# Computational Complexity of Hedonic Games on Sparse Graphs* 

Tesshu Hanaka ${ }^{1}$, Hironori Kiya ${ }^{2}$, Yasuhide Maei ${ }^{2}$, and Hirotaka Ono $^{2}$<br>${ }^{1}$ Chuo University, Tokyo, Japan<br>${ }^{2}$ Nagoya University, Nagoya, Japan


#### Abstract

The additively separable hedonic game (ASHG) is a model of coalition formation games on graphs. In this paper, we intensively and extensively investigate the computational complexity of finding several desirable solutions, such as a Nash stable solution, a maximum utilitarian solution, and a maximum egalitarian solution in ASHGs on sparse graphs including boundeddegree graphs, bounded-treewidth graphs, and near-planar graphs. For example, we show that finding a maximum egalitarian solution is weakly NP-hard even on graphs of treewidth 2, whereas it can be solvable in polynomial time on trees. Moreover, we give a pseudo fixed parameter algorithm when parameterized by treewidth.


## 1 Introduction

In this paper, we investigate the computational complexity of additively separable hedonic games on sparse graphs from the viewpoint of several solution concepts.

Given the set of agents, the coalition formation game is a model of finding a partition of the set of agents into subsets under a certain criterion, where each of the subsets is called a coalition. Such a partition is called a coalition structure. The hedonic game is a variant of coalition formation games, where each agent has the utility associated with his/her joining coalition. In the typical setting, if an agent belongs to a coalition where his/her favorite agents also belong to, his/her utility is high and he/she feels comfortable. Contrarily, if he/she does not like many members in the coalition, his/her utility must be low; since he/she feels uncomfortable, he/she would like to move to another coalition. Although the model of hedonic games is very simple, it is useful to represent many practical situations, such as formation of research team 2], formation of coalition government [19], clustering in social networks [3, 20, 21, multi-agent distributed task assignment 23], and so on.

The additively separable hedonic game (ASHG) is a class of hedonic games, where the utility forms an additively separable function. In ASHG, an agent has a certain valuation for each of the agents, which represents his/her preference. The valuation could be positive, negative or 0 . If the valuation of agent $u$ for agent $v$ is positive, agent $u$ prefers agent $v$, and if it is negative, agent $u$ does not prefer agent $v$. If it is 0 , agent $u$ has no interest for agent $v$. The utility of agent $u$ for $u$ 's joining coalition $C$ is defined by the sum of valuations of agent $u$ for other agents in $C$. This setting is considered not very but reasonably general. Due to this definition, it can be also defined by an edge-weighted directed graph, where the weight of edge $(u, v)$ represents the valuation of $u$ to $v$. If a valuation is 0 , we can remove the corresponding edge. Note that the undirected setting

[^0]is possible, and in the case the valuations are symmetric; the valuation of agent $u$ for agent $v$ is always equal to the one of agent $v$ for agent $u$.

In the study of hedonic games, several solution concepts are considered important and well investigated. One of the most natural solution concepts is maximum utilitarian, which is so-called a global optimal solution; it is a coalition structure that maximizes the total sum of the utilities of all the agents. The total sum of the utilities is also called social welfare. Another concept of a global optimal solution is maximum egalitarian. It maximizes the minimum utility of an agent among all the agents. That is, it makes the unhappiest agent as happy as possible. Nash-stability, envy-free and max envy-free are more personalized concepts of the solutions. A coalition structure is called Nash-stable if no agent has an incentive to move to another coalition from the current joining coalition. Such an incentive to move to another coalition is also called a deviation. Agent $u$ feels envious of $v$ if $u$ can increase his/her utility by exchanging the coalitions of $u$ and $v$. A coalition structure is envy-free if any agent does not envy any other agent. Furthermore, the best one among the envy-free coalition structures is also meaningful; it is an envy-free coalition structure with maximum social welfare. Some other concepts are also considered, though we focus on these concepts in this paper.

Of course, it is not trivial to find a coalition structure satisfying above mentioned solution concepts. Ballester studies the computational complexity for finding coalition structures of several concepts including the above mentioned ones 5. More precisely, he shows that determining whether there is a Nash stable, an individually stable, and a core stable coalition structure is NP-complete. In [24, Sung and Dimitrov show that the same results hold for ASHG. Aziz et al. investigate the computational complexity for many concepts including the above five solution concepts [4]. In summary, ASHG is unfortunately NP-hard for the above five solution concepts. These hardness results are however proven without any assumption about graph structures. For example, some of the proofs suppose that graphs are weighted complete graphs. This might be a problem, because graphs appearing in ASHGs for practical applications are so-called social networks; they are far from weighted complete graphs and known to be rather sparse or tree-like [1, 11]. What if we restrict the input graphs of ASHG to sparse graphs? This is the motivation of this research.

In this paper, we investigate the computational complexity of ASHG on sparse graphs from the above five solution concepts. The sparsity that we consider in this paper is as follows: graphs with bounded degree, graphs with bounded treewidth and near-planar graphs. The degree is a very natural parameter that characterizes the sparsity of graphs. In social networks, the degree represents the number of friends, which is usually much smaller than the size of network. The treewidth is a parameter that represents how tree-like a graph is. As Adcock, Sullivan and Mahoney pointed out in [1], many large social and information networks have tree-like structures, which implies the significance to investigate the computational complexity of ASHG on graphs with bounded treewidth. Near-planar graphs here are $p$-apex graphs. A graph $G$ is said to be $p$-apex if $G$ becomes planar after deleting $p$ vertices or fewer vertices. Near-planarity is less important than the former two in the context of social networks, though it also has many practical applications such as transportation networks. Note that all of these sparsity concepts are represented by parameters, i.e., treewidth, maximum degree and $p$-apex. In that sense, we consider the parameterized complexity of ASHG of several solution concepts in this paper.

This is not the first work that focuses on the parameterized complexity of ASHG. Peters presents that Nash-stable, Maximum Utilitarian, Maximum Egalitarian and Envy-free coalition structures can be computed in $2^{\mathrm{tw} \Delta^{2}} n^{O(1)}$ time, where tw is the treewidth and $\Delta$ is the maximum degree of an input graph $\sqrt[22]{ }$. In other word, it is fixed parameter tractable (FPT) with respect to treewidth and maximum degree. This implies that if both of the treewidth and the maximum degree are small, we can efficiently find desirable coalition structures. This result raises the following natural question: is finding these desirable coalition structures still FPT when parameterized by either the treewidth or the maximum degree?

This paper answers the question from various viewpoints. Different from the case parameterized by treewidth and maximum degree, the time complexity varies depending on the solution concepts.

Table 1: Complexity of ASHGs

| Concept | Time complexity to compute | Reference |
| :---: | :---: | :---: |
| Nash stable | NP-hard <br> PLS-complete (symm) <br> PLS-complete (symm, $\Delta=7$ ) <br> ${ }^{\text {tw }}{ }^{O(\mathrm{tw})} n$ (symm, FPT by treewidth) | $\begin{array}{r} \frac{24}{14} \\ {[\text { Th } 1} \\ {[\text { Cor } 1]} \\ \hline \end{array}$ |
| Max Utilitarian | $\begin{aligned} & \text { strongly NP-hard (symm) } \\ & \text { strongly NP-hard (symm, 3-apex) } \\ & \text { tw }{ }^{(\text {(tw) } n} \text { (FPT by treewidth) } \end{aligned}$ | $\left[\begin{array}{c} 4 \\ {[\operatorname{Th} 2} \\ \hline 3 \\ \hline \end{array}\right.$ |
| Max Egalitarian | ```strongly NP-hard weakly NP-hard (symm, 2-apex, vc = 4) weakly NP-hard (symm, planar, pw = 4, tw = 2) strongly NP-hard (symm) linear (symm, tree) P (tree) (twW)}\mp@subsup{)}{}{O(tw)}n\mathrm{ (pseudo FPT by treewidth)``` |  |
| Envy-free | trivial | 4 |
| Max Envy-free | weakly NP-hard (symm, planar, $\mathrm{vc}=2$, $\mathrm{tw}=2$ ) strongly NP-hard (symm) <br> linear (symm, tree) | $\begin{aligned} & {\left[\operatorname{Th} \cdot \frac{4}{7}\right.} \\ & {[\operatorname{Th}]} \\ & {[\operatorname{Th} \sqrt{8}} \end{aligned}$ |

For example, we can compute a maximum utilitarian coalition structure in $\mathrm{tw}^{O(\mathrm{tw})} n$ time, whereas computing a maximum egalitarian coalition structure is weakly NP-hard even for graphs with treewidth at most 2. Some other results of ours are summarized in Table 1.1. For more details, see Section 1.1 Also some related results are summarized in Section 1.2.

