PARITY IN GRAPH SHARING GAMES

PIOTR MICEK AND BARTOSZ WALCZAK

Abstract. Two players share a connected graph with non-negative weights on the vertices. They alternately take the vertices (one in each turn) and collect their weights. There are two variants of the rule they have to obey: one is that the taken (removed) part of the graph must be connected after each move, the other one is that the remaining part of the graph must be connected after each move. Both players want to maximize their final outcome. We present strategies for player 1 (starting the game) to get at least $1/4$ of the weight of any tree with an odd number of vertices in the game with taken part connected, and to get at least $1/4$ of the weight of any tree with an even number of vertices in the game with remaining part connected. The parity conditions are necessary: in either game player 1 cannot secure any positive fraction of the total weight on all trees. We suspect a kind of general parity phenomenon in these games, namely, that player 1 has a strategy to gather a substantial portion of any “simple enough” graph with an odd/even number of vertices in the game with taken/remaining part connected. The technique developed to construct our strategies may be worth independent interest.

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 $R$):** the rule is that after each move the remaining vertices form a connected subgraph.

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 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 can get at least $1/2$ 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 $4/9$ (see Figure 1). Winkler [7] conjectured in 2008 that Alice can secure at least $4/9$ of any pizza, and this has been proved by two independent groups of researchers.

**Theorem 1.1 ([2, 4]).** Alice can secure at least $4/9$ of a pizza.

As for pizzas, we will always measure Alice’s guaranteed gain on a given graph as a fraction of its total weight. When considering the game on general graphs (in both
variants), a natural question arises: Can Alice always guarantee herself a constant fraction \( c > 0 \) of the weight of any graph no matter how this weight is distributed? The answer is: No. For both games one can construct a sequence of weighted trees on which Alice’s guaranteed gain tends to zero. The interesting point is that the parity of the number of vertices makes a difference. Namely, such a sequence can consist of trees of only one parity (even for game T, odd for game R). If restricted to the other parity, Alice’s guaranteed gain on trees can be bounded away from zero, and all known constructions of graphs that are hopeless for her result in complex structure. The paper exposes and explores this parity phenomenon.

We show that in game T Alice can get at least \( \frac{1}{4} \) of any tree with an odd number of vertices. We also construct a sequence of graphs with an odd number of vertices and with Alice’s guaranteed outcome tending to zero. These graphs contain minors of arbitrarily large cliques. We conjecture that there is a function \( f(n) > 0 \) such that Alice can get at least \( f(n) \) of any graph with an odd number of vertices and with no \( K_n \)-minor. The result for trees confirms the conjecture for \( n = 3 \).

In game R Alice can secure at least \( \frac{1}{4} \) of any tree with an even number of vertices. We suspect that the exact constant is \( \frac{1}{2} \), which is an obvious upper bound. Here no sequence of graphs with an even number of vertices which is hopeless for Alice and which avoids arbitrarily large cliques as subgraphs is known.

Independent research concerning generalizations of the pizza game, also leading to game T and game R, has been carried out by Cibulka et al. [1]. They focus on connectivity and computational complexity issues. They also study game TR in which both the taken and the remaining part must be connected throughout the game. Game R on a tree has been also proposed by Rosenfeld [5].

Game T is dealt with in Sections 2–4, and game R in Section 5. Our main technique for constructing strategies in graph sharing games—the strategy on independent components—is exposed in Section 3. Although presented in the context of game T, this method is quite general and may be applicable as well to other combinatorial games in which two players collect values from a finite “board”. Main open problems arising from our study are summarized in Section 6.

2. Game T: Taken part connected

In the variant with taken part connected all vertices of the graph are available for Alice starting the game (this is not necessarily the case in game R). 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 [3].

Example 2.1. \( G_n = (V, E, w) \) 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.

Alice has no strategy to gather more than 1 from G_n in game T. Indeed, she starts with some a_i (collecting 1) or b_i, and clever Bob responds by taking the other 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.

