Zero Forcing in Generalized Petersen Graphs

Sarah Gibbons, Taylor Baumgard, Marcus Cisneros, Dr. Daniela Ferrero, Enrique Gomez-Leos, Nick Montana
Department of Mathematics, College of Science and Engineering, Texas State University

Abstract

Zero-forcing is a graph process introduced independently in linear algebra, quantum physics and electrical engineering, between the years 2002 and 2008. Due to its broad range of applications, it has attracted the interest of mathematicians, computer scientists, theoretical physicists and engineers.

A graph consists of a set of vertices together with some edges joining pairs of points. Assume the vertices of the graph are colored in blue and white. Iteratively apply the following color changing rule until it does not change the color of any vertex: if a blue vertex has exactly one white neighbor, its white neighbor turns blue. At the end, if all vertices are blue, the initial set of blue vertices is a zero-forcing set. The minimum number of vertices in a zero-forcing set is the zero-forcing number of the graph.

Graphs are ubiquitous network models, and their zero forcing numbers provide valuable network information. However, determining the forcing number of an arbitrary graph results in an NP-complete problem, so there is no efficient algorithm to compute it. Therefore, it is important to find families of graphs for which there is a formula for their zero-forcing number.

In this work, we construct zero-forcing sets in Generalized Petersen Graphs (GPG), prove they are minimal, and derive a formula for the zero-forcing number. In addition, we open up avenues for further research, since GPG are in the intersection of several important classes of graphs to which our results could be extended.

A Possible Code:

The pseudo code below outlines the method used to compute Z(G) for GP(n,k). The *outer, inner* and *marked* represent the outer vertices {u1...u5}, the inner vertices {v1...v5} and

```
loop outer :  ( i = 0; i < n; i++)
                    [ ( i + 1 ) mod n]
                    [ ( i + n - 1 ) mod n ]
                    inner [i]
loop inner :  ( j = 0; j < n; j++)
                    [ ( j + k ) mod n]
                    [ ( i + n- k ) mod n ]
                    inner [j]
...to finish
```

Listing 1 . Zero-forcing number pseudo-code. Computes the minimum zero forcing number of generalized petersen graphs.

Header

1 Zero Forcing

Let us define zero forcing formally. To do so, we need to give a formal definition of the changes in the set of vertices that are colored black as the color changing rule is applied. This is done by defining a sequence of sets $\{B_n(S)\}_{n\geq 0}$ that contains the black vertices when the color changing rule has been applied n times and the initial set of black vertices is S. Initially, only the vertices in the set B are black, so we define $B_0(B) = B$ (that is, if the color changing rule has not been used, the only black vertices are those intially black, which are those in the set B). Each time we apply the color changing rule, we look for a black vertex v with exactly one white neighbor that we will call w. How do we say this mathematically? If $B_n(B)$ is the set of black vertices at the moment of applying the color changing rule, we look for a vertex $v \in B_n(B)$ (this guarantees that v is black). We need to look at the neighbors of v and determine if there is exactly one white neighbor w. Since the neighbors of v are in the set $N(v)$ and currently the black vertices are those in $B_n(B)$, the set $N(v) \setminus B_n(B)$ contains all white neighbors of v. Then, to say that v has exaclty one white neighbor w means that $N(v) \setminus B_n(B) = \{w\}$. With this in mind we provide the following definition of the sequence $\{B_n(B)\}_{n\geq 0}$ and then define zero forcing.
Given a graph $G = (V, E)$ and a set $B \subseteq V$, we recursively define the sequence of sets $\{B_n(B)\}_{n\geq 0}$ using the following rules:

1. $B_0(B) = B$
2. $\forall n \geq 0$, $B_{n+1}(B) = B_n(B) \cup \{w \in V : \exists v \in B_n(B)$ such that $N(v) \setminus B_n(B) = \{w\}$

If there exists an integer m such that $B_m(B) = V$, we say that B is a *zero forcing set* of the graph G. Observe that for every $n \geq 0$, $B_{n-1}(B) \subseteq B_n(B) \subseteq V$, so it $B_m(B) = V$, then $B_n(B) = V$ for every $n \geq m$. A *minimum zero forcing set* is a zero forcing set that contains the least number of vertices, and its cardinality is $Z(G)$, the *zero forcing number* of the graph G.
The zero forcing game can also be described algorithmically:

2 Algorithm

Given a graph $G = (V, E)$ and a set $B \subseteq V$, we recursively define the sequence of sets $\{B_n(B)\}_{n\geq 0}$ using the following rules:

```
input: graph $G = (V, E)$, $B \subseteq V$ (the original set of black vertices);
output: winner or looser

1. while there is a vertex $v \in S$ such that $N(v) \cap B = \{w\}$, add $w$ to $B$
2. if $B = V$ then return winner else return looser
```

1 Background

Zero forcing is an interesting solitary game, because it can be mathematically proven that it does not matter how the game is played: the fact that someone wins or looses depends only on the set of vertices initially colored black. The goal of the zero forcing game is to find a minimum set of black vertices that guarantees that the player wins. Intuitively, the zero forcing number of a graph is the minimum number of vertices that must be initially colored in black to guarantee a win. Notice the term *minimum* for the initial number of black vertices; otherwise one can initially color all vertices in black.

