomes critical to look for *stable* outcomes, that is, outcomes which the agents are likely to accept, based on their preferences. In the context of ASHGs, the question then becomes: given an edge-weighted graph G representing the agents' preferences, can we find a stable partition of the agents into coalitions (possibly also optimizing some other social welfare goal)? The computational complexity of such questions has been amply studied [3, 5, 21, 27, 36, 37, 44] and several natural notions of stability have been proposed. In this paper we revisit the computational complexity of one of the most widely-studied such notions, which is called *core stability* ([35, 39, 43, 45]). Intuitively, a partition of n agents is called *core stable*, if a group of agents does not want to leave their coalitions to form a coalition together. More formally, given a partition $\mathcal{P}$ of the agents, a *blocking coalition* is a set of agents X such that all $v \in X$ have strictly higher utility in X than in the initial partition $\mathcal{P}$. Hence, if a blocking coalition X exists, the initial partition is *unstable*, because the agents of X would prefer to form a new coalition. A partition is then called core stable if no blocking coalition (of any size) exists. Notice that core stability is a very strong (and hence very desirable) notion of stability, compared with simpler notions, such as Nash stability (which only precludes divergence by a single agent).

Attractive though it may be from the game theory point of view, the notion of core stability presents some serious drawbacks from the point of view of computational complexity. In particular, deciding if an ASHG admits a core stable outcome is not just NP-hard, but in fact Σ_2^p-complete, that is, complete for the second level of the polynomial hierarchy [45], even if the preferences are symmetric (i.e., the input graph is undirected), has bounded degree, and edge weights are bounded by a constant [39]. Compared to simpler notions of stability, such as Nash stability (which is "only" NP-complete [23]), core stability is therefore highly intractable, and this strongly motivates the search for a better understanding of what the source of this intractability is and for ways to deal with it. The focus of this paper is on using notions of graph structure from parameterized complexity to achieve a more fine-grained understanding of the complexity of this problem. Throughout the paper we will concentrate on the case where agent preferences are *symmetric*, that is, the given graphs are undirected. Since most of our results are negative, this (natural) restriction only renders them stronger.

	CSV	CS
vc	coNP-complete ($\texttt{vc} = 2$) $(\texttt{vc}w_{\max})^{O(\texttt{vc})}\Delta^2 + O(\texttt{vc}n)$	Σ_2^p-complete ($\texttt{vc} = 12$)
vi	coW[1]-hard ($w_{\max} = n^{O(1)}$) $f(\texttt{vi} + w_{\max})n^{O(1)}$	
td	coW[1]-hard ($w \in \{-1, 1\}$)	
pw		$\nexists 2^{2^{o(\texttt{pw})}} n^{O(1)}$ ($\Delta = O(1)$)
tw	$(\Delta w_{\max})^{O(\texttt{tw})} n^{O(1)}$ $2^{O(\texttt{tw}\Delta)}(n + \log w_{\max})^{O(1)}$	$2^{2^{O(\Delta\texttt{tw})}} n^{O(1)}$

Figure 1 The complexity of Core Stability (CS) and Core Stability Verification (CSV) with respect to graph parameters: treewidth (tw), pathwidth (pw), tree-depth (td), vertex integrity (vi), and vertex cover (vc). We denote by $w_{\max}$ the maximum absolute weight. The hardness results are colored in red and the algorithmic results are colored in blue. The connection between the upper parameter p and the lower parameter q indicates that $q \leq p + 1$ holds for any graph G.

Our results. In this paper we present several results that improve and clarify the state of the art on the complexity of finding core stable outcomes in ASHGs (see Figure 1). We study two closely related problems: Core Stability (CS) and Core Stability Verification (CSV), which correspond to deciding if a core stable partition exists and deciding if a given partition is indeed core stable respectively. Intuitively, the reason CS is complete for the second level of the polynomial hierarchy (and not just NP-complete) is that CSV is also known to be intractable (coNP-complete [16, 43]). Our high-level aim is to understand which parts of the combinatorial structure of the input are responsible for the complexity of these two problems. In order to quantify the input structure we will use standard structural tools from the toolbox of parameterized complexity, such as the notions of treewidth and related parameters[1].

We begin our investigation with CSV and ask the question which restrictions on the input are likely to render the problem tractable (or conversely, what are the sources of the problem's intractability). We identify two possible culprits: the problem could become easy if we either impose restrictions on the graph structure, for example by requiring that the input be of low treewidth or degree, or if we impose restrictions on the allowed edge weights. Our results indicate that these two sources of intractability interact in non-trivial ways: placing restrictions of one type is typically not enough to render the problem tractable, but the problem does sometimes become tractable if we restrict both the graph structure and the allowed weights. More precisely, we show that:

- If we place absolutely no restrictions on the allowed weights, CSV remains hard even on severely restricted instances, that is, graphs of vertex cover 2 (Theorem 2). We find this rather surprising, as this class of graphs (which are essentially stars with one additional vertex) is rarely general enough to render problems intractable.
- One may be tempted to interpret the previous result as an artifact of the exponentially large weights we allow in the input. However, we show that even if we place the restriction that weights are polynomially bounded in the input size, CSV still remains quite hard from

[1] Throughout the paper we assume the reader is familiar with the basics of parameterized complexity, as given for example in [17].

the parameterized perspective, and more precisely coW[1]-hard parameterized by vertex integrity (Theorem 3). Note that graphs with small vertex integrity are graphs where there exists a small separator whose removal breaks down the graph into components of bounded size, so this parameterization is again rather restrictive because it usually easily renders most graph problems almost as tractable as parameterizing by vertex cover [25, 33].

- Finally, we show that even if we insist on weights only being selected from the set $\{-1, 1\}$, CSV is coW[1]-hard parameterized by tree-depth (Theorem 4).

Taken together these results show that CSV is an unusually intractable problem where hardness comes from a combination of two factors: the complexity of dealing with the edge weights and the complexity of dealing with the graph-theoretic structure of the input. We complement the above with several algorithms that paint a clearer picture of the complexity of CSV showing that: (i) CSV is polynomial-time solvable on trees (Theorem 5), hence Theorem 2 cannot be extended to graphs of vertex cover 1 (i.e., stars) (ii) CSV is FPT parameterized by vertex integrity plus the maximum edge weight (Theorem 7), so the hardness result of Theorem 4 cannot be extended to vertex integrity (iii) Theorem 4 is matched by an XP algorithm parameterized by treewidth with parameter dependence $(\Delta w_{\max})^{O(\mathtt{tw})}$, that is, an XP algorithm when weights are polynomially bounded (Theorem 8) (iv) the former algorithm can be improved to an FPT running time (even for unbounded weights) if we parameterize by $\mathtt{tw} + \Delta$ (this was already observed by Peters [38], who gave an algorithm with parameter dependence $2^{O(\Delta^2 \mathtt{tw})}$, but we improve this complexity to $2^{O(\Delta \mathtt{tw})}$ in Theorem 9).