Note that 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. This shows that even for very simple classes of graphs (caterpillars, subdivisions of stars) Alice cannot guarantee herself any positive constant gain 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 gain. The following construction has been also found by Valtr and participants of his doc-course [6].

Example 2.2. H_n = (V, E, w) is a weighted graph with 2n + 2n − 1 vertices V = \{a_1, \ldots, a_n, b_1, \ldots, b_n\} ∪ \{e_X : X \in P(b_1, \ldots, b_n) \setminus \emptyset\}. The neighborhoods of the vertices are: N(a_i) = \{b_i\}, N(b_i) = \{a_i\} ∪ \{e_X : b_i \not\in X\}, N(e_X) = X. Only the a_i’s have non-zero weight: w(a_i) = 1. The total weight is again n.

Alice can secure at most 1 on H_n in game T. For the proof suppose first that Alice starts with a_i (b_i). Then Bob responds by taking b_j (a_j). If n − 1 > 0, the subgraph induced by V_1 = \{a_j, b_j : j \neq i\} ∪ \{e_X : b_j \not\in X\} is isomorphic to H_{n−1}. In particular, |V_1| is odd and |V − V_1| is even. Since b_i 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, and Bob immediately follows with a_i. 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 \(b_i\) and the same argument shows that Bob can take all the \(a_j\)'s with \(j \neq i_1\).

The rich structure of \(H_n\) suggests that we can hope for effective strategies for Alice on “simple” graphs with an odd number of vertices. Such strategies indeed exist for trees, and we conjecture they exist for all graphs with a forbidden minor.

**Theorem 2.3.** Alice can secure at least \(\frac{1}{4}\) of the weight of any tree with an odd number of vertices in game \(T\).

**Conjecture 2.4.** 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 \(T\).

Cibulka et al. [1] have constructed a sequence of \(k\)-connected graphs with Alice’s guaranteed 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.

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

(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. Give Alice free hand to start the game and consider the situation after some 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 light enough, the following claim (proved later on in a stronger version) provides a universal strategy which guarantees Alice a substantial fraction of the total weight of the remaining part.

**Claim 2.5.** 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 to use it to derive a positive lower bound for Alice’s guaranteed gain on trees with an odd number of vertices. Then in Section 3 we construct a strategy witnessing Claim 2.5 and its strengthening Claim 3.1. We exploit this stronger version in Section 4 and achieve the bound \(\frac{1}{4}\) for odd trees thus proving Theorem 2.3.

**Proposition 2.6.** Alice can secure at least \(\frac{1}{6}\) of the weight of any tree with an odd number of vertices in game \(T\).

**Proof.** Let \(T\) 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 \(T\) a vertex \(v\) such that the components of \(T - \{v\}\) have weight at most \(\frac{1}{2}\). Such a vertex is called a center of \(T\). To this end pick any vertex \(v_0 \in T\). Either \(v_0\) is a center or exactly one component \(C\) of \(T - \{v_1\}\) has weight greater than \(\frac{1}{2}\). In the latter case choose the only neighbor of \(v_i\) in \(C\) to be \(v_{i+1}\). This way a simple path \(v_0, v_1, \ldots\) is constructed. Since \(T\) is finite, its center is finally found.
Alice starts with a center of $T$. Bob responds by taking some vertex $b$. Clearly, all components of the remaining part $T - \{v, b\}$ have weight at most $\frac{1}{2}$. Now Alice applies the strategy from Claim 2.5. 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 $T - \{v, b\}$. Therefore, her total gain is at least

$$w(v) + \frac{1}{2}(1 - w(v) - w(b) - \frac{1}{2}) \geq \frac{1}{4} - \frac{1}{4}w(b),$$

which is pretty much 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 $\max(\frac{1}{4} - \frac{1}{4}w(b), w(b)) \geq \frac{1}{4}$. □

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

**Proposition 2.7.** Let $G$ be a graph with an odd number of vertices, and fix a constant $c \in [0, 1]$. Suppose there is a connected subgraph $H$ of $G$ with $k$ vertices such that every connected component of $G - H$ has weight at most $c \cdot w(G)$. Then Alice can secure at least $\frac{k}{2} \cdot c$ of the total weight of $G$ in game $T$.

