A NOTE ON CONCURRENT GRAPH SHARING GAMES

Steven Chaplick[1]
Lehrstuhl für Informatik I, Universität Würzburg, Würzburg, Germany
steven.chaplick@uni-wuerzburg.de

Piotr Micek[2]
Theoretical Computer Science Department, Jagiellonian University, Kraków, Poland
piotr.micek@tcs.uj.edu.pl

Torsten Ueckerdt
Department of Mathematics, Karlsruhe Institute of Technology, Karlsruhe, Germany
torsten.ueckerdt@kit.edu

Veit Wiechert[3]
Institut für Mathematik, Technische Universität Berlin, Berlin, Germany
wiechert@math.tu-berlin.de

Received: 1/19/15, Revised: 10/5/15, Accepted: 1/1/16, Published: 1/29/16

Abstract

In the concurrent graph sharing game, two players, called 1^{st} and 2^{nd}, share the vertices of a connected graph with positive vertex-weights summing up to 1 as follows. The game begins with 1^{st} taking any vertex. In each proceeding round, the player with the smaller sum of collected weights so far chooses a non-taken vertex adjacent to a vertex which has been taken, i.e., the set of all taken vertices remains connected and one new vertex is taken in every round. (It is assumed that no two subsets of vertices have the same sum of weights.) One can imagine the players consume their taken vertex over a time proportional to its weight, before choosing a next vertex. In this note we show that 1^{st} has a strategy to guarantee vertices of weight at least 1/3 regardless of the graph and how it is weighted. This is best-possible already when the graph is a cycle. Moreover, if the graph is a tree 1^{st} can guarantee vertices of weight at least 1/2, which is clearly best-possible.

[1]S. Chaplick is supported by the ESF research project EUROGIGA GraDR.
[2]P. Micek is supported by the Mobility Plus program from The Polish Ministry of Science and Higher Education.
[3]V. Wiechert is supported by the Deutsche Forschungsgemeinschaft within the research training group 'Methods for Discrete Structures' (GRK 1408).

1. The Result

Imagine a pizza, sliced as usually into triangular pieces, not necessarily of the same size, and two players alternatingly taking slices in such a way that every slice, except the first one, is adjacent to a slice that was taken earlier. What is the fraction of the total size of the pizza that the first player can guarantee to get at least, independently of the number of slices and their sizes (weights)? This problem, the so-called *Pizza Problem*, posed by Peter Winkler was resolved in [1, 2] and it turns out that 1st can always guarantee to get at least 4/9 of the entire pizza and that this is best-possible. Considering a pizza to be a cycle with weights on its vertices, one can find work on similar games for trees [3, 4] and subdivision-free graphs [5].

The *concurrent graph sharing game* is a variant of the Pizza Problem introduced by Gao in [6] (as *the Pizza Race Game* and its generalizations). As before, a vertex-weighted graph is shared by 1st and 2nd taking one vertex at a time in such a way that the set of all taken vertices remains connected. The game begins with 1st taking any vertex. However, in each proceeding round, the player with the smaller sum of collected weights so far picks the next non-taken vertex. (We assume for now that no two subsets of vertices have the same sum of weights and discuss the situation without this assumption at the end of the paper.) One can imagine the players consume their taken vertex over a time proportional to its weight, before choosing a next vertex.

For convenience, assume that the weights of all the vertices in the graph sum up to 1. In [6] the author claims that for every weighted cycle 1st can guarantee to take vertices of total weight at least 2/5. However, his proof has a flaw and cannot be fixed. In fact, he starts by introducing a new vertex with vanishingly small weight between any two adjacent vertices of the given cycle, and claims that these vertices are irrelevant for the analysis of the game. But this is true only as long as 1st does not start with such a vertex. (When 1st starts with an original vertex Gao's argument for a 2/5 lower bound seems to be correct). Indeed, we show here that the maximum total weight that 1st can guarantee on every cycle is 1/3. In fact, our lower bound argument works for every graph, i.e., 1st can always guarantee to take vertices of total weight at least 1/3.

