The Zero Forcing Number of Circulant Graphs

Alex Bates* Laura Fink† Brenda K. Kroschel‡

August 31, 2015

Abstract

The zero forcing number of a graph $G$ is the cardinality of the smallest subset of the vertices of $G$ that forces the entire graph using a color change rule. This paper presents some basic properties of circulant graphs and later investigates zero forcing numbers of circulant graphs of the form $C[n, \{s, t\}]$, while also giving attention to propagation time for specific zero forcing sets.

1 Introduction

A graph is a set $G = \{V, E\}$ where $V$ is a set of vertices and $E$ is the set of edges expressed as unordered pairs such that an unordered pair containing two vertices denotes an edge between those two vertices. This paper deals exclusively with simple, undirected graphs. A simple graph denotes a graph where each vertex cannot have an edge with itself (eg. a loop) and does not allow more than one copy of an edge. An undirected graph denotes graphs for which edges have no orientation. The degree of a vertex $v$ of a graph $G$, denoted $\text{deg}(v)$, is the number of edges incident to that vertex.

![Graph](image)

Figure 1: The graph $\{\{0, 1, 2, 3, 4\}, \{(0, 1), (0, 2), (2, 3), (2, 4), (3, 4)\}\}$.

A vertex is said to share an edge with another vertex if the two are connected with an edge. This is also described by saying there is an edge between two vertices, or that two

---

*Center for Applied Mathematics, University of St. Thomas, St. Paul, MN 55105, USA
†Center for Applied Mathematics, University of St. Thomas, St. Paul, MN 55105, USA
‡Department of Mathematics, Mail #OSS 201, University of St. Thomas, 2115 Summit Avenue, St. Paul, MN 55105, USA (bkkroschel@stthomas.edu)
vertices are *neighbors, adjacent,* or \((v_1, v_2) \in E\). An edge is called *incident* to a vertex when it is attached to that vertex (ie: the edge \((v_1, v_2)\) is incident to the vertex \(v_1\)). Two vertices that are next to each other but not necessarily adjacent are called *consecutive* vertices.

The *color change rule* applies to graphs that initially have a certain set of vertices colored blue and the remaining vertices white. If a blue vertex has only one adjacent white vertex then that white vertex is *forced* or color-changed blue. This process is repeated until no more vertices can be forced blue. If every vertex is blue when the process stops, the graph is called *color-derived*. Every repetition of this process is considered one timestep or iteration.

A zero *forcing set* of a graph \(G\) is any subset of the vertices of \(G\) colored blue that can force the entire graph to its color-derived state via the color change rule. A *minimal zero forcing set* of \(G\) is a zero forcing set of \(G\) containing the fewest vertices. A graph \(G\) may have more than one minimal zero forcing set. The *zero forcing number* of a graph \(G\), denoted \(Z(G)\), is the cardinality of a minimal zero forcing set of \(G\).

Let \(G\) be a graph and \(S\) a zero forcing set of \(G\). The *propagation time* for \(S\) in \(G\), denoted \(pt(G, S)\), is the number of timesteps (or iterations) needed for \(S\) to force \(G\) to its color-derived state. The *minimal propagation time* of \(G\), denoted \(pt(G)\) is the minimum propagation time over all zero forcing sets.

![Graph Diagram]

*Figure 2: After two iterations of forcing, the graph in Figure 1 is now color-derived.*

Zero forcing was initially introduced in 2007 [1] for simple, undirected graphs. Zero forcing is of interest because the zero forcing number of a graph provides a bound for the
minimum rank of the matrix associated with that graph. Zero forcing has also garnered interest in recent years because of its relation to the spread of information in networks [2]. It is possible to compute the zero forcing number of a particular graph, but in many cases computation can take a considerable amount of time. To aid research in circulant graphs, the zero forcing numbers of all structurally unique circulant graphs (see section 2.2) up through size \( n = 29 \) were computed, the results of which can be found in Appendix B.

2 Circulant Graphs

A circulant graph, denoted \( C[n, \{s_1, s_2, \ldots, s_m\}] \), for which \( n > 2 \) and \( s_i \in \mathbb{Z}_n \) for \( i = 1, 2, \ldots, m \), is a graph consisting of \( n \) vertices such that each vertex \( v \) shares an edge with the vertices \( v + s_1, v - s_1, v + s_2, v - s_2, \ldots, v + s_m, v - s_m \mod n \). The set \( \{s_1, s_2, \ldots, s_m\} \) is referred to as the connection set of \( G \). At times, it is convenient to write the values of the connection set in order from least to greatest, however, this is not always possible nor necessary. All arithmetic done on the vertices of a circulant graph is mod \( n \).

For integers \( a, b, \) and \( c \), “\( a \) is equivalent to \( b \mod c \)” is denoted by \( a \equiv b \mod c \).

![Figure 3: \( C[10, \{1, 3\}] \)](image)

**Lemma 2.1.** The graphs \( G = C[n, \{s_1, s_2, \ldots, s_{i-1}, s_i, s_{i+1}, \ldots, s_m\}] \) and \( G' = C[n, \{s_1, s_2, \ldots, s_{i-1}, (n - s_i), s_{i+1}, \ldots, s_m\}] \), where \( 1 \leq i \leq m \), are identical.

**Proof.** To show that the two graphs are identical, it is sufficient to verify that they both have identical vertex sets and edge sets. It is obvious that they have identical vertex sets, since both \( G \) and \( G' \) are circulant graphs of size \( n \).

Let \( v \) denote an arbitrary vertex of \( G \) and \( G' \). Obviously, \( v \) is adjacent to \( v + s_j \) and \( v \) is adjacent to \( v - s_j \) for all \( j = 1, 2, \ldots, i-1, i+1, \ldots, m \) in both \( G \) and \( G' \). In \( G \), \( v \) is adjacent to \( v + s_i \) and \( v \) is adjacent to \( v - s_i \). In \( G' \), \( v \) is adjacent to \( v - (n - s_i) \equiv v + s_i \mod n \) implies that \( v \) is adjacent to \( v + s_i \), and \( v \) is adjacent to \( v + (n - s_i) \equiv v - s_i \mod n \) implies that \( v \) is adjacent to \( v - s_i \). Thus, \( v \) is adjacent to \( v + s_j \) and \( v \) is adjacent to \( v - s_j \) for all \( j = 1, 2, \ldots, m \) in both \( G \) and \( G' \). Therefore, the two graphs are identical. \( \blacksquare \)

Note that the lemma above shows that the connection \( s_j > \left[ \frac{n}{2} \right] \) can be rewritten as \( n - s_j \). Therefore, this justifies the claim that all \( s_j \) are in the set \( \{1, 2, \ldots, \left[ \frac{n}{2} \right]\} \).
2.1 Connected graphs

Certain circulant graphs are disconnected; they consist of multiple "copies" of other graphs that are placed on top of one another but do not share any edges. This paper gives little treatment to disconnected graphs, so it is necessary to identify those circulant graphs that are disconnected.