**Proof.** As before we scale the weights so that $w(G) = 1$. Alice plays in two phases. First she takes vertices of $H$ until all the $H$ is shared out. She can play so because $H$ is connected. This phase lasts at most $k$ turns, in which Bob takes some vertices $b_1, \ldots, b_k$ ($\ell \leq k$). Now, at the beginning of the second phase, all components of the remaining part have weight at most $c$. Alice applies the strategy from Claim 2.5 and guarantees herself at least $\frac{1}{2}(1 - w(H \cup \{b_1, \ldots, b_k\}) - c)$ on the remaining part. Thus, she gets at least $\frac{1}{2}(1 - w(b_1) - \ldots - w(b_k) - c)$ in total. A complementary strategy is to start with the heaviest vertex among $b_1, \ldots, b_k$. The better of these two strategies gives Alice at least

$$\max(\frac{1}{2}(1 - w(b_1) - \ldots - w(b_k) - c), w(b_1), \ldots, w(b_k)) \geq \frac{\ell + c}{\ell + 2}. □$$

### 3. Strategy for Components

Step back to the setting of Claim 2.5: some vertices of a graph $G$ have been shared out; the remaining part $R$ is odd and it is split into several connected components $C_1, \ldots, C_k$. Each component $C_i$ contains at least one vertex adjacent to the taken part $G - R$. We call such vertices roots of the component. If $G$ is a tree then every component is a tree and has a unique root.

The collection of rooted components $C_1, \ldots, C_k$ may be considered as a board for the further game. Each subsequent move of either player consists of choosing a $C_i$ and taking an available vertex in $C_i$. A vertex of $C_i$ is available if it is a root of $C_i$ or it is adjacent to an already taken vertex of $C_i$. 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 $C$. In this game, there are two players: 1st and 2nd, who take the vertices of $C$ alternately one by one (starting with 1st) according to the following rule: each vertex to be taken is a root of $C$ or is adjacent to an already taken vertex of $C$. Of course, the goal of both players is to collect as much as possible. The rooted game on $C$ exactly describes what can be played in the original game when $C$ is the only part of the graph that remains not taken.
The \(\ast\)-game on a rooted component \(C\) is a modification of the rooted game. It is played by two players: 1st and 2nd, who take the vertices of \(C\) 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 \(C\) or it is adjacent to an already taken vertex of \(C\) (just as in the rooted game on \(C\)).

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

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

\[\begin{array}{ll}
(i) & 2nd(C) - 1st(C), \text{if 2nd said } \textit{STOP}, \\
(ii) & \infty, \text{if } C \text{ has an even number of vertices and all of them have been taken.}
\end{array}\]

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

\[\begin{array}{ll}
(i) & 1st \text{ has a strategy in the } \ast\text{-game on } C \text{ such that after each his move } 2nd(C) - 1st(C) \leq \text{val}^\ast(C), \\
(ii) & 2nd \text{ has a strategy in the } \ast\text{-game on } C \text{ such that either he retains advantage till the entire } C \text{ has been shared out (} C \text{ must be even in this case) or he finally forces 1st to } \text{make a move after which } 2nd(C) - 1st(C) \geq \text{val}^\ast(C) \text{ (then he says } \textit{STOP}).
\end{array}\]

If \(\text{val}^\ast(C) = \infty\) then \(C\) has an even number of vertices and 2nd has a strategy in the \(\ast\)-game to continue the game till the entire \(C\) has been shared out.

Now we will show how to combine the strategies for the \(\ast\)-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 \(\ast\)-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 \(\ast\)-value of that component.

As this brief description suggests, we can strengthen Claim 2.5 if we take care for subgraphs of \(G\) that may become connected components of the remaining part at some point of the game. We call them \textit{subcomponents}. They allow an easy characterization: a subcomponent is a connected induced proper subgraph \(S\) of \(G\) such that \(G - S\) is also connected. The \textit{roots} of a subcomponent \(S\) are the vertices of \(S\) adjacent to \(G - S\). The definitions of the rooted game, the \(\ast\)-game and the \(\ast\)-value apply to subcomponents naturally.

Claim 3.1. Starting from a partially shared graph \(G\) whose remaining part \(R\) has an odd number of vertices, Alice has a strategy to gather at least

\[\min_S \frac{1}{2}(w(R) - \text{val}^\ast(S)),\]

where the minimum is taken over all subcomponents \(S\) of \(G\) such that \(S \subseteq R\) and \(S\) has an odd number of vertices.

Claim 2.5 follows directly from the above by a trivial bound \(\text{val}^\ast(S) \leq w(S) \leq w(C)\), where \(C\) is the connected component of \(R\) containing \(S\).
Proof of Claim 3.1. The argument goes by induction on the size of $R$. 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 $G$ with the new (smaller) remaining part.

Let $C_1, \ldots, C_k$ be the connected components of $R$. Alice builds her strategy on top of the strategies for the $*$-game on $C_1, \ldots, C_k$. Let $S_1^i(C_i)$ denote the strategy of 1st in the $*$-game on $C_i$ securing the value of the game to be at most $\text{val}^*(C_i)$. Let $S_2^i(C_i)$ denote the strategy of 2nd in the $*$-game on $C_i$ guaranteeing the value of the game to be at least $\text{val}^*(C_i)$. Finally, assume $\text{val}^*(C_1)$ is minimal among all $\text{val}^*(C_i)$. At least one component of $R$ has an odd number of vertices and therefore finite $*$-value, so $\text{val}^*(C_1)$ is finite.

Alice starts with the vertex from $C_1$ realizing the strategy $S_1^i(C_1)$. 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 $*$-game. Playing on $C_1$ Alice realizes $S_1^i(C_1)$ as 1st and Bob is put into 2nd’s shoes. On any other component $C_i$ Bob always plays as 1st and Alice realizes the strategy $S_2^i(C_i)$ 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 $R$) 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. $A(C_i)$ and $B(C_i)$ denote the weights collected so far by Alice and Bob, respectively, from the component $C_i$.

Case 1: Bob takes a vertex from $C_1$ and $B(C_1) \geq A(C_1)$.

Bob’s move is legal for 2nd in the $*$-game, so Alice can proceed with $S_1^i(C_1)$. We only need to argue that there is still something to take in $C_1$. If this is not the case then $C_1$ has an even number of vertices and Bob playing as 2nd maintained advantage all the time, which contradicts the fact that $\text{val}^*(C_1)$ is finite and that Alice stuck to $S_1^i(C_1)$.

Case 2: Bob takes a vertex from $C_1$ and $B(C_1) < A(C_1)$.

Alice playing as 2nd on $C_1$, for $i \neq 1$, obeyed the rules of the $*$-game and took the last vertex played there, which in particular means that

$$A(C_i) \geq B(C_i), \quad \text{for } i \neq 1.$$  

This together with $B(C_1) < A(C_1)$ ensures that Alice has gathered more than Bob from all the components.

Case 3: Bob takes a vertex from $C_i$ with $i \neq 1$.

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

$$A(C_j) \geq B(C_j), \quad \text{for } j \neq 1, i.$$  

She also took the last vertex played in $C_1$ and therefore by the property of $S_1^i(C_1)$ she has secured that

$$B(C_1) - A(C_1) \leq \text{val}^*(C_1).$$  

The STOP on $C_i$ ends the $*$-game on $C_1$. The strategy $S_2^i(C_i)$ realized by Alice guarantees that at this moment she has gathered at least $\text{val}^*(C_i)$ more than Bob from $C_i$:

$$\text{val}^*(C_i) \leq A(C_i) - B(C_i).$$  

Composing the inequalities we get

$$B(C_1) - A(C_1) \leq \text{val}^*(C_1) \leq \text{val}^*(C_i) \leq A(C_i) - B(C_i).$$
Therefore, Alice has secured at least as much as Bob from $C_1$, $C_i$ together, and she is ahead on all the other $C_j$’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 $C_i$’s except $C_1$ and obeying the rules of the $\star$-game has guaranteed that

