6-2011

The graph distance game

Wayne Goddard

Anne Sinko
College of Saint Benedict/Saint John's University, asinko@csbsju.edu

Peter J. Slater

Honghai Xu

The Graph Distance Game

Wayne Goddard
School of Computing and Dept of Mathematical Sciences
Clemson University, Clemson SC 29634 USA
goddard@clemson.edu

Anne Sinko
Department of Mathematics
Kenyon College, Gambier OH 43022 USA
sinkoa@kenyon.edu

Peter Slater
Computer Science Department and Mathematical Sciences Department
University of Alabama in Huntsville, Huntsville AL 35899 USA
pslater@cs.uah.edu, slaterp@math.uah.edu

Honghai Xu
Dept of Mathematical Sciences
Clemson University, Clemson SC 29634 USA
honghax@clemson.edu

Abstract

In the graph distance game, two players alternate in constructing a maximal path. The objective function is the distance between the two endpoints of the path, which one player tries to maximize and the other tries to minimize. In this note, we examine the distance game for various graphs, and provide general bounds, exact results for special graphs, and an algorithm for trees. Computer calculations suggest interesting conjectures for grids.

1 Introduction

There are many games in the combinatorial literature. In many, the winner is determined by who moves last, as studied for example in [2]. In others, the players compete to construct a desired goal, by taking one element at a time from the universe, as studied for example in [1]. Another idea is games where the players compete to maximize or minimize some quantity, such as the game chromatic number introduced in [3], the competition chromatic number introduced in [6], or graph competition independence introduced in [4]. We consider here a game that falls in the latter category. (See [5] for more on such competitive games.)

Consider the following game. A graph G is given. Two players alternate in constructing a path. The first player picks a vertex, the second player picks a neighbor of the first vertex, the first player picks a neighbor of the second vertex that has not yet been picked, and so on. This is continued until the path cannot be extended. One player tries to maximize the final distance from the start, and the other player tries to minimize this distance. That is, the value at the end of the game is the distance between the start and the finish, regardless of the path taken. For a trivial example, in the complete graph the value is always 1. We call this the *distance game*.

There are two versions, depending on which of the Minimizer or Maximizer moves first. We let $\mathcal{S}_m(G)$ denote the value of the game on graph G when the minimizer chooses the first vertex, and $\mathcal{S}_M(G)$ the value when the maximizer chooses the vertex. We call the first vertex of the path the *source*. Clearly if the graph is vertex transitive, all sources are equivalent. If it does not matter who goes first, then we drop the subscript and write $\mathcal{S}(G)$.

In this note, we explore the distance game for various graphs. We provide general bounds and exact results for simple graphs. We also show that the parameter can be calculated in a tree in linear time. Further, we determine the values for small grids, and present computer calculations that suggest some interesting conjectures for general grids.

2 Basics

There are obvious upper bounds based on the radius and diameter of the graph. (The second bound holds since the minimizer can start at a central vertex.)

Observation 1 *For any graph G, $\mathcal{S}_M(G) \leq \operatorname{diam}(G)$ and $\mathcal{S}_m(G) \leq \operatorname{rad}(G)$.*

One can obtain a slight improvement on these bounds for bipartite graphs by using a result about another game, given as an exercise in [7]. That exercise is about a game with the same idea—building a maximal path—but the objective of the game is that the person who makes the last move wins. Let us call the game LPW. The exercise shows that LPW is a win for the first player if and only if the graph does not have a perfect matching. We have a completely different objective function, but that result shows the following:

Observation 2 *Let graph G be bipartite. If G has no perfect matching, then $\mathcal{S}_M(G) > 1$. If G has a perfect matching and the diameter is even, then $\mathcal{S}_M(G) < \mathrm{diam}(G)$.*

Proof. If there is no perfect matching, then LPW is a win for the first player. So if the maximizer goes first, she can ensure she moves last. By bipartiteness, this means the path ends at a nonneighbor of the source. If there is a perfect matching, then LPW is a win for the second player. Therefore if the minimizer goes second, he can ensure he moves last, and thus the path ends at odd distance from the source. Since the diameter is even, this implies the result. □

These bounds apply to the complete bipartite graphs, for example, as we see next.