A walk in a graph $G$ is a finite sequence of vertices $v_0, v_1, ..., v_n$ and edges $e_1, e_2, ..., e_n$:

\[ v_0, e_1, v_1, e_2, ..., e_n, v_n, \]

where $e_i = (v_i - 1, v_i)$ for each $i$.

Definition 2.2. A graph is connected if for every pair of vertices $u$ and $v$, there is a walk from $u$ to $v$. [3]

The definition above is logically equivalent to the one below, but both are included here since they both have their own advantages and disadvantages in determining whether or not a graph is connected.

Definition 2.3. A graph is called disconnected if its vertex-set can be partitioned into two subsets, $V_1$ and $V_2$, that have no common element, in such a way that there is no edge with one endpoint in $V_1$ and the other in $V_2$; if a graph is not disconnected then it is connected. [4]

![Figure 4: A disconnected graph.](image)

Proposition 2.4. Let $m \geq 2$. An element $a \in \mathbb{Z}_m$ has a multiplicative inverse if and only if $\gcd(a, m) = 1$.

Let $n \geq 2$ and $b \in \mathbb{Z}_n$. The principal ideal generated by $b$ is $\langle b \rangle = \{kb | k \in \mathbb{Z}_n\}$.

Lemma 2.5. Let $A = \{(cg) \mod n | c \in \mathbb{Z}_n\}$ and $B = \{(cg + 1) \mod n | c \in \mathbb{Z}_n\}$ for $n, g > 1$ such that $\gcd(n, g) \neq 1$, then $A \cap B = \emptyset$.

Proof. By way of contradiction, suppose that $A \cap B \neq \emptyset$. Then, there exists $x \in A \cap B$ so that $x \equiv c_1g \equiv c_2g + 1 \mod n$ for some $c_1, c_2 \in \mathbb{Z}_n$. Then $g(c_1 - c_2) \equiv 1 \mod n$, which implies that $g$ has an inverse in $\mathbb{Z}_n$. But by Proposition 2.4, $g \in \mathbb{Z}_n$ has an inverse if and only if $\gcd(n, g) = 1$, which is a contradiction. Therefore, $A \cap B = \emptyset$. $\blacksquare$

Theorem 2.6. A circulant graph $G = C[n, \{s, t\}]$, where $n, s, t > 1$, is disconnected if and only if $\gcd(n, s, t) \neq 1$. 

4
Proof. To prove the backwards implication, let $G = C[n, \{s, t\}]$ with $gcd(n, s, t) = g \neq 1$. It will be shown that $G = C[n, \{s, t\}]$ is disconnected.

Define the sets $A = \{(cg) \mod n | c \in \mathbb{Z}_n\}$ and $B = \{(cg + 1) \mod n | c \in \mathbb{Z}_n\}$ as in Lemma 2.5. Note that $A$ and $B$ are subsets of the vertex set of $G$ and that $A = \langle g \rangle$. By the lemma, $A \cap B = \emptyset$.

It will be shown that there does not exist an edge with one endpoint in $A$ and the other in $B$. Consider an arbitrary vertex of $A$ and call it $v_a$. The vertex $v_a$ has the form $(cg) \mod n$ for some $c \in \mathbb{Z}_n$ and is adjacent to the vertices numbered $v_a - s, v_a + s, v_a - t,$ and $v_a + t \mod n$. Now $gcd(n, s, t) = g$, and both $s$ and $t$ are elements of $A$, which means that $v_a - s, v_a + s, v_a - t, v_a + t$ are also in $A$.

So, the vertices of $G$ have been partitioned into two distinct subgroups, $A$ and $B$, with $A \cap B = \emptyset$, in such a way that there is no edge with one endpoint in $A$ and the other in $B$. Therefore, $G$ is a disconnected graph.

To prove the forward implication, assume that a circulant graph $G$ is disconnected. It will be shown that $gcd(n, s, t) \neq 1$. By way of contradiction, suppose that $gcd(n, s, t) = 1$. Then, $gcd(n, s) = 1$ or $gcd(n, t) = 1$ or $gcd(s, t) = 1$.

Case 1: Assume that $gcd(n, s) = 1$ or $gcd(n, t) = 1$. Define $r = s$ if $gcd(n, s) = 1$ and $r = t$ if $gcd(n, t) = 1$. Consider two arbitrary vertices $u$ and $w$ of $G$. It will be shown that there is a walk from $u$ to $w$. If $u = w$, then there exists a walk $u, (u, u + r), u + r, (u + r, u), u$ in $G$ from $u$ to itself. If $u \neq w$, define $R = \{zr | z \in \mathbb{Z}_n\}$. Since $gcd(n, r) = 1$, $R = \mathbb{Z}_n$, which implies that $R$ contains all the vertices of $G$. This implies that $w = u + yr$ for some $y \in \mathbb{Z}_n$ so that there is a walk from $u$ to $w$. Then by Definition 2.2, $G$ is connected, which is a contradiction.

Case 2: Assume that $gcd(s, t) = 1$. Consider two arbitrary vertices $u$ and $w$ of $G$. It will be shown that there is a walk from $u$ to $w$. If $u = w$, then there exists a walk $u, (u, u + s), u + s, (u + s, u), u$ in $G$ from $u$ to itself. If $u \neq w$, define $d = (w - u) \mod n$. Now, $gcd(s, t) = 1$ implies that there exist $a, b \in \mathbb{Z}_n$ such that $as + bt = 1$, and $d(as + bt) = das + dbt = d$. Then there is a walk $u, (u, u + s), u + s, (u + s, u + 2s), u + 2s, ..., u + das, (u + das, u + das + t), u + das + t, (u + das + t, u + das + 2t), u + das + 2t, ..., u + das + dbt$ from $u$ to $u + das + dbt = u + d = w$. So, for every arbitrary pair of vertices of $G$ there exists a walk between those two vertices. Then by Definition 2.2, $G$ is connected, which is a contradiction.

$\blacksquare$
Example 2.1. The circulant graph $C[8, \{2, 4\}]$ is disconnected since $gcd(8, 2, 4) = 2 \neq 1$.

The graph on the left can be separated to make it look like the graph on the right.

2.2 Isomorphisms

Definition 2.7. Graphs $G = \{V_G, E_G\}$ and $G' = \{V_{G'}, E_{G'}\}$ are isomorphic or structurally equivalent (denoted $G \cong G'$) if there exists a bijective function $f : V_G \rightarrow V_{G'}$ that acts on both vertices and edges with the property that $(a_0, a_1) \in E_G$ if and only if $(f(a_0), f(a_1)) \in E_{G'}$. Such a function $f$ is called a graph isomorphism.