### 1.1 Our contribution

We first study (symmetric) Nash stable on bounded degree graphs. We show that the problem is PLS-complete even on graphs with maximum degree 7. PLS is a complexity class of a pair of an optimization problem and a local search for it. It is originally introduced to capture the difficulty of finding a locally optimal solution of an optimization problem. In the context of hedonic games, a deviation corresponds to an improvement in local search, and thus PLS or PLS-completeness is also used to model the difficulty of finding a stable solution.

We next show that Max Utilitarian is strongly NP-hard on 3-apex graphs, whereas it can be solved in time $\mathrm{tw}^{O(\mathrm{tw})} n$, and hence it is FPT when parameterized by treewidth tw. For Max Envy-Free, we show that the problem is weakly NP-hard on series-parallel graphs with vertex cover number at most 2 whereas finding an envy-free partition is trivial 4 .

Finally, we investigate the computational complexity of Max Egalitarian. We show that Max Egalitarian is weakly NP-hard on 2-apex graphs with vertex cover number at most 4 and planer graphs with pathwidth at most 4 and treewidth at most 2. Moreover, we show that Max Egalitarian and Max Envy-free are strongly NP-hard even if the preferences are symmetric. In contrast, an egalitarian and envy-free partition with maximum social welfare can be found in linear time on trees if the preferences are symmetric. Moreover, Max Egalitarian can be computed in polynomial time even if the preferences are asymmetric. In the end of this paper, we give a pseudo FPT algorithm when parameterized by treewidth.

### 1.2 Related work

The coalition formation game is first introduced by Dreze and Greenber 10 in the field of Economics. Based on the concept of coalition formation games, Banerjee, Konishi and Sönmez 6 and Bogomolnaia and Jackson 7 study some stability and core concepts on hedonic games. For the computational complexity on hedonic games, Ballester shows that finding several coalition structures including Nash stable, core stable, and individually stable coalition structures is NP-complete [5]. For ASHGs, Aziz et al. investigate the computational complexity of finding several desirable coalition structures 4 . Gairing and Savani $\sqrt{14}$ show that computing a Nash stable coalition structure is PLS-complete in symmetric AGHGs whereas Bogomolnaia and Jackson [7] prove that a Nash stable coalition structure always exists. In [22, Peters designs parameterized algorithms for computing some coalition structures on hedonic games with respect to treewidth and maximum degree.

## 2 Preliminaries

In this paper, we use the standard graph notations. For $G=(V, E)$, we define $n=|V|$ and $m=|E|$. For $V^{\prime} \subseteq V$, we denote by $G\left[V^{\prime}\right]$ the subgraph of $G$ induced by $V^{\prime}$. We denote the closed neighbourhood and the open neighbourhood of a vertex $v$ by $N[v]$ and $N(v)$, respectively. The degree of $v$ is denoted by $d(v)$. Moreover, the maximum degree of $G$ is denoted by $\Delta(G)$. For simplicity, we sometimes omit the subscript $G$.

### 2.1 Hedonic game

An additively separable hedonic game (ASHG) is defined on a directed edge-weighted graph $G=$ $(V, E, w)$. Each vertex $v \in V$ is called an agent. The weight of an edge $e=(u, v)$, denoted by $w_{e}$ or $w_{u v}$, represents the valuation of $u$ to $v$. An ASHG is said to be symmetric if $w_{u v}=w_{v u}$ holds for any pair of $u$ and $v$. Any symmetric ASHG can be defined on an undirected edge-weighted graph. We denote an undirected edge by $\{u, v\}$. Note that any edge of weight 0 is removed from a graph.

Let $\mathcal{P}$ be a partition of $V$. Then $C \in \mathcal{P}$ is called a coalition. We denote by $C_{u} \in \mathcal{P}$ the coalition to which an agent $u \in V$ belongs under $\mathcal{P}$, and by $E\left(C_{u}\right)$ the set of edges $\left\{(u, v) \cup(v, u) \in E \mid v \in C_{u}\right\}$. In ASHGs, the utility of an agent $u$ under $\mathcal{P}$ is defined as $\mathrm{u}_{\mathcal{P}}(u)=\sum_{v \in N(u) \cap C_{u}} w_{u v}$, which is the sum of weights of edges from $u$ to other agents in the same coalition. Also, the social welfare of $\mathcal{P}$ is defined as the sum of utilities of all agents under $\mathcal{P}$. Note that the social welfare equals to exactly twice the sum of weights of edges in coalitions.

Next, we define several concepts of desirable solution in ASHGs.
Definition 1 (Nash-stable). A partition $\mathcal{P}$ is Nash-stable if there exists no agent $u$ and coalition $C^{\prime} \neq C_{u}$ containing $u$, possibly empty, such that

$$
\sum_{v \in N(u) \cap C_{u}} w_{u v}<\sum_{v \in N(u) \cap C^{\prime}} w_{u v}
$$

As an important fact, in any symmetric ASHG, a partition with maximum social welfare is Nash-stable by using the potential function argument 7].
Proposition 1. In any symmetric $A S H G$, a partition with maximum social welfare is Nash-stable.
Thus, if we can compute a partition with maximum social welfare in a symmetric ASHG, then we also obtain a Nash-stable partition.

Definition 2 (Envy-free). We say an agent $u_{1} \in C_{u_{1}}$ envies $u_{2} \in C_{u_{2}}$ if the following inequality holds:

$$
\sum_{v \in N\left(u_{1}\right) \cap C_{u_{1}}} w_{u_{1} v}<\sum_{v \in N\left(u_{1}\right) \cap\left(C_{\left.u_{2} \backslash\left\{u_{2}\right\} \cup\left\{u_{1}\right\}\right)} w_{u_{1} v} .\right.}
$$

That is, $u_{1}$ envies $u_{2}$ if the utility of $u_{1}$ increases by replacing $u_{2}$ by $u_{1}$. A partition $\mathcal{P}$ is envy-free if any agent does not envy an agent.

Nash-stable, Envy-free, Max Envy-free, Max Utilitarian, and Max Egalitarian are the following problems: Given a weighted graph $G=(V, E, w)$, find a Nash-stable partition, an envy-free partition, an envy-free partition with maximum social welfare, a maximum utilitarian partition, and a maximum egalitarian partition, respectively.

### 2.2 Graph classes

A planar graph is a graph that can be drawn on the plane in such a way that its edges intersect only at their endpoints. For $p \geq 1$, a $p$-apex graph is a graph that can be planar by removing $p$ vertices or fewer vertices from it. Note that a planar graph is a $p$-apex graph for any $p \geq 1$. A graph $G$ is called series-parallel if every 2-connected component of $G$ can be constructed by applying series operation and parallel operation compositions recursively: The series operation entails subdividing an edge by a new vertex (replacing an edge by two edges in series). The parallel operation entails replacing an edge by two edges in parallel. It is well-known that a series-parallel graph is planar, and the class of series-parallel graph is equivalent to graphs with treewidth 2.

### 2.3 Graph parameters and parameterized complexity

For the basic definitions of parameterized complexity, such as the classes FPT and XP, refer to [8].
Definition 3 (Tree decomposition). A tree decomposition of an undirected graph $G=(V, E)$ is defined as a pair $\langle\mathcal{X}, T\rangle$, where $\mathcal{X}=\left\{X_{1}, X_{2}, \ldots, X_{N} \subseteq V\right\}$, and $T$ is a tree whose nodes are labeled by $I \in\{1,2, \ldots, N\}$, such that

1. $\bigcup_{i \in I} X_{i}=V$,
2. For all $\{u, v\} \in E$, there exists an $X_{i}$ such that $\{u, v\} \subseteq X_{i}$,
3. For all $i, j, k \in I$, if $j$ lies on the path from $i$ to $k$ in $T$, then $X_{i} \cap X_{k} \subseteq X_{j}$.

Here, $X_{i}$ is called a bag. The width of a tree decomposition is defined as $\min _{i \in I}\left|X_{i}\right|-1$, that is, minimum size of a bag minus one. Furthermore, the treewidth of $G$, denoted by $\operatorname{tw}(G)$, is minimum possible width of a tree decomposition of $G$. A tree decomposition $\langle\mathcal{X}, T\rangle$ is called a path decomposition if $T$ is a path. The pathwidth of $G$, denoted by $\operatorname{pw}(G)$, is minimum possible width of a path decomposition of $G$.

We introduce a special type of tree decomposition, a nice tree decomposition, introduced by Kloks [18]. It is a special binary tree decomposition which has four types of nodes, named leaf, introduce vertex, forget and join. In [8,9], Cygan et al. added a fifth type, the introduce edge node.

Definition 4 (Nice tree decomposition). A tree decomposition $\langle\mathcal{X}, T\rangle$ is called a nice tree decomposition if it satisfies the following:

1. $T$ is rooted at a designated node $r \in I$ satisfying $\left|X_{r}\right|=0$, called the root node,
2. Each node of the tree $T$ has at most two children,
3. Each node in $T$ has one of the following five types:

- A leaf node $i$ which has no children and its bag $X_{i}$ satisfies $\left|X_{i}\right|=0$,
- An introduce vertex node $i$ has one child $j$ with $X_{i}=X_{j} \cup\{v\}$ for a vertex $v \in V$,
- An introduce edge node $i$ has one child $j$ and labeled with an edge $(u, v) \in E$ where $u, v \in X_{i}=X_{j}$,
- A forget node $i$ has one child $j$ and satisfies $X_{i}=X_{j} \backslash\{v\}$ for a vertex $v \in V$,
- $A$ join node $i$ has two children nodes $j_{1}, j_{2}$ and satisfies $X_{i}=X_{j_{1}}=X_{j_{2}}$.

Any tree decomposition of width $\omega$ can be transformed into a nice tree decomposition of $\omega$ with $O(n)$ nodes in linear time [8].

A vertex cover $S$ is the set of vertices such that every edge has at least one vertex in $S$. The size of minimum vertex cover in $G$ is called vertex cover number, denoted by $\mathrm{vc}(G)$. The following proposition is a well-known relationship between treewidth, pathwidth, and vertex cover number.

Proposition 2. For any graph $G$, it holds that $\mathrm{tw}(G) \leq \mathrm{pw}(G) \leq \mathrm{vc}(G)$.
In (13, Fomin and Thilikos proved that for any planar graph $G, \operatorname{tw}(G) \leq 3.183 \sqrt{n}-1$ and a tree decomposition of such width can be computed in polynomial time. Using this fact, we obtain the following proposition for $p$-apex graphs.

Proposition 3. Let $p$ be some constant. For any p-apex graph $G$, $\operatorname{tw}(G) \leq 3.183 \sqrt{n}+p-1$. Moreover, a tree decomposition of such width can be computed in polynomial time.

Proof. In [13, Fomin and Thilikos proved that for any planar graph $G, \operatorname{tw}(G) \leq 3.183 \sqrt{n}-1$ and a tree decomposition of such width can be computed in polynomial time. Thus, we first guess $p$ vertices such that $G$ becomes planar by deleting them. Since we can check whether a graph is planar in time $O\left(n^{2}\right) \sqrt[17]{ }$, this can be done in polynomial time by using the brute forth. Now, we have a planar graph $G^{\prime}$ obtained from $G$ by deleting such $p$ vertices. Then we compute a tree decomposition of width $\operatorname{tw}\left(G^{\prime}\right) \leq 3.183 \sqrt{n}-1$ in polynomial time. Finally, we add $p$ vertices in $V(G) \backslash V\left(G^{\prime}\right)$ to each bag of a tree decomposition. The width of such a tree decomposition of $G$ is clearly at most $3.183 \sqrt{n}+p-1$.

Proposition 3 implies that there is a $2^{O(\sqrt{n} \log n)}$-time algorithm for any $p$-apex graph if there is a tw ${ }^{O(t w)}$-time or even an $n^{O(t w)}$-time algorithm. Therefore, Max Utilitarian and Max Egalitarian with restricted weights can be solved in time $2^{O(\sqrt{n} \log n)}$ on $p$-apex graphs from Theorems 3 and 10.

### 2.4 Problem list

In this subsection, we list problems used for the proofs in this paper.

- Max $k$-Cut: Given an undirected and edge-weighted graph $G=(V, E, w)$, find a partition $\left(V_{1}, V_{2}, \ldots, V_{k}\right)$ that maximizes $\sum_{u_{1} \in V_{i}, u_{2} \in V_{j}, V_{i} \neq V_{j}} w_{u_{1} u_{2}}$. Max 2-CuT is known as MAXCut.
- $k$-Coloring: Given an undirected graph $G=(V, E)$, determine whether there is a coloring $c: V \rightarrow\{1, \ldots, k\}$ such that $c(u) \neq c(v)$ for every $(u, v) \in E$.
- Partition: Given a finite set of integers $A=\left\{a_{1}, a_{2}, \ldots, a_{n}\right\}$ and $W=\sum_{i=1}^{n} a_{i}$, determine whether there is partition $\left(A_{1}, A_{2}\right)$ of $A$ where $A_{1} \cup A_{2}=A$ and $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=W / 2$.
- 3-Partition: Given a finite set of integers $A=\left\{a_{1}, \ldots, a_{3 n}\right\}$, determine whether there is partition $\left(A_{1}, \ldots, A_{n}\right)$ such that $\left|A_{i}\right|=3$ and $\sum_{a \in A_{i}} a=B$ for each $i$ where $B=\sum_{a \in A} a / n$.


## 3 Nash-stable

Any symmetric ASHG always has a Nash-stable partition by Proposition 1 However, finding a Nash-stable solution is PLS-complete [14. In this section, we prove that Nash-Stable is PLS-complete even on bounded degree graphs.


Figure 1: The constructed graph $G^{\prime}$ in the proof of Theorem 1.

Theorem 1. Symmetric NASH-STABLE is PLS-complete even on graphs with maximum degree $\Delta=7$.

Proof. We give a reduction from Local Max-Cut with flip. Local Max-Cut is a local search problem of Max-Cut. In the flip neighborhood, two solutions are neighbors if one can be obtained from the other by moving one element to the other set. Local Max-Cut with flip is PLS-complete on graphs with maximum degree $\Delta=5$ [12].

Given an edge weighted graph $G=(V, E, w)$, we construct $G^{\prime}=\left(V^{\prime}, E^{\prime}, w^{\prime}\right)$ as follows. Let $W=\sum_{e \in E}\left|w_{e}\right|+1$ and $M=n W+1$. First we set $w_{e}^{\prime}=-w_{e}$ for every $e \in E$. Then we add $P_{A}=\left\{p_{1}^{A}, \ldots, p_{n+1}^{A}\right\}$ and $P_{B}=\left\{p_{1}^{B}, \ldots, p_{n+1}^{B}\right\}$ that form paths of length $n$, respectively. For $1 \leq i \leq n$, we define $w_{p_{i}^{A} p_{i+1}^{A}}^{\prime}=w_{p_{i}^{B} p_{i+1}^{B}}^{\prime}=i(W+1)$ as the weight of $\left\{p_{i}^{A}, p_{i+1}^{A}\right\}$ and $\left\{p_{i}^{B}, p_{i+1}^{B}\right\}$. We connect each $p_{i}^{A}$ and $p_{i}^{B}$ to $v_{i} \in V$ by an edge of weight $W$, respectively. Finally, we connect $p_{n+1}^{A}$ and $p_{n+1}^{B}$ by an edge of weight $-3 M$. Note that $P_{A} \cup P_{B}$ forms a path of length $2 n+1$. We can observe that the degree of a vertex in $V$ is at most 7 and in $P_{A} \cup P_{B}$ is at most 3 .