The results above paint a comprehensive and rather negative picture on the complexity of CSV, which seems to imply that our main problem, that is, *finding* core-stable partitions, is likely to be even more intractable. We confirm this intuition by showing that CS remains Σ_2^p-complete even on graphs of bounded vertex cover (Theorem 10). One encouraging piece of news, however, is that we did manage to obtain an FPT algorithm when CSV is parameterized by $\mathtt{tw} + \Delta$, so this seems like a case worth considering for CS. Indeed, Peters [38] already showed that CS is FPT for this parameterization, without, however, giving an explicit algorithm (his argument was based on Courcelle's theorem). We improve upon this by giving an explicit algorithm whose dependence is *double-exponential* on $\mathtt{tw} + \Delta$, using the technique of reducing to ∃∀-SAT advocated in [31] (Theorem 11). Despite fixed-parameter tractability, it is fair to say that the running time of our algorithm is quite disappointing. Our main contribution in this part is to show that this is, unfortunately, likely to be optimal: even for instances of bounded degree, the existence of an algorithm with better than double-exponential dependence on treewidth would violate the ETH (Theorem 16). This shows another aspect where core-stability is significantly harder than Nash stability, which has "just" slightly super-exponential dependence in $\mathtt{tw} + \Delta$ [28]. Note that the phenomenon that problems complete for the second level of the polynomial hierarchy tend to have double-exponential complexity in treewidth has been observed before [22, 32, 34, 8]. Along the way, we provide a fine-grained analysis of the complexity of solving ∃∀-SAT parameterized by treewidth, which may be of independent interest.

Finally, we conclude our paper by considering one last relevant parameter: the size of the allowed blocking coalition. We say that a partition is k-core stable if no blocking coalition of size at most k exists. The concept of k-core stability was first proposed in [20] for handling more practical scenarios. For small values of k this is a natural variation of the problem, which could potentially render it more tractable – indeed, for k fixed, CSV is trivially in P and CS is trivially in NP. Unfortunately, we show that not much more is gained from these parameterizations: CSV is coW[1]-hard parameterized by k (even on unweighted graphs); while k-CS is NP-complete for all fixed $k \geq 3$, even on graphs of bounded maximum degree and with bounded weights.

The full version of this paper is available on arXiv, where all missing proofs can be found.

2 Preliminaries

We use standard graph-theoretic notation and focus on undirected graphs. An Additively Separable Hedonic Game (ASHG) is represented by a directed graph $G = (V, E)$, where vertices of V represent the agents, and a weight function $w : E \to \mathbb{Z}$. To simplify notation we will extend w to all pairs of vertices and assume that $w(uv) = 0$ whenever $uv \notin E$. We also assume that the self-utility of an agent is 0, that is, $w(uu) = 0$. If $w(uv) = w(vu)$ for every pair of u, v, an ASHG is called *symmetric* and the input graph is undirected. A partition $\mathcal{P}$ of V is a collection of disjoint subsets of V whose union includes all of V. We will call the sets of such a partition *coalitions*. Slightly abusing notation, we will write, for $u \in V$, $\mathcal{P}(u)$ to denote the set of $\mathcal{P}$ that contains u. The utility of an agent $u \in X$ in a set $X \subseteq V$ is defined as $\mathtt{ut}(X, u) = \sum_{v \in X} w(uv)$, while the utility of u in a partition $\mathcal{P}$ is defined as $\mathtt{ut}_{\mathcal{P}}(u) = \mathtt{ut}(\mathcal{P}(u), u) = \sum_{v \in \mathcal{P}(u)} w(uv)$. Even though we defined w as a function to the integers, we will sometimes allow rational edge weights, but with denominators sufficiently small that it will always be easy to obtain an equivalent integer instance by multiplying all weights by an appropriate integer. We use $w_{\max}$ to denote the maximum *absolute* weight of a given ASHG instance. Unless otherwise stated, we assume that w is given to us encoded in binary (and hence $w_{\max}$ may have value exponential in the input size).

We are chiefly interested in the following notion of stability.

▶ **Definition 1** (Core stability). *A partition $\mathcal{P}$ of an ASHG (G, w) is* core stable*, if there exists no $X \subseteq V(G)$ such that for all $u \in X$ we have $\mathtt{ut}(X, u) > \mathtt{ut}_{\mathcal{P}}(u)$.*

If the set X mentioned in Definition 1 does exist, then we say that $\mathcal{P}$ is unstable and that X is a *blocking coalition*. For fixed integer values of k, we will also study the notion of k-Core Stability: a partition is k-core stable if no blocking coalition of size at most k exists.

The two computational problems we are interested in are Core Stability (CS) and Core Stability Verification (CSV). In the former problem we are given as input an ASHG and are asked if there exists a core stable partition; in the latter we are also given a specific partition $\mathcal{P}$ and are asked if $\mathcal{P}$ is core stable.

We say that a partition $\mathcal{P}$ of $V(G)$ is *connected* if $G[P]$ is connected for every $P \in \mathcal{P}$. Notice that for both CSV and CS we may assume that the partition $\mathcal{P}$ we seek or we are given is connected, as replacing a disconnected coalition $P \in \mathcal{P}$ with a coalition for each of its components does not change $\mathtt{ut}_{\mathcal{P}}(u)$ for any $u \in V$ and hence does not affect stability.

Graph parameters and Parameterized Complexity. We assume the reader is familiar with the basics of parameterized complexity, such as the classes FPT and W[1], as given for example in [17]. We assume that the reader is also familiar with standard structural graph parameters. The parameters we will focus on are treewidth (tw), pathwidth (pw), tree-depth (td), vertex integrity (vi), and vertex cover (vc). For the definitions of treewidth and pathwidth, as well as the corresponding (nice) decompositions we refer the reader to [17]. The vertex integrity $\mathtt{vi}(G)$ of a graph G is defined as the minimum k such that there exists a set $S \subseteq V(G)$ (called a $\mathtt{vi}(k)$-set) such that the largest component of $G - S$ has order at most $k - |S|$. The tree-depth of a graph G is defined inductively as follows: an isolated vertex

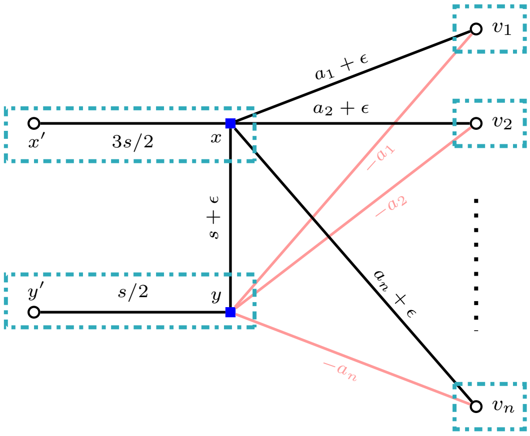

Figure 2 The graph (G, w) constructed in the proof of Theorem 2. The vertex cover is marked by blue squares. The initial partition is given by dot-dashed boxes.

has tree-depth 1; the tree-depth of a disconnected graph is the maximum of the tree-depth of its components; the tree-depth of a connected graph G is defined as $\min_{v \in V(G)} \texttt{td}(G - v) + 1$. The vertex cover of G is the size of the smallest set of vertices of G that intersects all edges.