Remark. Within the context of this paper, an isomorphism $f$ is understood to be a function that, given a graph $G$, acts on both the vertices and edges of $G$. Thus, such isomorphisms are able to act on a singular vertex, sets of vertices, and/or sets of edges. Hence, it is convenient to write $f : V_G \rightarrow V_{G'}$, $f : a \mapsto f(a)$, $f : E_G \rightarrow E_{G'}$, and/or $f : (a_0, a_1) \mapsto (f(a_0), f(a_1))$ as suits the context.

Isomorphic graphs, because they have the same structure, have the same zero forcing number. (This idea is intuitive, but a rigorous proof is given in the theorem below.) For this reason it is helpful to know which graphs are isomorphic so that their zero forcing numbers only have to be computed once. Within this paper, a circulant graph that is structurally unique is understood to be one that either has no graphs isomorphic to itself or, if there are graphs isomorphic to itself, has the smallest values of $s$ and $t$ possible in its connection set of all graphs isomorphic to itself.

Theorem 2.8. If $G_1 \cong G_2$ then $Z(G_1) = Z(G_2)$.

Proof. By way of contradiction, suppose there exist two graphs $G_1, G_2$ with graph isomorphism $f : V_{G_1} \rightarrow V_{G_2}$, and define $z_1 = Z(G_1)$, $z_2 = Z(G_2)$, $z_1 \neq z_2$. Denote the minimal zero forcing sets for $G_1$ and $G_2$ as $A = \{a_1, a_2, ..., a_{z_1}\}$ and $B = \{b_1, b_2, ..., b_{z_2}\}$, respectively.

Since $f$ is a graph isomorphism and $B$ is a zero forcing set for $G_2$, $f^{-1}(B) \subset V_{G_1}$ is a zero forcing set for $G_1$. 

6
But since \( f \) is an isomorphism, \( |f^{-1}(B)| = n < |A| \implies f^{-1}(B) \) is a smaller zero forcing set for \( G_1 \) than \( A \).

This is a contradiction, since \( A \) is the minimal zero forcing set for \( G_1 \). \[\boxed{}\]

A research question published in 1967 [5] stated as a matter of fact that two circulant graphs of size \( n \) are isomorphic if there exists a graph isomorphism \( f(x) = kx \) such that \( \gcd(k, n) = 1 \). Furthermore, it was conjectured that this sufficient condition was necessary. However, only three years later there was a counterexample given to this claim. [6]

**Theorem 2.9.** Let \( G \) be a circulant graph with vertex set \( V \), where \( |V| = n \geq 1 \), and edge set \( E \). Then the function \( f : \mathbb{Z}_n \to \mathbb{Z}_n \), defined by \( f(x) = kx \), where \( \gcd(k, n) = 1 \), is a graph isomorphism.

**Proof.** It is sufficient to show that \( f \) is bijective and that \((a, b)\) is an edge in \( G \) only if \((f(a), f(b))\) is an edge in the image of the edges of \( G \) for all \( a, b \in V \).

- **Injective:** Let \( a, c \in \mathbb{Z}_n \) and suppose \( f(a) = f(c) \).
  
  Then \( ka = kc \).
  
  Since \( k \in \mathbb{Z}_n \) and \( \gcd(k, n) = 1 \), by Proposition 2.4, there exists \( k^{-1} \in \mathbb{Z}_n \).
  
  Then \( ka = kc \) implies that \( k^{-1}ka = k^{-1}kc \), and then \( a = c \).

- **Surjective:** Consider \( b \in \mathbb{Z}_n \).
  
  Then there exists \( k^{-1}b \in \mathbb{Z}_n \), and so \( f(k^{-1}b) = k(k^{-1}b) = (kk^{-1})b = (1)b = b \).

- **Edge criterion:** Consider two arbitrary vertices \( v \) and \( w \) of \( G \). The fact that \((v, w)\) is an edge in \( G \) only if \((f(v), f(w))\) is an edge in the image of the edges of \( G \) follows immediately from the fact that \( f \) is a bijection. \[\boxed{}\]

**Observation 2.10.** Let \( n > 1 \). When operating on circulant graphs, the graph isomorphism \( f : \mathbb{Z}_n \to \mathbb{Z}_n \), defined by \( f(x) = kx \), where \( \gcd(k, n) = 1 \), sends \( C[n, \{s, t\}] \) to \( C[n, \{ks, kt\}] \).

**Proof.** Let \( G \) denote \( C[n, \{s, t\}] \) and let \( G' = \{V', E'\} \), where \( V' \) and \( E' \) are the images of the vertex and edge sets of \( G \) under \( f \), respectively. Since \( f \) is a bijection (see proof of Theorem 2.9), the size of the vertex set of \( G \) is equal to the size of the vertex set of \( G' \). Thus, their vertex sets are identical.

Consider an arbitrary vertex \( v \) of \( G \). Then \( v \) is adjacent to \( v + s, v - s, v + t, \) and \( v - t \). When \( f \) operates on \( G \), notice \( kv \) is adjacent to \( k(v + s) = kv + ks, k(v - s) = kv - ks, k(v + t) = kv + kt \), and \( k(v - t) = kv - kt \). Thus, for every vertex \( w \) of \( G' \), \( w \) is adjacent to \( w + ks, w - ks, w + kt, \) and \( w - kt \).

Therefore, \( G' \) is a circulant graph on \( n \) vertices with connections \( ks \) and \( kt \), i.e., \( C[n, \{k \cdot s, k \cdot t\}] \). \[\boxed{}\]

**Remark.** Given a circulant graph and graph isomorphism, the value of Observation 2.10 is that it allows one to operate directly on the connection set of the circulant graph via the isomorphism rather than having to do a bunch of “gory” work calculating vertices and edge connections via the operation.

**Theorem 2.11.** If \( \gcd(n, k) = 1 \), and \( n = tk + a \), for some \( 1 \leq a \leq k - 1 \), then \( C[n, \{1, \left\lfloor \frac{n}{k} \right\rfloor\}] \cong C[n, \{a, k\}] \).
Proof. Since $gcd(n, k) = 1$, it follows from Theorem 2.9 that the function $f : \mathbb{Z}_n \to \mathbb{Z}_n$, defined by $f(x) = kx$, is a graph isomorphism for ciculant graphs. Define $G = C[n, \{1, \lceil \frac{n}{k} \rceil \}]$. Since $n \equiv a \mod k$, $n - a \equiv 0 \mod k$, which implies that $k|\,(n-a)$. Thus, $C[n, \{1, \lceil \frac{n}{k} \rceil \}] = C[n, \{1, \frac{n-a}{k} \}]$. It follows from Observation 2.10, that operating on $G$ by $f$ yields $C[n, \{1, \frac{n-a}{k} \}] = C[n, \{k, n-a\}]$. Then Lemma 2.1 gives $C[n, \{1, \frac{n-a}{k} \}] = C[n, \{a, k \}]$. Therefore, $C[n, \{1, \frac{n-a}{k} \}] \sim C[n, \{a, k \}]$. ■

2.3 Specific Cases

Observation 2.12. If every vertex of a graph $G$ has the same degree, then the degree of the vertices of $G$ is a lower bound for $Z(G)$.