2 Applications

Zero forcing appears in the study of networks, which are naturally modeled by graphs. The nodes of the network correspond to the vertices of the graph, and the links between nodes correspond to the edges of the graph. Now, suppose that a network is large, and you need to place sensors in the nodes to control the traffic through the network. This problem appears in many applications, from the study of spreading of a disease, to determining the number of errors that a code can recover, control of quantum physics and electrical engineering. We will look at one application, in electrical engineering, simply to understand better what we are trying to accomplish in the study of zero forcing.

Electrical power networks must be continuously monitored in order to prevent blackouts and power surges. This is accomplished by placing sensors at selected network locations. Ideally, a sensor would be placed at each network node, but those sensor are expensive so this solution is not feasible due to its cost. In practice, power networks are monitored by placing sensors at some locations, and using their readings to calculate the voltage at other network points. As a consequence, a major problem in electrical engineering consists of determining the power network location where sensors must be placed to monitor the entire network using the minimum number of sensors. This problem can be modeled through the use of zero forcing.

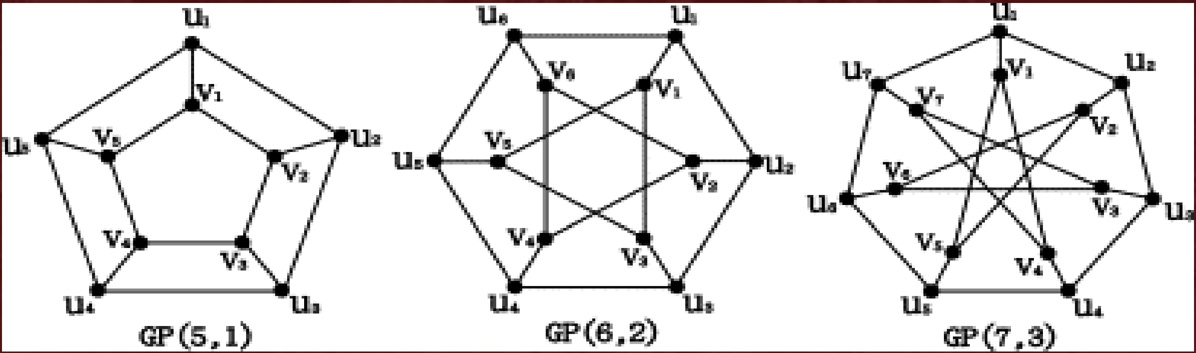
Generalized Petersen Graphs

Generalized Petersen Graphs (GP for short) are a specific family of simple, regular graphs that have the following properties: GP(n,k) has n “outer” vertices and n “inner” vertices creating a total of 2n vertices, the outer vertices we will label $\{u_0, u_1, u_2, \dots, u_n\}$ and the inner vertices we label $\{v_0, v_1, v_2, \dots, v_n\}$. There are three types of edges in GP(n, k):

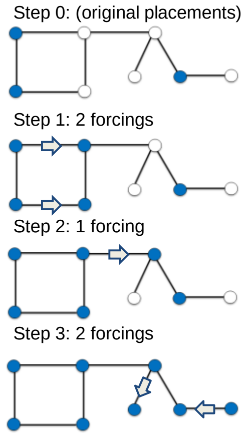
$$E\left(GP(n,k)\right) = \begin{Bmatrix} \{u_i, u_j\}\ i = 0, 1, \dots, n-1 \\ \{v_i, v_{(i+1) mod\ n}\}\ i = 0, 1, \dots, n-1 \\ \{u_i, u_{(i+k) mod\ n}\}\ i = 0, 1, \dots, n-1 \end{Bmatrix}$$

In summary, each “outer” vertex is connected to its 2 neighbors and its “inner” corresponding vertex while each “inner” vertex is connected to two other “inner” vertices determined by the number k in GP (n, k) and its corresponding “outer” vertex. As a result of this, in any GP (n, k) graph all vertices have degree 2 or 3. However, we consider GP(n,k) graphs with vertices of degree 2 to be trivial and thus, we will only be focusing on GP(n,k) graphs that are 3-regular.

Here are a few examples of Generalized Petersen Graphs:



Zero Forcing Visualized:



Some Regular Graphs: