Parity in graph sharing games

Piotr Micek, Bartosz Walczak*

Department of Theoretical Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

Abstract

Two players share a connected graph with non-negative weights on the vertices. They alternately take the vertices one by one and collect their weights. The rule they have to obey is that the taken part of the graph must be connected after each move. We present a strategy for the first player to get at least 1/4 of a tree with an odd number of vertices. The parity condition is necessary: it is easy to find even trees with the first player's guaranteed outcome tending to zero. In general there are odd graphs with arbitrarily small outcome of the first player, but all known constructions are intricate. We suspect a kind of general parity phenomenon, namely, that the first player can secure a substantial fraction of the weight of any K_n-minor-free graph with an odd number of vertices. We discuss analogies with another variant of this game, called the graph-grabbing game, where the players have to keep the remaining (not taken) part connected all the time.

Keywords: graph sharing, graph grabbing, pizza game, combinatorial games

1. Introduction

Graph sharing games are played on a finite connected graph with non-negative weights on the vertices (from now on, simply a graph). There are two players: Alice and Bob. Starting with Alice, they take the vertices alternately one by one and collect their weights. The vertices taken are removed from the graph. The choice of a vertex to be played in each move is restricted depending on the variant of the game:

- *Taken part connected* (*game T*): the rule is that after each move the vertices taken so far form a connected subgraph of the original graph;
- *Remaining part connected* (*game R*, *graph-grabbing game*): the rule is that after each move the remaining vertices form a connected subgraph.

*Corresponding author

Email addresses: Piotr.Micek@tcs.uj.edu.pl (Piotr Micek), Bartosz.Walczak@tcs.uj.edu.pl (Bartosz Walczak)

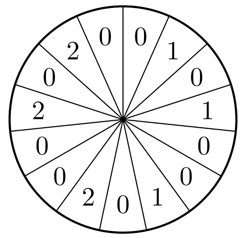

Figure 1: Alice can get at most $\frac{4}{9}$ of the pizza playing against clever Bob. Numbers stand for slice weights.

The game ends when all the vertices have been taken. Both players' goal is to maximize the total weight they have collected at the end.

The two variants of the game can be totally different in general. However, they coincide when the graph is a cycle. This case has been studied as the so-called pizza game: vertices are seen as slices of a pizza. In 1996 Brown asked whether Alice has a strategy to collect at least $\frac{1}{2}$ of the weight of any pizza. This can be easily confirmed for pizzas with an even number of slices: color alternately the slices with two colors and secure the heavier color. At first glance the case of pizzas with an odd number of slices looks better for Alice as she gets one slice more than Bob. Curiously things can get worse for her: there are examples where she can get only $\frac{4}{9}$ (see Figure 1). Winkler [1] conjectured in 2008 that Alice can secure at least $\frac{4}{9}$ of any pizza, and this has been proved by two independent groups of researchers.

Theorem 1.1 ([2, 3])**.** *Alice can secure at least* $\frac{4}{9}$ *of a pizza.*

As for pizzas, we measure Alice's outcome on a given graph as the fraction of the total weight of the graph that Alice can guarantee herself regardless of Bob's strategy. A natural question arises for both variants of the game: Is there a common constant $c > 0$ bounding from below Alice's outcome on any graph? The answer is: No. Simple examples show that such a lower bound cannot exist for either variant even if the game is played on trees instead of general graphs. The interesting point is that the parity of the number of vertices plays an important role here. In particular, Alice can guarantee herself a positive constant gain on trees with restricted parity of the number of vertices (odd for game T, even for game R).

The paper focuses on this parity phenomenon in game T. We provide a positive lower bound for Alice's outcome on *odd* trees.

Theorem 1.2. *Alice can secure at least* $\frac{1}{4}$ *of the weight of any tree with an odd number of vertices in game T.*

On the other hand, we construct a sequence of even trees with Alice's guaranteed outcome tending to zero. We also construct a sequence of odd graphs on which Alice's outcome tends to zero. These graphs contain minors of arbitrarily large cliques, which we believe are crucial.

Conjecture 1.3. *There is a function $f(n) > 0$ such that Alice can secure at least $f(n)$ of the weight of any graph with an odd number of vertices and with no K_n-minor in game T.*

The result for trees confirms the conjecture for $n = 3$. We have also proved the conjecture for $n = 4$, but our argument is too technical to be presented.

Game R is discussed by the authors in [4]. A similar kind of parity phenomenon holds in this variant, but the two parities switch their roles. Namely, Alice cannot expect anything playing on graphs with an *odd* number of vertices (consider a 3-vertex path with all the weight at the middle), and it is proved in [4] that Alice can collect at least $\frac{1}{4}$ of any tree with an *even* number of vertices. Seacrest and Seacrest [5] have improved this bound to $\frac{1}{2}$ (which is clearly best possible), thus proving our conjecture from [4]. For game R in general, there are graphs with an even number of vertices which are arbitrarily bad for Alice. The examples we know contain arbitrarily large cliques as subgraphs. It seems possible that Alice's outcome is bounded away from zero on graphs with an even number of vertices excluding K_n as a subgraph.

Independent research concerning generalizations of the pizza game, also leading to the two aforementioned variants, has been carried out by Cibulka et al. [6]. They focus on connectivity and computational complexity issues. In particular, they prove that deciding which player has a strategy to gather more than $\frac{1}{2}$ from a given graph in game R is PSPACE-complete. Whether the same is true for game T is left open.