Lemma 2.13. If a circulant graph $G = C[n, \{s, t \}]$, is such that $nt \neq 2$, then the degree of every vertex of $G$ is 4.

Proof. Consider an arbitrary vertex $v$ of $G$; $v$ shares an edge with $v-s$, $v+s$, $v-t$, and $v+t \mod n$. Now, $v-s \neq v-t$, and $v+s \neq v+t$, because $s \neq t$.

By way of contradiction, suppose that $v-s = v+t$ or $v-t = v+s$. Then $s+t \equiv 0 \mod n$, which is a contradiction because $\frac{n}{t} \neq 2$ implies that $1 \leq s < t < \frac{n}{2}$ which implies that $s+t < n$.

Suppose that $v-t = v+t$. Then $2t \equiv 0 \mod n$, which implies that $\frac{n}{t} = 2$, which is a contradiction.

Similarly, suppose $v-s = v+s$. Then $2s \equiv 0 \mod n$ which is a contradiction since $1 \leq s < t < \frac{n}{2}$.

Theorem 2.14. If $G = C[n, \{s, t \}]$ where $0 < s < t$, then $Z(G) \leq 2t$ and $pt(G, S) = \lceil \frac{n-2t}{2(t-s)} \rceil$.

Proof. Consider an arbitrary vertex $v$ in $Z_n$. This vertex $v$ has the adjacent vertices $v+s$, $v-s$, $v+t$, and $v-t$. Let the set $S_0 = \{(v+1)-t, ..., v, (v+1), ..., v+t\}$ of $2t$ consecutive vertices of $G$ be colored blue.

Consider the vertices in $S_0$ from $(v+1)-t$ to $v$. Of these vertices the $s$ number of vertices on the end of the set $S_0$ cannot force. These are the vertices from $(v+1)-t$ to $v-(t-s)$, ...
and each have two adjacent blue vertices that are $+s$ and $+t$ away, and two adjacent white vertices that are $-s$ and $-t$ away. So, each of these $s$ vertices have two white neighbors and cannot force at this time. However, the next $t-s$ vertices from $(v+1)-(t-s)$ to $v$ each force on the first iteration. These vertices can be denoted as $(v+1)-(t-s)+k$ for $k = 0, 1, ..., (t-s) - 1$. They each have three adjacent blue vertices, $(v+1)-(t-s)+k+s$, $(v+1)-(t-s)+k+s$, and $(v+1)-(t-s)+k-s$, and force their fourth adjacent vertex, $(v+1)-(t-s)+k-t$, in the first iteration.

Now consider the vertices from $(v+1)$ to $v+t$. Of these, the last $s$ vertices from $(v+1)+(t-s)$ to $v+t$ cannot force because they have two adjacent white vertices that are $+s$ and $+t$ away. The other $t-s$ vertices denoted $v+k$ where $k = 1, ..., t-s$ force their one adjacent white vertex, $v+k+t$, in the first iteration.

In general there are $2s$ blue vertices that cannot force at each iteration. The number of vertices which do force is, therefore, $2(t-s)$ in each iteration. The number of iterations necessary to color derive the entire graph $G$ is the number of vertices that are initially white, $(n-2t)$, divided by the number that force in each timestep, rounded upwards because timesteps are strictly integers. Propagation time is, therefore, $pt(G, S) = \lceil \frac{n-2t}{2^t(s-1)} \rceil$.

This theorem gives an upper bound for the zero forcing number for all circulant graphs $C[n, \{s, t\}]$. However, it is important to note that there are many circulant graphs of this form that have smaller zero forcing sets or sets which produce smaller propagation times. Some of the following theorems and propositions will cover such cases.

The following corollary is given without proof since it follows immediately from Theorem 2.14. Its value consists in the remark that follows.

**Corollary 2.14.1.** If $G = C[n, \{1, k\}]$ where $1 < k \leq \lfloor \frac{n}{2} \rfloor$, then $Z(G) \leq 2k$ and $pt(G, S) = \lceil \frac{n-2k}{2k-2} \rceil$.

Remark. From data that was computed as a reference at the beginning of the project, it appears that if $G = C[n, \{1, k\}]$ where $1 < k \leq \lfloor \frac{n}{2} \rfloor$ and $n > k^2$, then $Z(G) = 2k$. Furthermore, an appropriate zero forcing set for $G$ is $\{0, 1, ..., 2k-1\}$, and, as the corollary states, $pt(G, \{0, 1, ..., 2k-1\}) = \lceil \frac{n-2k}{2k-2} \rceil$.

**Proposition 2.15.** If $G = C[n, \{1, 2\}]$, then $Z(G) = 4$, $pt(G) \leq \lceil \frac{n-4}{2} \rceil$ for all $n > 4$.

*Proof.** Since $n > 4$, $\lceil \frac{n}{2} \rceil \neq 2$. Then by Lemma 2.13, the degree of every vertex of $G$ is equal to 4. It follows from Observation 2.12 that $Z(G) \geq 4$. Since 2.14 implies that $Z(G) \leq 4$, it is clear that $Z(G) = 4$. A set containing four vertices that forces the graph of $G$ will now be considered.

**Base Case:** Assume that the vertices $\{-k, -k+1, ..., 3+k\}$ are blue for $k = 0$. Now, $\{0, 1, 2, 3\}$ are blue. Since 1 is adjacent to $-1, 0, 2$, and 3 and three of these are blue, 1 forces $-1$ blue. Since 2 is adjacent to $0, 1, 3$, and 4 and three of these are blue, 2 forces 4 blue. Thus, $\{-k, -k+1, ..., 3+k\}$ are blue for $k = 1$ (and $k = 0$).

**Inductive Step:** Assume that the vertices $\{-m, -m+1, ..., 3+m\}$ are blue for some $m = j > 0$. Since $-j+1$ is adjacent to $-j-1, -j, -j+2$, and $-j+3$ and three of these are blue, $-j+1$ forces $-j-1$ blue. Since $j+2$ is adjacent to $j, j+1, j+3$, and $j+4$ and three of these are blue, $j+2$ forces $j+4$ blue. Thus, the set $\{-j-1, -j, ..., 4+j\}$ is blue, which implies that $\{-m, -m+1, ..., 3+m\}$ is blue for $m = j+1$. This process continues
until no more vertices of the graph may be forced. (Note: since two vertices are being forced at a time and the starting number of white vertices is \( n - 4 \), no more vertices may be forced after \( m = \lceil \frac{n-4}{2} \rceil \).

