Loop CCR Propagation time

Daniel Woldegiorgis *

May 10, 2020

Abstract

This project explores two loop color change rules: $CCR-Z_\ell$ and $CCR-\lfloor Z_\ell \rfloor$. The paper analyses the propagation times of different families of graphs for each respective color change rule. In addition to this, we also show results for the upper bound of the propagation time of $CCR-\lfloor Z_\ell \rfloor$ in terms of $CCR-Z_\ell$.

Keywords Propagation Time, Zero Forcing, Hopping, Loop CCR

1 Introduction

Define an (abstract) color change rule to be a set of conditions under which a vertex u can force a white vertex w to become blue in a graph whose vertices are colored white or blue.[2] Carlson generalizes the propagation time for a given abstract color change rule as follows: $\{pt_R(G;B) \mid$ B is a minimum R forcing set of G$\}$.[2]

We will first start off by analysing the $CCR-Z_\ell$ propagation time of several graphs, and then doing the same for $CCR-\lfloor Z_\ell \rfloor$. After this, we will look at a unifying result for the propagation times of these parameters. Finally, we will look at another student's proposal on studying propagation time for the hopping color change rule H.

2 Loop Color Change Rules

We will be discussing the propagation time and throttling numbers of two different color change rules that are defined as shown below. To make it easier to reference the different component rules of each color change rule, we clearly label (number) each one and will be referring to them as rule(1),(2), or (3) for the rest of this paper.

The color change rule $CCR-Z_\ell$ is defined as follows:

1. If u is blue and exactly one neighbor w of u is white, then change the color of w to blue.

*Dept. of Mathematics and Statistics, Williams College, Williamstown, MA, USA ({dww2}@williams.edu)

2. If w is white, w has a neighbor, and every neighbor of w is blue, then change the color of w to blue. [1]

For the remainder of this paper, let us for the sake of convenience refer to this color change rule as L.

The second color change rule CCR-$\lfloor Z_\ell \rfloor$ is defined as follows:

1. If u is blue and exactly one neighbor w of u is white, then change the color of w to blue.

2. If w is white, w has a neighbor, and every neighbor of w is blue, then change the color of w to blue.

3. If u is blue, all neighbors of u are blue, and u has not yet performed a force, then change the color of any white vertex w to blue, (this does not require that u have any neighbors).

Similarly, let us refer to the color change rule above as M for the rest of this paper.

The Color change rule for M can be thought of as L with hopping.

3 L Propagation Time

As in the proposal, I will first start by stating some general results about the Loop(L) propagation times of different families of graphs.

3.1 Complete Graph

Let $B \subseteq V(K_n)$ and $u, w \in V(K_n)$. Let u be a blue vertex and let w be a white vertex.

In order for u to be able to perform a force by rule(1), all but one of the neighbors of this blue vertex must be blue. ($|B| = n - 1$).

Similarly, in order for w to be able to force itself, all of its $n - 1$ neighbors must be blue. ($|B| = n - 1$).

Therefore, the minimum L forcing set of K_n has $n - 1$ vertices, and needs 1 timestep to propagate to the last remaining vertex. Therefore, $pt_L(K_n) = 1$.

3.2 Star Graph

For the star graph, the minimum L forcing set has size 1 simply containing the center vertex. This suffices because once the center is blue, then all of the leaf vertices can force themselves to become blue by rule(2) in 1 timestep. Therefore, $pt_L(K_{1,n-1}) = 1$.

3.3 Empty Graph

Let $B \subseteq V(\overline{K_n})$ and $u, w \in V(\overline{K_n})$. Let u be a blue vertex and let w be a white vertex.

a single vertex u cannot force any other vertex using rule(1) since all the vertices are isolated, and we need two vertices to be neighbors in order to perform a force using this rule. So, the zero forcing set according to this rule is $|B| = n$.

Again, since all the vertices are isolated, w cannot perform a force using rule(2) because none of the vertices have any neighbors. Therefore, this rule is invalid for empty graphs. ($|B| = n$)

Therefore, all of the vertices must initially be colored blue for us to obtain a zero forcing set. So, the minimum L forcing set of $\overline{K_n}$ has size n, and needs 0 timesteps to propagate. Therefore, $pt_L(\overline{K_n}) = 0$.

3.4 Wheel Graph

Let $t, u, v, w \in W_n$ where t, u and v are consecutive vertices (e.g. t is adjacent to u and u is adjacent to v) on the outer cycle of the wheel and w is the center vertex. Note that every vertex is connected to at least 3 other vertices. Therefore, we can note that the minimum L forcing set of W_n must have size ≥ 3, since for any of the vertices 1. to perform a force using rule(1) three vertices, including the center vertex, have to be connected to eachother OR 2. to perform a force using rule(2), all 3 of their neighbors have to be blue. Therefore, if we let $\{t, u, v\}$ be our initial L forcing set; then, since all of the neighbors of u are blue, it can force itself in the first timestep. Then, in the next timestep, t can force its white neighbor and v can force its white neighbor, and this propagates forward in each timestep where they force 2 new vertices coming around the outside cycle of the wheel graph in each timestep.