It is well known that for all graphs G we have $\texttt{tw}(G) \leq \texttt{pw}(G) \leq \texttt{td}(G) \leq \texttt{vi}(G) \leq \texttt{vc}(G) + 1$ where the second relationship is shown in [9] and the others follow immediately from the definition of the respective parameters. In terms of parameterized complexity these parameters therefore form a hierarchy: if a problem is FPT for a smaller parameter, then it is FPT for the larger ones and conversely if a problem is intractable for a large parameter, then it is intractable for a smaller one. We therefore say that larger parameters are more restrictive, with vertex cover being the most restrictive parameter we consider. We use $\Delta(G)$ to denote the maximum degree of a graph G and omit G when it is clear from the context.

3 Core Stability Verification

In this section we study the complexity of Core Stability Verification (CSV). What we discover is that this is an unusually intractable problem, even for quite restricted parameterizations. We complete the picture by giving complementing algorithms.

3.1 Hardness Results

We first prove the following three hardness results.

▶ **Theorem 2.** *Core Stability Verification is weakly coNP-complete on graphs of vertex cover number 2.*

Proof. First note that CSV is in coNP as it is polynomial to verify that a given coalition is blocking. We give a reduction from Partition. Given a set of positive integers $A = \{a_1, \ldots, a_n\}$, the Partition problem asks whether there exists a subset A' of A such that $\sum_{a \in A'} a = s/2$ where $s = \sum_{a \in A} a$. This problem is well-known to be weakly NP-complete [24].

We construct an instance of CSV. The construction is depicted in Figure 2. First, we create n vertices $v_1, \ldots, v_n$ corresponding to $a_1, \ldots, a_n \in A$ and four vertices x, y, x', y'. Then we add edges $v_i x$ of weight $a_i + \epsilon$, $v_i y$ of weight $-a_i$, xx' of weight $3s/2$, yy' of weight $s/2$, and xy of weight $s+\epsilon$. Here, without loss of generality, let ϵ be an integer sufficiently smaller than $\min_i a_i$; this can be achieved for example by multiplying all elements of A (and s) by n, and setting $\epsilon = 1$. Let (G, w) be the constructed graph. The partition $\mathcal{P}$ to verify consists of $\{x, x'\}$, $\{y, y'\}$ and singletons $\{v_1\}, \ldots, \{v_n\}$. Also note that $\{x, y\}$ is a vertex cover of G.

If there exists $A' \subseteq A$ such that $\sum_{a \in A'} a = s/2$, then the coalition $X = \{v_i : a_i \in A'\} \cup \{x, y\}$ blocks $\mathcal{P}$. To see this, observe that the utility of each vertex in X increases by at least ϵ and thus X is a blocking coalition of $\mathcal{P}$.

Conversely, suppose that there exists a blocking coalition X of $\mathcal{P}$. Clearly, X contains neither x' nor y'. If X does not contain x, no vertex can have positive utility in X. Thus, they also do not join X as they have non-negative utility in $\mathcal{P}$. Thus, X must contain x. To increase the utility of x, y must be contained in X. In particular, if y was not contained in X, then the utility of x would be at most $s + n\epsilon < 3s/2$. Since y must have utility more than $s/2$ in X, it holds that $\sum_{v_i \in X} a_i \leq s/2$. Finally, since x obtains at least $s+\epsilon$ utility in X from $y \in X$, and x has $3/2s$ utility in P, we have that $s + \epsilon + \sum_{v_i \in X} a_i > 3/2s$, that is $\sum_{v_i \in a_i} > 1/2s - \epsilon$. Since ϵ is sufficiently smaller than $\min_i a_i$, this implies that $\sum_{v_i \in X} a_i = s/2$ and hence there exists a subset A' of A such that $\sum_{a \in A'} a = s/2$. ◀

We use a similar but more involved construction to reduce Bin Packing to CSV.

▶ **Theorem 3.** *Core Stability Verification is coW[1]-hard parameterized by vertex integrity, even if all weights are bounded by a polynomial in the input size.*

We prove the third hardness result by a reduction from Bounded Degree Deletion.

▶ **Theorem 4.** *Core Stability Verification is coW[1]-hard parameterized by tree-depth, even if all weights are in $\{-1, 1\}$.*

3.2 Algorithms

In this section, we prove the algorithmic results complementing the hardness results of the previous section.

▶ **Theorem 5.** *Core Stability Verification is polynomial time solvable on trees.*

▶ **Theorem 6.** *Core Stability Verification can be solved in time $(\mathtt{vc} w_{\max})^{O(\mathtt{vc})}\Delta^2 + O(\mathtt{vc}n)$.*

Proof. Given a graph (G, w) and a partition $\mathcal{P}$ of $V(G)$, we check whether there is a blocking coalition $X \subseteq V(G)$ of $\mathcal{P}$. In the algorithm, we first compute a minimum vertex cover S of size $\mathtt{vc}$ in time $O(1.25284^{\mathtt{vc}} + \mathtt{vc}n)$ [29]. Then we guess the intersection S' of X and S. The number of possible candidates of the intersection S' is at most $2^{|S|}$. Let $I' \subseteq V(G) \setminus S$ be the set of vertices in $V(G) \setminus S$ such that each vertex $u \in I'$ satisfies $\sum_{v \in N_G(u) \cap S'} w(uv) > \mathtt{ut}_{\mathcal{P}}(u)$. The vertices in I' could form a blocking coalition of $\mathcal{P}$ by cooperating with the vertices in S'. In order for X to become a blocking coalition, all the vertices in S' must have larger utility in X than their utility in $\mathcal{P}$ after some vertices in I' joined X. This condition can be represented as an Integer Linear Program (ILP) as follows:

$$\sum_{v \in I'} w(uv)x_v + \sum_{v \in N_G(u) \cap S'} w(uv) \geq \mathtt{ut}_{\mathcal{P}}(u) + 1 \quad \forall u \in S' \tag{1}$$

$$x_v \in \{0, 1\} \quad \forall v \in I' \tag{2}$$

where the variable x_v represents whether vertex $v \in I'$ joins X. On the left hand side of (1), $\sum_{v \in N_G(u) \cap S'} w(uv)$ represents the contribution of edge weights in S' to the utility of u. Clearly, the ILP is feasible if and only if there is a blocking coalition X because the utility of each agent in X strictly increases. Note that we suppose that each edge weight is an integer.

Here, the feasibility check of an ILP $\{Ax = b, x \geq 0, x \in \mathbb{Z}^n\}$ can be solved in time $(m \cdot \Lambda)^{O(m)} \cdot ||b||_\infty^2$ where $A \in \mathbb{Z}^{n \times m}$, $b \in \mathbb{Z}^m$, and Λ is an upper bound on each absolute value of an entry in A [19]. By adding a slack variables in $\mathbb{Z}$ to each inequality in (1), the ILP can be transformed into the form $\{Ax = b, x \geq 0, x \in \mathbb{Z}^n\}$ where $n = |I'| + |S'|$ and $m = |S'|$. Since the maximum absolute value of coefficients of variables is $w_{\max}$, the range of $\mathtt{ut}_{\mathcal{P}}(u)$ and $\sum_{v \in N_G(u) \cap S'} w(uv)$ is $[-w_{\max}\Delta, w_{\max}\Delta]$, the ILP can be solved in time $(\mathtt{vc}w_{\max})^{O(\mathtt{vc})}(w_{\max}\Delta)^2 = (\mathtt{vc}w_{\max})^{O(\mathtt{vc})}\Delta^2$. Thus, the total running time is $O(1.25284^{\mathtt{vc}} + \mathtt{vc}n) + 2^{\mathtt{vc}}(\mathtt{vc}w_{\max})^{O(\mathtt{vc})}\Delta^2 = (\mathtt{vc}w_{\max})^{O(\mathtt{vc})}\Delta^2 + O(\mathtt{vc}n)$. ◀

▶ **Theorem 7.** *Core Stability Verification is in FPT parameterized by* $\mathtt{vi} + w_{\max}$.