Secondly, Gao asks whether 1st can guarantee any positive fraction of the total weight if the game is played on a tree. We show here with an easy strategy stealing argument that, playing on trees, 1st can always guarantee to take vertices of total weight at least 1/2, which is clearly best-possible.

An *instance* of the concurrent graph sharing game is a pair (G, w) of a graph $G = (V, E)$ and positive real vertex weights $w : V \to (0, 1]$ with $\sum_{v \in V} w(v) = 1$. For a subset $A \subseteq V$ of vertices we denote $w(A) = \sum_{a \in A} w(a)$. For a vertex $a \in V$, let F_a and S_a be the subsets of vertices that 1st and 2nd take when 1st starts with

a, and from then on both players play optimally subject to maximizing $w(F_a)$ and $w(S_a)$, respectively. Thus, $F_a \sqcup S_a = V$ for all $a \in V$. The *value of an instance* (G, w) is the maximum total weight $v(G, w)$ of vertices that 1st can guarantee to take in this instance. In particular, $v(G, w) = \max_{a \in V(G)} w(F_a)$.

Theorem 1. *For the concurrent graph sharing game we have*

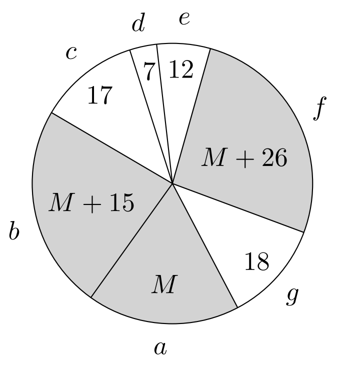
$$\inf_{(G,w)} v(G, w) = 1/3 \quad \text{and} \quad \inf_{(G,w),\ G \text{ is a tree}} v(G, w) = 1/2.$$

Recall that in the above theorem we consider only instances (G, w) in which no two disjoint subsets of vertices have the same weight. However, as explained below, we use this hypothesis only for a strategy stealing argument proving the lower bound of 1/2 for trees. We remark that a similar strategy stealing is part of Gao's proof.

Proof. To prove $\inf_{(G,w)} v(G, w) \geqslant 1/3$, let (G, w) be any instance of the concurrent graph sharing game. If there is a vertex $a \in V(G)$ with $w(a) \geqslant 1/3$, then clearly $v(G, w) \geqslant w(F_a) \geqslant 1/3$. On the other hand, if $w(a) < 1/3$ for all $a \in V(G)$, then at the moment 1st can take no further vertex (because all vertices are already taken), 2nd's current vertex has weight less than 1/3. So for every $a \in V(G)$ we have $w(S_a) - w(F_a) < 1/3$. Together with $w(F_a) + w(S_a) = 1$ this implies that $v(G, w) \geqslant w(F_a) > 1/3$.

Next we shall prove $\inf_{(G,w)} v(G, w) \leqslant 1/3$ by providing for every $\varepsilon > 0$ an instance (G, w_ε) with $v(G, w_\varepsilon) \leqslant 1/3 + \varepsilon$. Consider the cycle G consisting of seven vertices a, b, c, d, e, f, g in this cyclic order and corresponding vertex-weights $M, M + 15, 17, 7, 12, M + 26, 18$, where $M = M(\varepsilon) \gg 95$ is large enough. This instance (G, w_ε) is depicted in Figure 1 in form of a pizza. It contains three pieces with weight at least M, which we call *heavy*. For 2nd to get at least two heavy pieces (and therefore roughly 2/3 of the entire pizza) he moves according to the table on the right and then takes the last heavy piece when it is his turn again. E.g., when 1st starts with d, 2nd takes e, and if 1st continues with c, 2nd takes f, and then 2nd is guaranteed to get the last heavy piece (either a or b in this case).

