The Lights Out Game on Directed Graphs

T. Elise Dettling* Darren B. Parker†

March 30, 2025

MR Subject Classifications: 05C20, 05C40, 05C50, 05C57, 05C78
Keywords: Lights out, light-switching game, feedback arc sets, linear algebra.

Abstract

We study a version of the lights out game played on directed graphs. For a digraph D, we begin with a labeling of $V(D)$ with elements of $\mathbb{Z}_k$ for $k \geq 2$. When a vertex v is toggled, the labels of v and any vertex that v dominates are increased by 1 mod k. The game is won when each vertex has label 0. We say that D is k-Always Winnable (also written k-AW) if the game can be won for every initial labeling with elements of $\mathbb{Z}_k$. We prove that all acyclic digraphs are k-AW for all k, and we reduce the problem of determining whether a graph is k-AW to the case of strongly connected digraphs. We then determine winnability for certain tournaments with feedback arc sets that arc-induce directed paths or directed star digraphs.

1 Introduction

The lights out game was originally an electronic game created by Tiger Electronics in 1995. The idea behind the game has since been extended to several light-switching games on graphs. Some of these extensions are direct generalizations of the original game, like the σ^+-game in [Sut89] and the neighborhood lights out game developed independently in [GP13] and [Ara12]. This was generalized further to a matrix-generated version in [KP24]. Other versions are explored in [Pel87], [CMP09], and [PZ21].

In each version of the game, we begin with some labeling of the vertices, usually by elements of $\mathbb{Z}_k$ for some $k \geq 2$. We play a given game by toggling the vertices, which changes the labels of some vertices according to whether or not they are adjacent to the toggled vertex. The game is won when we achieve some desired labeling, usually where each vertex has label 0.

In this paper, we study a directed graph version of the neighborhood lights out game. The game begins with a digraph D and an initial *labeling* of the vertices with elements

*Department of Mathematics, Grand Valley State University, Allendale, Michigan 49401-6495, dettlint@mail.gvsu.edu,

†Department of Mathematics, Grand Valley State University, Allendale, Michigan 49401-6495, parkerda@gvsu.edu

of $\mathbb{Z}_k$, which we primarily express as a function $\lambda : V(D) \to \mathbb{Z}_k$. The game is played by toggling vertices. Each time a vertex v is toggled, the label of v and each vertex it dominates (i.e. each w such that $vw \in A(D)$) is increased by 1 modulo k. The object of the game is to toggle the vertices so that all vertices have label 0. We call this game the k-lights out game (note that in [KP24], we call this the (N, k)-lights out game, where N is the neighborhood matrix of the graph; we drop the N in our notation here, since we only play one version of the game in this paper). If it is possible to win the game when we begin with the labeling λ, we call λ a k-winnable labeling. We say that D is k-Always Winnable (or k-AW) if every labeling of $V(D)$ is k-winnable.

The basic question we explore is for which digraphs D and natural numbers k is D k-AW. In Section 3, we solve this problem for acyclic digraphs and reduce the problem of determining k-winnability on digraphs to determining k-winnability on strongly connected digraphs. We then turn our attention to strongly connected tournaments. In Section 4, we look at *feedback arc sets*, focusing on what we call *nonconsecutive feedback arc sets* that do not include arcs between consecutive vertices in the vertex ordering. In Section 5, we determine precisely when a strongly connected tournament is k-AW in the cases where the tournament has a nonconsecutive feedback arc set that arc-induces either a directed path or a directed star.

For the most part, we use digraph notation and terminology as in [BJG09]. A digraph D consists of a set of *vertices* $V(D)$ along with a set $A(D)$ of ordered pairs of vertices called *arcs*. For $v, w \in V(D)$, we can express the arc (v, w) as vw or $v \to w$. In this arc, we call v the *tail* and w the *head* of the arc and say that v dominates w. The *outdegree* of a vertex is the number of vertices it dominates, and the *score-list* of a digraph is a multiset of the outdegrees of its vertices. If $B \subseteq A(D)$, we define the subgraph *arc-induced* by B (denoted D_B) to have arc set B and vertex set consisting of all vertices incident with an arc in B.

2 Linear Algebra

Our most useful technique in analyzing when a lights out game can be won is reducing the act of winning the game to solving a system of linear equations. We do this by focusing on what happens to the label of each vertex in the course of playing the game. The final label of a vertex v is determined by taking its original label and adding the number of times v and each vertex that dominates v are toggled. Note that the final label depends only on the number of times each vertex is toggled, not on the order in which they are toggled. Also, since our labels are elements of $\mathbb{Z}_k$, the final label is dependent only on the number of times we toggle a given vertex mod k. Thus, if we define a variable for each vertex that represents the number of times that vertex is toggled, we can express the final label in terms of these variables, and we can consider the variables as elements of $\mathbb{Z}_k$. Setting all the final labels equal to zero gives us a system of linear equations that is equivalent to winning the game.

There are two ways that we apply this method to the study of the neighborhood lights out game. One way is analogous to a method used in the neighborhood game on graphs as studied in [AF98], [AMW14], [GP13], [KP24], and [EEJ+10]. For a digraph D, suppose we order the vertices $v_1, v_2, \ldots, v_n$. We can express an initial labeling $\lambda : V(D) \to \mathbb{Z}_k$ as a column vector $\mathbf{b}$, where $\mathbf{b}[j] = \lambda(v_j)$. If each vertex v_i is toggled x_i times mod k, we can also express this as a column vector $\mathbf{x}$, where $\mathbf{x}[i] = x_i$. Since the label of each v_j is increased by x_i mod k when $i = j$ and when v_i dominates v_j, and is unchanged by each v_i that does not dominate v_j, the resulting label for each v_j is $\lambda(v_j) + \sum_{i=1}^{n} a_{ij} x_i$, where $a_{ij} = 1$ if either $i = j$ or $v_i v_j \in A(D)$, and $a_{ij} = 0$ otherwise. We then determine how to win the game by setting each label equal to 0, which results in a system of linear equations. The toggling $\mathbf{x}$ wins the game precisely when $\sum_{i=1}^{n} a_{ij} x_i = -\lambda(v_j)$ for all j. If we let $N = [a_{ij}]$, winning the k-lights out game is equivalent to finding a solution to the matrix equation $N\mathbf{x} = -\mathbf{b}$. Note that if A is the adjacency matrix of D, then $N = A + I_n$, which we call the *neighborhood matrix* of D. We get the following.