▶ **Theorem 8.** *Core Stability Verification can be computed in time* $(\Delta w_{\max})^{O(\mathtt{tw})} n^{O(1)}$.

▶ **Theorem 9.** *Core Stability Verification can be computed in time* $2^{O(\mathtt{tw}\Delta)}(n + \log w_{\max})^{O(1)}$.

4 Core Stability

In this section we study the complexity of Core Stability (CS). We first show that CS remains Σ_2^p-complete even on graphs of bounded vertex cover number (Theorem 10).

The second part of this section is dedicated to extending our understanding of the complexity of CS parameterized by $\mathtt{tw} + \Delta$. We give an algorithm for CS running in time $2^{2^{O(\Delta\mathtt{tw})}} n$ (Theorem 11) improving on the previous algorithm based on Courcelle's Theorem by Peters [38]. In order to avoid having to formulate a tedious dynamic programming algorithm, we instead obtain our algorithm via a reduction to ∃∀-SAT, which is known to be solvable in double-exponential time (in treewidth) [15]. We complement these results by giving an ETH based lower bound of $2^{2^{o(\mathtt{pw})}}$ on graphs of bounded degree (Theorem 16). This shows that the double-exponential dependence of our algorithm on treewidth is in fact inevitable, and confirms a pattern shown by other Σ_2^p-complete problems [32].

4.1 Core stability on graphs of bounded vertex cover number

In this section we prove the following result.

▶ **Theorem 10.** *Core Stability is* Σ_2^p*-complete on graphs of vertex cover number 12.*

To obtain this result we use a variation of the constructions used to prove Theorem 2 and Theorem 3; however reducing from an appropriate variant of Partition namely ∃∀-partition. To control how core stable partitions behave we use a gadget utilizing the non-core stable graph given in [3]. Details of this auxiliary gadgets construction and behaviour is omitted due to space limitation.

4.2 Core stability parameterized by maximum degree and treewidth

We first prove the algorithmic result of the section.

▶ **Theorem 11.** *Core Stability can be solved in time* $2^{2^{O(\Delta\mathtt{tw})}} n^{O(1)}$.

Before we prove Theorem 11, let us sketch our high-level strategy. Given an instance of Core Stability, we want to produce an equivalent instance ϕ of $\exists\forall$-SAT, such that ϕ has treewidth roughly $\Delta\mathtt{tw}$, where Δ and $\mathtt{tw}$ are the maximum degree and treewidth of the original instance. We could then use the known (double-exponential) algorithm for $\exists\forall$-SAT [15] to solve our problem. Intuitively, we would then attempt to use the existential part of ϕ to encode the "there exists a partition" part of the problem, and the universal part to encode the "all blocking coalitions fail to be blocking" part.

Fundamentally, this strategy is sound and works in a relatively straightforward way for the universal part: we use a boolean variable for each vertex (to encode whether it belongs in the potential blocking coalition) and to check that a coalition fails to be blocking for a vertex v we need to place a constraint on v and all its (at most Δ) neighbors. This means that a tree decomposition of ϕ should be constructible from a tree decomposition of the square of the original graph, which would have width at most $\Delta\mathtt{tw}$.

Where we run into some more difficulties, however, is in encoding the existential part. Intuitively, this is because encoding the partition of the vertices of a bag into coalitions requires a super-linear number of bits, hence it is not sufficient to define a variable for each vertex. Indeed, to simplify things, we define a variable for each *pair* of vertices that appear together in a bag, encoding whether they are together in a coalition. This means that the treewidth of the formula ϕ we construct is in fact not $O(\Delta\mathtt{tw})$, but actually can only be upper-bounded by $O(\mathtt{tw}^2 + \Delta\mathtt{tw})$.

Nevertheless, we insist on obtaining an algorithm that is double-exponential "only" in $\Delta\mathtt{tw}$, and not in $\mathtt{tw}^2$. In order to circumvent this difficulty we observe that the term that is super-linear in treewidth only depends on existentially quantified variables. Thankfully, we manage to show, via an argument that is more careful than that of [15], that $\exists\forall$-SAT has a time complexity that only needs to be double-exponential in the number of *universally quantified* variables of each bag (Proposition 12). Using this, we are able to show that the second exponent of the running time is "only" $O(\Delta\mathtt{tw})$, which as we show later is optimal, even when $\Delta = O(1)$, under the ETH.

Let us now give some more details. We first recall that $\exists\forall$-SAT is a variant of the SAT problem where we aim to decide the satisfiability of a given quantified Boolean formula (QBF) ϕ which is of the form $\exists x_1 \dots \exists x_k \forall y_1 \dots \forall y_\ell \psi$ where ψ is a DNF formula on variables $x_1, \dots, x_k, y_1, \dots, y_\ell$. Two common ways of associating structure of satisfiability problems is to consider the primal or incidence graph of the formula. The primal graph of a formula ϕ (in CNF or DNF) is a graph on the set of variables of ϕ where two variables are adjacent if they appear in the same clause. Similarly, the incidence graph is a bipartite graph on the set of variables and clauses of ϕ where a variable is adjacent to all clauses it appears in. For convenience, we use a variant of $\exists\forall$-SAT. We say that a QBF ϕ is in $\exists 3\,\mathrm{CNF}\,\forall\,\mathrm{DNF}$ if ϕ can be written as

$$\phi = \exists x_1 \dots \exists x_k \bigwedge_{i=1}^{k'} d_i \quad \forall y_1 \dots \forall y_\ell \bigvee_{i=1}^{\ell'} c_i$$

for some $k, k', \ell, \ell' \in \mathbb{N}$ where d_i are disjunctive clause over variables $x_1, \dots, x_k$ containing at most 3 literals per clause and c_i are conjunctive clauses over variables $x_1, \dots, x_k, y_1, \dots, y_\ell$. We present an algorithm for $\exists 3\,\mathrm{CNF}\,\forall\,\mathrm{DNF}$ in several steps. First, we give an algorithm with an improved running time than that of [15].

▶ **Proposition 12.** *There is an algorithm that takes as input an instance of* $\exists\forall$-SAT $\exists x \forall y \phi(x, y)$*, where* x, y *are tuples of boolean variables,* ϕ *is in 3-*DNF*, and a tree decomposition of the primal graph of* ϕ *where each bag contains at most* $t_\exists$ *existentially quantified variables and at most* $t_\forall$ *universally quantified variables and decides if the input is satisfiable in time* $2^{O(t_\exists + 2^{t_\forall})}|\phi|^{O(1)}$.

