Discussiones Mathematicae
Graph Theory 32 (2012) 181–185

Note

THE FIRST PLAYER WINS THE ONE-COLOUR TRIANGLE AVOIDANCE GAME ON 16 VERTICES

Przemysław Gordinowicz

Institute of Mathematics
Technical University of Lodz
Łódź, Poland

e-mail: pgordin@p.lodz.pl

and

Paweł Prałat

Department of Mathematics
West Virginia University
Morgantown, WV 26506–6310, USA

e-mail: pralat@math.wvu.edu

Abstract

We consider the one-colour triangle avoidance game. Using a high performance computing network, we showed that the first player can win the game on 16 vertices.

Keywords: triangle avoidance game, combinatorial games.

2010 Mathematics Subject Classification: 05C57, 05C35.

1. Introduction

In this note, we consider the one-colour graph avoidance game. Let G be a fixed graph on n_0 vertices, and let $n \geq n_0$ be an integer. The game between two players, first and second, starts with n isolated vertices. In each turn, a player draws a new edge. Players alternate turns, starting with the first player. We deal with simple graphs only so it is forbidden to create parallel edges and loops. Both players have the same goal, namely to try to avoid creating G as a subgraph. Since $n \geq n_0 = |V(G)|$, it is unavoidable that eventually one player is forced to create a copy of G and loses the game. Since this is a two-person, full information game with no ties, either the first player or the second player has a winning strategy (namely, the strategy to achieve a maximal G-free graph, that is, a G-free graph with the property that the addition of any edge creates a copy of G).

A well-known, interesting, and highly nontrivial game is when the players try to avoid creating a triangle (see [2] for more details on this and other variations). The outcomes for $n \leq 9$ were reported by Seress [6]. For several years only two more values were known (namely, for $n = 10$ and $n = 11$) and it was conjectured that the first player wins if, and only if, $n \equiv 2 \pmod 4$. However, Cater, Harary, and Robinson [1] managed to show, using computer support, that the conjecture fails for $n = 12$. The second author of this note, showed that the first player wins the game on 13, 14, and 15 vertices [5]. The total computational requirements to solve the triangle avoidance game for $n = 15$ were estimated to be 1,440 CPU hours (2 months) on a 2.2GHz computer. The other results noted were easier to obtain.

It remains an open problem to show who has a winning strategy for any n. On the other hand, it has been shown that the first player wins the connected variant of the game if, and only if, n is even [6]. In this short note we report that the first player wins the game on 16 vertices, which supports the conjecture that this is the case for each $n \geq 12$. (We do admit that the conjecture is rather bold; the only reason to pose it, beyond the values in Table 1, is that almost all combinatorial games are first player win [7].) The total computational requirements to analyze the game for $n = 16$ we estimate to be 12,500 CPU hours (≈ 1.46 years). (Two independent experiments have been done so, in fact, the process took roughly 25,000 CPU hours.) Generating all triangle-free graphs and the preparation process took us roughly 2,200 CPU hours (≈ 90 days). The number of triangle-free graphs on 17 vertices is about $3 \cdot 10^{13}$ and generating them all would take about one year on a 4GHz computer [4]. Since the generating process takes a tiny fraction of all time required to solve the problem, it seems that there is no hope to solve the game by exhaustive computation for $n \geq 17$. We estimate the total computational time required to analyze the game for $n = 17$ to be 47 CPU years. Moreover, there would be a problem with disk space. During the process of solving the game for $n = 16$, disk space of approximately 20TB was needed. The game for $n = 17$ would require much more disk space.

2. Tools Used to Obtain the Result

In order to obtain this result, we generated a family $\mathcal{H}$ of all non-isomorphic triangle-free graphs on n vertices using Brendan McKay's `nauty` software package [3] for computing automorphism groups of graphs and digraphs. Let $h(n) = |\{G : G \in \mathcal{H}\}|$ denote the number of such graphs and let $e(n) = \max\{|E(G)| : G \in \mathcal{H}\}$ be the number of edges in the densest graph in this family. Let us note that $e(n) = \lfloor n^2/4 \rfloor$ by Mantel's theorem. It is clear that all graphs with $e(n)$ edges have the property that the next player to move loses the game. We call those graphs *previous player wins* graphs, and denote corresponding subfamily by $P_{e(n)}$. Now, we partition the set of all graphs on $e(n) - 1$ edges into *previous player wins* graphs ($P_{e(n)-1}$) and *next players wins* ones ($N_{e(n)-1}$). In order for a graph G to be in $N_{e(n)-1}$, it is required that there is an edge $e \notin E(G)$ such that after adding e to G we get a graph which is in $P_{e(n)}$. (The next player should draw e to force the opponent to give up.) Since this is also a sufficient condition,

$$\begin{aligned} N_{e(n)-1} &= \{G \in \mathcal{H} : G = H \setminus \{e\} \text{ for some } H \in P_{e(n)} \text{ and } e \in E(H)\}, \\ P_{e(n)-1} &= \{G \in \mathcal{H} : |E(G)| = e(n) - 1\} \setminus N_{e(n)-1}. \end{aligned}$$