The constructions of graphs with arbitrarily small Alice's outcome announced above are presented in Section 2. Section 3 describes the ideas behind our strategies for Alice. Section 4 exposes the main ingredient in these strategies—the strategy on independent components. Although presented in the context of game T, this concept is more general and may also be applicable to game R as well as to other combinatorial games in which two players collect values from a finite board. The proof of Theorem 1.2 is contained in Section 5. Main open problems arising from our study are summarized in Section 6.

2. Upper bounds

In the variant with taken part connected all vertices of the graph are available for Alice at the start of the game. Obviously, Alice can pick the heaviest vertex thus securing at least $1/|V|$ of the total weight with her very first move. In general she cannot be sure to get much more. The following example was presented to the authors by Kierstead [7].

Example 2.1. $\mathbf{G}_n$ is a weighted graph with $2n$ vertices $V = \{a_1, \ldots, a_n, b_1, \ldots, b_n\}$. The b_i's form a clique: $b_i \; E \; b_j$ for all $i \neq j$. The only neighbor of each a_i is b_i. The weights are distributed on the a_i's: $w(a_i) = 1$ and $w(b_i) = 0$, and thus the total weight is n (see Figure 2).

Alice has no strategy to gather more than 1 from $\mathbf{G}_n$. Indeed, she starts with some a_i (collecting 1) or b_i, and clever Bob responds by taking the other

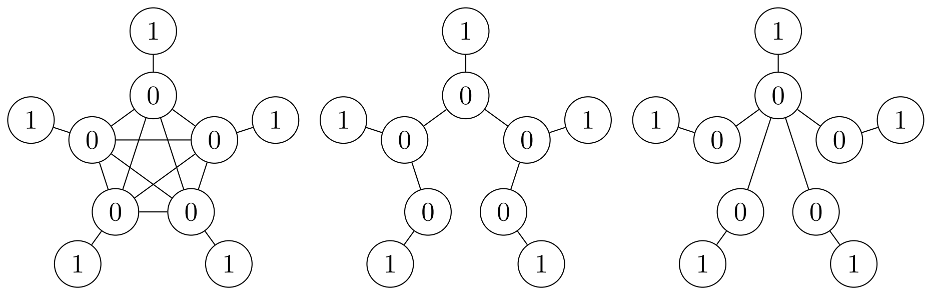

Figure 2: $\mathbf{G}_5$ and related examples

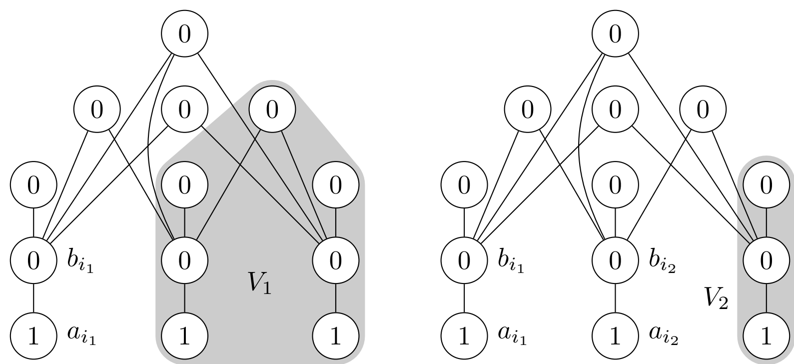

Figure 3: $\mathbf{H}_3$: Alice is forced to enter sets V_1 and V_2.

vertex. Then, in all subsequent moves Alice is forced to take some vertex of the clique, say b_j, and Bob responds by playing a_j.

In the above example the clique on the b_i's can be replaced by any connected graph (a path, a star, etc.), and the argument continues to work[1]. This shows that even for very simple classes of graphs (caterpillars, subdivided stars) Alice cannot guarantee herself any positive constant outcome as the size of the graph goes to infinity. However, all examples constructed this way have an even number of vertices. Things get more complicated if we ask for a sequence of graphs with an *odd* number of vertices and arbitrarily small Alice's guaranteed outcome. The following construction has been also found by Valtr and participants of his doc-course [8].

Example 2.2. $\mathbf{H}_n$ is a weighted graph with $2n+2^n-1$ vertices $V = \{a_1, \ldots, a_n, b_1, \ldots, b_n\} \cup \{c_X : X \in \mathcal{P}\{b_1, \ldots, b_n\} - \{\emptyset\}\}$. The neighborhoods of the vertices are: $N(a_i) = \{b_i\}$, $N(b_i) = \{a_i\} \cup \{c_X : b_i \in X\}$, $N(c_X) = X$. Only the a_i's have non-zero weight: $w(a_i) = 1$. The total weight is again n (see Figure 3).

[1]Switching weights 0 to 1 and 1 to 0 in $\mathbf{G}_n$ we obtain an analogous example of a graph which is arbitrarily bad for Alice in game R as presented in [4]. There, however, the clique on the b_i's is crucial and cannot be replaced by any other graph.

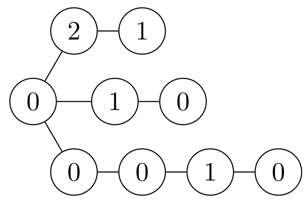

Figure 4: Alice gets at most 2 out of 5 on this tree.