▶ **Proposition 13.** *There is an algorithm that takes as input an* $\exists 3\,\mathrm{CNF}\,\forall\,\mathrm{DNF}$-SAT *instance* ϕ *and a tree decomposition of its incidence graph of width* t *that contains at most* $t_\forall$ *universally quantified variables in each bag and at most* 2 *clauses in each bag, and decides* ϕ *in time* $2^{O(t2^{t_\forall})}|\phi|$.

Proof of Theorem 11. We prove Theorem 11 by a reduction to $\exists 3\,\mathrm{CNF}\,\forall\,\mathrm{DNF}$-SAT. Let (G, w) be an instance of CS and (T, β) be a rooted tree decomposition of G. Here, we denote the bag of a node $t \in T$ by $\beta(t)$. First we let (G', w') be the graph obtained from (G, w) by adding edges uv of weight 0 for every pair of vertices u, v appearing in a bag together. It is straightforward to see that (G, w) is a YES-instance of CS if and only if (G', w') is a YES-instance of CS. Additionally, G' is chordal and (T, β) is a tree decomposition of G'.

We construct an instance ϕ of $\exists 3\,\mathrm{CNF}\,\forall\,\mathrm{DNF}$-SAT. We introduce a variable x_e for every $e \in E(G')$ and a variable y_u for every vertex $u \in V(G')$. An assignment $\alpha_X : \{x_e : e \in E(G')\} \to \{0,1\}$ represents a subset of $E(G')$ and an assignment $\alpha_Y : \{y_u : u \in V(G')\} \to \{0,1\}$ represents a subset of $V(G')$. Intuitively, a partition of $V(G')$ corresponds to a set of edges, i.e., by including all edges that are incident to vertices from the same part. For every $t \in V(T)$ we define a formula ϕ_t essentially expressing transitivity. We use formula ϕ_t to enforce sufficient criteria for the set of edges represented by an assignment $\alpha_X : \{x_e : e \in E(G')\} \to \{0,1\}$ to correspond to a partition of $V(G')$. Let

$$\phi_t = \bigwedge_{u_1,u_2,u_3 \in \beta(t)} \big(x_{u_1u_2} \wedge x_{u_2u_3}\big) \to x_{u_1u_3} = \bigwedge_{u_1,u_2,u_3 \in \beta(t)} \Big(\neg x_{u_1u_2} \vee \neg x_{u_2u_3} \vee x_{u_1u_3}\Big).$$

For a fixed partition $\mathcal{P}$ represented by some assignment $\alpha_X : \{x_e : e \in E(G')\} \to \{0,1\}$ our formula needs to ensure that no blocking coalitions exist. This is realized by guaranteeing that for each assignment $\alpha_Y : \{y_u : u \in V(G')\} \to \{0,1\}$ the set corresponding to α_Y is not blocking. Intuitively, the formula ϕ_u defined below ensures that the utility of vertex u in $\mathcal{P}$ is at least as large as the utility of u in the coalition represented by α_Y. To realize this, we make sure that the set N of neighbors of u which are in the coalition represented by α_Y and the set $\tilde{N}$ of neighbors of u which are in the same part as u in $\mathcal{P}$ satisfy $\sum_{v \in N} w(uv) \le \sum_{v \in \tilde{N}} w(uv)$. For $u \in V(G')$ let

$$\phi_u = \bigvee_{\substack{N, \tilde{N} \subseteq N_{G'}(u),\\ \sum_{v\in N} w(uv) \le \sum_{v \in \tilde{N}} w(uv)}} \Bigg(y_u \wedge \bigwedge_{v \in N} y_v \wedge \bigwedge_{v \in N_{G'}(u)\setminus N} \neg y_v \wedge \bigwedge_{v \in \tilde{N}} x_{uv} \wedge \bigwedge_{v \in N_{G'}(u)\setminus \tilde{N}} \neg x_{uv}\Bigg).$$

We now define the formula ϕ to be

$$\phi = \exists x_{e_1} \dots \exists x_{e_m} \bigwedge_{t \in V(T)} \phi_t \quad \forall y_{v_1} \dots \forall y_{v_n} \big(\neg y_{v_1} \wedge \dots \wedge \neg y_{v_n}\big) \vee \bigvee_{u \in V(G')} \phi_u.$$

Observe that ϕ is in $\exists 3\,\mathrm{CNF}\,\forall\,\mathrm{DNF}$. In the following we prove that (G', w') is a YES-instance of CS if and only if ϕ is satisfiable.

First assume that $\mathcal{P}$ is a core stable partition of $V(G')$. We define an assignment $\alpha_X : \{x_e : e \in E(G')\} \to \{0,1\}$ by setting $\alpha_X(x_e) = 1$ if e is incident to two vertices residing in the same part of $\mathcal{P}$ and $\alpha_X(x_e) = 0$ otherwise. This assignment satisfies ϕ_t for every $t \in V(T)$ as for any three vertices $u_1, u_2, u_3 \in \beta(t)$ it holds that if $\alpha_X(x_{u_1u_2}) = 1$ and $\alpha_X(x_{u_2u_3}) = 1$, then u_1, u_2 and u_3 must reside in the same part of $\mathcal{P}$ and hence $\alpha_X(x_{u_1u_3}) = 1$. Furthermore, consider any assignment $\alpha_Y : \{y_u : u \in V(G')\} \to \{0,1\}$ and let $X = \{u \in V(G') : \alpha_Y(y_u) = 1\}$. We have to argue that the formula $\big(\neg y_{v_1} \wedge \dots \wedge \neg y_{v_n}\big) \vee \bigvee_{u \in V(G')} \phi_u$ is satisfied under the assignments α_X and α_Y. In case that $X = \emptyset$, the clause $(\neg y_{v_1} \wedge \cdots \wedge \neg y_{v_n})$ is satisfied. On the other hand, if $X \neq \emptyset$, then there must be a vertex $u \in X$ whose utility in $\mathcal{P}$ is at least as large as its utility in X as the set X cannot be a blocking coalition. Hence, for the sets $N = N_{G'}(u) \cap X$ and $\tilde{N} = N_{G'}(u) \cap P$, where $P \in \mathcal{P}$ is the part containing u, we have that $\sum_{v \in N} w(uv) \leq \sum_{v \in \tilde{N}} w(uv)$. Therefore, ϕ_u contains the clause $(y_u \wedge \bigwedge_{v \in N} y_v \wedge \bigwedge_{v \in N_{G'}(u) \setminus N} \neg y_v \wedge \bigwedge_{v \in \tilde{N}} x_{uv} \wedge \bigwedge_{v \in N_{G'}(u) \setminus \tilde{N}} \neg x_{uv})$ and this clause is satisfied under the assignment α_X and α_Y by choice of N and $\tilde{N}$. This shows that ϕ is satisfiable.