Proposition 2.1. [GP13] Let D be a digraph, let N be the neighborhood matrix of D, and let $\mathbf{b}$ be an initial labeling of $V(D)$.

1. $\mathbf{b}$ is k-winnable if and only if there is a solution to the matrix equation $N\mathbf{x} = -\mathbf{b}$.

2. D is k-AW if and only if N is invertible over $\mathbb{Z}_k$.

The second way that we encounter linear algebra is through the following process.

- We prove that given any initial labeling, we can toggle the vertices in such a way that we get a labeling that has a more desirable form.

- Starting with a labeling in the more desirable form, we determine vertices where the number of toggles is unknown, and assign each of the number of toggles of these vertices a variable.

- We use the labeling that results from the toggling above to generate a system of linear equations.

- We use linear algebra to determine when a solution exists.

This strategy often leads to a system of linear equations that has fewer variables or is otherwise simpler to solve than the system generated by the neighborhood matrix.

In both instances, we need to determine whether a system of equations has a solution regardless of the initial labeling. That requires knowing when the matrix associated with that system of equations is invertible. Since all entries of our matrices and column vectors are elements of the commutative ring $\mathbb{Z}_k$, we need some understanding of how linear algebra works over commutative rings. We use the following facts.

Proposition 2.2. [Bro93] Let A be an $n \times n$ matrix over a commutative ring R.

1. A is invertible if and only if $\det(A)$ is a unit in R.

2. If we take a multiple of one row of A and add it to another row, then the determinant of the resulting matrix is $\det(A)$.

3. If we switch two rows of A, the resulting matrix has determinant $-\det(A)$.
4. If we multiply one row of A by $r \in R$, the resulting matrix has determinant $r\det(A)$.

Moreover, chapter 4 in [Bro93] discusses a generalization of rank for matrices over commutative rings. Recall that an ideal of a commutative ring R is a subset of R that is closed under multiplication by any element of R. Also, if $S \subseteq R$, the annihilator of S is the set of all $x \in R$ such that $xs = 0_R$ for all $s \in S$. This gives us the following generalization of the rank of a matrix over a field.

Definition. Let A be an $m \times n$ matrix over a commutative ring R. For each $t \geq 1$, let I_t be the ideal generated by the $t \times t$ minors of A. The rank of A (denoted $\mathrm{rk}(A)$) is the maximum value of t such that the annihilator of I_t is $\{0_R\}$.

As one would expect from the rank of a matrix, we get the following result.

Theorem 2.3. [McC48] The $n \times n$ matrix A has an inverse if and only if $\mathrm{rk}(A) = n$.

3 Acyclic Digraphs and Strongly Connected Components

Our techniques for determining whether the lights out game can be won depend heavily on a strategic ordering of the vertices. All of our toggling strategies begin with some variation of the following algorithm.

Algorithm 1. Given a digraph D,

- Order the vertices of D in a particular way. Say the ordering is $v_1, v_2, \ldots, v_n$.
- Toggle v_1 until it has label 0.
- Recursively, once v_m has been toggled, toggle v_{m+1} until it has label 0.
- Continue toggling in this way until v_n has been toggled.

Most often, this algorithm by itself will not win the game, but it often gets us closer to determining whether or not the game can be won.

Recall that a *walk* in D is a sequence of vertices $v_1 v_2 \cdots v_n$, where each $v_i v_{i+1}$ is an arc. A *cycle* is a closed walk (i.e. where $v_1 = v_n$) where there are no repeated vertices except the first and last vertex. The strategy suggested by Algorithm 1 is quite effective on *acyclic digraphs*, which are digraphs that contain no cycles. It is straightforward to prove (see [BJG09, Prop. 2.1.3 (p. 33)]) that every acyclic digraph has an ordering $v_1, v_2, \ldots, v_n$ of the vertices (called an *acyclic ordering*) such that whenever $v_i v_j \in A(D)$, then $i < j$. This helps us prove the following.

Theorem 3.1. Let D be an acyclic digraph. Then D is k-AW for all $k \geq 2$.

Proof. Let $\lambda : V(D) \to \mathbb{Z}_k$ be any labeling of $V(D)$. As noted above, there is an acyclic ordering $v_1, v_2, \ldots, v_n$ of $V(D)$. We apply Algorithm 1 to this ordering. No vertex can dominate any vertex that comes before it in the ordering, so every time we toggle a vertex, it does not affect the labeling of the vertices before it. Thus, an easy induction proves that after toggling each v_m, this leaves $v_1, v_2, \ldots, v_m$ all having label 0. Applying this result to $m = n$ gives us the zero labeling for $V(D)$, and thus D is k-AW. □

We now look at connectivity in digraphs. We say that a digraph D is *strongly connected* if for every $v, w \in A(D)$, there is both a walk from v to w and a walk from w to v. On any digraph D, a *strong component* is a maximal subdigraph that is strongly connected.

Proposition 3.2. [BJG09, p. 17] Let D be a digraph, and let $D_1, D_2, \ldots, D_t$ be the strong components of D.

1. The sets $V(D_1), V(D_2), \ldots, V(D_t)$ form a partition of $V(D)$.

2. We can order the D_i in such a way that if $v \in V(D_i)$ and $w \in V(D_j)$ with $vw \in A(D)$ and $i \neq j$, then $i < j$. This is called an *acyclic ordering* of the strong components.

This helps us reduce the problem of determining winnability on digraphs to determining winnability on strongly connected digraphs.