Alice can secure at most 1 on $\mathbf{H}_n$. For the proof suppose first that Alice starts with a_{i_1} (b_{i_1}). Then Bob responds by taking b_{i_1} (a_{i_1}). If $n-1>0$, the subgraph induced by $V_1 = \{a_j, b_j : j \neq i_1\} \cup \{c_X : b_{i_1} \notin X\}$ is isomorphic to $\mathbf{H}_{n-1}$. In particular, $|V_1|$ is odd and $|V - V_1|$ is even. Since b_{i_1} is taken, all vertices in $V - V_1$ are available. Therefore, as long as Alice plays in $V - V_1$, Bob can respond also in $V - V_1$. Alice is eventually forced to enter V_1, which is possible only by taking some b_{i_2}, and Bob immediately follows with a_{i_2}. If $n-2>0$ then we define $V_2 = \{a_j, b_j : j \neq i_1, i_2\} \cup \{c_X : b_{i_1}, b_{i_2} \notin X\}$ and continue with the same argument. And so on. This way Bob wins all the remaining a_i's. If Alice starts with some c_X then Bob takes any available b_{i_1} and the same argument shows that Bob can take all the a_j's with $j \neq i_1$.

Cibulka et al. [6] have constructed a sequence of k-connected graphs with Alice's outcome tending to zero for any given k. They have also observed that Examples 2.1 and 2.2 from this paper lead to such sequences consisting of k-connected graphs of either parity: just replace each 0-vertex with an odd clique of 0-vertices and each original edge with a complete bipartite graph.

The best upper bound known for odd trees is $\frac{2}{5}$ (see Figure 4).

3. Strategies

Here comes the general idea behind the strategies we develop for Alice in game T:

(i) Split the graph into connected components of small weight by taking very few (a constant number of) vertices; this is possible for example for trees.
(ii) Play simultaneously on these independent light components securing $\frac{1}{2}$ of the total weight of all of them except one.

Fix an arbitrary weighted graph with an odd number of vertices. Let Alice play the game as she chooses and consider the state of the game after Bob's move. Clearly, an even number of vertices have been taken. Thus, the remaining part is odd and maybe it is split into several connected components. If only these components are balanced (that is, none of them contains most of the remaining weight), the following lemma (proved later on in a stronger version) provides a universal strategy which guarantees Alice a substantial fraction of the total remaining weight.

Lemma 3.1. *Starting from a partially shared graph with an odd number of remaining vertices, Alice has a strategy to gather at least $\frac{1}{2}$ of the total weight of all connected components of the remaining part except the heaviest one.*

Note that the claimed strategy works with no assumptions on the components except the right parity of the total number of vertices. Now we are going to show how it is used to derive a positive lower bound for Alice's guaranteed gain on trees with an odd number of vertices. Then in Section 4 we construct a strategy witnessing Lemma 3.1 and its strengthening Lemma 4.1. We exploit this stronger version in Section 5 and achieve the bound $\frac{1}{4}$ for odd trees thus proving Theorem 1.2.

Proposition 3.2. *Alice can secure at least $\frac{1}{6}$ of the weight of any tree with an odd number of vertices.*

Proof. Let $\mathbf{T}$ be a tree with an odd number of vertices. For convenience the weights are scaled so that they sum up to 1. First locate in $\mathbf{T}$ a vertex v such that the components of $\mathbf{T} - \{v\}$ have weight at most $\frac{1}{2}$. Such a vertex is called a *center* of $\mathbf{T}$. To this end pick any vertex $v_0 \in \mathbf{T}$. Either v_i is a center or exactly one component $\mathbf{C}$ of $\mathbf{T} - \{v_i\}$ has weight greater than $\frac{1}{2}$. In the latter case choose the only neighbor of v_i in $\mathbf{C}$ to be v_{i+1}. This way a simple path $v_0, v_1, \dots$ is constructed. Since $\mathbf{T}$ is finite, its center is finally found.

Alice starts with v a center of $\mathbf{T}$. Bob responds by taking some vertex b. Clearly, all components of the remaining part $\mathbf{T} - \{v, b\}$ have weight at most $\frac{1}{2}$. Now Alice applies the strategy from Lemma 3.1. As a result, since all components have weight at most $\frac{1}{2}$, she gets at least $\frac{1}{2}(1 - w(v) - w(b) - \frac{1}{2})$ on $\mathbf{T} - \{v, b\}$. Therefore, her total gain is at least

$$w(v) + \tfrac{1}{2}(1 - w(v) - w(b) - \tfrac{1}{2}) \geqslant \tfrac{1}{4} - \tfrac{1}{2}w(b),$$

which is good if b has a small weight. Otherwise, Alice can use a simple complementary strategy: start with b and do anything afterwards. The better of these two strategies gives Alice at least $\max(\frac{1}{4} - \frac{1}{2}w(b), w(b)) \geqslant \frac{1}{6}$. □

The above argument works as well for graphs with an odd number of vertices and with a small (constant-size) connected balancing set.