Therefore, we utilize 1 timestep for u to force itself, and all subsequent timesteps consume 2 new white vertices each time ($\lceil \frac{n-4}{2} \rceil$. (Here $n - 4$ is the remaining white vertices after coloring 3 of them blue and having u force itself). Therefore, $pt_L(W_n) = 1 + \lceil \frac{n-4}{2} \rceil$.

4 M Propagation Time

I will first start by stating some general results about the $CCR - \lfloor z \rfloor(M)$ propagation times of different families of graphs.

4.1 Complete Graph

Let $B \subseteq V(K_n)$ and $u, w \in V(K_n)$. Let u be a blue vertex and let w be a white vertex.

In order for u to be able to perform a force by rule(1) or a hop by rule(3), either all but one, or all of the neighbors respectively of this blue vertex must be blue. ($|B| = n - 1$).

Similarly, in order for w to be able to force itself by rule(2), all of its $n - 1$ neighbors must be blue. ($|B| = n - 1$).

Therefore, the minimum M forcing set of K_n has $n - 1$ vertices, and needs 1 timestep to propagate to the last remaining vertex. Therefore, $pt_M(K_n) = 1$.

4.2 Star Graph

For the star graph, the minimum M forcing set has size 1 simply containing the center vertex. This suffices because once the center is blue, then all of the leaf vertices can force themselves to become blue by rule(2) in 1 timestep. Therefore, $pt_M(K_{1,n-1}) = 1$.

4.3 Empty Graph

Let $B \subseteq V(\overline{K_n})$ and $u, w \in V(\overline{K_n})$. Let u be a blue vertex and let w be a white vertex.

a single vertex u cannot force any other vertex using rule(1) since all the vertices are isolated, and we need two vertices to be neighbors in order to perform a force using this rule. So, the zero forcing set according to this rule is $|B| = n$.

Again, since all the vertices are isolated, w cannot perform a force using rule(2) because none of the vertices have any neighbors. Therefore, this rule is invalid for empty graphs. ($|B| = n$)

However, if we use rule(3), given a single vertex u that is blue, we can have it force another vertex. After this, this new blue vertex then forces another white vertex, and so on until the whole graph is colored blue. Therefore, we have a zero forcing set B of size 1.

So, all in all, the minimum M forcing set of $\overline{K_n}$ has size 1, and needs $n-1$ timesteps to propagate. Therefore, $pt_M(\overline{K_n}) = n - 1$.

4.4 Wheel Graph

Let $t, u, v, w \in W_n$ where t, u and v are consecutive vertices (e.g. t is adjacent to u and u is adjacent to v) on the outer cycle of the wheel and w is the center vertex. Note that every vertex is connected to at least 3 other vertices. Therefore, we can note that the minimum M forcing set of W_n must have size ≥ 3, since for any of the vertices 1. to perform a force using rule(1) three vertices, including the center vertex, have to be connected to eachother OR 2. to perform a force using rule(2), all 3 of their neighbors have to be blue OR 3. to perform a hop from a vertex that has neighbors by rule(3), all of the 3 or more neighbors must be blue. Therefore, if we let $\{t, u, v\}$ be our initial M forcing set; then, since all of the neighbors of u are blue, it can force itself in the first timestep. Then, in the next timestep, t can force its white neighbor and v can force its white neighbor, and u can hop and force the farthest vertex from both t and v. In the next iteration, the vertices forced by t and v can now force using rule(1), AND vertices t and v can now perform hop and force the two vertices on either side of the vertex that was previously forced by a hop. And this process continues forward in each timestep where 2 forces are performed using rule(1) plus 2 forces are performed using rule(3). Therefore, 4 new vertices are consumed in each timestep.

Therefore, we utilize 1 timestep for u to force itself, and all subsequent timesteps consume 4 new white vertices each time ($\lceil \frac{n-3}{4} \rceil$). Therefore, $pt_M(W_n) = 1 + \lceil \frac{n-3}{4} \rceil$.

4.5 Related Results

A unifying theorem that relates the propagation times of L and M can be stated as follows.