Theorem 3.3. Let D be a digraph, and let $D_1, D_2, \ldots, D_t$ be the strong components of D. Then D is k-AW if and only if each D_i is k-AW.

Proof. Assume that the ordering $D_1, D_2, \ldots, D_t$ is an acyclic ordering of the strong components. We first assume each D_i is k-AW and prove that D is k-AW. So let λ be a labeling of $V(D)$ with labels in $\mathbb{Z}_k$. We then win the game by applying the following variation on Algorithm 1.

- Toggle the vertices of D_1 so that all vertices have label 0. This can be done since D_1 is k-AW.

- Recursively, once the vertices of D_m have been toggled so as to leave each vertex in D_i with label 0 for each $1 \leq i \leq m$, we toggle the vertices of D_{m+1} until all vertices in D_{m+1} have label 0. As above, this can be done since D_{m+1} is k-AW. Note that the vertices in D_{m+1} do not dominate any vertices in any of the D_i for $1 \leq i \leq m$, so all vertices in D_i for $1 \leq i \leq m+1$ now have label 0.

- Continue toggling in this way until the vertices in D_t have been toggled.

As noted in the algorithm, once we toggle the vertices in D_t, all vertices in D_i for $1 \leq i \leq t$ have label 0. Since the $V(D_i)$ partition $V(D)$ by Proposition 3.2(1), all vertices in D have label 0, which makes λ k-winnable. Since λ was arbitrary, D is k-AW.

We now assume that D is k-AW and prove that each D_i is k-AW. We actually prove something slightly stronger. We prove that for each i,

- Every labeling λ on $V(D_i)$ is k-winnable.
- For any labeling π on $V(D)$, if we apply the winning toggles to each D_i in order (i.e. D_1 first, D_2 second, etc), then at the end of toggling the vertices of D_i, all vertices in D_j have label 0 for all $1 \leq j \leq i$.

We prove this by induction. For $i = 1$, let λ be any labeling of D_1. We can extend λ to $V(D)$ by defining $\lambda(v)$ to be any element of $\mathbb{Z}_k$ (it does not matter which element) for all $v \notin V(D_1)$. Since D is k-AW, this extended labeling is k-winnable. Thus, we can toggle the vertices of D so that every vertex has label 0. Consider the toggles of this winning strategy that are done on $V(D_1)$. None of the vertices outside of $V(D_1)$ dominate any of the vertices in $V(D_1)$. Thus, in order for all vertices of D (including all vertices of D_1) to have label 0 at the end of the winning toggling, all vertices of D_1 must have label 0 after we have toggled the vertices in D_1. Thus, λ is k-winnable, and so D_1 is k-AW.

Note that if π is any labeling of D, the above argument can be used to prove that any winning toggling on D that we restrict to $V(D_1)$ must result in every vertex of D_1 having label 0. This completes the base case.

For induction, let λ be a labeling of D_i, and extend λ to all of $V(D)$ by defining $\lambda(v) = 0$ for all $v \notin V(D_i)$. Our induction hypothesis implies that our winning toggling must include, for all $1 \leq m \leq i-1$, all vertices in D_m having labeling 0 after their vertices are toggled. But since they begin with their vertices having label 0 to begin with, the toggling for each D_m must be a solution to $N_m \mathbf{x}_m = \mathbf{0}$, where N_m is the neighborhood matrix of D_m. But each D_m is k-AW, and so each N_m is invertible. It follows that each $\mathbf{x}_m = \mathbf{0}$, and so none of the vertices in any of the D_i's are toggled. Since no vertices in $V(D_m)$ for $m > i$ dominate any vertices in D_i, the toggles of vertices in D_i must result in all vertices in D_i having label 0. Thus, λ on $V(D_i)$ is k-winnable, and so D_i is k-AW.

It now suffices to prove that for all labelings π on D, when we apply a winning toggling to vertices of D_m, $1 \leq m \leq i$, in order, the vertices of each D_m have label 0 once the vertices in D_m have been toggled. This is true for $1 \leq m \leq i-1$ by induction. Once this has been done, we apply the argument in the previous paragraph to prove that after the vertices of D_i have been toggled, the vertices of D_i all have label 0. $\square$

It follows from Theorem 3.3 that once we have determined which strongly connected digraphs are k-AW, we can apply this to the strong components of any digraph to determine whether or not it is k-AW. So we concentrate our efforts on strongly connected digraphs.

4 Using Feedback Arc Sets to Determine Winnability

The remainder of our paper will be focused on winnability in strongly connected tournaments. An important tool in our results will be the use of feedback arc sets. We use the following definitions.

Definition. Let D be a digraph, and let $\sigma = v_1, v_2, \ldots, v_n$ be an ordering of the vertices in D.

1. The *feedback arc set with respect to* σ is the set of all arcs $v_j v_i \in A(D)$ such that $i < j$. When the ordering of the vertices is clear, we simply call it a *feedback arc set*.

2. A *minimum feedback arc set* is a feedback arc set of minimum cardinality among all possible orderings of the vertices of D.

3. If an arc vw is in a feedback arc set S, we call v a *tail feedback vertex* and w a *head feedback vertex*.

There are other definitions of feedback arc sets, most of which are equivalent. One is given in [IN04], which defines a feedback arc set as a set of arcs that when reversed makes the resulting graph acyclic. These are called *reversing sets* in [BHI+95], a term apparently no longer in use. The standard definition of feedback arc sets appears to be the one given in [BJG09] and [Kud22], which replaces *reversed* in the above definition with *removed*. This definition is not quite equivalent to the others (e.g. under this definition, any set of arcs in an acyclic digraph would be considered a feedback arc set in the removal definition but not necessarily in the other definitions). However, minimum feedback arc sets are equivalent in all definitions.

If we have a feedback arc $v_{i+1}v_i$ between consecutive vertices in an ordering of the vertices of D, it is clear that reversing the order of v_i and v_{i+1} does not create any new feedback arcs, and thus has fewer feedback arcs than the original ordering. Thus, when we have a minimum feedback arc set, we get the following.