Proposition 3.3. *Let $\mathbf{G}$ be a graph with an odd number of vertices, and fix a constant $c \in [0, 1)$. Suppose there is a connected subgraph $\mathbf{H}$ of $\mathbf{G}$ with k vertices such that every connected component of $\mathbf{G} - \mathbf{H}$ has weight at most $c \cdot w(\mathbf{G})$. Then Alice can secure at least $\frac{1-c}{k+2}$ of the total weight of $\mathbf{G}$.*

Proof. As before we scale the weights so that $w(\mathbf{G}) = 1$. Alice plays in two phases. First she takes vertices of $\mathbf{H}$ until the entire $\mathbf{H}$ is taken, which is possible since $\mathbf{H}$ is connected. This phase lasts at most k turns, in which Bob takes some vertices $b_1, \dots, b_\ell$ ($\ell \leqslant k$). Now, at the beginning of the second phase, all components of the remaining part have weight at most c. Alice applies the strategy from Lemma 3.1 and guarantees herself at least $\frac{1}{2}(1 -$

$w(\mathbf{H} \cup \{b_1, \ldots, b_\ell\}) - c)$ on the remaining part. Thus, she gets at least $\frac{1}{2}(1 - w(b_1) - \ldots - w(b_\ell) - c)$ in total. A complementary strategy is to start with the heaviest vertex among $b_1, \ldots, b_\ell$. The better of these two strategies gives Alice at least

$$\max\bigl(\tfrac{1}{2}(1 - w(b_1) - \ldots - w(b_\ell) - c), w(b_1), \ldots, w(b_\ell)\bigr) \geqslant \tfrac{1-c}{k+2}. \qquad \square$$

4. Strategy for components

Let us return to the situation expressed in Lemma 3.1: some vertices of a graph $\mathbf{G}$ have been taken; the remaining part $\mathbf{R}$ is odd and it is split into several connected components $\mathbf{C}_1, \ldots, \mathbf{C}_k$. Each component $\mathbf{C}_i$ contains at least one vertex adjacent to the taken part $\mathbf{G} - \mathbf{R}$. We call such vertices *roots* of the component. If $\mathbf{G}$ is a tree then every component is a tree and has a unique root.

The collection of rooted components $\mathbf{C}_1, \ldots, \mathbf{C}_k$ may be considered as a board for the further game. Each subsequent move of either player consists of choosing a $\mathbf{C}_i$ and taking an available vertex in $\mathbf{C}_i$. A vertex of $\mathbf{C}_i$ is available if it is a root of $\mathbf{C}_i$ or it is adjacent to an already taken vertex of $\mathbf{C}_i$. This exhibits the crucial property of components: the set of allowed moves in a component depends only on the taken part of that component. We will make use of this property constructing a good strategy for Alice on the entire collection of components which combines some local strategies defined for each component separately. These local strategies will be expressed in terms of an auxiliary game on the components.

First we define the *rooted game* on a rooted component $\mathbf{C}$. In this game, there are two players: 1st and 2nd, who take the vertices of $\mathbf{C}$ alternately one by one (starting with 1st) according to the following rule: each vertex to be taken is a root of $\mathbf{C}$ or is adjacent to an already taken vertex of $\mathbf{C}$. Of course, the goal of both players is to collect as much as possible. The rooted game on $\mathbf{C}$ exactly describes what can be played in the original game when $\mathbf{C}$ is the only part of the graph that remains not taken.

The $\star$-*game* on a rooted component $\mathbf{C}$ is a modification of the rooted game. It is played by two players: 1st and 2nd, who take the vertices of $\mathbf{C}$ alternately one by one (starting with 1st) according to the following rules:

(i) A vertex can be taken only if it is a root of $\mathbf{C}$ or it is adjacent to an already taken vertex of $\mathbf{C}$ (just as in the rooted game on $\mathbf{C}$).

(ii) 2nd instead of taking a vertex may say *STOP*, which immediately ends the game.

(iii) 2nd when taking a vertex must *retain advantage*, that is, the total weight gathered by 2nd after his move must be at least the weight gathered by 1st so far.

A single $\star$-game ends when 2nd says *STOP* or when $\mathbf{C}$ has an even number of vertices and all of them have been taken (if $\mathbf{C}$ has an odd number of vertices and 1st has taken the last vertex then 2nd says *STOP*). Let 1st($\mathbf{C}$) and 2nd($\mathbf{C}$) denote the total weights gathered from $\mathbf{C}$ by 1st and 2nd respectively. The value of a single $\star$-game on $\mathbf{C}$ is

(i) $2\mathrm{nd}(\mathbf{C}) - 1\mathrm{st}(\mathbf{C})$ if 2nd said *STOP*,
(ii) ∞ if $\mathbf{C}$ has an even number of vertices and all of them have been taken.

The $\star$-*value* of $\mathbf{C}$, denoted by $\mathrm{val}^\star(\mathbf{C})$, is the minimum value of a $\star$-game on $\mathbf{C}$ that 1st can guarantee or, equivalently, the maximum value of a $\star$-game on $\mathbf{C}$ that can be guaranteed by 2nd. Clearly, if $\mathrm{val}^\star(\mathbf{C})$ is finite then

(i) 1st has a strategy in the $\star$-game on $\mathbf{C}$ such that after each his move $2\mathrm{nd}(\mathbf{C}) - 1\mathrm{st}(\mathbf{C}) \leqslant \mathrm{val}^\star(\mathbf{C})$,
(ii) 2nd has a strategy in the $\star$-game on $\mathbf{C}$ such that either he retains advantage till the entire $\mathbf{C}$ has been taken ($\mathbf{C}$ must be even in this case) or he finally forces 1st to make a move after which $2\mathrm{nd}(\mathbf{C}) - 1\mathrm{st}(\mathbf{C}) \geqslant \mathrm{val}^\star(\mathbf{C})$ (then he says *STOP*).