3 Solved Graphs

Observation 3
(a) For a path, $\mathcal{S}_M(P_n) = n - 1$ and $\mathcal{S}_m(P_n) = \lfloor n/2 \rfloor$.
(b) For a cycle, $\mathcal{S}(C_n) = 1$.
(c) For complete bipartite graph: $\mathcal{S}_m(K_{r,s}) = 1$; and $\mathcal{S}_M(K_{r,s}) = 1$ if $r = s$, and 2 otherwise.

Proof. We prove (c). If $r = s$ then the path will use up all vertices and end on the opposite side to the source, no matter what. If $r \neq s$, then the path always ends on the larger side. So the only freedom anybody has is when the first player chooses the source. The minimizer wants the source on the smaller side, and the other way around for the maximizer. □

The result on complete bipartite graphs has a simple extension to complete multipartite graphs:

Observation 4 *For nonbipartite complete multipartite graphs with n vertices and largest partite set of size m,*
(a) $\mathcal{S}_m(G) = 1$; and
(b) $\mathcal{S}_M(G) = 1$ if $m < n/2$, and $\mathcal{S}_M(G) = 2$ otherwise.

PROOF. (a) The minimizer's strategy is the following: let A be any partite set other than the largest. He chooses a vertex of A as the source, and thereafter the minimizer takes a vertex in A whenever possible. The only way the minimizer is unable to take a vertex in A and A is not used up, is that the path is currently at a vertex of A. It follows that every alternate vertex of the path is in A until A is used up; and the path continues.

(b) If $m > n/2$, then the maximizer chooses a vertex in the largest partite set B as the source. No matter what happens thereafter, the path will end in B. If $m = n/2$, then the maximizer again chooses a vertex in the largest partite set B. At her second move, since G is not bipartite, she is able to choose a vertex not in B, and thereafter, no matter what happens, the path will end in B. If $m < n/2$, then the minimizer follows the strategy of choosing a vertex in the same partite set as the source whenever possible. By the same argument as above, the minimizer is able to use up these vertices before the end of the path. □

3.1 Dense Graphs

We observed earlier that the value of the game for the complete graph is trivially 1. It is not surprising that one gets a similar result for all very dense graphs:

Theorem 5 *For any graph G with n vertices and minimum degree δ, if $\delta \geq (4n-4)/5$, then $\mathcal{S}(G) = 1$.*

PROOF. The diameter of such a graph is (at most) 2. The minimizer adopts the following strategy: move to a non-neighbor of the source whenever possible (else make any move). We argue that he is able to use up all such vertices before the end of the path.

Suppose the path starts at v. Let X be the non-neighbors of v. Note that $|X| \leq n-1-\delta$. Any maximal path has length at least δ. So let P be the portion of the path constructed consisting of the first $\delta - 2$ vertices. The minimizer has at least $\delta/2 - 2$ moves (ignoring the first move, if he starts, but counting the last vertex of P if it's minimizer's move there). Suppose some vertex $x \in X$ is neither in P nor the vertex immediately after it. Then the minimizer moves to X at most $|X| - 1$ times. But x has at most $n - \delta - 2$ non-neighbors excluding the source. Since

$$(\delta/2 - 2) - ((n - 1 - \delta) - 1) > n - \delta - 2,$$

there is a point where P is at a neighbor of x with minimizer to move and minimizer does not choose a vertex of X, a contradiction. $\square$

This bound can probably be improved.

4 Grids and Related Graphs

Consider the $k \times m$ grid $\mathcal{G}_{k,m}$. We number the rows from 1 to k (from top to bottom), and the columns from 1 up to m (from left to right). The vertex in row i and column j is labeled (j, i). We use $\mathcal{C}_j$ to denote column j

4.1 Grids with two rows

Theorem 6 $\mathcal{S}_M(\mathcal{G}_{2,m}) = 1$.