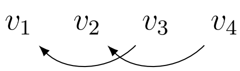
Proposition 4.1. Let D be a digraph, let S be a minimum feedback arc set, and let $\sigma = v_1, v_2, \ldots, v_n$ be an ordering of the vertices whose feedback arc set is S. If $v_j v_i \in S$, then $j - i \geq 2$.

This helps motivate the following definition.

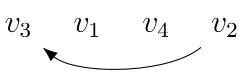
Definition. Let D be a digraph. We say that a feedback arc set S is *nonconsecutive* if no arc in S is incident to consecutive vertices in the ordering of the vertices.

Note that all minimum feedback arc sets are nonconsecutive by Proposition 4.1, but it is possible for nonconsecutive feedback arc sets not to be minimum, as we see in the following example.

Example 4.2. Consider the following tournament on four vertices. Note that only the feedback arcs are shown. Any arcs not shown go from left to right (e.g. the arc v_1v_4).



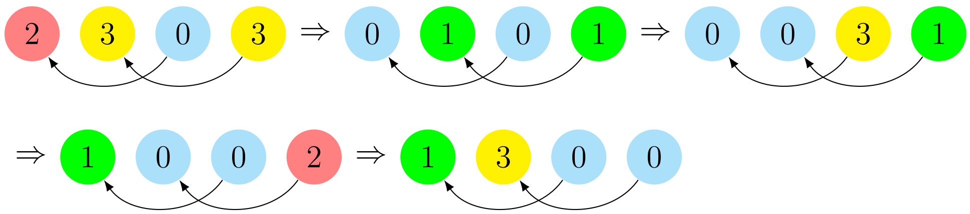
By re-ordering the vertices v_3, v_1, v_4, v_2, we get the following.



Note that both feedback arc sets are nonconsecutive, but only the second one is minimum.

The next example shows what happens in the k-lights out game when we apply Algorithm 1 to a nonconsecutive feedback arc set.

Example 4.3. We play the 4-lights out game on the same graph as the previous example. The sequence of tournaments with labeled vertices below shows what happens when we apply Algorithm 1 to the vertices in order.



Playing the 4-lights out game means every label is in $\mathbb{Z}_4$. The first digraph in the upper left is the initial labeling. We toggle the first vertex twice to get its label to 0. This increases the second and fourth vertices' labels by 2 mod 4, as shown in the second digraph. We get the third digraph by toggling the second vertex three times. We get the fourth digraph by toggling the third vertex once. Finally, we get the last graph by toggling the last vertex twice.

Note that the only vertices with a nonzero label are the first two vertices, which happen to be the head feedback vertices.

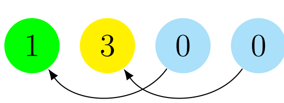
The following lemma shows how this process works in general.

Lemma 4.4. Let D be a digraph with labeling $\lambda : V(D) \to \mathbb{Z}_k$, and let S be the feedback arc set with respect to σ. Then the vertices of D can be toggled in such a way that the only vertices with nonzero labels are head feedback vertices.

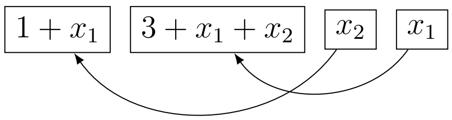
Proof. We apply Algorithm 1 to D with the labeling λ. Let $v \in V(D)$ be a vertex that is not a head feedback vertex. After we finish toggling v, it has a label of 0. Since no vertex toggled after v dominates v, this label will not change for the remainder of the algorithm. Thus, the final labeling for v is 0, which completes the proof. Note that if any tail feedback vertex is toggled as part of the algorithm, it may cause the final label of each head vertex it dominates to have a nonzero final labeling. □

Our general approach to determining whether a tournament is k-AW is to first toggle the vertices so that the labeling is in the form given in Lemma 4.4, and then interpret winning the game as a system of linear equations. We demonstrate this by continuing where we left off in Example 4.2.

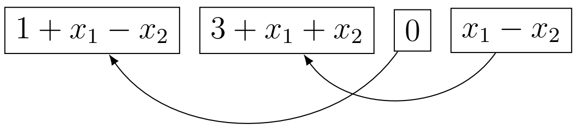
Example 4.5. Consider the following from the end of Example 4.2.



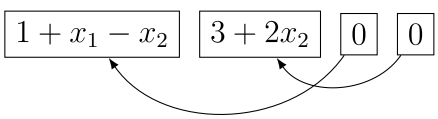
Let x_1 be the number of times we toggle the first vertex, and let x_2 be the number of times we toggle the second vertex. We end up with the following labeling.



The third vertex is not dominated by any vertices that have not yet been toggled, so we must toggle it $-x_2$ times to have any chance of winning the game. This gives us



The fourth vertex is the last vertex to be toggled, so we must toggle it $x_2 - x_1$ times to have any chance of winning the game. We get



The game is now over, so we win precisely when $1+x_1-x_2 \equiv 0 \pmod 4$ and $3+2x_2 \equiv 0 \pmod 4$. Note that the second congruence can never be satisfied since $3+2x_2$ will always be odd. Thus, this game cannot be won. Notice that we needed to solve a system of two equations with two unknowns (one unknown for each head feedback vertex). If we used the neighborhood matrix, we would have four equations and four unknowns.

5 Winnability in Strongly Connected Tournaments

In this section, we focus our attention on tournaments that are strongly connected. A *tournament* is a digraph D where for every distinct pair of vertices $v, w \in V(D)$, either $vw \in A(D)$ or $wv \in A(D)$, but not both. Our results in this section will depend on the structure of the subdigraph D_S that is arc-induced from a nonconsecutive feedback arc set S. Note that since all minimum feedback arc sets are nonconsecutive, the conclusions of our results apply to minimum feedback arc sets as well. We consider subdigraphs D_S that belong to the following classes.

Definition. Let D be a digraph.