In any Nash-stable partition in $G^{\prime}, p_{n+1}^{A}$ and $p_{n+1}^{B}$ must be in different coalitions. If not so, the utility of $p_{n+1}^{A}$ is at most $-3 M+n(W+1) W<0$, and it has an incentive to deviate to a singleton because the utility in a singleton is 0 . Moreover, in any Nash-stable partition of $G^{\prime}$, every vertex in $P_{A}$ always belong to the same coalition. Otherwise, there is an agent $p_{i}^{A}$ with utility at most $(i-1)(W+1)+W=i(W+1)-1$. Then the utility of $p_{i}^{A}$ can be increased to at least $i(W+1)$ by deviating to the coalition to which $p_{i+1}^{A}$ belongs. Similarly, every vertex in $P_{B}$ always belong to the same coalition in any Nash-stable partition of $G^{\prime}$. Therefore, in any Nash-stable partition, there exist a coalition $C_{A}$ containing all vertices included in $P_{A}$ and a coalition $C_{B}$ containing all vertices included in $P_{B}$. Furthermore, if a vertex in $V$ does not belong to neither $C_{A}$ nor $C_{B}$, the utility is at most 0 . Since the utility of the vertex in $C_{A}$ or $C_{B}$ is more than 0 , it must belong to either $C_{A}$ or $C_{B}$. Thus, any Nash-stable partition $\mathcal{P}^{*}$ in $G^{\prime}$ has exactly two coalitions $C_{A}$ containing $P_{A}$ and $C_{B}$ containing $P_{B}$.

Here, we can observe that any Nash-stable partition in $G^{\prime}$ is a local optimal solution of Local Max-Cut in $G$. If not so, there is a vertex $v \in V$ that can increase the weight of a cut in $G$ by flipping $v$ from the current set to the other. In $G^{\prime}$, such a vertex deviates to the other coalition because the utility increases. Conversely, we are given a local optimal cut ( $C_{1}, C_{2}$ ) of Local Max-Cut. Then a partition $\left(C_{1} \cup P_{A}, C_{2} \cup P_{B}\right)$ in $G^{\prime}$ is a NASH-Stable partition. This is because any vertex in $P_{A}$ and $P_{B}$ does not deviate from the current coalition. Moreover, suppose that there is a vertex $v$ in $C_{1} \cup C_{2}=V$ deviates to the other coalition to increase the utility. This implies that there is a vertex $v$ in $G$ that increases the weight by flipping $v$. This contradicts the optimality of $\left(C_{1}, C_{2}\right)$.

## 4 Max Utilitarian

Theorem 2. Max Utilitarian is strongly NP-hard on 3-apex graphs even if the preferences are symmetric.

Proof. Let $G=(V, E)$ be an unweighted graph where $|V|=n$ and $|E|=m$. We first confirm that Max 3-Cut is NP-hard for planar graphs. This can be done by a reduction from 3-COLORING on planar graphs which is NP-complete [16]. If an unweighted graph $G$ is 3 -colorable, it is clear that $G$ has a 3 -cut of size $m$ because for every edge, its endpoints have different colors. On the other hand, if an unweighted graph $G$ has a 3 -cut of size $m$, it is obviously 3-colorable.

Then we give a reduction from Max 3-Cut to Max Utilitarian. Given an unweighted planar graph $G=(V, E)$ of an instance of Max 3-Cut, we add three super vertices $S=\left\{s_{1}, s_{2}, s_{3}\right\}$ such that and each $s_{i}$ is connected to all vertices in $G$ by three edges of weight $m$. Moreover, we connect $s_{1}, s_{2}, s_{3}$ to each other by edges of weight -6 mn , and hence $S$ forms a clique. Finally, for each edge $e \in E$, we define the weight $w_{e}=-1$. Let $G^{\prime}$ be the constructed graph.

In the following, we show that there is a 3 -cut of size $k$ in $G$ if and only if there is a partition $\mathcal{P}$ with social welfare $2(3 m n-m+k)$. Given a 3 -cut $\left(V_{1}, V_{2}, V_{3}\right)$ of size $k$ in $G$, we construct a partition $\mathcal{P}=\left\{V_{1} \cup\left\{s_{1}\right\}, V_{2} \cup\left\{s_{2}\right\}, V_{3} \cup\left\{s_{3}\right\}\right\}$. Since the number of edges in $E$ in coalitions is $m-k$, any edge in $G[S]$ is not contained in coalitions, and every edge between $s \in S$ and $v \in V$ is contained in coalitions, the social welfare of $\mathcal{P}$ is $2(3 m n-m+k)$.

Conversely, we are given a partition $\mathcal{P}$ of social welfare $2(3 m n-m+k)$. If a coalition contains at least two vertices in $S$, the social welfare is at most $6 m-6 m n<0$. Thus, each $s_{i}$ does not belong to the same coalition. If there is $v \in V$ that does not belong to a coalition containing $s \in S$, then the social welfare is at most $2 \cdot 3 m(n-1)=6 m n-6 m<2(3 m n-m+k)$. Hence, $V$ must be partitioned into three sets adjacent to either $s_{1}, s_{2}$ or $s_{3}$. Since the social welfare of $\mathcal{P}$ is $2(3 m n-m+k)$, every edge between $s \in S$ and $v \in V$ is contained in a coalition, and the weight of an edge in $E$ is -1 , the number of edges in $E$ in coalitions is $m-k$. This implies that there is a 3 -cut of size $k$.

Theorem 3. Given a tree decomposition of width tw, Max Utilitarian can be solved in time $\mathrm{tw}^{O(\mathrm{tw})} n$.

Proof. Our algorithm is based on dynamic programming on a tree decomposition for connectivity problems such as Steiner tree 8]. In our dynamic programming, we keep track of all the partitions in each bag.

We define the recursive formulas for computing the social welfare of each partition $\mathcal{P}$ in the subgraph based on a subtree of a tree decomposition. Let $\mathcal{P}_{i}$ be a partition of $X_{i}$. We denote by $A_{i}\left[\mathcal{P}_{i}\right]$ the maximum social welfare in the subgraph $G_{i}$ such that $X_{i}$ is partitioned into $\mathcal{P}_{i}$. Notice that $A_{r}[\emptyset]$ in root node $r$ is the maximum social welfare of $G$. We denote a parent node by $i$ and its child node by $j$. For a join node, we write $j_{1}$ and $j_{2}$ to denote its two children.

Leaf node: In leaf nodes, we set $A_{i}[\emptyset]=0$.

Introduce vertex $v$ node: In an introduce vertex $v$ node $i$, let $C_{v} \in \mathcal{P}_{i}$ be the coalition including $v$. We notice that the social welfare is increased by edges between $v$ and vertices in the coalition including $v$. Thus, the recursive formula is defined as: $A_{i}\left[\mathcal{P}_{i}\right]=A_{j}\left[\mathcal{P}_{j}\right]+\sum_{u \in N(v) \cap C_{v}} w_{u v}+$ $\sum_{u \in N(v) \cap C_{v}} w_{v u}$ where $\mathcal{P}_{j}=\mathcal{P}_{i} \backslash\left\{C_{v}\right\} \cup\left\{C_{v} \backslash\{v\}\right\}$.

Forget $v$ node: In a forget $v$ node, we only take a partition with maximum social welfare when we forget $v$ because $v$ does not affect the social welfare hereafter. Thus, the recursive formula is defined as: $A_{i}\left[\mathcal{P}_{i}\right]=\max _{\mathcal{P}_{j} \in \mathcal{D}_{j}} A_{j}\left[\mathcal{P}_{j}\right]$ where $\mathcal{D}_{j}=\left\{\mathcal{P}_{j} \mid \mathcal{P}_{j} \backslash\left\{C_{v}\right\} \cup\left\{C_{v} \backslash\{v\}\right\}=\mathcal{P}_{i}\right\}$.