$$A(C_i) \geq B(C_i), \quad \text{for } i \neq 1.$$  

She took the last vertex of $C_1$ playing as 1st, so by the property of $S^*_1(C_1)$ she has secured that

$$B(C_1) - A(C_1) \leq \text{val}^*(C_1).$$

Summing up for all components of $R$ we get

$$B(R) - A(R) \leq \text{val}^*(C_1), \quad B(R) + A(R) = w(R),$$

$$A(R) \geq \frac{1}{2}(w(R) - \text{val}^*(C_1)).$$

4. Improved strategy for odd trees

Now we show how to use Claim 3.1 to derive a strategy for Alice securing $\frac{1}{3}$ of any tree with an odd number of vertices. The core idea is the same as in the proof of Proposition 2.6 (which yields $\frac{1}{5}$), 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 $S$ be any subcomponent of a tree $T$, that is, any proper subtree of $T$ such that $T - S$ is connected. It has a unique root—the vertex adjacent to $T - S$. Recall the rule of the rooted game on $S$: the root is taken first, and each next taken vertex is adjacent to some previously taken vertex. Compared to the $\star$-game on $S$, there are no further restrictions on the moves of 2nd and no STOPs. Define $\text{val}_1(S)$ to be the maximum gain of 1st in the rooted game on $S$, that is, the maximum total weight of the vertices taken by 1st that he can secure. Similarly, define $\text{val}_2(S)$ to be the maximum gain of 2nd in the rooted game. Clearly, $\text{val}_1(S) + \text{val}_2(S) = w(S)$.

The following explains the relationship of these to the $\star$-value of $S$.

**Lemma 4.1.** If $S$ has an odd number of vertices then $\text{val}^*(S) \leq \text{val}_2(S)$.

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

**Proof of Theorem 2.3.** Let $T$ be a tree with an odd number of vertices. For convenience we scale the weights on $T$ so that they sum up to 1. We prove by induction on the number of vertices of $T$ that Alice can secure at least $c \cdot w(T) = c$ on $T$.

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 $\frac{4}{5}$. Consider three cases:

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

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

$$c \cdot w(T - E) + \text{val}_2(E) \geq c \cdot w(T - E) + c \cdot w(E) \geq c.$$  

**Case 2:**

(i) Every even subcomponent $E$ of $T$ has $\text{val}_1(E) \geq (1 - c)w(E)$.

(ii) There is an odd subcomponent $O$ of $T$ with $w(O) \leq \frac{1}{2}$ and $\text{val}_2(O) \geq c$.

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

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

$$\text{val}_1(E) \geq (1 - c)w(E) = (1 - c)(w(T) - w(O)) \geq \frac{1}{2}(1 - c).$$

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

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

In this setting we apply the argument from the proof of Proposition 2.6, but in place of Claim 2.5 we plug Claim 3.1.

Alice starts with $v$ a center of $T$. Bob responds by taking some vertex $b$. Since all components of the remaining part $T - \{v, b\}$ have weight at most $\frac{1}{2}$, for each $O$ an odd subcomponent of $T - \{v, b\}$ we have $\text{val}_2(O) \leq c$. Therefore, by Lemma 4.1, $\text{val}^*(O) \leq c$.

Now, Alice applies the strategy from Claim 3.1. As a result she gets at least $\frac{1}{2}(1 - w(v) - w(b) - c)$ from $T - \{v, b\}$. Therefore, her total gain is at least

$$w(v) + \frac{1}{2}(1 - w(v) - w(b) - c) \geq \frac{1}{2}(1 - w(b) - c).$$

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

$$\max\left(\frac{1}{2}(1 - w(b) - c), w(b)\right) \geq \frac{1}{2}(1 - c) \geq c,$$

where the latter inequality holds for $c \leq \frac{1}{3}$. 