On the other hand, assume that ϕ is satisfiable and let $\alpha_X : \{x_e : e \in E(G')\} \to \{0, 1\}$ be an assignment such that $\bigwedge_{t \in V(T)} \phi_t$ as well as $\forall y_{v_1} \ldots \forall y_{v_n} (\neg y_{v_1} \wedge \cdots \wedge \neg y_{v_n}) \vee \bigvee_{u \in V(G')} \phi_u$ is satisfied under α_X. We let $E_X = \{e \in E(G') : \alpha_X(x_e) = 1\}$ and define a partition $\mathcal{P}$ by letting every part of $\mathcal{P}$ correspond to the vertices of a connected component of the graph $(V(G'), E_X)$. To show that the partition $\mathcal{P}$ is core stable we use the following claim.

▷ Claim 14. For every edge $uv \in E(G')$ it holds that $uv \in E_X$ if and only if u and v are contained in the same part of $\mathcal{P}$.

Towards a contradiction assume that $\mathcal{P}$ is not core stable and let X be a blocking coalition. We define an assignment $\alpha_Y : \{y_u : u \in V(G')\} \to \{0, 1\}$ by setting $\alpha_Y(y_u) = 1$ if $u \in X$ and $\alpha_Y(y_u) = 0$ otherwise. By assumption, the DNF formula $(\neg y_{v_1} \wedge \cdots \wedge \neg y_{v_n}) \vee \bigvee_{u \in V(G')} \phi_u$ is satisfied under assignment α_X and α_Y. As X cannot be empty there is at least one $u \in V(G')$ such that $\alpha_Y(y_u) = 1$ and hence the clause $(\neg y_{v_1} \wedge \cdots \wedge \neg y_{v_n})$ cannot be satisfied under the assignment α_Y which implies that some clause in $\bigvee_{u \in V(G')} \phi_u$ must be satisfied. Let $u \in V(G')$ and $N, \tilde{N} \subseteq N_{G'}(u)$ with $\sum_{v \in N} w(uv) \leq \sum_{v \in \tilde{N}} w(uv)$ such that the clause $(y_u \wedge \bigwedge_{v \in N} y_v \wedge \bigwedge_{v \in N_{G'}(u) \setminus N} \neg y_v \wedge \bigwedge_{v \in \tilde{N}} x_{uv} \wedge \bigwedge_{v \in N_{G'}(u) \setminus \tilde{N}} \neg x_{uv})$ is satisfied under assignments α_X and α_Y. This implies that $N = N_{G'}(u) \cap X$ and $\tilde{N} = \{v \in V(G') : uv \in E_X\}$. Since by Claim 14, $\{v \in V(G') : uv \in E_X\} = N_{G'}(u) \cap P$, where P is the part of $\mathcal{P}$ containing u, this implies that the utility of u in X is less or equal to the utility of u in $\mathcal{P}$. As this contradicts our assumption that X is a blocking coalition it follows that $\mathcal{P}$ is core stable.

▷ Claim 15. The formula ϕ has incidence treewidth at most $\Delta(G')\mathtt{tw}(G') + \mathtt{tw}(G')^2 + 2$. Furthermore, we can construct a decomposition of that width which contains at most 2 clauses per bag and at most $(\Delta(G') + 1)\mathtt{tw}(G')$ universally quantified variables per bag.

As (T, β), G' and the formula ϕ can be computed in time $\mathcal{O}(\mathtt{tw}(G')^2 n)$, combining the reduction with the algorithm from Proposition 13 yields a $2^{2^{\mathcal{O}(\Delta \mathtt{tw})}} n^{O(1)}$ time algorithm. ◀

Finally, we prove our ETH based lower bound.

▶ **Theorem 16.** *Unless the ETH fails, there is no algorithm for* Core stability *running in time $2^{2^{o(\mathtt{pw})}} n^{\mathcal{O}(1)}$ even if G has bounded degree and weights are constant.*

We give an overview of our construction. Given an instance ϕ of (3, 3)-SAT (each variable appears at most 3 times) we construct a graph (G, w) and show that ϕ is satisfiable if and only if (G, w) is core stable. Firstly, we use a gadget based on the non-core stable graph given in [3] to control potential core stable partitions. The auxiliary gadget is a non-core stable graph H that we attach with positive weight edges at a set S of vertices. Because H is not core stable any partition must place some vertex of H in a part with some $s \in S$. Using negative weight edges we can control this behaviour even further. We achieve that $\{h, s\}$ must be in any core stable partition for one vertex $s \in S$ where $h \in V(H)$ is a fixed vertex of H.

We now describe the construction. Every variable x_i, $i \in [n]$ of ϕ is represented by two vertices y_i and $\neg y_i$ and for each $i \in [n]$ we attach one auxiliary gadget at $\{y_i, \neg y_i\}$. We add further vertices (details follow in the next paragraph). Each of these additional vertices v either has its private auxiliary gadget attached at $\{v\}$ or is not connected to any auxiliary gadgets. We define a special set of partitions of $V(G)$ which we refer to as candidate partitions as follows. Any candidate partition $\mathcal{P}$ has to contain either $\{h, y_i\}$ or $\{h, \neg y_i\}$ (but not both) where h is a vertex of the respective auxiliary gadget. For any other vertex v without an attached auxiliary gadget (excluding vertices of auxiliary gadgets) any candidate parition $\mathcal{P}$ has to contain $\{v\}$. For any vertex v with an auxiliary gadget attached at $\{v\}$ any candidate partition $\mathcal{P}$ has to contain the set $\{h, v\}$ where h is a vertex of respective auxiliary gadget. Using the properties of the auxiliary gadget, we can argue that any core stable partition has to be a candidate partition. By the construction, there is a correspondence between assignments and candidate partitions, i.e., $\alpha(x_i) = 1$ if and only if $\{h, y_i\} \in \mathcal{P}$ for candidate partition $\mathcal{P}$ and the corresponding assignment α.

Any blocking coalition of a candidate partition allows us to find a clause which is not satisfied under the assignment which corresponds to the candidate partition and vice versa. This is realized as follows. We take $2m + 1$ cycles $U^1, \ldots, U^m, V^1, \ldots, V^m, Z$ of length $3n$ where $2^m \leq 3n$ is the number of clauses of ϕ. By choosing suitable edge weights, we enforce that any blocking coalition of any candidate partition contains Z and either U^k or V^k (but not both) for every $k \in [m]$. For now, we call any set containing Z, U^k or V^k (but not both) for every $k \in [m]$ (and some other vertices we neglect here) a candidate blocking coalition. We number the clauses of ϕ in such a way that each candidate blocking coalition corresponds to a clause. More sprecifically, the j-th clause corresponds to the candidate blocking coalition X in which $U^k \in X$ if and only if the k-th bit of j in binary is 0.

Each vertex in Z corresponds to the appearance of a variable. Assume $z \in Z$ corresponds to the appearance of variable x_i in clause c_j. We connect z to either y_i or $\neg y_i$ dependent on whether x_i appears negated in c_j. We connect z to either U^k or V^k for every $k \in [m]$ dependent on whether the k-th bit of j in binary is 0 or 1 using a gadget we call clause selection gadget. The gadget enforces that z obtains a $+1$ towards its total utility in X if and only if the candidate blocking coalition X does not encode the clause c_j. By choice of edge weights, we ensure that vertex z can only be convinced to join a blocking coalition if it either gets utility $+1$ from its clause selection gadget or utility $+1$ from y_i (or $\neg y_i$, resp.). On the other hand, y_i ($\neg y_i$, resp.) can only be convinced to join a blocking coalition if it appears as a singleton in the partition we are trying to block and hence the corresponding literal is false. In conclusion, for any candidate partition $\mathcal{P}$ there is a blocking coalition X if and only if for the clause corresponding to X each literal is false. Hence, (G, w) is core stable if and only if ϕ is satisfiable.