Figure 2: The constructed graph $H$.

Join node: A join node $i$ has two child nodes $j_{1}, j_{2}$ where $X_{i}=X_{j_{1}}=X_{j_{2}}$. The social welfare of each partition $X_{i}$ is the sum of the corresponding partition of $X_{j_{1}}=X_{j_{2}}$. Therefore, the recursive formula for a join node is defined as: $A_{i}\left[\mathcal{P}_{i}\right]=A_{j_{1}}\left[\mathcal{P}_{i}\right]+A_{j_{2}}\left[\mathcal{P}_{i}\right]-\sum_{C \in \mathcal{P}_{i}} \sum_{u, v \in C}\left(w_{u v}+w_{v u}\right)$. The last term means subtracting the double counting of edges.

Because the size of each DP table is $\mathrm{tw}^{O(\mathrm{tw})}$, we can compute the recursive formulas in time $\mathrm{tw}^{O(\mathrm{tw})}$. As the result, the total running time is $\mathrm{tw}^{O(\mathrm{tw})} n$.

By Proposition 1. symmetric Nash-stable is also solvable in time $\mathrm{tw}^{O(\mathrm{tw})} n$.
Corollary 1. Given a tree decomposition of width tw, symmetric Nash-Stable can be solved in time $\mathrm{tw}^{O(\mathrm{tw})} n$.

## 5 Max Envy-free and Max Egalitarian

In [4], Aziz et al. show that finding an envy-free partition is trivial because a partition of singletons is envy-free. However, finding a maximum envy-free partition is much more difficult than finding an envy-free partition.

Theorem 4. Max Envy-free is weakly NP-hard on series-parallel graphs of vertex cover number 2 even if the preferences are symmetric.

Proof. We give a reduction from Partition, which is weakly NP-complete 15. Without loss of generality, we suppose $a_{1} \leq a_{2} \leq \ldots \leq a_{n}$ and $a_{n}<W / 2$.

Given a set of positive integers $A=\left\{a_{1}, a_{2}, \ldots, a_{n}\right\}$, we build the corresponding vertex set $V_{A}=\left\{v_{a_{1}}, v_{a_{2}}, \ldots, v_{a_{n}}\right\}$. Then we construct an edge-weighted complete bipartite graph $K_{2, n}=$ $\left(V_{A} \cup U, E\right)$ where $U=\left\{u_{1}, u_{2}\right\}$. For each edge $\left\{v_{a_{i}}, u_{j}\right\} \in E$, we set the weight $w_{v_{a_{i}} u_{j}}=a_{i}$. Finally, we add $e_{u}=\left\{u_{1}, u_{2}\right\}$ of weight $-2 W-1$. Let $H=\left(V_{A} \cup U, E \cup\left\{e_{u}\right\}\right)$ be the constructed graph. Note that $H$ is a series parallel graph and $\mathrm{vc}(H)=2$ (see Fig. 2).

We show that an instance of Partition is a yes-instance if and only if there is an env-free partition with social welfare at least $2 W$ in $H$.

Given a partition $\left(A_{1}, A_{2}\right)$ of $A$ such that $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=W / 2$, let $V_{A_{1}}$ and $V_{A_{2}}$ be the corresponding vertex set to $A_{1}$ and $A_{2}$, respectively. In short, $V_{A_{i}}=\left\{v_{a} \mid a \in A_{i}\right\}$ for $i \in\{1,2\}$. Let $\mathcal{P}=\left\{V_{A_{1}} \cup\left\{u_{1}\right\}, V_{A_{2}} \cup\left\{u_{2}\right\}\right\}$ be a partition in $H$. By the definition of $H$, we have $\mathrm{u}_{\mathcal{P}}\left(v_{a}\right)=a$ for every $v_{a} \in V_{A}$ and $u_{\mathcal{P}}(u)=W / 2$ for every $u \in U$. Let $C_{i}=V_{A_{i}} \cup\left\{u_{i}\right\}$ for $i \in\{1,2\}$. For an agent $v_{a} \in C_{i}$, consider $C_{i}^{\prime}=C_{j} \backslash\{w\} \cup\left\{v_{a}\right\}$ for any $w \in C_{j} \neq C_{i}$. In this case, the utility of $v_{a}$ is at most $a$. Moreover, consider $C_{1}^{\prime}=C_{2} \backslash\{w\} \cup\left\{u_{1}\right\}$ for any $w \in C_{2}$. Then the utility of $u_{1}$ is at most $W / 2$. Therefore, $\mathcal{P}$ is envy-free. Also, the social welfare of $\mathcal{P}$ is $W / 2+W / 2+\sum_{a \in A} a=2 W$.

Conversely, we are given an envy-free partition $\mathcal{P}$ with social welfare at least $2 W$ in $H$. If $\mathcal{P}$ has a coalition that contains both $u_{1}$ and $u_{2}$, the social welfare of $\mathcal{P}$ is strictly less than $2 W$.


Figure 3: The constructed graph $H^{\prime}$.


Figure 4: The constructed graph $H^{\prime \prime}$.

Thus, there are two coalitions $C_{1}, C_{2}$ in $\mathcal{P}$ such that $u_{1} \in C_{1}$ and $u_{2} \in C_{2}$. For each $a \in A$, one of $v_{a} \in C_{1}, v_{a} \in C_{2}$, and $v_{a} \in C^{\prime}$ holds where $C^{\prime} \subseteq V_{A}$ does not contain $u_{1}$ and $u_{2}$. This implies that the social welfare of $\mathcal{P}$ is at most $2 W$ because at most one of $\left\{v_{a}, u_{1}\right\}$ and $\left\{v_{a}, u_{2}\right\}$ contributes to the social welfare. If there is an agent $v_{a}$ in neither $C^{\prime}$, the social welfare of $\mathcal{P}$ is strictly less than $2 W$. Therefore, it holds that either $v_{a} \in C_{1}$ or $v_{a} \in C_{2}$ for each $v_{a} \in V_{A}$ in $\mathcal{P}$. Suppose that $u_{\mathcal{P}}\left(u_{1}\right)>\mathrm{u}_{\mathcal{P}}\left(u_{2}\right)$. Then $u_{2}$ envies $u_{1}$ by the definition of $H$. This contradicts the assumption. Now, we have $\mathrm{u}_{\mathcal{P}}\left(u_{1}\right)=\mathrm{u}_{\mathcal{P}}\left(u_{2}\right)=W / 2$. Let $A_{i}=\left\{a \in A \mid v_{a} \in N\left(u_{i}\right) \subseteq V_{A}\right\}$ for $i \in\{1,2\}$. Then a partition $\left\{A_{1}, A_{2}\right\}$ satisfies that $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=W / 2$.

Next, we show that Max Egalitarian is weakly NP-hard on series-parallel graphs of pathwidth 4. Note that the class of series-parallel graph is equivalent to graphs with treewidth 2.

Theorem 5. In the symmetric hedonic games, Max Egalitarian is weakly NP-hard on seriesparallel graphs of pathwidth 4 even if the preferences are symmetric.
Proof. We give a reduction from Parition as in the proof of Theorem 4, though we adopt a bit different graph from $H$. For each $v_{a_{i}}$ in $V_{A}$, we create two copies of $v_{a_{i}}$, denoted by $v_{a_{i}}^{\prime}$ and $v_{a_{i}}^{\prime \prime}$ respectively, such that they form a clique. For each $a_{i} \in A$, we define the weights $w_{v_{a_{i}} v_{a_{i}}^{\prime}}=w_{v_{a_{i}} v_{a_{i}}^{\prime \prime}}=W / 4-a_{i} / 2$ and the weight $w_{v_{a_{i}} v_{a_{i}}^{\prime \prime}}=W / 4+a_{i} / 2$. Without loss of generality, we can assume that each $a_{i}$ is even. Let $V_{A}^{\prime}=\left\{v_{a}^{\prime} \mid a \in A\right\}$ and $V_{A}^{\prime \prime}=\left\{v_{a}^{\prime \prime} \mid a \in A\right\}$ and let $H^{\prime}$ be the constructed graph (see Fig. 3). For the graph $H^{\prime}$, if we set $X_{i}=\left\{u_{1}, u_{2}, v_{a_{i}}, v_{a_{i}}^{\prime}, v_{a_{i}}^{\prime \prime}\right\}$ for each $a_{i}$ and connect $X_{i}$ and $X_{i+1}$ by an edge, we can construct a path decomposition of width at most 4. Also, it is easy to show that $H^{\prime}$ is a series-parallel graph, and hence the treewidth of $H^{\prime}$ is 2 .