If $\mathrm{val}^\star(\mathbf{C}) = \infty$ then $\mathbf{C}$ has an even number of vertices and 2nd has a strategy in the $\star$-game to continue the game till the entire $\mathbf{C}$ has been taken.

Now we will show how to combine the strategies for the $\star$-game into a single strategy for Alice on the entire collection of components. The general scheme is that Alice starts in the component with the minimum $\star$-value and (in most cases) leads the game to a point at which she gains advantage after Bob's move. Then we recompute the components and their corresponding local strategies according to the new remaining part, and we apply the same argument inductively for this new collection of components. At the very end, when only one component remains, Alice loses no more than the $\star$-value of that component.

As this brief description suggests, we can strengthen Lemma 3.1 if we take care for subgraphs of $\mathbf{G}$ that may become connected components of the remaining part at some point of the game. We call them *subcomponents*. They allow an easy characterization: a subcomponent is a connected induced proper subgraph $\mathbf{S}$ of $\mathbf{G}$ such that $\mathbf{G} - \mathbf{S}$ is also connected. The *roots* of a subcomponent $\mathbf{S}$ are the vertices of $\mathbf{S}$ adjacent to $\mathbf{G} - \mathbf{S}$. The definitions of the rooted game, the $\star$-game and the $\star$-value apply to subcomponents naturally.

Lemma 4.1. *Starting from a partially shared graph* $\mathbf{G}$ *whose remaining part* $\mathbf{R}$ *has an odd number of vertices, Alice has a strategy to gather at least*

$$\min_{\mathbf{S}} \tfrac{1}{2}(w(\mathbf{R}) - \mathrm{val}^\star(\mathbf{S})),$$

where the minimum is taken over all subcomponents $\mathbf{S}$ *of* $\mathbf{G}$ *such that* $\mathbf{S} \subseteq \mathbf{R}$ *and* $\mathbf{S}$ *has an odd number of vertices.*

Lemma 3.1 follows directly from the above by a trivial bound $\mathrm{val}^\star(\mathbf{S}) \leqslant w(\mathbf{S}) \leqslant w(\mathbf{C})$, where $\mathbf{C}$ is the connected component of $\mathbf{R}$ containing $\mathbf{S}$.

Proof of Lemma 4.1. The argument goes by induction on the size of $\mathbf{R}$. If only Alice manages to gather at least as much as Bob after some Bob's move, the conclusion follows directly from the inductive hypothesis applied to $\mathbf{G}$ with the new (smaller) remaining part.

Let $\mathbf{C}_1, \ldots, \mathbf{C}_k$ be the connected components of $\mathbf{R}$. Alice builds her strategy on top of the strategies for the $\star$-game on $\mathbf{C}_1, \ldots, \mathbf{C}_k$. Let $S_1^\star(\mathbf{C}_i)$ denote the strategy of 1st in the $\star$-game on $\mathbf{C}_i$ securing the value of the game to be at most $\mathrm{val}^\star(\mathbf{C}_i)$. Let $S_2^\star(\mathbf{C}_i)$ denote the strategy of 2nd in the $\star$-game on $\mathbf{C}_i$ guaranteeing the value of the game to be at least $\mathrm{val}^\star(\mathbf{C}_i)$. Finally, assume $\mathrm{val}^\star(\mathbf{C}_1)$ is minimal among all $\mathrm{val}^\star(\mathbf{C}_i)$. At least one component of $\mathbf{R}$ has an odd number of vertices and therefore finite $\star$-value, so $\mathrm{val}^\star(\mathbf{C}_1)$ is finite.

Alice starts with the vertex from $\mathbf{C}_1$ realizing the strategy $S_1^\star(\mathbf{C}_1)$. From now on, as long as Bob maintains advantage, she responds in the same component as Bob plays. We keep an invariant that the players share each component according to the rules of the $\star$-game. Playing on $\mathbf{C}_1$ Alice realizes $S_1^\star(\mathbf{C}_1)$ as 1st and Bob is put into 2nd's shoes. On any other component $\mathbf{C}_i$ Bob always plays as 1st and Alice realizes the strategy $S_2^\star(\mathbf{C}_i)$ as 2nd. We will show that if this invariant cannot be kept any more, the game has reached a point at which Alice has gathered at least as much as Bob (from the entire $\mathbf{R}$) and therefore we can apply the inductive hypothesis for the rest of the game.

Let us analyze Bob's possible moves and their consequences in detail. $A(\mathbf{C}_i)$ and $B(\mathbf{C}_i)$ denote the weights collected so far by Alice and Bob, respectively, from the component $\mathbf{C}_i$.

Case 1: Bob takes a vertex from $\mathbf{C}_1$ and $B(\mathbf{C}_1) \geqslant A(\mathbf{C}_1)$.