5 k-Core Stability

In this section we consider the complexity of finding and verifying k-core stable partitions, when the size of the allowed blocking coalitions k is a parameter. Even though the two problems do become easier when k is a fixed constant (because we can check all possible blocking coalitions in polynomial time), we show that it is likely that not much more can be gained from this assumption: k-CSV is coW[1]-hard parameterized by k (Theorem 17), while k-CS is NP-complete even if $k \geq 3$ is a fixed constant (Theorem 19). On the positive side, we do show that finding 2-core stable partitions is in P, but it is worth noting that the fact that we consider undirected graphs is crucial to obtain even this small tractable case.

▶ **Theorem 17.** *k-Core Stability Verification is in XP when parameterized by k whereas coW[1]-hard even on unweighted graphs.*

Proof. The upper bound can be easily shown by brute force. That is, given a coalition structure $\mathcal{P}$, for each coalition X of size at most k, we check if each agent v in X has higher utility than in $\mathcal{P}$. The running time of brute force is $n^{O(k)}$.

Then we show that k-CSV is W[1]-hard even on unweighted graphs. We give a reduction from k-Clique. Given a graph G, we attach $k-2$ pendant vertices for each vertex in $V(G)$. Let P_v be the set of pendant vertices for v. We set $\mathcal{P} = \{P_v \cup \{v\} : v \in V(G)\}$ as a coalition structure to verify.

In the following, we show that there exists a k-clique in G if and only if there exists a blocking coalition for $\mathcal{P}$. Let C be a clique of size k in G. For C, each vertex in C has the utility $k-1$. Since $v \in V$ has the utility $k-2$ in $\mathcal{P}$, C is a blocking coalition for $\mathcal{P}$. Conversely, let X be a blocking coalition for $\mathcal{P}$. Since vertices in $\bigcup_{v \in V(G)} P_v$ have maximum utility 1 in $\mathcal{P}$, they do not join X. Thus, X is a subset of $V(G)$. Since the utility of $v \in V(G)$ is $k-2$ in $\mathcal{P}$ and $|X| = k$, X is a clique of size k. ◀

▶ **Theorem 18.** *Every graph admits a 2-core stable partition and 2-Core Stability can be solved in polynomial time.*

Proof. Given a weighted graph (G, w), start with the partition where every vertex is a singleton. Order the positive-weight edges in non-increasing order $e_1, e_2, \ldots, e_m$. For each e_i, do the following: if the endpoint of e_i are currently singletons, merge them into a cluster of size 2; otherwise move to the next edge. The resulting partition $\mathcal{P}$ is 2-core stable because if there was a blocking coalition of size 2, it would have to induce an edge $e_i = uv$. However, when e_i is considered, at least one of u, v was not a singleton. Therefore, the utility of that vertex must be larger in $\mathcal{P}$ than in the coalition $\{u, v\}$ contradicting the assumption that $\{u, v\}$ is a blocking coalition. ◀

▶ **Theorem 19.** *For any fixed $k \geq 3$, k-Core Stability is NP-complete on bounded degree graphs, even if the weights are constant.*

6 Conclusion

The general tenor of our results indicates that core stability is an algorithmically highly intractable notion: even for very restricted input structures, obtaining efficient algorithms seems out of reach; and even for the few cases where positive fixed-parameter tractability results can be obtained, complexity lower bounds still push the parameter dependence to prohibitive levels. Despite the above, we believe that a promising avenue for future research may be the further investigation of k-core stability. Even though we have shown that parameterizing the problem by k alone does not help, it would be interesting to ask whether parameterizing at the same time by both k and a structural parameter (such as treewidth) could help us evade the lower bounds that apply to each case individually. Finally, investigating the parameterized complexity of core stability in other variants of hedonic games such as fractional hedonic games [2, 26, 20] is another promising direction.

References

1 Alessandro Aloisio, Michele Flammini, and Cosimo Vinci. The impact of selfishness in hypergraph hedonic games. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1766–1773. AAAI Press, 2020. doi:10.1609/AAAI.V34I02.5542.

2 Haris Aziz, Florian Brandl, Felix Brandt, Paul Harrenstein, Martin Olsen, and Dominik Peters. Fractional hedonic games. *ACM Trans. Economics and Comput.*, 7(2):6:1–6:29, 2019. doi:10.1145/3327970.

3 Haris Aziz, Felix Brandt, and Hans Georg Seedig. Computing desirable partitions in additively separable hedonic games. *Artif. Intell.*, 195:316–334, 2013. doi:10.1016/j.artint.2012.09.006.

4 Haris Aziz and Rahul Savani. Hedonic games. In *Handbook of Computational Social Choice*, pages 356–376. Cambridge University Press, 2016. doi:10.1017/CBO9781107446984.016.

5 Coralio Ballester. NP-completeness in hedonic games. *Games Econ. Behav.*, 49(1):1–30, 2004. doi:10.1016/J.GEB.2003.10.003.

6 Nathanaël Barrot, Kazunori Ota, Yuko Sakurai, and Makoto Yokoo. Unknown agents in friends oriented hedonic games: Stability and complexity. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 1756–1763. AAAI Press, 2019. doi:10.1609/aaai.v33i01.33011756.

7 Nathanaël Barrot and Makoto Yokoo. Stable and envy-free partitions in hedonic games. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 67–73. ijcai.org, 2019. doi:10.24963/ijcai.2019/10.

8 Ivan Bliznets and Markus Hecher. Tight double exponential lower bounds. In Xujin Chen and Bo Li, editors, *Theory and Applications of Models of Computation - 18th Annual Conference, TAMC 2024, Hong Kong, China, May 13-15, 2024, Proceedings*, volume 14637 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2024. doi:10.1007/978-981-97-2340-9_11.

9 Hans L Bodlaender, John R Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995. doi:10.1006/JAGM.1995.1009.

10 Niclas Boehmer and Edith Elkind. Individual-based stability in hedonic diversity games. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1822–1829. AAAI Press, 2020. doi:10.1609/AAAI.V34I02.5549.

11 Anna Bogomolnaia and Matthew O Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002. doi:10.1006/GAME.2001.0877.

12 Felix Brandt, Martin Bullinger, and Anaëlle Wilczynski. Reaching individually stable coalition structures. *ACM Trans. Economics and Comput.*, 11:4:1–4:65, 2023. doi:10.1145/3588753.

13 Martin Bullinger and Stefan Kober. Loyalty in cardinal hedonic games. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 66–72. ijcai.org, 2021. doi:10.24963/ijcai.2021/10.

14 Katarína Cechlárová. Stable partition problem. In *Encyclopedia of Algorithms*, pages 2075–2078. Springer, 2016. doi:10.1007/978-1-4939-2864-4_397.