\hfill $\square$

5. **Game R: Remaining part connected**

When the rule demands to keep the remaining part connected, possibly (and very likely) not all vertices are allowed to be taken at the very first move. For instance, in case of a tree only the leaves are available. It is not hard to find examples of weighted graphs of either parity that are very bad for Alice.

**Example 5.1.** $P_3$ is a weighted path with three vertices: the middle one has weight 1 an the two others have weight 0. Clearly, Alice cannot guarantee herself anything on $P_3$ in game R.

**Example 5.2.** $G_n' = (V, E, w)$ is a weighted graph similar to $G_n$ from Example 2.1 but with switched weights. Thus, $V = \{a_1, \ldots, a_n, b_1, \ldots, b_n\}$, the $b_i$’s form a clique, and each $a_i$ has only one neighbor $b_i$. The weights are: $w(a_i) = 0$ and $w(b_i) = 1$. Alice has no strategy to gather more than 1 from $G_n'$ in game R.
In contrast to Example 2.1, here the clique on the $b_i$'s cannot be replaced by any simpler graph. It would be interesting to see an example of a $K_n$-free graph on which Alice cannot guarantee herself more than $\frac{1}{2}$ of the total weight.

Cibulka et al. [1] have provided sequences of highly connected graphs of either parity with Alice’s guaranteed gain tending to zero.

**Theorem 5.3.** Alice can secure at least $\frac{1}{4}$ of the weight of any tree with an even number of vertices in game $R$.

**Proof.** A two-colored tree is a tree whose vertices are colored black or white so that no two adjacent vertices have the same color. We are going to prove that given a two-colored tree $T$ with an even number of vertices Alice can secure at least $\frac{1}{2}$ of the total weight of a color she chooses. This yields at least $\frac{1}{4}$ of the total weight of $T$ for Alice.

The proof goes by induction on the number of vertices of $T$ (only even numbers). For a tree with two vertices the statement is trivial. To get through the induction step we construct a strategy for Alice which leads the game to a point (after some Bob's move) at which the weight of black vertices she has taken is at least the weight of black vertices taken by Bob. We distinguish two cases.

*Case 1:* $T$ has a black leaf.

This is an easy case. Alice takes the heaviest available black vertex. With this move she does not uncover any new black vertex, so Bob can only respond with a white vertex or a non-heavier black vertex. In both scenarios Alice gains no less than Bob from the black part of the graph, and for the rest of the tree the induction hypothesis can be applied.

*Case 2:* All leaves of $T$ are white.

First observe that there must be a black vertex of degree greater than 2: if all black vertices have degree 2 then the total number of edges is even, so $T$ has an odd number of vertices, which contradicts the assumption. Let $K$ be the subtree of $T$ spanned by all black vertices of degree greater than 2. We call $K$ the core of $T$, and we call connected components of $T - K$ simply components. The root of a component is its only vertex adjacent to the core. A similar argument with counting edges shows that a component with a white root has an odd number of vertices and a component with a black root has an even number of vertices (see Figure 4). Since all leaves of the core are black and have degree greater than 2, at least two of their (white) neighbors must be roots of components. Therefore, taking a vertex from the core is possible only after at least two components with an odd number of vertices have been entirely shared out.

![Figure 4](image-url)
Let $C_1, \ldots, C_k$ be the components, and assume $C_1$ is a component with an odd number of vertices and the least total weight of black vertices among all components with an odd number of vertices.

Alice starts with any vertex from $C_1$. The following invariants are to be kept till the moment the induction hypothesis is applied:

(i) For $i \neq 1$, all vertices in $C_i$ available for Bob are white.
(ii) No component with an odd number of vertices other than $C_1$ has been entirely shared out.

Subsequent moves of Alice depend on what Bob has just played. Observe that by (ii) no vertex from the core is available. Thus, Bob can choose only from the vertices in the components.

**Case 1:** Bob takes a vertex from $C_1$.

Alice takes another vertex of $C_1$. Such a vertex exists as $C_1$ has an odd number of vertices. Clearly, both invariants are preserved.