1. We say that D is a *directed path* if we can order $V(D) = \{v_1, v_2, \ldots, v_n\}$ in such a way that $A(D) = \{v_i v_{i+1} : 1 \le i \le n-1\}$
2. If $s, t \ge 0$, we say that D is an (s,t)-*directed star* if we have $V(D) = \{v, v_1, v_2, \ldots, v_s, w_1, w_2, \ldots, w_t\}$ and $A(D) = \{v_i v, v w_j : 1 \le i \le s \text{ and } 1 \le j \le t\}$.

Note that if we ignore the directions of the arcs in a directed star, it becomes an ordinary star graph.

Our first main result determines winnability of strongly connected tournaments that have nonconsecutive feedback arc sets whose arcs induce directed paths. Any strongly connected tournament that has a feedback arc set that arc-induces a directed path is called an *upset tournament*, including those with consecutive feedback arcs. Upset tournaments have been studied in [BRLS12], [BL83], and [PS98].

In determining winnability in our upset tournaments without consecutive feedback arcs, we encounter the following sequence of matrices. For each $n \geq 1$, we define the $n \times n$ matrix A_n as follows. Let $A_1 = [2]$ and $A_2 = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}$. For each $n \geq 3$, we define

$$A_n = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 1 & 1 \\ 0 & 0 & \dots & 0 & -1 & 2 \end{bmatrix}$$

Also, recall the sequence of Fibonacci numbers F_n given by $F_0 = 0$, $F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for all $n \geq 2$.

Lemma 5.1. For each $n \geq 1$, $\det(A_n) = F_{n+2}$.

Proof. For $n = 1, 2$, the result is easy to check. For $n \geq 3$, we row-reduce A_n to an echelon form that has the same determinant as A_n and has determinant F_{n+2}.

We first prove by induction that for each $1 \leq i \leq n-1$, we can row-reduce A_n to a matrix with the same determinant as A_n such that

- For $1 \leq m \leq i-1$, the leading entry of row m is 1 and is located in column m.
- Row i will be $[0 \ \dots \ 0 \ \ F_{i+1} \ \ F_i \ \ 0 \ \dots \ 0]$, where F_{i+1} is in column i.

For $i = 1$, the first two entries of row 1 are both 1 (i.e. F_2 and F_1), which proves the base case. If we assume our result for i, we already have the leading entry of 1 in row m, column m for $1 \leq m \leq i-1$. For the inductive step, we need only concern ourselves with rows i and $i+1$. Note that these rows are currently the following, where each leading entry is in column i.

$$\begin{bmatrix} 0 & \dots & 0 & F_{i+1} & F_i & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 1 & 1 & 0 & \dots & 0 \end{bmatrix}$$

We now multiply the lower row by -1 and switch the two rows. By Proposition 2.2, this multiplies the determinant by $-1 \cdot -1 = 1$, so the determinant is unchanged. We get

$$\begin{bmatrix} 0 & \dots & 0 & 1 & -1 & -1 & 0 & \dots & 0 \\ 0 & \dots & 0 & F_{i+1} & F_i & 0 & 0 & \dots & 0 \end{bmatrix}$$

We now add $-F_{i+1}$ times row i to row $i+1$. By Proposition 2.2, this leaves the determinant unchanged. We get

$$\begin{bmatrix} 0 & \dots & 0 & 1 & -1 & -1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & F_i + F_{i+1} & F_{i+1} & 0 & \dots & 0 \end{bmatrix}$$

Since $F_i + F_{i+1} = F_{i+2}$, the result follows.

We now apply this result to $i = n-1$. What we have at that point is a matrix whose first $n-2$ rows are in echelon form with 1's on the diagonal, and whose last two rows are

$$\begin{bmatrix} 0 & \dots & 0 & F_n & F_{n-1} \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}$$

As before, we multiply the last row by -1 and switch rows (which again leaves the determinant unchanged). We get

$$\begin{bmatrix} 0 & \dots & 0 & 1 & -2 \\ 0 & \dots & 0 & F_n & F_{n-1} \end{bmatrix}$$

We now add $-F_n$ times row $n-1$ to row n to get

$$\begin{bmatrix} 0 & \dots & 0 & 1 & -2 \\ 0 & \dots & 0 & 0 & 2F_n + F_{n-1} \end{bmatrix}$$

We have $2F_n + F_{n-1} = F_n + (F_n + F_{n-1}) = F_n + F_{n+1} = F_{n+2}$. This results in an upper triangular matrix. We find the determinant by multiplying the diagonal entries to get $1 \cdots 1 \cdot F_{n+2} = F_{n+2}$. This proves the lemma. □

This leads us to our first main theorem.

Theorem 5.2. Let D be a strongly connected tournament with nonconsecutive feedback arc set S such that D_S is a directed path. If $|S| = m$, then D is k-AW if and only if $\gcd(k, F_{m+2}) = 1$.

Proof. Let $\sigma = v_1, v_2, \dots, v_n$ be an ordering of $V(D)$ such that S is the feedback arc set with respect to σ. Since D_S is a directed path with m arcs, there are vertices $v_{i_1}, v_{i_2}, \dots, v_{i_{m+1}}$ such that $S = \{v_{i_{t+1}} v_{i_t} : 1 \leq t \leq m\}$. Since S is nonconsecutive, $i_{t+1} - i_t \geq 2$ for each t, and so v_{i_t+1} is not a feedback vertex for any $1 \leq t \leq m$. Also, since D is strongly connected, $i_1 = 1$ and $i_{m+1} = n$.