Proof. Let C be the unique hamiltonian cycle. The minimizer's strategy is to always choose the next vertex along C going clockwise. The only thing to note is that when we reach the other vertex in the same column as the source, it is maximizer's turn to move; so at the next vertex, it is minimizer's turn and he can ensure that the final neighbor of the source is not used prematurely. See Figure 1 for an example path so constructed. $\square$

Theorem 7 $\mathcal{S}_m(\mathcal{G}_{2,m}) = 2\lfloor m/4 \rfloor + 1$.

Proof. The radius of the grid is $r = \lfloor m/2 \rfloor + 1$. This provides the upper bound when m is congruent to 0 or 1 modulo 4.

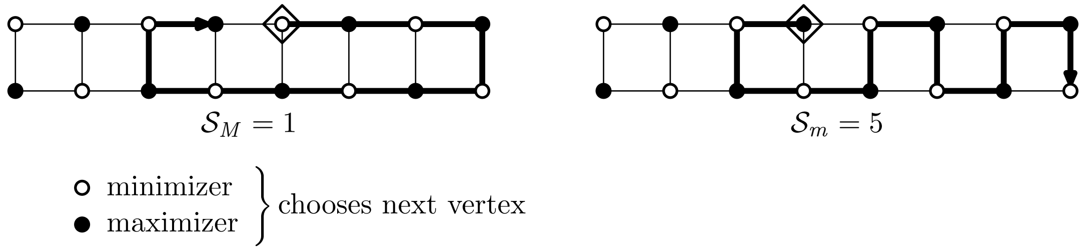

Figure 1: Example paths on the 2×8 grid

So assume m is congruent to 2 or 3 modulo 4. Then r is even. The upper bound in this case is provided by the following observation. For a vertex v, we define its eccentric vertices as those at maximum distance from it.

Claim. *If graph G is bipartite with $\delta(G) \geq 2$, the radius is even, and there exists a central vertex v whose eccentric vertices are mutually distance at least* 3 *apart, then $\mathcal{S}_m(G) < \mathrm{rad}(G)$.*

Proof of claim. The minimizer employs the strategy of starting at v, and then whenever possible, he moves to an eccentric vertex of v (else makes any move). If the path is to end at an eccentric vertex of v, say w, then by the bipartiteness and the radius being even, it is the maximizer who first moves to a neighbor of w, say y. When that occurs, the minimizer immediately chooses w, and the path continues since w has at least two neighbors. Because the eccentric vertices are far enough apart, there can be at most one eccentric vertex adjacent to y; so this does not interfere with the strategy elsewhere. This proves the claim.

The maximizer can achieve the stated value by the following strategy. After the minimizer chooses the source, start by going along C towards the nearest degree-2 vertex; continue around the hamiltonian cycle C (it does not matter if the minimizer uses a chord of C) until we reach the other vertex in the same column as the source. It will now be minimizer's move. So he must move to a new column, and then the maximizer takes the other vertex in that column. From there on, she causes the path to repeatedly snake until it ends in the last column after the maximizer has moved. See Figure 1 for an example path so constructed.

So the distance from the source to the finish is odd. Indeed, it is whichever is odd out of x or $x+1$, where x is the horizontal distance from the source column to the farthest column. Since x is always at least $\lfloor m/2 \rfloor$, we get the lower bounds from the theorem. $\square$

4.2 Grids with three rows

Our first result handles the case where the maximizer moves first in the three-row grid with an odd number of columns:

Theorem 8 $\mathcal{S}_M(\mathcal{G}_{3,m}) = m+1$ *for odd* m.