Bob's move is legal for 2nd in the $\star$-game, so Alice can proceed with $S_1^\star(\mathbf{C}_1)$. We only need to argue that there is still something to take in $\mathbf{C}_1$. If this is not the case then $\mathbf{C}_1$ has an even number of vertices and Bob playing as 2nd maintained advantage all the time, which contradicts the fact that $\mathrm{val}^\star(\mathbf{C}_1)$ is finite and that Alice stuck to $S_1^\star(\mathbf{C}_1)$.

Case 2: Bob takes a vertex from $\mathbf{C}_1$ and $B(\mathbf{C}_1) < A(\mathbf{C}_1)$.

Alice playing as 2nd on $\mathbf{C}_i$, for $i \neq 1$, obeyed the rules of the $\star$-game and took the last vertex played there, which in particular means that

$$A(\mathbf{C}_i) \geqslant B(\mathbf{C}_i), \quad \text{for } i \neq 1.$$

This together with $B(\mathbf{C}_1) < A(\mathbf{C}_1)$ ensures that Alice has gathered more than Bob from all the components.

Case 3: Bob takes a vertex from $\mathbf{C}_i$ with $i \neq 1$.

If $S_2^\star(\mathbf{C}_i)$ tells Alice to take a vertex in $\mathbf{C}_i$, she does so and the invariant is kept. The only interesting case is when $S_2^\star(\mathbf{C}_i)$ says *STOP*. Then Alice playing as 2nd on $\mathbf{C}_j$ with $j \notin \{1, i\}$ obeyed the rules of the $\star$-game and took the last vertex played there, which yields

$$A(\mathbf{C}_j) \geqslant B(\mathbf{C}_j), \quad \text{for } j \neq 1, i.$$

She also took the last vertex played in $\mathbf{C}_1$ and therefore by the property of $S_1^\star(\mathbf{C}_1)$ she has secured that

$$B(\mathbf{C}_1) - A(\mathbf{C}_1) \leqslant \mathrm{val}^\star(\mathbf{C}_1).$$

The *STOP* on $\mathbf{C}_i$ ends the $\star$-game on $\mathbf{C}_i$. The strategy $S_2^\star(\mathbf{C}_i)$ realized by Alice guarantees that at this moment she has gathered at least $\mathrm{val}^\star(\mathbf{C}_i)$ more than Bob from $\mathbf{C}_i$:

$$\mathrm{val}^\star(\mathbf{C}_i) \leqslant A(\mathbf{C}_i) - B(\mathbf{C}_i).$$

Composing the inequalities we get

$$B(\mathbf{C}_1) - A(\mathbf{C}_1) \leqslant \mathrm{val}^\star(\mathbf{C}_1) \leqslant \mathrm{val}^\star(\mathbf{C}_i) \leqslant A(\mathbf{C}_i) - B(\mathbf{C}_i).$$

Therefore, Alice has secured at least as much as Bob from $\mathbf{C}_1$, $\mathbf{C}_i$ together, and she is ahead on all the other $\mathbf{C}_j$'s as well.

Case 4: Bob does not move any more, since Alice took the last vertex of the graph.

Alice playing as 2nd on all the $\mathbf{C}_i$'s except $\mathbf{C}_1$ and obeying the rules of the $\star$-game has guaranteed that

$$A(\mathbf{C}_i) \geqslant B(\mathbf{C}_i), \quad \text{for } i \neq 1.$$

She took the last vertex of $\mathbf{C}_1$ playing as 1st, so by the property of $S_1^\star(\mathbf{C}_1)$ she has secured that

$$B(\mathbf{C}_1) - A(\mathbf{C}_1) \leqslant \mathrm{val}^\star(\mathbf{C}_1).$$

Summing up for all components of $\mathbf{R}$ we get

$$B(\mathbf{R}) - A(\mathbf{R}) \leqslant \mathrm{val}^\star(\mathbf{C}_1), \qquad B(\mathbf{R}) + A(\mathbf{R}) = w(\mathbf{R}),$$
$$A(\mathbf{R}) \geqslant \tfrac{1}{2}(w(\mathbf{R}) - \mathrm{val}^\star(\mathbf{C}_1)).$$
□

5. Improved strategy for odd trees

Now we show how to use Lemma 4.1 to derive a strategy for Alice securing $\frac{1}{4}$ of any tree with an odd number of vertices. The core idea is the same as in the proof of Proposition 3.2 (which yields $\frac{1}{6}$), with the only difference that instead of an entire component Alice loses only the $\star$-value of some subcomponent. The improvement follows from a somewhat straightforward induction on the number of vertices in the tree which is used when the $\star$-value of some subcomponent is too large.

Let $\mathbf{S}$ be any subcomponent of a tree $\mathbf{T}$, that is, any proper subtree of $\mathbf{T}$ such that $\mathbf{T} - \mathbf{S}$ is connected. It has a unique root—the vertex adjacent to $\mathbf{T} - \mathbf{S}$. Recall the rule of the rooted game on $\mathbf{S}$: the root is taken first, and each next taken vertex is adjacent to some previously taken vertex. Compared to the $\star$-game on $\mathbf{S}$, there are no further restrictions on the moves of 2nd and no *STOP*s. Define $\mathrm{val}_1(\mathbf{S})$ to be the maximum gain of 1st in the rooted game on $\mathbf{S}$, that is, the maximum total weight of the vertices taken by 1st that he can secure. Similarly, define $\mathrm{val}_2(\mathbf{S})$ to be the maximum gain of 2nd in the rooted game. Clearly, $\mathrm{val}_1(\mathbf{S}) + \mathrm{val}_2(\mathbf{S}) = w(\mathbf{S})$. The following explains the relationship of these to the $\star$-value of $\mathbf{S}$.