Let $\lambda : V(D) \to \mathbb{Z}_k$ be an arbitrary labeling. We proceed similarly as in Example 4.5. By Lemma 4.4, we can assume $\lambda(v) = 0$ for all v that are not head feedback vertices. For each $1 \leq t \leq m$, let x_t be the number of times we toggle v_{i_t}. Then λ is winnable if and only if we can choose x_t so that all vertices have label 0. We toggle the vertices in order according to σ. So we first toggle $v_{i_1} = v_1$ x_1 times. Since v_1 dominates every vertex except v_{i_2}, this increases the label of every vertex except v_{i_2} by x_1 and leaves the label of v_{i_2} unchanged. There are no remaining untoggled vertices that dominate v_2, so v_2 must be toggled $-x_1$ times to have any chance of winning the game. This leaves all vertices v after v_2 with their original label $\lambda(v)$ except for $v = v_{i_2}$, which now has label $\lambda(v_{i_2}) - x_1$. Each v_i with $2 < i < i_2$ now has label 0, and since they induce no cycles, they cannot be toggled without permanently making at least one of their labels nonzero. To have any chance of winning the game, they must not be toggled at all. Next, when v_{i_2} gets toggled x_2 times, that leaves us with $v_1 = v_{i_1}$ having label $\lambda(v_{i_1}) + x_1 + x_2$ and v_{i_2} having label $\lambda(v_{i_2}) - x_1 + x_2$. Note that when v_{i_2} is toggled, every vertex v_i with $i > i_2$ has label $\lambda(v_i) + x_2$, except v_{i_3}, whose label remains $\lambda(v_{i_3})$.

We apply induction to toggling the remaining vertices. We have quite a few induction hypotheses. First, after we toggle v_{i_r} for $2 \leq r \leq m-1$, the labels of all vertices v_i for $1 \leq i \leq i_r$ are as follows.

- If v_i is not a head feedback vertex, then the label is 0.
- The label for $v_{i_1} = v_1$ is $\lambda(v_{i_1}) + x_1 + x_2$.
- If $2 \leq t \leq r-1$, then the label for v_{i_t} is $\lambda(v_{i_t}) - x_{t-1} + x_t + x_{t+1}$. Note: This part of the induction hypothesis will be clear once we complete the inductive step.
- v_{i_r} has a label of $\lambda(v_{i_r}) - x_{r-1} + x_r$.

Our induction hypothesis also assumes that the labels of all vertices v_i with $i > i_r$ have label $\lambda(v_i) + x_r$, except $v_{i_{r+1}}$, whose label remains $\lambda(v_{i_{r+1}})$.

Now we are ready to continue toggling. Our next toggle is v_{i_r+1}, which by induction has label x_r. Similarly as when we toggled v_2 above, v_{i_r+1} must be toggled $-x_r$ times. As in the base case, all vertices v_i with $i > i_r$ are returned to their original labels except for $v_{i_{r+1}}$, which now has label $\lambda(v_{i_{k+1}}) - x_k$. As in the base case, all vertices that are not feedback vertices between v_{i_r} and $v_{i_{r+1}}$ have label 0, so they are not toggled at all. Then $v_{i_{r+1}}$ is toggled x_{r+1} times. This gives v_{i_r} a label of $\lambda(v_{i_r}) - x_{r-1} + x_r + x_{r+1}$ and $v_{i_{r+1}}$ a label of $\lambda(v_{i_{r+1}}) - x_r + x_{r+1}$. Furthermore, this increases the labels of all vertices v_i with $i > i_{r+1}$ by x_{r+1}, so they have label $\lambda(v_i) + x_{r+1}$. Since $v_{i_{r+2}}v_{i_{r+1}} \in A(D)$, $v_{i_{r+2}}$ still has label $\lambda(v_{i_{r+2}})$. This completes the induction argument.

Once we have toggled v_m, we apply this process one more time to finish off the game. Since v_{i_m} was toggled x_m times, that gives v_{i_m+1} a label of x_m. As before, v_{i_m+1} is toggled $-x_m$ times. This leaves all vertices that are not head feedback vertices with a label of 0, except $v_{i_{m+1}} = v_n$ which is left with a label of $-x_m$. In order to end up with a label of 0, $v_{i_{m+1}}$ is toggled x_m times, leaving v_{i_m} with a label of $\lambda(v_{i_m}) - x_{m-1} + 2x_m$.

At this point, all labels must be 0, and so we must have

- $\lambda(v_{i_1}) + x_1 + x_2 = 0$
- $\lambda(v_{i_t}) - x_{t-1} + x_t + x_{t+1} = 0$ for $2 \leq t \leq m-1$
- $\lambda(v_{i_m}) - x_{m-1} + 2x_m = 0$

Notice that if we write this system of linear equations in matrix form, we get $A_m\mathbf{x} = -\mathbf{b}$, where $\mathbf{x}[t] = x_t$, $\mathbf{b}[t] = \lambda(v_{i_t})$, and A_m is the matrix from Lemma 5.1. Since the vector

$\mathbf{b}$ is arbitrary (it comes from the arbitrary labeling λ), the system of linear equations always has a solution if and only if A_m is invertible. Thus, D is k-AW if and only if A_m is invertible, which by Proposition 2.2 occurs precisely when $\det(A_m)$ is a unit in $\mathbb{Z}_k$. Since $\det(A_m) = F_{m+2}$ by Lemma 5.1, this occurs precisely when $\gcd(k, F_{m+2}) = 1$. This completes the proof. $\square$

We now consider feedback arc sets S where D_S is a directed star. Here is some helpful terminology.

Definition. Let D be a digraph, let $\sigma = v_1, v_2, \ldots, v_n$ be an ordering of $V(D)$, and let S be the feedback arc set with respect to σ. A *head feedback interval* (resp. tail feedback interval) is a nonempty subset of $V(D)$ of the form $\{v_i, v_{i+1}, \ldots, v_{j-1}, v_j\}$, where each v_t with $i \le t \le j$ is a head feedback vertex (resp. tail feedback vertex), and both v_{i-1} and v_{j+1} are not head feedback vertices (resp. tail feedback vertices). A *feedback interval* is a set of vertices that is either a head feedback interval or a tail feedback interval (or both).