Proof. The proof is by induction on m, with the added stipulation that maximizer always starts in the top-left corner $(1,1)$. The claim is true for $m=1$ (as we have a path). So assume true for $m-2$ and test for m. Let G denote the whole grid and G' the grid without the rightmost two columns. Our basic strategy for the maximizer is to play the optimal strategy for G' as long as possible.

This means that, if the minimizer never voluntarily moves out of G', that we will reach the bottom right corner $(m-2,3)$ of G' with the path unable to continue in G'. In particular, this means that the square $(m-2,2)$ has been used. By bipartiteness, it is then minimizer's move. It is easy for the maximizer to end the game in the corner $(m,3)$, because it is her turn at both $(m-1,1)$ and $(m-1,3)$ (if the path reaches the latter) where she moves up and right respectively.

The other possibility is that the minimizer moves out of G' before the game on G' finishes. This must be a move from $(m-2,1)$, by bipartiteness. Further, since the path could have been forced to $(m-2,3)$ if we were playing in G', by planarity, we must have reached $(m-2,1)$ from its left. So after the minimizer moves to $(m-1,1)$, the maximizer plays right, the minimizer can only go down, and the maximizer plays left. If the minimizer goes down, then maximizer wins immediately by going right. But if the minimizer goes left to $(m-2,2)$, then maximizer can resume the strategy on G' and force the path to $(m-2,3)$, as before. Thereafter two right moves are forced and so the path ends at $(m,3)$. $\square$

Next we consider the remaining cases for grids with three rows. The proof idea is similarly inductive.

Lemma 1

(a) If $\mathcal{S}_m(\mathcal{G}_{3,m-2}) \le m/2 - 2$*, then* $\mathcal{S}_m(\mathcal{G}_{3,m}) \le \mathcal{S}_m(\mathcal{G}_{3,m-2})$*.*

(b) If $\mathcal{S}_M(\mathcal{G}_{3,m-2}) \le m/2 - 3$*, then* $\mathcal{S}_M(\mathcal{G}_{3,m}) \le \mathcal{S}_M(\mathcal{G}_{3,m-2})$*.*

(c) $\mathcal{S}_m(\mathcal{G}_{3,m}) \ge \mathcal{S}_m(\mathcal{G}_{3,m-2})$*.*

(d) $\mathcal{S}_M(\mathcal{G}_{3,m}) \ge \mathcal{S}_M(\mathcal{G}_{3,m-2})$*.*

Proof. (a) Let $d = \mathcal{S}_m(\mathcal{G}_{3,m-2})$. Let $G = \mathcal{G}_{3,m}$ and let G' be G without the rightmost two columns. Our basic strategy for the minimizer is to play his strategy $\mathcal{T}$ for G'. By symmetry of G', we can assume the source is in the first $\lceil (m-2)/2 \rceil$ columns of G (and in particular is not in $\mathcal{C}_{m-2}$).

There are two events that require a change: (α) the maximizer moves out of G', or (β) the strategy $\mathcal{T}$ calls for a move by the minimizer up or down from $(m-2,2)$. If neither (α) nor (β) occurs, then the path will reach a point where it cannot continue in G' and is within distance d of the source; since $\lceil (m-2)/2 \rceil + (m/2-2) < m-2$, it follows that the path is to the left of $\mathcal{C}_{m-2}$, and is actually finished.

So we need to consider two cases depending on which event occurs first.

1. *Event α occurs.* With the maximizer to move, the path was in $\mathcal{C}_{m-2}$. Assume it was in the top or bottom row, say the top. Then since (β) did not occur, the previous vertex in the path was $(m-3,1)$. Further, since the path in G' does not end here, vertex $(m-2,2)$ is unused. Then in G, the minimizer moves the path $(m-1,1) \to (m,1) \to (m,2) \to (m,3) \to (m-1,3) \to (m-1,2) \to (m-2,2)$, where the maximizer's moves are all forced, and the path continues as if the game was being played in G', and ends within d of the source.