Those operations can be done easily with the support of the `nauty` software package to compute a canonical labeling for each triangle-free graph encountered, so that only one isomorphic copy of each is explored in the game tree.

n	$w(n)$	$e(n)$	$h(n)$
3	2	2	3
4	2	4	7
5	2	6	14
6	1	9	38
7	2	12	107
8	2	16	410
9	2	20	1,897
10	1	25	12,172
11	2	30	105,071
12	1	36	1,262,180
13	1	42	20,797,002
14	1	49	467,871,369
15	1	56	14,232,552,452
16	1	64	581,460,254,000

Table 1. Triangle avoidance game.

Now, we can determine the families P_i and N_i ($i = e(n) - 2, e(n) - 3, \ldots, 0$) recursively. If the only graph with no edge in $\mathcal{H}$ (the empty graph) is in N_0, then the first player wins the game (we put $w(n) = 1$); otherwise the second player has a winning strategy ($w(n) = 2$). A UNIX script used to solve the problem can be found in [8]. Below we present the results of our program (Table 1 and Table 2 in the Appendix section).

3. Appendix

i	$\lvert P_i\rvert$	$\lvert N_i\rvert$	i	$\lvert P_i\rvert$	$\lvert N_i\rvert$
0	0	1	33	15,005,640,656	38,493,524,091
1	1	0	34	6,642,333,038	31,554,879,683
2	0	2	35	7,821,394,916	16,897,642,360
3	4	0	36	2,868,189,306	11,780,092,401
4	0	9	37	2,960,556,173	5,087,150,776
5	18	1	38	874,742,979	3,280,151,880
6	1	44	39	879,858,398	1,162,753,034
7	101	4	40	218,656,966	748,486,734
8	3	264	41	218,040,317	226,777,897
9	682	18	42	51,703,473	148,101,829
10	9	1,935	43	46,002,839	41,890,058
11	5,561	79	44	11,983,510	25,914,396
12	30	17,194	45	8,370,238	7,646,012
13	54,311	352	46	2,561,848	4,071,091
14	102	179,422	47	1,348,857	1,343,422
15	600,280	1,715	48	482,097	590,349
16	377	2,033,291	49	200,097	219,866
17	6,792,986	15,922	50	80,487	81,862
18	2,312	22,221,923	51	28,525	33,610
19	68,755,572	850,340	52	12,479	11,217
20	89,676	206,172,723	53	4,066	4,977
21	520,377,177	50,893,327	54	1,909	1,581
22	5,855,066	1,457,842,619	55	603	752
23	2,379,102,854	1,060,655,670	56	307	231
24	153,164,036	7,208,782,220	57	96	120
25	6,640,506,030	7,625,458,377	58	53	34
26	1,313,210,807	23,595,969,182	59	16	21
27	13,048,146,475	25,987,132,204	60	11	6
28	4,752,262,803	49,979,012,609	61	3	4
29	18,967,193,628	49,532,711,838	62	2	1
30	8,977,248,881	67,444,470,537	63	1	1
31	20,082,656,309	55,897,791,178	64	1	0
32	9,875,275,876	57,523,172,470		124,403,496,235	457,056,757,765

Table 2. Triangle avoidance game on 16 vertices — more details.

Acknowledgement

This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network SHARCNET, Ontario, Canada (www.sharcnet.ca): 10,808 CPUs. The programs used to obtain the result can be downloaded from [8].

References

[1] S.C. Cater, F. Harary and R.W. Robinson, *One-color triangle avoidance games*, Congr. Numer. **153** (2001) 211–221.

[2] F. Harary, *Achievement and avoidance games for graphs*, Ann. Discrete Math. **13** (1982) 111–119.

[3] B.D. McKay, `nauty` Users Guide (Version 2.4), `http://cs.anu.edu.au/~bdm/nauty/`.

[4] B.D. McKay, personal communication.

[5] P. Prałat, *A note on the one-colour avoidance game on graphs*, J. Combin. Math. and Combin. Comp. **75** (2010) 85–94.

[6] Á. Seress, *On Hajnal's triangle-free game*, Graphs and Combin. **8** (1992) 75–79.

[7] D. Singmaster, *Almost all partizan games are first person and almost all impartial games are maximal*, J. Combin. Inform. System Sci. **7** (1982) 270–274.

[8] A UNIX script and programs written in C/C++ used to solve the problem, `http://www.math.wvu.edu/~pralat/index.php?page=publications`.

Received 8 December 2010
Revised 7 March 2011
Accepted 8 March 2011