In the following, we show that there exists a partition $\left(A_{1}, A_{2}\right)$ such that $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=$ $W / 2$ if and only if there exists a partition $\mathcal{P}^{\prime}$ in $H^{\prime}$ such that $\min _{v \in V\left(H^{\prime}\right)} \mathrm{u}_{\mathcal{P}^{\prime}}(v)=W / 2$.

Given a partition $\left(A_{1}, A_{2}\right)$ of $A$ such that $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=W / 2$, we set $V_{A_{i}}=\left\{v_{a} \in\right.$ $\left.V_{A} \mid a \in A_{i}\right\}, V_{A_{i}}^{\prime}=\left\{v_{a}^{\prime} \in V_{A}^{\prime} \mid a \in A_{i}\right\}$, and $V_{A_{i}}^{\prime \prime}=\left\{v_{a}^{\prime \prime} \in V_{A}^{\prime \prime} \mid a \in A_{i}\right\}$ for $i \in\{1,2\}$. Let $\mathcal{P}=\left\{V_{A_{1}} \cup V_{A_{1}}^{\prime} \cup V_{A_{1}}^{\prime \prime} \cup\left\{u_{1}\right\}, V_{A_{2}} \cup V_{A_{2}}^{\prime} \cup V_{A_{2}}^{\prime \prime} \cup\left\{u_{2}\right\}\right\}$ be a partition in $H^{\prime}$. By the definition of $H^{\prime}$, we have $u_{\mathcal{P}}(v)=W / 2$ for every $v \in V\left(H^{\prime}\right)$.

Conversely, we are given a partition $\mathcal{P}$ such that $\min _{v \in V\left(H^{\prime}\right)} \mathrm{u}_{\mathcal{P}^{\prime}}(v)=W / 2$. If $\mathcal{P}$ has a coalition that contains both $u_{1}$ and $u_{2}$, the utilities of $u_{1}$ and $u_{2}$ are less than 0 . Thus, we suppose that there are two coalitions $C_{1}, C_{2}$ in $\mathcal{P}$ such that $u_{1} \in C_{1}$ and $u_{2} \in C_{2}$. For each $a \in A$, if $v_{a}, v_{a}^{\prime}$, and $v_{a}^{\prime \prime}$ belong to a different coalition from the other two, the utility of each is strictly less than $W / 2$. Moreover, if $v_{a} \in V_{A}$ belongs to neither $C_{1}$ nor $C_{2}$, $\mathrm{u}_{\mathcal{P}}\left(v_{a}\right)<W / 2$. Thus, $v_{a}, v_{a}^{\prime}$, and $v_{a}^{\prime \prime}$ belong to the same coalition of either $C_{1}$ or $C_{2}$.

Since the utilities of $u_{1}$ and $u_{2}$ are $W / 2$, it holds that $\sum_{v \in N\left(u_{1}\right)} w_{u_{1} v}=\sum_{v \in N\left(u_{2}\right)} w_{u_{2} v}=W / 2$. Therefore, if we set $A_{i}=\left\{a \in A \mid v_{a} \in N\left(u_{i}\right)\right\},\left(A_{1}, A_{2}\right)$ is a partition satisfying $\sum_{a \in A_{1}} a=$ $\sum_{a \in A_{2}} a=W / 2$.

Note that the pathwidth and the treewidth of $H^{\prime}$ are bounded, but the vertex cover number is not bounded. We can similarly show that Max Egalitarian is also weakly NP-hard on bounded vertex cover number graphs by using the reduced graph $H^{\prime \prime}$ in Fig. 4

Theorem 6. Max Egalitarian is weakly NP-hard on 2-apex graphs of vertex cover number 4 even if the preferences are symmetric.

Proof. We give a reduction from Parition as in the proof of Theorem4. Without loss of generality, we suppose $a_{n}<W / 2$. To show this, we modify the graph $H^{\prime}$ in Theorem 4 . We add two super vertices $u_{3}$ and $u_{4}$ connecting to all vertices in $V_{A}$, respectively. For each edge $\left\{a_{i}, u_{j}\right\}$ where $i \in\{1, \ldots, n\}$ and $j \in\{3,4\}$, we set the weight $w_{a_{i} u_{j}}=W / 2-a_{i}$. Let $H^{\prime \prime}$ be the constructed graph (see Fig. 4). We can observe that $H^{\prime \prime}$ is a 2-apex graph because a graph obtained from $H^{\prime \prime}$ by deleting $u_{3}, u_{4}$ is planar. Moreover, $\left\{u_{1}, u_{2}, u_{3}, u_{4}\right\}$ is a vertex cover of size four in $H^{\prime \prime}$.

Then, we show that there exists a partition $\left(A_{1}, A_{2}\right)$ such that $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=W / 2$ if and only if there exists a partition $\mathcal{P}$ in $H^{\prime \prime}$ such that $\min _{v \in V\left(H^{\prime}\right)} \mathrm{u}_{\mathcal{P}}(v)=W / 2$.

Given a partition $\left(A_{1}, A_{2}\right)$ of $A$ such that $\sum_{a \in A_{1}} a=\sum_{a \in A_{2}} a=W / 2$, we set $V_{A_{i}}=\left\{v_{a} \in\right.$ $\left.V_{A} \mid a \in A_{i}\right\}$ for $i \in\{1,2\}$. Let $\mathcal{P}=\left\{V_{A_{1}} \cup\left\{u_{1}, u_{3}\right\}, V_{A_{2}} \cup\left\{u_{2}, u_{4}\right\}\right\}$ be a partition in $H^{\prime \prime}$. By the definition of $H^{\prime}$, we have $\mathrm{u}_{\mathcal{P}}(v)=W / 2$ for every $v \in V\left(H^{\prime}\right)$. For $j \in\{1,2\}$, it holds that $u_{\mathcal{P}}\left(u_{j}\right)=\sum_{a \in A_{j}} a=W / 2$. Since $a_{n}<W / 2$, it holds that $\left|A_{1}\right|,\left|A_{2}\right| \geq 2$ and then we have $u_{\mathcal{P}}\left(u_{j}\right) \geq W / 2$. Moreover, since any $v_{a}$ has either $u_{1}$ and $u_{3}$ or $u_{2}$ and $u_{4}$ as neighbors, $u_{\mathcal{P}}\left(v_{a}\right)=W / 2$ holds.

Conversely, we are given a partition $\mathcal{P}$ such that $\min _{v \in V\left(H^{\prime \prime}\right)} \mathrm{u}_{\mathcal{P}}(v)=W / 2$. If $\mathcal{P}$ has a coalition that contains both $u_{1}$ and $u_{2}$, the utilities of $u_{1}$ and $u_{2}$ are less than 0 . We suppose that there are two coalitions $C_{1}, C_{2}$ in $\mathcal{P}$ such that $u_{1} \in C_{1}$ and $u_{2} \in C_{2}$.

Since the utilities of $u_{1}$ and $u_{2}$ are $W / 2$, it holds that $\sum_{v \in N\left(u_{1}\right)} w_{u_{1} v}=\sum_{v \in N\left(u_{2}\right)} w_{u_{2} v}=W / 2$. Therefore, if we set $A_{i}=\left\{a \in A \mid v_{a} \in N\left(u_{i}\right)\right\},\left(A_{1}, A_{2}\right)$ is a partition satisfying $\sum_{a \in A_{1}} a=$ $\sum_{a \in A_{2}} a=W / 2$. Note that $N\left(u_{1}\right)=N\left(u_{2}\right) \subseteq V_{A}$.