   Assume then the maximizer moves $(m-2,2) \to (m-1,2)$. Then there are two possibilities. Assume first that the previous move was in $\mathcal{C}_{m-2}$, say from $(m-2,1)$. Then the minimizer moves the path $(m-1,1) \to (m,1) \to (m,2) \to (m,3) \to (m-1,3) \to (m-2,3)$, and the path continues as if the game was being played in G'. Assume second that the previous vertex was $(m-3,2)$. Then the other two vertices in $\mathcal{C}_{m-2}$ are unused. Then the minimizer makes any move and the path continues back to a vertex in $\mathcal{C}_{m-2}$ and $\mathcal{T}$ is resumed. By the assumption, the maximizer cannot force the path back to the final vertex in $\mathcal{C}_{m-2}$, since in G' the path would end there, and that vertex is more than d from the source.

2. *Event β occurs.* Say $\mathcal{T}$ calls for minimizer to move up $(m-2,2) \to (m-2,1)$. Then instead, the minimizer plays right $(m-2,2) \to (m-1,2)$. Then, no matter which of the three options the maximizer plays next, the minimizer can force the path back to $(m-2,1)$, and the path continues as if the game was being played in G'. As above, the maximizer cannot force the path back to $(m-2,3)$ if untaken, since that would contradict the assumption about the final vertex.

This completes the proof.

(b) The proof is identical to that of (a), except that we can only assume the source is in the first $\lceil m/2 \rceil$ columns of G.

(c,d) The proof is almost the same as the above parts. That is, the maximizer plays the same strategy on $\mathcal{G}_{3,m-2}$ and adjusts it as above if events (α) or (β) occur, where now (α) means the minimizer moves out of G', and (β) means the strategy $\mathcal{T}$ calls for a move by the maximizer up or down from $(m-2,2)$.

The difference is that the path may indeed return to the final vertex of $\mathcal{C}_{m-2}$, say x. However, if so, the path in G' would finish here, so x is distance at least d from the source. Now, in G the path continues. The only vertex in the final two columns that can be closer than x to the source, is the one in $\mathcal{C}_{m-1}$ in the opposite row, say y. But in both cases α and β it is easily checked that the path cannot later end at y in G. $\square$

Theorem 9
(a) $\mathcal{S}_M(\mathcal{G}_{3,m}) = 1$ *for even* m.
(b) $\mathcal{S}_m(\mathcal{G}_{3,m}) = 1$ *for even* m *and* $\mathcal{S}_m(\mathcal{G}_{3,m}) = 2$ *for odd* $m \geq 3$.

Proof. (a) The case of $m = 2$ is easily argued by hand. The upper bounds for cases $m = 4$ and $m = 6$ can be checked by hand, or by computer, or by a more careful induction than in the above lemma. We omit the calculations. So assume $m \geq 8$. Then the result follows by induction using the above lemma.

(b) The proof is similar to (a). Handle small m separately, and use the above lemma for $m \geq 8$. $\square$

4.3 General grids

We used a computer to calculate the values of the game for other small grids. See Table 1.

MAXimizer moves first

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	4	1	6	1	8	1	10	1	12	1	14	1	16	1	18	1	20	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	–	–	–	–
5	1	6	1	8	1	10	1	12	1	14	1	16	–	–	–	–	–	–	–
6	1	1	1	1	1	1	1	1	1	–	–	–	–	–	–	–	–	–	–
7	1	8	1	10	1	12	1	14	–	–	–	–	–	–	–	–	–	–	–

minimizer moves first

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	1	1	3	3	3	3	5	5	5	5	7	7	7	7	9	9	9	9	11
3	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
4	3	1	3	3	5	5	5	5	7	7	7	7	9	9	9	–	–	–	–
5	3	2	3	2	5	2	5	2	5	2	5	2	–	–	–	–	–	–	–
6	3	1	5	5	5	5	7	7	7	–	–	–	–	–	–	–	–	–	–
7	3	2	5	2	5	2	5	2	–	–	–	–	–	–	–	–	–	–	–