Theorem 4.1. $pt_M(G) \leq pt_L(G)$

Proof. Let $B \subseteq V(G)$ be a minimum L forcing set of G such that $pt_L(G; B) = pt_L(G)$. Now, we know from the definition of each respective color change rule that every L forcing set is also an M forcing set. Therefore, B is also an M forcing set. So, this gives us the inequalities below:

$$pt_M(G) \leq pt_M(G; B) \leq pt_L(G; B) = pt_L(G).$$

Therefore, $pt_M(G) \leq pt_L(G)$. □

5 Concluding Remarks

We have looked at different families of graphs and how the two different color change rules affect the propagation time that we expect to get. A natural next step is to look into the throttling numbers of L and M, and to see if they have any relationships to other color change rules that we have seen in class or otherwise.

6 Hopping Propagation Time and Throttling

As in the proposal, I will first start by stating some general results about the Hopping propagation times of different families of graphs.

6.1 Complete Graphs

$pt_H(K_n) = 0$. Let $u \in V(K_n)$. Now, since u can only perform a hop if all neighbors of u are blue, then the minimum H forcing set of K_n has n vertices in it, and they color the graph in 0 timesteps. Therefore, $pt_H(K_n) = 0$.

6.2 Stars

$pt_H(K_{1,n-1}) = n - 2$. Let $u, v \in K_{1,n-1}$ where u is a leaf vertex and v is the center vertex. To start, we can note that the minimum H forcing set of $K_{1,n-1}$ must have size ≥ 2. This is because every vertex is connected to at least one other vertex and hence cannot perform a force with just one vertex colored. Now, let's show that the minimum H forcing set of a star has size 2.

For this, let's consider $\{u, v\}$ to be our initial H forcing set. Then, since all of u's neighbors are blue, then u can force any other leaf in the star. This vertex can then force any other vertex since it's only neighbor(v) is also blue. This process continues until the whole graph is colored blue. Therefore, with an initial minimum H forcing set with 2 vertices, it will take $n - 2$ timesteps for the entire graph to turn blue. So, $pt(G; B) = n - 2$.

6.3 Empty Graphs

$pt_H(\overline{K_n}) = n - 1$. Once again, let $u, v \in V(\overline{K_n})$. Now, if we color one of these vertices u in the graph, then we can perform a hop to one other vertex v, which can then hop to some other vertex, and so on until the entire graph is blue. Therefore, the minimum H forcing set of $\overline{K_n}$ has size 1, and needs $n - 1$ timesteps to propagate. Therefore, $pt_H(\overline{K_n}) = n - 1$.

6.4 Wheels

$pt_H(W_n) = n - 4$. Let $t, u, v, w \in W_n$ where t, u and v are consecutive vertices (e.g. t is adjacent to u and u is adjacent to v) on the outer cycle of the wheel and w is the center vertex. Note that every vertex is connected to at least 3 other vertices. Therefore, we can note that the minimum H forcing set of W_n must have size > 3, since for any of them to perform a hop, all 3 of their neighbors including themselves have to be blue. Now, let's show that the result above holds true.

For this, let $\{t, u, v, w\}$ be our initial H forcing set. Then, since all of the neighbors of u are blue, then it can hop and force a vertex. However, it must force either the neighbor of t or the neighbor of v for the hop to be able to continue further. In the next iteration, v can now force either the neighbor of t or the new vertex that was forced in the first iteration. As we continue in this way, there will always be one vertex that has not yet performed a hop and has all of it's neighbors being blue. Therefore, the hopping will continue one vertex at a time until the rest of the graph is forced. Since, we started with an intial set of 4 vertices, then it will take $n - 4$ hops to force everything to be blue. Therefore, $pt_H(W_n) = n - 4$.

6.5 Other results

One of the conjectures that was stated by my colleague was that $pt_{\lfloor z \rfloor}(G) \leq pt_H(G)$. I have outlined a proof of this result below.

Theorem 6.1. $pt_{\lfloor z \rfloor}(G) \leq pt_H(G)$

Proof. Let $B \subseteq V(G)$ be a minimum H forcing set of G such that $pt_H(G; B) = pt_H(G)$. Now, we know from the definition of each respective color change rule that every H forcing set is also a $\lfloor z \rfloor$ forcing set. Therefore, B is also a $\lfloor z \rfloor$ forcing set. So, this gives us the inequalities below:

$$pt_{\lfloor z \rfloor}(G) \leq pt_{\lfloor z \rfloor}(G; B) \leq pt_H(G; B) = pt_H(G).$$

Therefore, $pt_{\lfloor z \rfloor}(G) \leq pt_H(G)$ □

References

[1] F. Barioli, W. Barrett, S. Fallat, H. T. Hall, L. Hogben, B. Shader, P. van den Driessche and H. van der Holst, Parameters related to tree-width, zero forcing, and maximum nullity of a graph, J. Graph Theory 72 (2013), 146–177.

[2] J. Carlson. Throttling for zero forcing and variants. Australian Journal of Combinatorics,Volume 75(1) (2019), Pages 96–112