It will now be shown that every vertex of \( G \) is blue. From the induction above, the vertex set \( \{-m, -m + 1, \ldots, 3 + m\} \) is blue for \( m = \lceil \frac{n-4}{2} \rceil \). Considering the use of modular arithmetic, this is equal to the set \( \{0, 1, \ldots, 2m + 3\} \). It will be shown that this is the entire vertex set of \( G \).

**Case 1:** If \( n \) is odd, then \( m = \lceil \frac{n-4}{2} \rceil = \frac{n-3}{2} \), which implies that \( 2m + 3 = n \equiv 0 \mod n \). So, consider the set \( \{0, 1, \ldots, 2m + 2\} \). Since \( 2m + 3 = n \), \( 2m + 2 = n - 1 \), so this set is actually \( \mathbb{Z}_n \), which is the vertex set of \( G \).

**Case 2:** If \( n \) is even, then \( m = \lceil \frac{n-4}{2} \rceil = \frac{n-4}{2} \), which implies that \( 2m + 3 = n - 1 \). So, \( \{0, 1, \ldots, 2m + 3\} \) is actually \( \mathbb{Z}_n \), which is the vertex set of \( G \).

Since all the vertices of \( G \) have been forced blue by a set containing 4 elements in \( \lceil \frac{n-4}{2} \rceil \) iterations of forcing, \( pt(G) \leq \lceil \frac{n-4}{2} \rceil \).

A proof of the following theorem was written in 2010 [7]. The proposition that follows adds a proof for the propagation time for the particular set that was specified in the 2010 theorem.

**Theorem 2.16.** If \( G = C[k^2, \{1, k\}] \), then \( Z(G) \leq 2k - 1 \).

**Proposition 2.17.** If \( G = C[k^2, \{1, k\}] \) and \( Z(G) \leq 2k - 1 \), then \( pt(G, S) = k - 1 \) for \( S = \{(0)k + 0, (1)k + 0, \ldots, (k - 1)k + 0, (1)k + 1, (2)k + 1, \ldots, (k - 1)k + 1\} \).

**Proof.** In the proof for zero forcing number [7], each vertex is denoted \( jk + i \) where \( j \) represents a column, and \( i \) represents a row. This can be represented with a lattice structure as seen in the figure.

![Figure 6: Lattice structure for \( C_{36}(1, 6) \)](image)

The proof in [7] specifies a zero forcing set consisting of two consecutive rows minus one vertex colored blue, or equivalently \( S = \{(0)k + 0, (1)k + 0, \ldots, (k - 1)k + 0, (1)k + 1, (2)k + 1, \ldots, (k - 1)k + 1\} \). The one vertex in these two rows which is not blue in \( S \) is vertex \((0)k + 1\). Two separate cases are considered to show how vertices are forced.

**Case 1:** If \( n \) is even, forcing progresses as follows. Let \( x \) be the number of colored vertices in a row. The two rows with blue vertices will each force \( x - 2 \) vertices in their next
consecutive row. This will continue with the two outermost rows that have blue vertices forcing vertices in the next consecutive rows, until each row has at least one blue vertex. For the case $n$ is even, this occurs in $\frac{k-2}{2}$ timesteps. When there are blue vertices in every row, the row $i = \frac{k}{2}$ has only one blue vertex, which is in column $j = \frac{k}{2}$. In the next timestep, the vertex at row $i = \frac{k}{2}$, column $j = \frac{k}{2} - 1$, is forced and now there are two columns that are entirely blue. The total number of timesteps necessary to color two columns blue is $\frac{k-2}{2} + 1$.

From the point at which two columns are colored, at each timestep the outermost two entirely blue columns will each force the remaining white vertices in the next consecutive column. Since there are first two completely blue columns out of the $k$ columns, and two more are fully colored each timestep, the graph is color-derived in another $\frac{k-2}{2}$ iterations.
The total propagation time for \( n \) even is \( \frac{k-2}{2} + 1 + \frac{k-2}{2} = (k-2) + 1 = k-1 \) timesteps.

**Case 2:** If \( n \) is odd, again the two rows with blue vertices will each force \( x - 2 \) vertices in their next consecutive row. This continues with the two outermost rows that have blue vertices forcing vertices in the next consecutive rows, until each row except one has at least one blue vertex. Two rows have forced at each timestep and reaching the point at which all but one row has at least one blue vertex takes \( \frac{k-3}{2} \) timesteps. There are two blue vertices in row \( i = \frac{k-1}{2} \) which cannot force because they both have two adjacent white vertices. However, forcing continues with the four vertices in row \( i = \frac{k+1}{2} \) forcing two vertices in \( j = \frac{k-1}{2} \), at which point column \( j = \frac{k-1}{2} \) is entirely blue. In the next time step the remaining two vertices in columns \( i = \frac{k-3}{2} \) and \( j = \frac{k+1}{2} \) are forced by vertices in column \( j = \frac{k-1}{2} \).

There are now three entirely blue columns and the two outermost blue columns will force the remaining white vertices in their next consecutive column. Since there are three colored columns out of \( k \) total, and two more are fully colored blue at each timestep, the graph will be color-derived in another \( \frac{k-3}{2} \) iterations.

The total propagation time for \( n \) odd is then \( \frac{k-3}{2} + 1 + \frac{k-3}{2} = (k-3) + 2 = k-1 \). For circulant graphs \( C[k^2, \{1, k\}] \) and \( Z(G) \leq 2k-1 \), with zero forcing set consisting of two consecutive rows minus one colored blue, \( pt(G, S) = k-1 \).

**Lemma 2.18.** If a circulant graph \( G = C[n, \{s, t\}] \) is such that \( \frac{n}{t} = 2 \) and \( s < t \), then the degree of every vertex of \( G \) is 3.

**Proof.** Suppose \( \frac{n}{t} = 2 \). Consider an arbitrary vertex \( v \) of \( G \); \( v \) shares an edge with \( v-s \), \( v+s \), \( v-t \), and \( v+t \) mod \( n \). It will be shown that \( v-t \equiv v+t \) mod \( n \) and that the rest are distinct. Obviously, \( v-s \neq v-t \), and \( v+s \neq v+t \), since \( s \neq t \).

Now \( \frac{n}{t} = 2 \) implies that \( n = 2t \). Then \( 2t \equiv 0 \) mod \( n \). Then \( v + 2t \equiv v \) mod \( n \), which means that \( v + t \equiv v - t \) mod \( n \).