Lemma 5.1. *If* $\mathbf{S}$ *has an odd number of vertices then* $\mathrm{val}^\star(\mathbf{S}) \leqslant \mathrm{val}_2(\mathbf{S})$.

Proof. Every $\star$-game on $\mathbf{S}$ ends with a *STOP*. 2nd has a strategy in the $\star$-game on $\mathbf{S}$ such that at the moment he says *STOP* his advantage over 1st (and therefore his gain on $\mathbf{S}$) is at least $\mathrm{val}^\star(\mathbf{S})$. The legal moves of 1st in the rooted game and in the $\star$-game are the same. Therefore, since 2nd has a strategy to collect at least $\mathrm{val}^\star(\mathbf{S})$ in the $\star$-game, which is more restrictive for him, he can also gather at least $\mathrm{val}^\star(\mathbf{S})$ in the rooted game on $\mathbf{S}$. □

Proof of Theorem 1.2. Let $\mathbf{T}$ be a tree with an odd number of vertices. For convenience we scale the weights on $\mathbf{T}$ so that they sum up to 1. We prove by induction on the number of vertices of $\mathbf{T}$ that Alice can secure at least $c \cdot w(\mathbf{T}) = c$ on $\mathbf{T}$. To get through the induction step we will bound the constant c from above. The largest c for which the argument works will turn out to be $\frac{1}{4}$. Consider three cases:

Case 1: There is an even subcomponent $\mathbf{E}$ of $\mathbf{T}$ with $\mathrm{val}_2(\mathbf{E}) \geqslant c \cdot w(\mathbf{E})$.

As $\mathbf{T} - \mathbf{E}$ has an odd number of vertices, by the induction hypothesis Alice can guarantee herself at least $c \cdot w(\mathbf{T} - \mathbf{E})$ on $\mathbf{T} - \mathbf{E}$. We now construct a strategy for Alice on $\mathbf{T}$. She starts in $\mathbf{T} - \mathbf{E}$ according to her best strategy on $\mathbf{T} - \mathbf{E}$. Every time Bob plays in $\mathbf{T} - \mathbf{E}$, Alice responds in $\mathbf{T} - \mathbf{E}$ following that strategy. At some point Bob decides to take a first vertex from $\mathbf{E}$: it must be the root of $\mathbf{E}$. Every time Bob takes a vertex from $\mathbf{E}$, Alice responds in $\mathbf{E}$ according to the strategy for 2nd in the rooted game on $\mathbf{E}$ that guarantees $\mathrm{val}_2(\mathbf{E})$. The parities of $\mathbf{E}$ and $\mathbf{T} - \mathbf{E}$ are so chosen that Alice makes the last move in both parts. This way Alice's total gain on $\mathbf{T}$ is at least

$$c \cdot w(\mathbf{T} - \mathbf{E}) + \mathrm{val}_2(\mathbf{E}) \geqslant c \cdot w(\mathbf{T} - \mathbf{E}) + c \cdot w(\mathbf{E}) = c.$$

Case 2:

(i) Every even subcomponent $\mathbf{E}$ of $\mathbf{T}$ has $\mathrm{val}_1(\mathbf{E}) \geqslant (1-c)w(\mathbf{E})$.
(ii) There is an odd subcomponent $\mathbf{O}$ of $\mathbf{T}$ with $w(\mathbf{O}) \leqslant \frac{1}{2}$ and $\mathrm{val}_2(\mathbf{O}) \geqslant c$.

Let $\mathbf{E} = \mathbf{T} - \mathbf{O}$. Clearly, $\mathbf{E}$ is an even subcomponent of $\mathbf{T}$. The strategy for Alice on $\mathbf{T}$ is the following. She starts with the root of $\mathbf{E}$. Bob has two options: he can take another vertex of $\mathbf{E}$ or the root of $\mathbf{O}$. If he takes a vertex from $\mathbf{E}$, Alice responds in $\mathbf{E}$ sticking to the strategy of 1st on $\mathbf{E}$ which guarantees $\mathrm{val}_1(\mathbf{E})$. If Bob takes a vertex from $\mathbf{O}$, Alice responds in $\mathbf{O}$ realizing the strategy of 2nd on $\mathbf{O}$ which guarantees $\mathrm{val}_2(\mathbf{O})$. Alice continues this procedure till $\mathbf{E}$ or $\mathbf{O}$ runs out of vertices. We claim that already at that moment Alice has secured at least c.

If $\mathbf{O}$ has been entirely taken before $\mathbf{E}$ then Alice has realized the strategy of 2nd on $\mathbf{O}$ collecting $\mathrm{val}_2(\mathbf{O}) \geqslant c$. Now, suppose the entire $\mathbf{E}$ has been taken before $\mathbf{O}$. Then Alice has realized the strategy of 1st on $\mathbf{E}$ taking at least

$$\mathrm{val}_1(\mathbf{E}) \geqslant (1-c)w(\mathbf{E}) = (1-c)(w(\mathbf{T}) - w(\mathbf{O})) \geqslant \tfrac{1}{2}(1-c).$$

Thus, she has secured c if only $\frac{1}{2}(1-c) \geqslant c$, which holds for $c \leqslant \frac{1}{3}$.