15 Hubie Chen. Quantified constraint satisfaction and bounded treewidth. In Ramón López de Mántaras and Lorenza Saitta, editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, pages 161–165. IOS Press, 2004.

16 Jiehua Chen, Gergely Csáji, Sanjukta Roy, and Sofia Simola. Hedonic games with friends, enemies, and neutrals: Resolving open questions and fine-grained complexity. In Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh, editors, *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pages 251–259. ACM, 2023. doi:10.5555/3545946.3598644.

17 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

18 Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer, and Gerhard J. Woeginger. Group activity selection problem with approval preferences. *Int. J. Game Theory*, 47(3):767–796, 2018. doi:10.1007/s00182-017-0596-4.

19 Friedrich Eisenbrand and Robert Weismantel. Proximity results and faster algorithms for integer programming using the Steinitz lemma. *ACM Trans. Algorithms*, 16(1):5:1–5:14, 2020. doi:10.1145/3340322.

20 Angelo Fanelli, Gianpiero Monaco, and Luca Moscardelli. Relaxed core stability in fractional hedonic games. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 182–188. ijcai.org, 2021. doi:10.24963/ijcai.2021/26.

21 Michele Flammini, Bojana Kodric, Gianpiero Monaco, and Qiang Zhang. Strategyproof mechanisms for additively separable and fractional hedonic games. *J. Artif. Intell. Res.*, 70:1253–1279, 2021. doi:10.1613/JAIR.1.12107.

22 Florent Foucaud, Esther Galby, Liana Khazaliya, Shaohua Li, Fionn Mc Inerney, Roohani Sharma, and Prafullkumar Tale. Tight (double) exponential bounds for NP-complete problems: Treewidth and vertex cover parameterizations. *CoRR*, abs/2307.08149, 2023. doi:10.48550/arXiv.2307.08149.

23 Martin Gairing and Rahul Savani. Computing stable outcomes in symmetric additively separable hedonic games. *Math. Oper. Res.*, 44(3):1101–1121, 2019. doi:10.1287/MOOR.2018.0960.

24 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

25 Tatsuya Gima, Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, and Yota Otachi. Exploring the gap between treedepth and vertex cover through vertex integrity. *Theor. Comput. Sci.*, 918:60–76, 2022. doi:10.1016/j.tcs.2022.03.021.

26 Tesshu Hanaka, Airi Ikeyama, and Hirotaka Ono. Maximizing utilitarian and egalitarian welfare of fractional hedonic games on tree-like graphs. In Weili Wu and Jianxiong Guo, editors, *Combinatorial Optimization and Applications - 17th International Conference, COCOA 2023, Hawaii, HI, USA, December 15-17, 2023, Proceedings, Part I*, volume 14461 of *Lecture Notes in Computer Science*, pages 392–405. Springer, 2023. doi:10.1007/978-3-031-49611-0_28.

27 Tesshu Hanaka, Hironori Kiya, Yasuhide Maei, and Hirotaka Ono. Computational complexity of hedonic games on sparse graphs. In *PRIMA*, volume 11873 of *Lecture Notes in Computer Science*, pages 576–584. Springer, 2019. doi:10.1007/978-3-030-33792-6_43.

28 Tesshu Hanaka and Michael Lampis. Hedonic games and treewidth revisited. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 64:1–64:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ESA.2022.64.

29 David G. Harris and N. S. Narayanaswamy. A faster algorithm for vertex cover parameterized by solution size. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPIcs*, pages 40:1–40:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.STACS.2024.40.

30 Ayumi Igarashi, Kazunori Ota, Yuko Sakurai, and Makoto Yokoo. Robustness against agent failure in hedonic games. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 364–370. ijcai.org, 2019. doi:10.24963/ijcai.2019/52.

31 Michael Lampis, Stefan Mengel, and Valia Mitsou. QBF as an alternative to Courcelle's theorem. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings*, volume 10929 of *Lecture Notes in Computer Science*, pages 235–252. Springer, 2018. doi:10.1007/978-3-319-94144-8_15.

32 Michael Lampis and Valia Mitsou. Treewidth with a quantifier alternation revisited. In Daniel Lokshtanov and Naomi Nishimura, editors, *12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria*, volume 89 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.IPEC.2017.26.

33 Michael Lampis and Valia Mitsou. Fine-grained meta-theorems for vertex integrity. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation, ISAAC 2021, December 6-8, 2021, Fukuoka, Japan*, volume 212 of *LIPIcs*, pages 34:1–34:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ISAAC.2021.34.

34 Dániel Marx and Valia Mitsou. Double-exponential and triple-exponential bounds for choosability problems parameterized by treewidth. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 28:1–28:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.28.

35 Kazunori Ohta, Nathanaël Barrot, Anisse Ismaili, Yuko Sakurai, and Makoto Yokoo. Core stability in hedonic games among friends and enemies: Impact of neutrals. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 359–365. ijcai.org, 2017. doi:10.24963/ijcai.2017/51.

36 Martin Olsen. Nash stability in additively separable hedonic games and community structures. *Theory Comput. Syst.*, 45(4):917–925, 2009. doi:10.1007/S00224-009-9176-8.

37 Martin Olsen, Lars Bækgaard, and Torben Tambo. On non-trivial Nash stable partitions in additive hedonic games with symmetric 0/1-utilities. *Inf. Process. Lett.*, 112(23):903–907, 2012. doi:10.1016/J.IPL.2012.08.016.

38 Dominik Peters. Graphical hedonic games of bounded treewidth. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 586–593. AAAI Press, 2016. doi:10.1609/AAAI.V30I1.10046.

39 Dominik Peters. Precise complexity of the core in dichotomous and additive hedonic games. In Jörg Rothe, editor, *Algorithmic Decision Theory - 5th International Conference, ADT 2017, Luxembourg, Luxembourg, October 25-27, 2017, Proceedings*, volume 10576 of *Lecture Notes in Computer Science*, pages 214–227. Springer, 2017. doi:10.1007/978-3-319-67504-6_15.

40 Debraj Ray. *A game-theoretic perspective on coalition formation.* Oxford University Press, 2007.

41 Walid Saad, Zhu Han, Tamer Basar, Mérouane Debbah, and Are Hjørungnes. Hedonic coalition formation for distributed task allocation among wireless agents. *IEEE Trans. Mob. Comput.*, 10(9):1327–1344, 2011. `doi:10.1109/TMC.2010.242`.

42 Jakub Sliwinski and Yair Zick. Learning hedonic games. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2730–2736. ijcai.org, 2017. `doi:10.24963/ijcai.2017/380`.

43 Shao Chin Sung and Dinko Dimitrov. On core membership testing for hedonic coalition formation games. *Oper. Res. Lett.*, 35(2):155–158, 2007. `doi:10.1016/j.orl.2006.03.011`.

44 Shao Chin Sung and Dinko Dimitrov. Computational complexity in additive hedonic games. *Eur. J. Oper. Res.*, 203(3):635–639, 2010. `doi:10.1016/J.EJOR.2009.09.004`.

45 Gerhard J. Woeginger. A hardness result for core stability in additive hedonic games. *Math. Soc. Sci.*, 65(2):101–104, 2013. `doi:10.1016/j.mathsocsci.2012.10.001`.