Aziz et al. show that asymmetric Max Egalitarian is strongly NP-hard [4. We show that symmetric Max Envy-free and symmetric Max Egalitarian remain to be strongly NP-hard. To show this, we give a reduction from 3-Partition, which is strongly NP-complete 15 .

Theorem 7. Max Envy-free and Max Egalitarian are strongly NP-hard even if the preferences are symmetric.

Proof. We first explain a reduction for Max Envy-free. For $H$ in Theorem 4, we set $n=3 m$. Then we add $m-2$ super vertices $\left\{u_{3}, \ldots, u_{m}\right\}$ connecting to every vertex $v_{a_{i}}$ in $V_{A}$ by an edge of weight $a_{i}$. Moreover, we connect $u_{p}$ and $u_{q}$ with an edge of weight $-m B$ for $p, q \in\{1, \ldots, m\}$ and $p \neq q$ where $B=\sum_{a \in A} a / n$. Note that $\left\{u_{1}, \ldots, u_{m}\right\}$ forms a clique. The number of vertices in the constructed graph is $m+3 m=4 m$. It is easily seen that there is a partition $\left(A_{1}, \ldots, A_{m}\right)$ such that $\left|A_{i}\right|=B$ for each $i$ if and only if there is an envy-free partition with social welfare at least $2 m B$ in $H$ as in the proof of Theorem 4.

Next, we explain a reduction for Max Egalitarian. For $H^{\prime}$ in Theorem5, we set $n=3 m$. We create $m$ super vertices $\left\{u_{3}, \ldots, u_{m}\right\}$ connecting to every vertex $v_{a_{i}}$ in $V_{A}$ by an edge of weight $a_{i}$ in $H^{\prime}$. Moreover, we connect $u_{p}$ and $u_{q}$ with an edge of weight $-m B$ for $p, q \in\{1, \ldots, m\}$ and $p \neq q$. Finally, we change the weights of $\left\{v_{a_{i}}, v_{a_{i}}^{\prime}\right\}$ and $\left\{v_{a_{i}}, v_{a_{i}}^{\prime}\right\}$ to $B / 2-a_{i} / 2$ and the weight of $\left\{v_{a_{i}}^{\prime}, v_{a_{i}}^{\prime \prime}\right\}$ to $B / 2+a_{i} / 2$. Then we can observe that there is a partition $\left(A_{1}, \ldots, A_{m}\right)$ such that $\left|A_{i}\right|=B$ for each $i$ if and only if there is a partition partition in $H^{\prime}$ such that $\min _{v \in V\left(H^{\prime}\right)} \mathrm{u}_{\mathcal{P}^{\prime}}(v)=B$ as in the proof of Theorem 5

Since $\operatorname{tw}(G) \leq \mathrm{vc}(G)$, Max Envy-Free is weakly NP-hard on graphs of $\operatorname{tw}(G)=2$ by Theorem 4. Also, Max Egalitarian is weakly NP-hard on graphs of $\operatorname{tw}(G)=2$ by Theorem 5 . However, we show that symmetric Max Envy-Free and symmetric Max Egalitarian on trees, which are of treewidth 1, are solvable in linear time. Indeed, we can find an envy-free and maximum egalitarian
partition with maximum social welfare. Such a partition consists of connected components of a forest obtained by removing all negative edges from an input tree.

Theorem 8. Symmetric Max Envy-free and symmetric Max Egalitarian are solvable in linear time on trees.

Note that linear-time solvability does not hold for asymmetric cases, though asymmetric Max Egalitarian on trees can be solved in near-linear time.

Theorem 9. Max Egalitarian can be solved in time $O(n \log W)$ on trees.
Proof. In this proof, we design an algorithm for Max Egalitarian on trees with self-loops, which is a slightly wider class of graphs. Given a tree $T$ with self-loops and a non-negative value $W$, our algorithm determines whether there exists a partition of $T$ such that the utility of every agent is at least $W$ in linear time. The idea is that we can immediately answer "No" by focusing on a leaf, or we can reduce the given $T$ to a smaller tree $T^{\prime}$ with self-loops whose answer is equivalent to the one for $T$.

Given a tree $T$, we consider a leaf $u$ and its adjacent vertex $v$. Let $w(v), w(u, v)$ and $w(v, u)$ be the weights of self-loop of $v$, edges $(u, v)$ and $(v, u)$, respectively. We consider the following four cases: (i) $w(u)<W$ and $w(u)+w(u, v)<W$ hold, (ii) $w(u)<W$ and $w(u)+w(u, v) \geq W$ hold, (iii) $w(u) \geq W$ and $w(u)+w(u, v)<W$ hold, and (iv) $w(u) \geq W$ and $w(u)+w(u, v) \geq W$ hold. In case (i), $u$ can be isolated or can be with $v$, but in any cases, $u$ 's utility is smaller than $W$; the answer is obviously "No". In case (ii), in order to give $u$ utility at least $W, u$ and $v$ should belong to a same coalition, which implies that $v$ receives utility $w(v, u)$. This can be interpreted that $u$ is contracted into $v$ and the weight of self-loop $(v, v)$ is updated to $w(v)+w(v, u)$. In case (iii), in order to guarantee at least $W$ utility for $u, u$ should be isolated. Then, we simply consider the problem for $T^{\prime}$ obtained from $T$ by deleting $u$. In case (iv), $u$ can have utility at least $W$ whichever $u$ belongs to a same coalition with $v$. We then consider two subcases: (iv-1) $w(v, u)>0$ and (iv-2) $w(v, u)<0$ (if $w(v, u)=0, u$ does not affect the partition. We can ignore $v$ ). If (iv-1), $v$ can reserve utility $w(v, u)$ by belonging to a same coalition with $u$; we can apply the same argument with (ii). If (iv-2), it is better that $v$ does not belong to a same coalition with $u$; we can apply the same argument with (iii).

By the above observation, we can immediately say "No", or obtain $T^{\prime}$ with one smaller vertices. Since the above check can be done in $O(1)$, the decision problem can be done in $O(n)$ time, where $n$ is the number of vertices. By applying the binary search, we can obtain a maximum egalitarian partition.

Theorems 5 and 6 mean that Max Egalitarian is weakly NP-hard even on bounded treewidth graphs. On the other hand, we show that there is a pseudo FPT algorithm for Max Egalitarian when parameterized by treewidth.
Theorem 10. Given a tree decomposition of width tw, Max Egalitarian can be solvable in time $(\mathrm{tw} W)^{O(\mathrm{tw})} n$ where $W=\max _{u \in V} \sum_{v \in N(u)}\left|w_{u v}\right|$.
Proof. Let $V_{i}$ be the set of vertices in $X_{i}$ or the descent of $X_{i}$ on a tree decomposition. Then we define DP tables of our dynamic programming.

Let $\mathcal{P}_{i}$ be a partition of $X_{i}$ and $\mathbf{u}_{i}$ be a $\left|X_{i}\right|$-dimensional vector whose elements take from $-W$ to $W$, called a utility vector of $X_{i}$. For $v \in X_{i}$, the element $\mathbf{u}_{i}(v)$ represents the utility of $v$ in $G\left[V_{i}\right]$. Finally, we define $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]$ for each bag $X_{i}$ by using $\mathcal{P}_{i}$ and $\mathbf{u}_{i}$ as the maximum minimum utility of an agent in $V_{i} \backslash X_{i}$ in $G\left[V_{i}\right]$. The value of $A_{r}[\emptyset, \emptyset]$ is an optimal value for Max Egalitarian in $G$. In the following, we define the recursive formulas for computing $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]$ on a nice tree decomposition.

Leaf node: We initialize DP tables for each leaf node $i$ as $A_{i}[\emptyset, \emptyset]=W+1$. Note that the maximum minimum utility is at most $W$ and once we execute the recursive formula in a forget node, $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]$ becomes at most $W$.