Table 1: Values for grid

This data suggests the following conjecture:

Conjecture 1 *For $k \times m$ grid $\mathcal{G}_{k,m}$:*

$\mathcal{S}_M = 1$ *if* km *is even, and* $k+m-2$ *if* km *is odd.*
$\mathcal{S}_m = 2$ *if* km *odd.*

4.4 Other computer calculations

We also calculated values for the torus, being the cartesian product of two cycles. See Table 2. It is to be noted that the data supported a similar conjecture for $\mathcal{S}_m$ being 2 when the number of vertices of the graph is odd, until we calculated the value for the 5×11 torus. This suggests that the earlier conjectures for grids might simply be patterns that only hold for small numbers.

MAXimizer moves first

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	2	1	2	1	3	1	3	1	4	1	4	1	5	1	5
4	1	1	1	1	1	1	1	1	1	1	1	–	–	–	–
5	2	1	3	2	3	2	3	3	4	–	–	–	–	–	–
6	1	1	2	1	2	1	2	–	–	–	–	–	–	–	–
7	3	1	3	2	4	3	–	–	–	–	–	–	–	–	–

minimizer moves first

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	1	2	2	3	2	3	2	3	2	4	2	5	2	5	2
4	2	3	4	5	5	5	6	5	7	7	8	–	–	–	–
5	2	4	2	4	2	4	2	4	3	–	–	–	–	–	–
6	3	5	4	5	4	5	6	–	–	–	–	–	–	–	–
7	2	5	2	4	2	5	–	–	–	–	–	–	–	–	–

Table 2: Values for torus

For another family, we consider the rook's graph, being the cartesian product of two complete graphs (also known as the line graph of the complete bipartite graph). This graph has diameter 2, so the only question is whether the value is 1 or 2. See Table 3.

We also calculated the values for the first five hypercubes Q_k by computer. The value is 1 if maximizer goes first; The value is $1, 1, 3, 3, 3$ if minimizer goes first.

MAXimizer moves first		2	3	4	5	6	7	8	9	10
	2	1	1	1	1	1	1	1	1	1
	3	1	2	2	2	2	2	2	2	–
	4	1	2	2	2	2	2	–	–	–
	5	1	2	2	2	–	–	–	–	–

minimizer moves first		2	3	4	5	6	7	8	9	10
	2	1	2	2	2	2	2	2	2	2
	3	2	1	2	2	2	2	2	2	–
	4	2	2	2	2	2	2	–	–	–
	5	2	2	2	2	–	–	–	–	–

Table 3: Values for rook's graph $K_r \times K_s$

5 Trees

We saw above that $\mathcal{S}_M(\mathcal{G}_{2,m}) \leq \mathcal{S}_m(\mathcal{G}_{2,m})$. In other words, both minimizer and maximizer prefer to be the second player on such a grid. For trees, however, both minimizer and maximizer always prefer to go first.

Theorem 10 *For all trees T on at least three vertices, $\mathcal{S}_M(T) > \mathcal{S}_m(T)$.*

PROOF. If x and y are leaves, then we call them "nearby" if the path between them has (at most) one vertex v of degree more than 2.

Claim: if leaves x and y are nearby, then $\mathcal{S}_m(T) \leq 1 + \lfloor d(x,y)/2 \rfloor \leq 1 + \lceil d(x,y)/2 \rceil \leq \mathcal{S}_M(T)$, where $d(x,y)$ is the distance between them.

To show the upper bound on $\mathcal{S}_m$, let the minimizer choose as source a vertex closest to the center of the x–y path P, subject to the constraint that the vertex is in the opposite partite set to v. The constraint ensures that if the growing path ever reaches v, then it is minimizer's move there, and thus minimizer can keep the path inside P. So the final length of the path is at most the distance from the source to the farther of x and y.