**Case 2:** Bob takes a vertex $v$ from $C_i$ with $i \neq 1$.

If $v$ is not the root of $C_i$ then Bob’s move uncovers exactly one black vertex in $C_i$ which is now a new leaf. Alice takes it. It may be the root of $C_i$ but then $C_i$ has an even number of vertices, so taking the last vertex of $C_i$ does not violate (ii). All vertices that remain available in $C_i$ are white exactly as (i) states.

If $v$ is the root of $C_i$ then $C_i$ has an odd number of vertices and all of them have been taken. Note that all black vertices taken by Bob are in $C_1$, and Alice has taken all black vertices from $C_i$. Since the weight of black vertices in $C_i$ is at least the weight of black vertices in $C_1$, Alice has collected at least as much as Bob from the black part. For the remaining tree the induction hypothesis is applied.

Have you noticed an analogy between the strategies on components above and in the proof of Claim 3.1? Here it goes. First observe that the rooted game on a rooted tree-component $C$ can be defined as well in the variant with remaining part connected, by the following rule: a vertex of $C$ can be taken only if this does not disconnect any remaining vertex of $C$ from the root of $C$. It describes what can be played in $C$ as long as some vertices still remain in $T - C$. This yields the corresponding $\ast$-game—conditions (ii) and (iii) in the definition of $\ast$-game remain unchanged. This new $\ast$-game keeps all key properties of the old one.

Now suppose the weights of all white vertices of $T$ are set to zero. Then for each component $C$ of $T - K$ we have $\text{val}^\ast(C) = w(C)$ if $C$ is odd, and $\text{val}^\ast(C) = \infty$ if $C$ is even. Alice’s strategy from the above proof, which leads to her advantage after some Bob’s move, is exactly the same as the strategy from Claim 3.1 (with the new variant of $\ast$-game). After Alice has gained advantage she can play inductively on the rest of the tree thus securing $\frac{1}{2}$ of the total (modified) weight.

We can try a similar reasoning for the original weights (without two-coloring). Suppose we choose a core so that Alice, following her $\ast$-dependent strategy on the resulting components, can always gain advantage before Bob can enter the core. Then she gets at least $\frac{1}{2}$ of the taken part and hopefully she can play inductively on the rest of the tree. The problem is that such a core may not exist. In order to prove that Alice can secure at least $\frac{1}{2}$ of the entire tree, we need some global strategy that would solve the case where choosing a core is not possible.

**Conjecture 5.4.** Alice can secure at least $\frac{1}{2}$ of the weight of any tree with an even number of vertices in game $R$.

The conjecture is confirmed for all subdivisions of stars, that is, for all trees with at most one vertex of degree greater than 2.
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 on the class of graphs with an odd number of vertices excluding a $K_n$-minor (see Conjecture 2.4). This is confirmed for odd trees. On the way to verify this belief one can inspect other specific classes of graphs with a forbidden minor (and an odd number of vertices) for which the question remains open: outerplanar graphs, planar graphs, graphs with bounded treewidth (however, $k$-trees are solved by Proposition 2.7). The exact value of Alice’s maximum guaranteed gain for odd trees lies between $\frac{1}{4}$ (by Theorem 2.3) and $\frac{2}{5}$ (witnessed by the tree in Figure 5).

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. We also conjecture that Alice can secure at least $\frac{1}{2}$ of any tree with an even number of vertices. This has been confirmed for all subdivisions of stars.

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. We wonder whether it can be adapted to construct effective strategies for other known combinatorial games.

Complexity-type problems concerning graph sharing games have been raised in [1]. In particular, it has been proved that deciding which player has a strategy to gather more than $\frac{1}{2}$ from a given graph is PSPACE-complete for game R and for game TR. Whether the same is true for game T is left open.

References


Department of Theoretical Computer Science, Jagiellonian University, Lojasiewicza 6, 30-348 Kraków, Poland
E-mail address: Piotr.Micek@tcs.uj.edu.pl
URL: http://tcs.uj.edu.pl/Micek
E-mail address: Bartosz.Walczak@tcs.uj.edu.pl
URL: http://tcs.uj.edu.pl/Walczak