Now suppose that \( v-s = v+t \) or \( v-t = v+s \). Then \( s + t \equiv 0 \) mod \( n \), which is impossible since \( 1 \leq s < t = \frac{n}{2} \) so that \( s + t < n \).

If \( v-s = v+s \), then \( 2s \equiv 0 \) mod \( n \), which also cannot be since \( 1 \leq s < t = \frac{n}{2} \) so that \( 2s < n \).

Thus, \( v - t = v + t \), but are distinct from \( v-s \) and \( v+s \) and thus, \( v \) has exactly three adjacent vertices. \( \blacksquare \)

**Proposition 2.19.** Given a connected graph \( G = C[n, \{s, t\}] \) in which \( n > 4 \), \( \frac{n}{t} = 2 \), and \( s < t \), \( Z(G) = 4 \).

**Proof.** It follows from Lemma 2.18 that \( \frac{n}{t} = 2 \) implies that the degree of every vertex of \( G \) is equal to 3. Then by Observation 2.12, \( Z(G) \geq 3 \).

Pick an arbitrary vertex of \( G \) and call it \( v \). It follows from the proof of Lemma 2.18 that \( v \) shares and edge with \( v-s \), \( v+s \), and \( v+t \) mod \( n \). The vertex \( v \) and its neighbors will now be used to force other vertices of the graph. It will be shown that any set of exactly three vertices containing \( v \) will not force the entire graph. Let \( v \) and two of its neighbors be blue.

**Case 1:** Let \( v \), \( v+s \), and \( v+t \) be blue. Then \( v \) can force \( v-s \) blue, but no more vertices can be forced blue because \( v+s \), \( v+t \), and \( v-s \) each have two white neighbors. For example, the vertices \( v+2s \) and \( v+s+t \) are adjacent to \( v+s \) are both white. The others are similar. By symmetry, \( v \), \( v-s \), and \( v+t \) also each have two white neighbors.
Case 2: Let \( v, v-s, \) and \( v+s \) be blue. Then \( v \) can force \( v+t \) blue, but no more vertices can force others blue because \( v-s, v+s, \) and \( v+t \) each have two white neighbors.

Since none of the 3-vertex sets containing \( v \) can force the entire graph of \( G \) blue, consider a 4-vertex set that forces the entire graph.

**Base Case:** Assume that \( \{-ks,(k+1)s,t-ks,t+(k+1)s\} \) are blue for \( k = 0 \), that is, \( \{0,s,t,t+s\} \) are blue. The vertex 0 is adjacent to \( -s, s, \) and \( t \). Since \( s \) and \( t \) are blue, 0 forces \( -s \). Similarly, since two of the three neighbors of \( s, t, \) and \( t+s \) are blue and one is white, these force 2\( s, t-s, \) and \( t+2s \), respectively. Thus, all vertices in the set \( \{-ks,(k+1)s,t-ks,t+(k+1)s\} \) are blue for \( k = 0 \) and \( k = 1 \).

**Inductive Step:** Assume that \( \{-js,(j+1)s,t-js,t+(j+1)s\} \) for all \( j \leq k \) for \( k \geq 0 \). It will be shown that these four vertices are forced with \( j = k + 1 \). Now, \( \{-ks,(k+1)s,t-ks,t+(k+1)s\} \) are blue. The vertex \( -ks \) is adjacent to \( -(k+1)s, (k+1)s, \) and \( t-ks \). Since \( (k+1)s \) and \( t-ks \) are blue, \( -ks \) forces \( -(k+1)s \). Similarly, since two of the three neighbors of \( (k+1)s, t-ks, \) and \( t+(k+1)s \) are blue and one is white, these force \( (k+2)s, t-(k+1)s, \) and \( t+(k+2)s \), respectively, until no more vertices of the graph may be forced. (Note: since four vertices are being forced at a time and the starting number of white vertices is \( n-4 \), no more vertices may be forced after \( k = \left\lceil \frac{n-4}{4} \right\rceil \).)

After having performed the induction as described above, it will now be shown that every vertex of \( G \) is blue. Consider an arbitrary vertex of \( G \) and call it \( x \). Since \( G \) is a connected circulant graph, there exists a walk from 0 to \( x \). This implies that \( x = at + bs \) for some \( 0 \leq a \leq 1 \) and \( b \geq 0 \). (Note: the restriction “\( 0 \leq a \leq 1 \)” is imposed because \( 2t = n \equiv 0 \) \( \mod n \).) So either \( x = bs \) or \( x = t+bs \) for some \( b \geq 0 \). Since the vertices \( -ks \) and \( t-ks \) are blue for all \( k \) in \( \{0,1,\ldots, \left\lceil \frac{n-4}{4} \right\rceil \} \), \( x = bs \) or \( x = t+bs \) must be blue, which implies that \( x \) must be blue. Thus, every vertex of \( G \) must be blue.

Since the vertex set \( \{v,v \pm s,v+t,v+t \pm s\} \) is a zero forcing set for the graph of \( G \) and no smaller zero forcing set exists, it is concluded that \( Z(G) = 4 \).

### 3 Future Research

One goal of this project was to find a general rule for the zero forcing numbers and propagation times of all circulant graphs of the form \( C[n, \{s,t\}] \). Although the results that were obtained give some boundaries for zero forcing numbers for all graphs of this type, there are many more cases and patterns which ought to be explored. Some questions were also raised about how a zero forcing set might be manipulated or augmented to create a particular propagation time or range of propagation times. Alternatively, since the study of zero forcing is very closely related to the study of minimum rank in matrices, exploring this problem from a linear algebra perspective could be an important part of future research. A better understanding of zero forcing numbers could also be achieved in further research through rigorous study of the structure of circulant graphs.
References


The code that follows was written and run in Octave 3.6.4 and was of much aid in the study of zero forcing number of particular circulant graphs.

```matlab
function out = makeC(n,k)
    A = zeros(n,n);
    for el = k
        k = [k,n-el];
    end
    for i = 1:n
        for j = 1:n
            for l = k
                if mod(abs(j-i),n) == l
                    A(i,j) = 1;
                end
            end
        end
    end
    out = A;
end

function out = totatives(n)
    out = [1];
    if isprime(n)==1
        out = 1:n-1;
    else
        for i = 2:n-1
            if gcd(n,i)==1
                out = [out,i];
            end
        end
    end
end

function out = iscirc(G)
    if size(G)(1)!=size(G)(2)
        error('G not square matrix, bro. ')
    else
        n = size(G)(1);
    end
    t = G(:,);
```
for j = 1:n
    t(j,:) = circshift(G(j,:),[0,n-j]);
end
if rank(t) <= 1
    out = 1;
else
    out = 0;
end

function out = isconn(G)
    if size(G)(1) != size(G)(2)
        error('G is not square matrix, bro.');
    else
        n = size(G)(1);
    end