Case 3: Every odd subcomponent $\mathbf{O}$ of $\mathbf{T}$ with $w(\mathbf{O}) \leqslant \frac{1}{2}$ has $\mathrm{val}_2(\mathbf{O}) \leqslant c$.

In this setting we apply the argument from the proof of Proposition 3.2, but in place of Lemma 3.1 we plug Lemma 4.1.

Alice starts with v a center of $\mathbf{T}$. Bob responds by taking some vertex b. Since all components of the remaining part $\mathbf{T} - \{v, b\}$ have weight at most $\frac{1}{2}$, for each $\mathbf{O}$ an odd subcomponent of $\mathbf{T} - \{v, b\}$ we have $\mathrm{val}_2(\mathbf{O}) \leqslant c$. Therefore, by Lemma 5.1, $\mathrm{val}^\star(\mathbf{O}) \leqslant c$.

Now, Alice applies the strategy from Lemma 4.1. As a result she gets at least $\frac{1}{2}(1 - w(v) - w(b) - c)$ from $\mathbf{T} - \{v, b\}$. Therefore, her total gain is at least

$$w(v) + \tfrac{1}{2}(1 - w(v) - w(b) - c) \geqslant \tfrac{1}{2}(1 - w(b) - c).$$

A complementary strategy for Alice (good for large weight of b) is to start with b and do anything afterwards. The better of these two strategies gives Alice at least

$$\max(\tfrac{1}{2}(1-w(b)-c), w(b)) \geqslant \tfrac{1}{3}(1-c) \geqslant c,$$

where the latter inequality holds for $c \leqslant \frac{1}{4}$. □

6. Concluding remarks and open problems

The most intriguing problems arise when considering graphs with

- an odd number of vertices in game T,
- an even number of vertices in game R.

There are basically two kinds of questions we can ask for both games. One concerns the border line separating graphs with a reasonable strategy for Alice from those being hopeless for her. An expected result is that Alice has a good strategy on all graphs with "simple enough" structure and the right parity of the number of vertices. The other kind of problem is to determine precisely the maximum fraction of the graph that Alice can guarantee herself on all graphs from a given class. Here trees are the first natural candidate to study.

We believe that in game T Alice's guaranteed outcome may be bounded away from zero for the class of graphs with an odd number of vertices excluding a K_n-minor (see Conjecture 1.3). This is confirmed for odd trees ($n = 3$) and for odd graphs of treewidth 2 ($n = 4$). On the way to verify this belief one can inspect other specific classes of graphs with a forbidden minor (and with an odd number of vertices) for which the question remains open: planar graphs, graphs with treewidth $k \geqslant 3$ (however, k-trees are solved by Proposition 3.3). The exact value of Alice's maximum guaranteed gain on odd trees lies between $\frac{1}{4}$ (by Theorem 1.2) and $\frac{2}{5}$ (witnessed by the tree in Figure 4).

In game R all known examples of graphs with an even number of vertices and very small Alice's guaranteed outcome contain a large clique. Thus, it seems possible that Alice has a strategy which guarantees her some positive constant fraction of any K_n-free graph with an even number of vertices. Alice's maximum guaranteed gain for even trees in game R is known to be $\frac{1}{2}$.

The concept of the $\star$-game is independent of particular rules of the game as long as the moves consist in collecting values from the board. It is not a coincidence that the strategies for Alice in game R presented in [4] proceed along very similar lines as the proof of Lemma 4.1—they can be reformulated in terms of components and the $\star$-game adapted appropriately to the rules of game R. We wonder whether this concept can be useful to construct effective strategies for other known combinatorial games.

References

[1] P. M. Winkler, problem posed at Building Bridges, a conference in honour of 60th birthday of László Lovász, Budapest, 2008.

[2] J. Cibulka, J. Kynčl, V. Mészáros, R. Stolař, P. Valtr, Solution to Peter Winkler's pizza problem, in: J. Fiala, J. Kratochvíl, M. Miller (Eds.), Combinatorial Algorithms, Vol. 5874 of Lecture Notes in Computer Science, Springer, 2009, pp. 356–367.

[3] K. Knauer, P. Micek, T. Ueckerdt, How to eat $\frac{4}{9}$ of a pizza, Discrete Mathematics 311 (16) (2011) 1635–1645.

[4] P. Micek, B. Walczak, A graph-grabbing game, Combinatorics, Probability and Computing 20 (4) (2011) 623–629.

[5] D. E. Seacrest, T. Seacrest, Grabbing the gold, submitted.

[6] J. Cibulka, J. Kynčl, V. Mészáros, R. Stolař, P. Valtr, Graph sharing games: complexity and connectivity, in: J. Kratochvíl, A. Li, J. Fiala, P. Kolman (Eds.), Theory and Applications of Models of Computation, Vol. 6108 of Lecture Notes in Computer Science, Springer, 2010, pp. 340–349.

[7] H. A. Kierstead, private communication.

[8] P. Valtr, participants of a doc-course in Berlin, 2009, private communication.