Note that if D_S is a directed star, the feedback intervals $F_1, F_2, \ldots, F_t$ can be ordered so that if $i < j$, then all vertices in F_i come before all vertices in F_j. Also, the vertices in the head feedback intervals come before the vertices in the tail feedback vertices, and the set containing the central vertex is both a head feedback interval and a tail feedback interval when both head feedback intervals and tail feedback intervals exist. Thus, we can order the feedback intervals as above, but we can now write them as $H_1, H_2, \ldots, H_r, \{v\}, T_1, T_2, \ldots, T_s$, where the H_i are the head feedback intervals, the T_j are the tail feedback intervals, and v is the central vertex. In this case, we have $T_j \to v \to H_i$.

As we saw in Lemma 4.4, we can toggle the vertices so that all but the head feedback vertices have label 0. Thus, the more head feedback vertices we have (i.e. the more vertices in the H_i above), the more difficult it is to determine winnability in the tournament. The following lemma shows that by slightly re-ordering the vertices, we can eliminate all but one head feedback vertex without affecting the number of feedback intervals.

Lemma 5.3. Let D be a tournament, and let S be a feedback arc set such that D_S is a directed star. Then the vertices can be ordered so that the resulting feedback arc set S' satisfies the following.

1. $D_{S'}$ is a $(t, 0)$-directed star for some $t \in \mathbb{N}$.

2. $D_{S'}$ has the same number of feedback intervals as D_S.

3. S' is nonconsecutive.

Proof. For the ordering of the vertices that gives us the feedback arc set S, let the head feedback intervals be $H_1, H_2, \ldots, H_r$; let the tail feedback intervals be $T_1, T_2, \ldots, T_s$; and let the central vertex be v as in the discussion preceding the lemma statement. Since we have r head feedback intervals, s tail feedback intervals, plus the feedback interval $\{v\}$, there are $r + s + 1$ feedback intervals altogether. For each $1 \le i \le r - 1$, define H_i' to be the set of vertices that come after the vertices in H_i and before the vertices in H_{i+1}

in the ordering of the vertices. We define H'_r to be the set of vertices that come after the vertices in H_r and before v.

Now consider the ordering of the vertices that is identical to the one given except that v is moved to being just before the first vertex in H_1. This has the following effect on the feedback arcs.

- All tail feedback vertices from the original ordering (i.e. the vertices in the T_i) remain tail feedback vertices in the new ordering.
- All head feedback vertices from the original ordering (i.e. the vertices in each the H_i) are no longer feedback vertices at all in the new ordering, since v dominates them and comes before them in the new ordering.
- All vertices that came before v in the original ordering but were *not* feedback vertices (i.e. the vertices in each H'_i) are now *tail* feedback vertices, which makes each H'_i a tail feedback interval.

This means that in the new ordering, there are no head feedback intervals, and each H'_i and T_j is a tail feedback interval. This gives us $r+s$ tail feedback intervals. Adding in the feedback interval $\{v\}$, we get $r+s+1$ feedback intervals altogether, which is identical to the total in the original ordering.

Finally, we prove that S' is nonconsecutive. In the case that all H_i are empty, we have $S' = S$, which is nonconsecutive. Thus, we can assume $H_1 \neq \emptyset$. In S', v is the only head feedback vertex. Since $H_1 \neq \emptyset$, the vertex that immediately follows v is in H_1. It follows that v dominates w, and so S' is nonconsecutive. □

We are now ready to prove our result for when D_S is a directed star.

Theorem 5.4. Let D be a strongly connected tournament, let $\sigma = v_1, v_2, \dots, v_n$ be an ordering of the vertices, and let S be the feedback arc set with respect to σ such that S is nonconsecutive and D_S is a directed star. Let m be the number of feedback intervals of S. Then D is k-AW if and only if $\gcd(k, m) = 1$.

Proof. By Lemma 5.3, since D_S is a directed star and D is strongly connected, we can assume v_1 is the central vertex of D_S; that all feedback intervals are $\{v_1\}$ and the tail feedback intervals $T_1, \dots, T_s$; and that the last vertex in D is the last vertex in T_s. Since m is the number of feedback intervals, it follows that $m = s + 1$. We proceed similarly as in Theorem 5.2. Let $\lambda : V(D) \to \mathbb{Z}_k$ be an arbitrary labeling. By Lemma 4.4, we can assume λ is zero on all vertices except the lone head feedback vertex v_1. Let x be the number of times we toggle v_1. As in Theorem 5.2, we determine when there exists x so that all vertices have label 0.

We toggle the vertices in order according to σ. After v_1 is toggled x times, the label of v_1 is $x + \lambda(v_1)$, the labels of all tail feedback vertices (which dominate v_1) are 0, and the labels of the remaining vertices (which are dominated by v_1) are x. Since no remaining untoggled vertices dominate v_2, then v_2 must be toggled $-x$ times. This results in v_1 still having label $x + \lambda(v_1)$, all tail feedback vertices having label $-x$, and all other vertices having label 0. Note that none of the other vertices after v_1 and before T_1 can be toggled, since there are no other vertices that dominate them that we can toggle to get their labels back to 0.

We now use induction to prove that for each $1 \le i \le s$, when we have finished toggling all vertices before the vertices in T_i, the following conditions hold.

- v_1 has label $ix + \lambda(v_1)$.
- All tail feedback vertices in $T_i, T_{i+1}, \dots, T_s$ have label $-x$.
- All other vertices in D have label 0.

We have just proved this for $i = 1$. We now assume these conditions hold for $i = \ell$, where $1 \le \ell \le s-1$ and prove they hold for $i = \ell+1$. By induction, the first vertex u in T_ℓ has label $-x$, and all vertices that dominate u have already been completely toggled. Thus, u must be toggled x times. This increases the label of v_1 and all vertices after u by x. Thus, v_1 has label $(\ell+1)x + \lambda(v_1)$, all tail feedback vertices have label 0, and all vertices after u that are not tail feedback vertices have label x. Since there are no remaining untoggled vertices that dominate any of the vertices in T_ℓ, the remaining vertices in T_ℓ are not toggled. This brings us to the first vertex w after the vertices in T_ℓ. As stated above, w has label x, and all vertices that dominate w have finished being toggled, so w must be toggled $-x$ times. This brings the labels of the vertices in $T_{\ell+1}, T_{\ell+2}, \dots, T_s$ back to $-x$ and all other labels back to 0, which completes the induction.

The only vertices that are left to toggle are the vertices in T_s. By our induction argument above, at this point v_1 has label $sx + \lambda(v_1)$, all other vertices before T_s have label 0, and the vertices of T_s have label $-x$. As in the inductive step above, we toggle the first vertex in T_s x times. This brings the labels of all vertices in T_s to 0, increases the label of v_1 by x to $(s+1)x + \lambda(v_1)$, and completes the toggling. At this point, all vertices except perhaps v_1 have label 0, so D is k-AW if and only if the equation $(s+1)x + \lambda(v_1) = 0$ has a solution in $\mathbb{Z}_k$ for all possible values of $\lambda(v_1)$. This occurs precisely when $\gcd(k, s+1) = 1$. Since $m = s+1$, this completes the proof. □

Recall from Proposition 2.1 that a digraph being k-AW is equivalent to the neighborhood matrix being invertible. Theorem 2.3 states that the neighborhood matrix being invertible is equivalent to the matrix having rank equal to the number of rows (or columns) in the matrix, which happens to be the order of the graph. We put these facts together with Proposition 2.2, Theorem 5.2, and Theorem 5.4 to get the following.

Corollary 5.5. Let D be a strongly connected tournament of order n with neighborhood matrix N.

1. If D has a nonconsecutive feedback arc set that arc induces a directed path with m arcs, then the following are equivalent.
   (a) $\det(N)$ is a unit in $\mathbb{Z}_k$.
   (b) $\mathrm{rk}(N) = n$.

(c) $\gcd(k, F_{m+2}) = 1$.

2. If D has a nonconsecutive feedback arc set that arc induces a directed star with m feedback intervals, then the following are equivalent

   (a) $\det(N)$ is a unit in $\mathbb{Z}_k$.

   (b) $\mathrm{rk}(N) = n$.

   (c) $\gcd(k, m) = 1$.

6 Conclusion

We would like to classify winnability in all digraphs. However, here are some questions that are perhaps more manageable.

- Theorem 3.3 reduces the question of being k-AW to strongly connected graphs. However, little about connectivity was used in the results in Section 5. Perhaps stronger results can be obtained with more sophisticated use of strong connectivity.

- Theorem 5.2 settled the question of being k-AW for some upset tournaments. However, it leaves open the winnability issue for upset tournaments that have a nonconsecutive feedback arc set that induces a directed path. It would be nice to settle this issue.

- Finally, it appears that for a feedback arc set S, the isomorphism class of D_S is significant (e.g. the result for directed paths was very different from the result for directed stars). However, even within isomorphism classes, there can be other issues that affect winnability (e.g. in directed stars, whether and how the head feedback vertices and tail feedback vertices are organized into feedback intervals). What other quirks in the ordering of vertices can affect winnability?

References

[AF98] M. Anderson and T. Feil, *Turning Lights Out with Linear Algebra*, Math. Mag. **71** (1998), 300–303.

[AMW14] C. Arangala, M. MacDonald, and R. Wilson, *Multistate lights out*, Pi Mu Epsilon J. **14** (2014), 9–18.

[Ara12] C. Arangala, *The $4 \times n$ multistate lights out game*, Math. Sci. Int. Res. J. **1** (2012), 10–13.

[BHI+95] J-P. Barthélemy, Olivier Hudry, Garth Isaak, F.S. Roberts, and B. Tesman, *The reversing number of a digraph*, Disc. Appl. Math. **60** (1995), 39–76.

[BJG09] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms, and Applications (Second Edition)*, Springer, 2009.

[BL83] R.A. Brualdi and Q. Li, *Upsets in round robin tournaments*, J. Combinatorial Theory Ser. B **35** (1983), 62–77.

[BRLS12] D. E. Brown, S. Roy, J. R. Lundgren, and D. J. Siewert, *Boolean rank of upset tournament matrices*, Linear Algebra Appl. **436** (2012), 3239–3246.

[Bro93] W. C. Brown, *Matrices over Commutative Rings*, Marcel Dekker, New York, 1993.

[CMP09] D. Craft, Z. Miller, and D. Pritikin, *A solitaire game played on 2-colored graphs*, Discrete Math. **309** (2009), 188–201.

[EEJ+10] S. Edwards, V. Elandt, N. James, K. Johnson, Z. Mitchell, and D. Stephenson, *Lights out on finite graphs*, Involve **3** (2010), 17–32.

[GP13] A. Giffen and D. B. Parker, *On generalizing the "lights out" game and a generalization of parity domination*, Ars Combin. **111** (2013), 273–288.

[IN04] G. Isaak and D. Narayan, *A classification of tournaments having an acyclic tournament as a minimum feedback arc set*, Inf. Process. Lett. **92** (2004), 107–111.

[KP24] L. Keough and D. B. Parker, *Extremal questions for the lights out game*, Discuss. Math. Graph Theory **44** (2024), 997–1021.

[Kud22] R. Kudelić, *Feedback Arc Set: A History of the Problem and Algorithms*, Springer, Cham, Switzerland, 2022.

[McC48] N. H. McCoy, *Rings and Ideals*, vol. 8, Math. Assoc. of Am., 1948.

[Pel87] D. Pelletier, *Merlin's magic square*, Amer. Math. Monthly **94** (1987), 143–150.

[PS98] J. L. Poet and B. L. Shader, *Short score certificates for upset tournaments*, Electron. J. Combin. **5** (1998), #R24.

[PZ21] D. B. Parker and Vasily Zadorozhnyy, *A group labeling version of the lights out game*, Involve **14** (2021), 541–554.

[Sut89] K. Sutner, *The σ-game and cellular automata*, Amer. Math. Monthly **97** (1989), 24–34.