    % We start at vertex 1 because it's convenient.
    verts = [1];
    t = touches(verts,G);

    while ~isequal(verts,t)
        verts = unique([verts,t]);
        t = touches(verts,G);
    end

    out = isequal(verts,1:n);
end

function out = findij(G)
    if iscirc(G) != 1
        error('G is not circulant, bro.');
    else
        n = size(G)(1);
    end

    out = [0,0];
    k = floor(n/2)+1;
    b = touches(1,G);
function out = structequiv(A)
    if size(A)(1)~=size(A)(2)
        error('A not square matrix, bro. ')
    else
        n = size(A)(1);
    end
    out = [];
    isA = iscirc(A);
    if isA
        Aij = findij(A);
        tots = totatives(n);
        for j = tots(2:(length(tots)/2))
            posij = mod(Aij*j,n);
            for k = 1:length(posij)
                if posij(k) > floor(n/2)
                    posij(k) = n-posij(k);
                end
            end
            posij = sort(posij);
            out = [out;posij];
        end
        out = sortrows(out);
    end
end

function out = isstructequiv(n1,i1,j1,n2,i2,j2)
    out = 0;
    if n1~=n2
        return
    end
    if (i1 == i2 && j1 == j2) || (i1 == j2 && j1 == i2)
        out = 1;
    return
    end
    trying = structequiv(makeC(n1,[i1,j1]));
    for row = 1:size(trying)(1)
        if isequal([i2,j2],[trying(row,:)])
            out =1;
        end
    end
end

out = b(1:ceil(length(b)/2))-1;
function [out, p] = try2force(G, CV)
    if size(G)(1)~=size(G)(2)
        error('G not square matrix, bro.')</n    else
        n = size(G)(1);
    end

    V = zeros(1,n); V(CV) = 1;
    newV = V;
    pt = 0;
    while length(CV)!=n
        for j = CV
            adj = find(G(j,:)==1);
            adjNCV = zeros(1,n);
            adjNCV(adj) = 1;
            adjNCV = adjNCV.*[V-1];
            if sum(adjNCV) == -1 && V(j) == 1
                CV = [find(adjNCV==-1),CV];
                newV = zeros(1,n); newV(CV) =
                continue
            end
        end
        if sum(newV)==sum(V)
            break
        else
            V = newV;
            CV = unique(CV);
            pt+=1;
        end
    end

    if length(CV) == n
        success = 1;
    end
return
end
else
    success = 0;
end

if nargout == 0
    out = success;
elseif nargout == 1
    out = CV;
elseif nargout == 2
    out = CV;
    p = pt;
end
end

function out = next_d2b(a)
    out = a;
    out(end)+=1;
    i = 0;
    lenout = length(out);
    while i<=lenout-1
        if out(end-i)==2
            out(end-i) = 0;
            out(end-(i+1))+=1;
        end
        i+=1;
    end
end

function [out, p, ZFS] = ZFN(G,startnumCVs)
    %startnumCVs is equivalent to minimum number of CVs needed to force just one vertex.
    if size(G)(1)!=size(G)(2)
        error('G not square matrix,bro. ')
    else
        n = size(G)(1);
    end
    if nargin < 2
if nargout == 3
    [out, p, ZFS] = ZFN(G,1);
    return
elseif nargout == 2
    [out, p] = ZFN(G,1);
    return
else
    out = ZFN(G,1);
    return
end

%Below is the business end of the code.

s = [];
if startnumCVs == 1
    startnumCVs = min(sum(G));
end

out = p = Inf;
circtrue = iscirc(G);

if circtrue && length(ourij = findij(G))==2;  %
    everything inside the ifcirctrue statement is
    looking for patterns
    %looking for lines here in the for statement.
    for k = 2:floor(n/2)
        if n>k*k
            if isstructequiv(n,ourij(1),
                ourij(2),n,1,k)
                for t = totatives(n)
                    posij = mod([1,
                        k]*t,n);
                    for spot =
                        length(posij
                    )
                        if
                            posij
                            (spot
                            ) >
                            floor
                            (n
                            /2)
                            posij

20
(spot)
= n - posij

posij = sort(posij);

if isequal([posij],[ourij])
break
end

end

posij = sort(posij);

out = 2*k;
p = ceil((n-2*k)/(2*k-2));

ZFS = sort(mod([1:2*k]*t,n));
ZFS = ZFS- ZFS(1)+1;

return

else
break
end

end

end

justZFS = 0;
if circtrue
ourijs = ';
for b = ourij
ourijs = [ourijs,int2str(b),',']
end
ourijs =ourijs(1:end-1);
filename = ['C',int2str(n),',(',ourijs,').csv']
];

if circtrue && exist(filename)
    update = csvread(filename);
    a = update(4,:);
    suma = sum(a);
    out = update(1,:)(update(1,:)!=-1);
    p = update(2,:)(update(2,:)!=-1);
    ZFS = update(3,:)(update(3,:)!=-1);
else
    a = zeros(1,n);
    a(end) = 1;
    suma = sum(a);
end

if justZFS
    writes = 0;
tic
    while suma < n
        if suma < startnumCVs || suma > out
            a = next_d2b(a);
            suma = sum(a);
            continue
        end
        nCV = [1:n].*[a];
        nCV = nCV(nCV!=0);
        [tCV,newp] = try2force(G,nCV);
        lentCVeqn = length(tCV)==n;
        if newp < p && lentCVeqn
            ['propagation time should be',
             newp]
            error([['findings data wrong at '
             ',n','
             ,{"',ourijs,'"}']].)
        elseif lentCVeqn
            ZFS = nCV-min(nCV)+1;
            %return
        end
    end
if toc > writes*5*60 && circttrue && exist('ZFS')
    writes +=1;
    tempfindings = -ones(4,n);
    tempfindings(4,:) = a;
    tempfindings(3,1:length(ZFS))= ZFS;
    tempfindings(2,1) = p;
    tempfindings(1,1) = out;
    csvwrite(filename,[tempfindings ]);
end

a = next_d2b(a);
suma = sum(a);
end

else
    writes = 0;
tic
    while suma < n
        if suma < startnumCVs || suma > out
            a = next_d2b(a);
            suma = sum(a);
            continue
        end

        nCV = [1:n].*[a];
        nCV = nCV(nCV!=0);
        [tCV,newp] = try2force(G,nCV);
        if length(tCV)==n
            if suma<out
                out = suma;
                p = newp;
                ZFS = nCV-min(nCV)+1;
            elseif suma==out && newp < p
                p = newp;
                ZFS = nCV-min(nCV)+1;
            end