Introduce vertex $v$ node: Let $C_{v} \in \mathcal{P}_{i}$ be a coalition that contains $v$ in an introduce $v$ node $i$. Note that $C_{v}$ may contain only $v$, that is, $C_{v}=\{v\}$. In an introduce $v$ node, an agent $v$ is added to a coalition. This changes the utilities of agents in $C_{v}$. Also, the utility of $v$ in $G\left[V_{i}\right]$ is the sum of weight of edges between $v$ and agents in $C_{v}$. Since every agent in $X_{j}$ also appears in $X_{i}$, the maximum minimum utility of an agent in $V_{i} \backslash X_{i}$ in $G\left[V_{i}\right]$ does not change. Therefore, we define the recursive formula as follows: $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]=A_{j}\left[\mathcal{P}_{j}, \mathbf{u}_{j}\right]$, where $\mathcal{P}_{j}=\mathcal{P}_{i} \backslash\left\{C_{v}\right\} \cup\left\{C_{v} \backslash\{v\}\right\}, \mathbf{u}_{j}(u)=\mathbf{u}_{i}(u)-w_{u v}$ for $u \in C_{v} \backslash\{v\}, \mathbf{u}_{j}(u)=\mathbf{u}_{i}(u)$ for other $u$ 's in $X_{i} \backslash\{v\}$, and $\sum_{u \in N(v) \cap C_{v}} w_{v u}=\mathbf{u}_{i}(v)$. Otherwise, we define $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]=-\infty$ as an invalid case.

Forget $v$ node: In a forget $v$ node, if a vertex $v$ is forgotten, it never appears in $X_{i}$ and its ancestors on the decomposition tree. This implies that the utility of $v$ does not change hereafter. Namely, the maximum minimum utility among forgotten agent is stored in $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]$ in some sense. Thus what we need to do here is to update the minimum by comparing the previous maximum minimum utility with the utility of the newly forgotten agent, which can be the new minimum. Taking the maximum among $\mathcal{P}_{j}$ and $\mathbf{u}_{j}$, this can be interpreted as the following recursive formula:

$$
A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]=\max _{\mathcal{P}_{j}, \mathbf{u}_{j}} \min \left\{A_{j}\left[\mathcal{P}_{j}, \mathbf{u}_{j}\right], \mathbf{u}_{j}(v)\right\}
$$

where $\mathbf{u}_{j}(u)=\mathbf{u}_{i}(u)$ for $u \in X_{i}$ and $\mathcal{P}_{j} \backslash\left\{C_{v}\right\} \cup\left\{C_{v} \backslash\{v\}\right\}=\mathcal{P}_{i}$. The condition $\mathcal{P}_{j} \backslash\left\{C_{v}\right\} \cup\left\{C_{v} \backslash\right.$ $\{v\}\}=\mathcal{P}_{i}$ means that the coalition to which an agent belongs in node $j$ is the same as the coalition to which an agent belongs in node $i$.

Join node: For two children $j_{1}, j_{2}$ of a join node $i$, it holds that $X_{i}=X_{j_{1}}=X_{j_{2}}$. To update $A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]$ in a join node, we first take the minimum of $A_{j_{1}}\left[\mathcal{P}_{i}, \mathbf{u}_{j_{1}}\right]$ and $A_{j_{2}}\left[\mathcal{P}_{i}, \mathbf{u}_{j_{2}}\right]$. Note that the maximum minimum utility among forgotten agent until $X_{i}$ is the minimum of ones until the children nodes. Here, for every agent $v \in X_{i}, \mathbf{u}_{i}(v)=\mathbf{u}_{j_{1}}(v)+\mathbf{u}_{j_{2}}(v)-\sum_{u \in N(v) \cap C_{v}} w_{v u}$ must hold. The subtraction avoids the double counting of edges. Then taking the maximum among $\mathbf{u}_{j_{1}}$ and $\mathbf{u}_{j_{2}}$ satisfying the above condition, the recursive formula can be defined as follows:

$$
A_{i}\left[\mathcal{P}_{i}, \mathbf{u}_{i}\right]=\max _{\mathbf{u}_{j_{1}}+\mathbf{u}_{j_{2}}=\mathbf{u}_{i}^{\prime}} \min \left\{A_{j_{1}}\left[\mathcal{P}_{i}, \mathbf{u}_{j_{1}}\right], A_{j_{2}}\left[\mathcal{P}_{i}, \mathbf{u}_{j_{2}}\right]\right\}
$$

where each element $\mathbf{u}_{i}^{\prime}(v)$ of $\mathbf{u}_{i}^{\prime}$ is defined as $\mathbf{u}_{i}^{\prime}(v)=\mathbf{u}_{i}(v)-\sum_{u \in N(v) \cap C_{v}} w_{v u}$.
Since the size of a DP table of each bag is $(\operatorname{tw} W)^{O(\mathrm{tw})}$ and each recursive formula can be computed in time $(\mathrm{tw} W)^{O(\mathrm{tw})}$, the total running time is $(\mathrm{tw} W)^{O(\mathrm{tw})} n$.

Theorem 10 implies that if $W$ is bounded by a polynomial in $n$, Max Egalitarian can be computed in time $n^{O(\mathrm{tw})}$.

## References

[1] A. B. Adcock, B. D. Sullivan, and M. W. Mahoney. Tree-like structure in large social and information networks. In ICDM 2013, pages 1-10, 2013.
[2] J. Alcalde and P. Revilla. Researching with whom? stability and manipulation. Journal of Mathematical Economics, 40(8):869-887, 2004.
[3] H. Aziz, F. Brandt, and P. Harrenstein. Fractional hedonic games. In AAMAS 2014, pages 5-12, 2014.
[4] H. Aziz, F. Brandt, and H. G. Seedig. Computing desirable partitions in additively separable hedonic games. Artificial Intelligence, 195:316 - 334, 2013.
[5] C. Ballester. NP-completeness in hedonic games. Games and Economic Behavior, 49(1):1-30, 2004.
[6] S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. Social Choice and Welfare, 18(1):135-153, 2001.
[7] A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. Games and Economic Behavior, 38(2):201-230, 2002.
[8] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Parameterized Algorithms. Springer International Publishing, 2015.
[9] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In FOCS 2011, pages 150-159, 2011.
[10] J. H. Dreze and J. Greenberg. Hedonic coalitions: Optimality and stability. Econometrica, 48(4):987, 1980.
[11] R. I. M. Dunbar. Neocortex size as a constraint on group size in primates. Journal of Human Evolution, 22(6):469-493, 1992.
[12] R. Elsässer and T. Tscheuschner. Settling the complexity of local max-cut (almost) completely. In ICALP 2011, pages 171-182, 2011.
[13] F. V. Fomin and D. M. Thilikos. A simple and fast approach for solving problems on planar graphs. In STACS 2004, pages 56-67, 2004.
[14] M. Gairing and R. Savani. Computing stable outcomes in hedonic games. In SAGT 2010, pages 174-185, 2010.
[15] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman \& Co., New York, NY, USA, 1979.
[16] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer. Some simplified np-complete graph problems. Theoretical Computer Science, 1(3):237-267, 1976.
[17] K. Kawarabayashi, Y. Kobayashi, and B. Reed. The disjoint paths problem in quadratic time. Journal of Combinatorial Theory, Series B, 102(2):424-435, 2012.
[18] T. Kloks. Treewidth, Computations and Approximations, volume 842 of Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 1994.
[19] M. Le Breton, I. Ortuño-Ortin, and S. Weber. Gamsons law and hedonic games. Social Choice and Welfare, 30(1):57-67, 2008.
[20] P. J. McSweeney, K. Mehrotra, and J. C. Oh. A game theoretic framework for community detection. In ASONAM 2012, pages 227-234, 2012.
[21] M. Olsen. Nash stability in additively separable hedonic games and community structures. Theory of Computing Systems, 45(4):917-925, 2009.
[22] D. Peters. Graphical hedonic games of bounded treewidth. In AAAI 2016, pages 586-593, 2016.
[23] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes. Hedonic coalition formation for distributed task allocation among wireless agents. IEEE Transactions on Mobile Computing, 10(9):1327-1344, 2010.
[24] S. C. Sung and D. Dimitrov. Computational complexity in additive hedonic games. European Journal of Operational Research, 203(3):635-639, 2010.


[^0]:    *This work was partially supported by JSPS KAKENHI Grant Numbers JP17K19960, 17H01698, 19K21537.