To show the lower bound on $\mathcal{S}_M$, let the maximizer choose as source whichever of x or y is farthest from v. Then no matter what happens, the path continues for at least $1 + \lceil d(x,y)/2 \rceil$ steps.

Thus $s_m(T) = s_M(T)$ requires every pair of nearby vertices to have exactly the same distance D, and D must be even, and v must be the center of the path. But then, if maximizer starts with the source as a leaf as before, then when the path reaches v it is maximizer's move and so maximizer can ensure the path has length at least D. Hence we need $D/2 + 1 = D$, that is $D = 2$. But if the tree has diameter more than 2, then the maximizer can make another choice at v and so $\mathcal{S}_M \geq 3$; on the other hand if the tree has diameter 2, then the tree is a star and $\mathcal{S}_m = 1$. In each case, $\mathcal{S}_m(T) < \mathcal{S}_M(T)$. $\square$

5.1 Tree algorithm

In this section we show that there is a linear-time algorithm for calculating the value of the distance game in trees.

Given a particular source, it is easy to compute the value of the result by the obvious minimax algorithm. For each vertex v, define $f(v)$ as the length of the down path starting at v when minimizer goes first after v (and players alternate after that), and g the same except that maximizer goes first after v. Do a postorder traversal of the tree calculating the two values. Leaves have values $f = g = 0$. At each vertex v, we get:

$$f(v) = 1+\min\{\, g(c) : c \text{ is child of } v \,\} \quad \text{and} \quad g(v) = 1+\max\{\, f(c) : c \text{ is child of } v \,\}.$$

Then the value $\mathcal{S}$ at the root/source is f or g, depending on who is the first player. This takes linear time.

Now to extend this to a full algorithm, we have to consider all other sources. For each vertex v and every neighbor c of v, define $f(v, c)$ as the length of the path starting at v when minimizer chooses the neighbor of v subject to the constraint that it is not c, and $g(v, c)$ the same except that maximizer chooses the neighbor. For example, the original $f(v)$ is $f(v, c)$ where c is v's parent. So after the postorder traversal, continue with a preorder traversal of the tree that calculates $f(v, c)$ at each vertex v for every neighbor c. This gives us enough information to calculate the result of each vertex being the source, and we can in linear-time perform a traversal to find the best source.

However, in order to obtain a linear-time algorithm, we need to calculate at each vertex v the values of all $f(v,c)$ and $g(v,c)$ in time proportional to the number of neighbors. But this can be done. The point is that we start by calculating which neighbor has the highest and second-highest f-value and which has the lowest and second-lowest g-value. Then, when we calculate $f(v,c)$ at v, it is 1 more than the minimum g-values of its neighbors, c excluded. This minimum must be either the minimum or second-minimum value among the original g-values. The value of g is similarly calculated. Thus the overall algorithm runs in linear time. $\square$

6 Conclusion

We considered the value of the game for some special graphs and obtained rather primitive bounds. We were unable to prove the general conjecture about grids. We were also unable to determine the general complexity of the game.

References

[1] József Beck. *Combinatorial Games: Tic-tac-toe theory.* Cambridge University Press, Cambridge, 2008.

[2] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for your Mathematical Plays. Vol. 1.* A K Peters Ltd., Natick, MA, second edition, 2001.

[3] U. Faigle, U. Kern, H. Kierstead, and W. T. Trotter. On the game chromatic number of some classes of graphs. *Ars Combin.*, 35:143–150, 1993.

[4] J.B. Phillips and P.J. Slater. Graph competition independence and enclaveless parameters. *Congr. Numer.*, 154:79–100, 2002.

[5] S.J. Seo and P.J. Slater. Competition parameters of a graph. *AKCE Int. J. Graphs Comb.*, 4:183–190, 2007.

[6] M. Walsh. The competition chromatic numbers of a graph. *Congr. Numer.*, 164:153–158, 2003.

[7] Douglas B. West. *Introduction to Graph Theory (Second edition).* Prentice Hall Inc., Upper Saddle River, NJ, 2000.