            verts = nCV;
            s = [s;CV];  % This gets us the full list of successes
        end
end
if toc > writes*5*60 && circtru &
exist('ZFS')
    writes +=1;
    tempfindings = -ones(4,n);
    tempfindings(4,:) = a;
    tempfindings(3,1:length(ZFS))=
        ZFS;
    tempfindings(2,1) = p;
    tempfindings(1,1) = out;
    csvwrite(filename,[tempfindings ]);
end

    a = next_d2b(a);
    suma = sum(a);
end

if exist(filename)
delte(filename);
end

1 machinenumber = 0;
2
3 if exist('findings.csv') ==2 && machinenumber == 0
4    findings = csvread('findings.csv');
5 end
6
7 n = 4
8 spot = 1;
9 starti = startj = 1;
10 forbidden = [];
11 while n != findings(end,1)+1
12     n
13     for i = starti:floor(n/2)
14         for j = startj:floor(n/2)
15             if i!=j
16                 truth1 = 1;
17                 for k = 1:size(forbidden)(1)
18                     if isequal([forbidden(k ,:)],i,j)
19                         truth1 = 0;
20                 end
21         end
22     end
23 end
if !isconn(G = makeC(n,[i,j]))
    && isequal(findings(spot,1:3),[n,i,j])
    findings = [findings(1:spot-1,:);findings(spot+1:end,:)];
    forbidden = [forbidden;j,i;structequiv(G);fliplr(structequiv(G))];
    csvwrite('findings.csv',findings);
end
end

elseif !truth1 && isequal(findings(spot,1:3),[n,i,j])
    findings = [findings(1:spot-1,:);findings(spot+1:end,:)];
    forbidden = [forbidden;j,i;structequiv(G);fliplr(structequiv(G))];
    csvwrite('findings.csv',findings);
end

elseif truth1 && isequal(findings(spot,1:3),[n,i,j])
    spot += 1;
    forbidden = [forbidden;j,i;structequiv(G);fliplr(structequiv(G))];
end
elseif truth1 && isconn(G = makeC(n,[i,j]))
    forbidden = [forbidden;j,i;structequiv(G);fliplr(structequiv(G))];
    [Z,p] = ZFN(G);
    findings = [findings(1:
spot-1,:);n,i,j,Z,p;
findings(spot:end,:)
];
[n,i,j,Z,p]
csvwrite('findings.csv'
,findings);
spot+= 1;
end
end

if spot <= size(findings)(1)
if findings(spot,1)!n
break
else
continue
end
else
break
end
end
if spot <= size(findings)(1)
if findings(spot,1)!n
break
else
continue
end
else
break
end
end
forbidden = [];
starti = 1;
startj = 1;
n+=1;
end
%

if exist('findings.csv')=2 && machinenumber == 0
findings = csvread('findings.csv');
startn = findings(end,1);
n = startn;
starti = findings(end,2)+1;
startj = findings(end,3)+1;
nidx = find(findings(:,1)==18);
forbidden = findings(nidx,2:3);
elseif exist('findings.csv')==2 && machinenumber != 0
findings = csvread('findings.csv');
startn = findings(end,1);
n = startn+machinenumber;
starti = startj = 1;
forbidden = [];
else
startn = 4;
starti = startj = 1;
findings = [];
forbidden = [];
end

n = findings(end,1);
starti = findings(end,2);
startj = findings(end,3)+1;
idx = find(findings(:,1)==n);
forbidden = fliplr(findings(idx,2:3));
while n<Inf %formerly 'for n = startn:30'

    for i = starti:floor(n/2)
        for j = startj:floor(n/2)
            if i!=j
                truth = 1;
                for k = 1:size(forbidden)(1)
                    if isequal([forbidden(k ,:)],[i,j])
                        truth = 0;
                        break
                    end
                end
                if truth && isconn(G = makeC(n ,[i,j]))
                    forbidden = [forbidden ;[j,i];structequiv(G )];fliplr(structequiv (G))];
                    [Z,p] = ZFN(G);
                    findings = [findings;n,
\textbf{B The Data}

The data below was generated by the script \texttt{onescript.m} and saved as a .csv file. The data below can be read as follows: a circulant graph of the form $C[n,\{s,t\}]$ has zero forcing number $Z$ and propagation time $pt$.

\begin{table}[h]
\centering
\begin{tabular}{cccc}
\hline
$n$ & $s$ & $t$ & $Z$ & $pt$
\hline
4 & 1 & 2 & 3 & 1 \\
5 & 1 & 2 & 4 & 1 \\
6 & 1 & 2 & 4 & 1 \\
6 & 1 & 3 & 4 & 1 \\
6 & 2 & 3 & 3 & 1 \\
7 & 1 & 2 & 4 & 2 \\
8 & 1 & 2 & 4 & 2 \\
8 & 1 & 3 & 6 & 1 \\
8 & 1 & 4 & 4 & 1 \\
9 & 1 & 2 & 4 & 3 \\
9 & 1 & 3 & 5 & 2 \\
10 & 1 & 2 & 4 & 3 \\
10 & 1 & 3 & 6 & 1 \\
10 & 1 & 4 & 6 & 2 \\
10 & 1 & 5 & 4 & 2 \\
10 & 2 & 5 & 4 & 2 \\
\hline
\end{tabular}
\end{table}
24   2  3  6  9  
24   3  4  8  8  
24   3  8  6  3  
25   1  2  4  11 
25   1  3  6  5  
25   1  4  8  3  
25   1  5  9  4  
25   1  7  8  9  
25   1  9  6  10 
25   1 10 10 2  
25   1  2  4  11 
26   1  3  6  5  
26   1  4  8  3  
26   1  5  10  2 
26   1  6  8  5  
26   1  7  8  5  
26   1  8  6  10 
26   1 10 8 9  
26   1 12 14 6  
26   1 13 4 6  
26   1 2 13 4 6 
27   1  2  6  6  
27   1  4  8  4  
27   1  5  10  3 
27   1  6  8  10 
27   1  8 10 9  
27   1  9  6  4  
27   112 6 111  
28   1  2  4  12 
28   1  3  6  6  
28   1  4  8  4  
28   1  5  10  3 
28   1  6 10 3  
28   1  7  8  3  
28   1  8  8  10 
28   1 10 6 11  
28   1 12 8 5  
28   1 13 16 3  
28   1 14 4 6  
29   1  2  4  13 
29   1  3  6  6  
29   1  4  8  4  
29   1  5  10  3 
29   1  8  8  11 

31
<table>
<thead>
<tr>
<th>29</th>
<th>1</th>
<th>9</th>
<th>6</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>1</td>
<td>12</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>6</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>9</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>10</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>