We now consider instances in which the underlying graph is a tree. To prove that $\inf_{(G,w),\ G \text{ tree}} \geqslant 1/2$ let (G, w) be any instance where G is a tree. If $|V(G)| = 1$, then clearly $v(G, w) = 1$. Otherwise, for each vertex $a \in V(G)$, let $b(a) \in Y_a$ be the first vertex 2nd takes when 1st starts with a. As $|E(G)| < |V(G)|$ there exists an edge aa' such that $b(a) = a'$ and $b(a') = a$. Consider the games in which 1st starts with a and a', respectively, and both players play optimally. In the former game 1st starts with a and 2nd answers with a', while in the latter it is the other way around. In particular, from that moment on both games are identical, but the roles of 1st and 2nd are switched. It follows that $w(X_a) = w(Y_{a'})$ and with $w(Y_{a'}) = 1 - w(X_{a'})$ we conclude $v(G, w) \geqslant \max\{w(X_a), w(X_{a'})\} \geqslant 1/2$.



1st moves	2nd moves	1st's outcome
$a \to g$	$b \to f$	$M+54$
$a \to c$	$b \to d$	$M+47$
b	cda	$M+45$
c	db	$M+55$
$d \to f$	$e \to gab$	$M+50$
$d \to c$	$e \to f$	$M+57$
e	f	$M+69$
f	gab	$M+62$
g	a	$M+45$

Figure 1: A pizza (vertex-weighted cycle) with total weight $3M + 95$ in which 1st cannot guarantee to get more than $M + 69$, for any M large enough. Divide each weight by $3M + 95$ to get a total sum of weights equal to 1.

To see that $\inf_{(G,w),\ G \text{ is a tree}} \leqslant 1/2$, consider for every $\varepsilon > 0$ a tree consisting of a single edge ab with $w(a) = (1-\varepsilon)/2$ and $w(b) = (1+\varepsilon)/2$. □

We remark that the assumption that no two subsets of vertices have the same sum of weights is crucial for the strategy stealing argument we used for the trees. Indeed, if both players may finish their current vertex at the *same* time and in such cases, say, 1st always takes the next vertex, then the best 1st can guarantee on any tree is 1/3, instead of 1/2. For the upper bound see Figure 2. On the other hand, note that the general lower bound of 1/3 remains valid, no matter how those "ties" are broken.

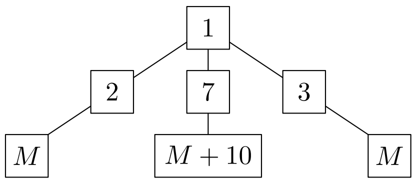


Figure 2: A vertex-weighted tree with total weight $3M + 23$ in which 1st cannot guarantee to get more than $M + 20$, for any M large enough. Divide each weight by $3M + 23$ to get a total sum of weights equal to 1.

Lastly, we remark that in the original Pizza Problem most of the effort was to find a strategy for 1st to get at least 4/9 of the pizza. The concurrent graph sharing games considered in this paper turned out to be somewhat simpler in the analysis. Indeed, here the strategy for 1st to get 1/3 of the total weight in any graph is obvious and the difficulty was to believe that this is best-possible already for cycles.

Tight examples were found by working out the sequence of moves in an optimal strategy for 2nd, which led to a system of linear constraints for the vertex-weights whose optimization gave a best-possible scenario for 2nd.

References

[1] J. Cibulka, J. Kynčl, V. Mészáros, R. Stolař, and P. Valtr. Solution to Peter Winkler's pizza problem, *Fete of Combinatorics and Computer Science*, *Bolyai Soc. Math. Stud.* **20** (2010), 69-93.

[2] K. Knauer, P. Micek, and T. Ueckerdt. How to eat $\frac{4}{9}$ of a pizza, *Discrete Math.* **311** (2011), 1635-1645.

[3] P. Micek and B. Walczak. A graph-grabbing game, *Combin. Probab. Comput.* **20** (2011), 623-629.

[4] D. Seacrest and T. Seacrest. Grabbing the gold, *Discrete Math.* **312** (2012), 1804-1806.

[5] A. Gagol, P. Micek, and B. Walczak. Graph sharing game and the structure of weighted graphs with a forbidden subdivision, *arXiv-preprint* (2014) http://arxiv.org/abs/1411.6727.

[6] K. Gao. Pizza race problem, *arXiv-preprint* (2012) http://arxiv.org/abs/1212.2525.