NON-RESILIENCE OF RESILIENT DISTRIBUTED
CONSENSUS IN MULTI-AGENT SYSTEMS

by

Leon Khalyavin

APPROVED BY SUPERVISORY COMMITTEE:

___________________________
Waseem Abbas, Chair

___________________________
Stephen Yurkovich

___________________________
Justin Ruths
This thesis is dedicated to my family and friends that supported me through this journey.
NON-RESILIENCE OF RESILIENT DISTRIBUTED
CONSENSUS IN MULTI-AGENT SYSTEMS

by

LEON KHALYAVIN, BSEE, MSEE

THESIS
Presented to the Faculty of
The University of Texas at Dallas
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN
SYSTEMS ENGINEERING AND MANAGEMENT

THE UNIVERSITY OF TEXAS AT DALLAS
August 2023
ACKNOWLEDGMENTS

I would firstly like to thank my advisor, Dr. Waseem Abbas. His enthusiasm about the topic of distributed control systems has really sparked an interest in me of pursuing this field. In addition, his clear explanations and his effort to push me to learn as much as possible has really helped me grow as a researcher and appreciate the field of academia a lot more. Most importantly though, his kindness and dedication to be a great teacher has really made me feel passion for researching, and his guidance has helped me go through my early academic journey.

In addition, I would like to thank the Systems Engineering Department at UT Dallas, and I would like to acknowledge the support of Amazon Robotics, who through a generous gift to the department of Systems Engineering made much of this work possible. After being lost about the direction I wanted to pursue with my career, the department has granted me a lot of support with helping me understand more about myself and the direction I want to take my career. I especially want to acknowledge and thank Dr. Stephen Yurkovich. He is one of the first professor who introduced me to the entire field of control systems and connected me with Dr. Waseem Abbas to find this field of distributed systems. In addition, he has instilled confidence in me in pursuing this career and has helped me navigate the difficult aspects of academia. I would like to also thank Dr. Justin Ruths for introducing me to the topic of network systems in creating one of the most engaging courses I have taken.

I would also like to thank my advisor, Brenda Rains, from the department. She has really helped me realize and hone in on what I am interested in and what I should be doing in my career. The goal of any advisor is to make as much impact and support as she has provided not only me, but the rest of the students in the Systems Engineering Department.

I want to thank my close family and close friends. From my family, my mom has made some of the biggest impact on me. She taught me to love learning and has helped me get through
some of my most stressful times. I would also like to thank my dad for supporting my decisions that I have been making throughout my career. I would also like to give a special thank to my aunt and uncle who have been supportive throughout my academic journey so far.

Finally, my friends have also made a huge impact on getting me through university. I especially want to thank two of my closest friends, Savi and Rooshi, for getting me to do my work and getting me through my most stressful times. I could not have asked for a better group to be around, but they already know.

July 2023
This thesis explores the resilient distributed consensus in networks that lack the necessary structural robustness to achieve consensus in the presence of malicious agents. While existing solutions provide robustness conditions for consensus among normal agents, they fail to evaluate network performance comprehensively when the graph’s robustness is insufficient. To address this limitation, we introduce the concept of non-convergent nodes, representing agents unable to attain consensus with any arbitrary agent due to malicious agents in the network. This notion allows us to classify graphs based on their robustness levels and assess partial performance. This study initially establishes the \((r, s)\)-robustness of commonly encountered graphs, such as complete, complete bipartite, 1-D distance, and circulant graphs. Our approach facilitates easier identification of robustness and enables us to gain insights into the behavior of non-convergent nodes. By understanding the dynamics of these non-convergent nodes, we can establish more relaxed conditions for converging subgraphs, which are the subgraphs that are guaranteed to converge. This knowledge enhances our understanding of resilient algorithms and their behavior in practical scenarios. Furthermore, we present graphs with given robustness levels, including \((F + 1, 1)\), \((F, F)\), and \((F + 1, F)\) robustness, and determine the maximal number of non-convergent nodes associated with each graph. This quantification of non-resilience sheds light on the impact of graph robustness.
on the network’s ability to achieve consensus. Surprisingly, we find that graphs with the same structural robustness may exhibit varying degrees of non-resilience, leading to different network performance outcomes. Through numerical evaluations, we demonstrate that our approach provides a comprehensive resilience perspective beyond the conventional binary view of success or failure in the face of malicious agents. By quantifying network performance under sub-optimal robustness conditions and identifying converging subgraphs, our study opens up new possibilities for designing more resilient consensus algorithms.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>xi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>xii</td>
</tr>
<tr>
<td>CHAPTER 1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Motivation and Focus</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Related Work</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Contributions and Organization</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 2</td>
<td>RESILIENT CONSENSUS</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Resilient Distributed Consensus</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>W-MSR Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>$(r, s)$-Robustness</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Robustness of Common Graphs</td>
<td>13</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Complete Graph</td>
<td>13</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Complete Bipartite Graph</td>
<td>14</td>
</tr>
<tr>
<td>2.4.3</td>
<td>1-D Distance Graph</td>
<td>16</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Circulant Graph</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>Limitations of the W-MSR Algorithm</td>
<td>21</td>
</tr>
<tr>
<td>CHAPTER 3</td>
<td>QUANTIFYING NON-RESILIENCE</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>Notion of Non-Convergence</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>Resilience Lower Bound for Graphs</td>
<td>24</td>
</tr>
<tr>
<td>3.2.1</td>
<td>$(F + 1, 1)$-Robust Case</td>
<td>26</td>
</tr>
<tr>
<td>3.2.2</td>
<td>$(F, F)$-Robust Case</td>
<td>29</td>
</tr>
<tr>
<td>3.2.3</td>
<td>$(F + 1, F)$-Robust Case</td>
<td>35</td>
</tr>
<tr>
<td>3.3</td>
<td>Illustrations and Simulations</td>
<td>39</td>
</tr>
<tr>
<td>CHAPTER 4</td>
<td>DISCUSSION AND CONCLUSION</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Expanding on Non-Convergence</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Subgraph Convergence</td>
<td>50</td>
</tr>
</tbody>
</table>
4.3 Conclusion ................................................................. 55
REFERENCES .................................................................. 58
BIOGRAPHICAL SKETCH .................................................. 62
CURRICULUM VITAE
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>$G$ is robust enough against a single malicious agent, whereas $G_1, G_2, G_3, G_4$ are not. By measuring the number of non-convergent nodes (red), we characterize the non-resilience in graphs whose robustness is below the required threshold.</td>
</tr>
<tr>
<td>2.1</td>
<td>Three conditions for $(r, s)$-robustness. Blue nodes have at least $r$ neighbors outside of their subset. (a) $</td>
</tr>
<tr>
<td>2.2</td>
<td>Eight nodes, four from $S_1$ (red) and four from $S_2$ (blue). By rotating the partition, a partition exists where $S_1$ and $S_2$ are evenly divided.</td>
</tr>
<tr>
<td>3.1</td>
<td>The number of non-convergent nodes (highlighted red) in (a), (b), and (c) are 6, 3, and 1, respectively.</td>
</tr>
<tr>
<td>3.2</td>
<td>$G = E_2 \oplus C_7^{1,2}$ is $(4, 1)$-robust.</td>
</tr>
<tr>
<td>3.3</td>
<td>(a) A set $S$ of four nodes contains two nodes (red), each of which has two neighbors outside of $S$. (b) A set $S$ contains five nodes, of which four nodes (red) have at least one neighbor outside of $S$.</td>
</tr>
<tr>
<td>3.4</td>
<td>$G = K_7 \oplus C_7$ is $(F, F)$-robust graph for $F = 5$.</td>
</tr>
<tr>
<td>3.5</td>
<td>$G = E_6 \oplus C_6^{1,2}$ is $(F + 1, F)$-robust graph for $F = 3$.</td>
</tr>
<tr>
<td>3.6</td>
<td>$G$ is $(3, 3)$-robust and $v_6$ is non-convergent for $F = 3$.</td>
</tr>
<tr>
<td>3.7</td>
<td>$v_7$ (green) is a non-convergent node. For every node $v_i \neq v_7$, there is an attack consisting of $F = 3$ nodes (as in (c) and (d)) preventing $v_7$ and $v_i$ from converging to a common point.</td>
</tr>
<tr>
<td>3.8</td>
<td>Non-convergent nodes (colored red) in $(3, 3)$-robust, $(4, 1)$-robust, and $(4, 3)$-robust graphs.</td>
</tr>
<tr>
<td>3.9</td>
<td>(a) $(2, 2)$-robust graph. (b) Number of non-convergent nodes increases with $F$.</td>
</tr>
<tr>
<td>4.1</td>
<td>Graph, $G$, which is not $(2, 2)$-robust, with $F = 1$ adversaries, but subgraph ${u_1, \cdots, u_4}$ which is subgraph $(2, 2)$-robust with adversary at node 6. (a) depicts the graph and (b) plots the convergence in time.</td>
</tr>
</tbody>
</table>
LIST OF TABLES

3.1 The number of non-convergent nodes in graphs with various robustness considering $F = 3$ malicious nodes. ........................................... 44
1.1 Motivation and Focus

In distributed systems, the presence of just a few malicious nodes can severely disrupt the operation of the system. Attacks on the system can severely disrupt the system’s performance, whether the attacks are due to intelligent adversaries or faulty nodes. Distributed systems have vast applications ranging from multi-agent robotics to data processing, and these applications require robust systems so that they are able to safely and accurately perform. For example, consider the distributed consensus in multi-agent systems, a canonical problem with several applications across various domains, including networked control systems, multi-robot systems, and sensor networks. The primary goal here is to ensure that all agents update their local states in a way that eventually converges to a common state. A simple Linear Consensus Protocol (LCP), where each agent updates its state by averaging its neighbors’ states, solves the problem (e.g., [Jadbabaie et al., 2003; Olfati-Saber et al., 2007; Ren et al., 2007]). However, with just a single malicious agent that does not adhere to the LCP can prevent the other agents, and therefore, the entire graph from achieving consensus e.g., [LeBlanc et al., 2013]. The objective of resilient algorithms are such that the network systems are able to withstand a certain number of malicious agents and guarantee desired performance.

In the current state, research in resilient distributed algorithms have provided conditions in order for the network system to perform under the desired behavior. These conditions are often quite strict, such that it can be impractical or difficult to implement such systems into the real world. The theories also state that if a system satisfies the robustness conditions, then it will be able to converge, but they often do not explore what occurs then those conditions are not met.
Our goal is to better understand resilience and the dynamics of when such conditions are not satisfied. Figure 1.1 demonstrates how currently, by not having the proper robustness, we cannot understand anything about the dynamics of the system. We will dive deep into the notion of non-resilience and be able to quantify how much and what parts of the graph are malicious. We not only explore this new notion of non-convergence, but we further explore resilience on a systemic level, including finding the resilience of common graphs and analyzing how the notion of non-convergence can be applied to better understand new resilient sub-structures in graphs.

Figure 1.1: $G$ is robust enough against a single malicious agent, whereas $G_1, G_2, G_3, G_4$ are not. By measuring the number of non-convergent nodes (red), we characterize the non-resilience in graphs whose robustness is below the required threshold.

We provide a better understanding and more holistic view of resilient algorithm’s performance and the dynamics of the graph. Initially, we introduce this idea of resilience and
the strict conditions it imposes on graphical structure. Because resilience can be difficult
to find, we provide a more practical usage of resilience, we provide the resilience of some
common graphs. Assume there is a graph that is very robust, however, just a small portion
of the graph does not satisfy the robustness conditions. By exploring what happens when
those conditions are not met, we can understand what, if anything, can be preserved. This
lays out the main question that we try to address:

*How can we evaluate the performance of a network that fails to meet the robustness
threshold for guaranteeing consensus when facing $F$ malicious agents?*

We introduce the notion of *non-convergent*, which refers to an agent that can fail to
reach consensus with any arbitrary normal agent in the system. By using this new notion
of *non-convergence*, we demonstrate how we can still preserve portions of 'non-resilient'
graphs, trends of non-convergence with graph robustness, and analysis on how the notion
of non-convergence can be applied to further the research into the study of 'non-resilient'
graphs.

1.2 Related Work

Distributed systems rely on agents updating their values by utilizing their own information,
information from neighbors, and an update protocol. In *resilient* distributed systems, the
goal is to design strategies that filter out the information from malicious nodes and utilize
the information from other normal nodes. Resilience is enhanced by making sure the agents
follow a proper algorithm, such that it filters the bad values, and the graph has a proper
structure, ensuring that each good agent is able to communicate with other good agents.
Based on these principles, various resilient distributed strategies and algorithms have been
proposed to tackle distributed optimization problems like consensus ([LeBlanc et al. 2013];
[Dibaji and Ishii 2015]; [Senejohnny et al. 2019]; [Usevitch and Panagou 2020]; [Wang et al.
2022]; [Ishii et al. 2022]; [Yan et al. 2022]; [Ramos et al. 2022]; [Abbas et al. 2022]), diffusion
Li et al., 2019; Yu et al., 2022; Safi et al., 2022), estimation (Mitra et al., 2019; Mitra and Sundaram, 2019; An and Yang, 2021; Chen et al., 2018), learning and optimization (Li et al., 2020; Yang and Bajwa, 2019; Yang et al., 2020; Mitra et al., 2020; Sundaram and Gharesifard, 2018; Su and Vaidya, 2020; Zhao et al., 2019). In particular, the Weighted-Mean-Subsequence-Reduced (W-MSR) algorithm, presented in (LeBlanc et al., 2013), stands out as a widely used resilient distributed consensus approach. The algorithm works by having each agent ignore the extreme values when updating their own value with the update protocol. This leverages the graphical structure, or the \((r,s)\)-robustness, to guarantee consensus of agents implementing the W-MSR algorithm in the face of \(F\) malicious agents (LeBlanc et al., 2013; Usevitch and Panagou, 2020; Saldana et al., 2017; Pirani et al., 2022; Ishii et al., 2022; Renganathan et al., 2021; Dibaji and Ishii, 2017; Sundaram and Gharesifard, 2018; Rezaee et al., 2021; Abbas et al., 2018; Chen et al., 2018; Shang, 2018; Saulnier et al., 2017; Wen et al., 2023; Wu et al., 2021; Lu and Jia, 2023).

Current research focuses on the performance of when the robustness conditions have been met. For example, the W-MSR algorithm guarantees consensus among non-adversarial agents if the number of adversaries is bounded by \(F\) and the network meets the required robustness conditions, \((F + 1, F + 1)\)-robust. Evaluating partial performance is crucial in understanding how the system behaves and being able to implement such system, as the number of adversarial nodes can easily exceed \(F\). Currently, it is not possible to determine the number of agents that can or cannot achieve consensus when the graph is less robust than required. The current all or nothing approach needs to be more comprehensive to understand the system better for theoretical and practical applications. Strict robustness conditions can become impractical (Abbas et al., 2018; Dibaji et al., 2019; Zhang et al., 2015), especially when designing systems with an unknown \(F\) or when the number of malicious agents can easily exceed \(F\).

We address these shortcomings in the current state of research by laying out a new, and more continuous way, of analyzing resilience in a graph. This allows us to find a
more accurate way of ordering networks based on their resilience, rather than treating them equally. This insight can allow us to better understand what happens when these robustness conditions are not met, and what trade-offs we can make to create networks that better fit our needs.

1.3 Contributions and Organization

Some of the major contributions of this thesis include:

1. We formulate or improve on a generalized $(r,s)$-robustness for common graphs used. These graphs include complete graph, complete bipartite graph, 1-D distance graph, and circulant graph. By identifying the generalized $(r,s)$-robustness for these graphs, we can develop quicker ways of identifying robustness of these graphs given that we can identify the graph structure faster than the co-NP-complete difficulty of calculating robustness.

2. We propose a new notion of ‘non-convergence’ to characterize how non-resilient nodes are in a graph. A non-convergent node refers to a normal node where attacks exist that prevent the node from converging to any other normal node in the graph. By using this new metric, we are able to compare graphs that fail to meet the required robustness for resilient consensus with $F$ malicious agents and evaluate the partial performance of such graphs.

3. For a given graph robustness, we construct graphs that maximize the number of non-convergent nodes. This creates a lower bound on the resilience, or upper bound on a new notion of ‘local non-resilience’. This analysis reveals the extent that at which adversaries can disrupt graphs that do not meet the required $(F+1, F+1)$-metric. We construct cases for the $(F+1, 1),(F, F)$, and $(F+1, F)$-robust cases using a combination of circulant, empty, and complete graphs and their joins. This quantifies the worst
case that graphs can experience when in the presence of more adversaries than initially designed for.

4. We highlight the correlation between non-convergent nodes and the graphical robustness. We then provide a detailed numerical evaluation of our approach, which illustrates our results. Additionally, we provide new potential research directions for the usage of non-convergent nodes to evaluate the graph robustness.

5. Using the notion of non-convergent nodes, we propose a new idea of creating ‘converging subgraphs’. We list out a potential direction for creating such converging subgraphs and the importance of understanding the dynamics of non-convergent nodes to design better versions of converging subgraphs.

This thesis has also resulted in a journal article, *On the Non-resiliency of Resilient Distributed Consensus in Multiagent Networks* by Leon Khalyavin and Waseem Abbas. This paper has been submitted to IEEE Transactions on Control of Network Systems and is currently submitted to be reviewed.

The rest of the thesis is organized as follows. Chapter 2 involves around understanding resilient consensus and the problem formulation. In Section 2.4, we provide the robustness of common graphs and their respective proofs for complete, complete bipartite, 1-D distance, and circulant graphs. Chapter 3 introduces this new notion of non-convergence, which is explained in detail in Section 3.1. In Section 3.2, we provide the lower bound on resilience for 
\((F + 1, 1), (F, F), \text{ and } (F + 1, F)\)-robust graphs. Section 3.3 provides illustrations of how non-convergence varies within robustness and trends within non-convergence. Finally, chapter 4 concludes the thesis by discussing possible research directions. Section 4.1 provides the benefits and short-comings of the current definition of non-convergence and ways that it can be improved upon while Section 4.2 describes how we can use the notion of non-convergence to understand the convergence of subgraphs, where the original graph does not satisfy the 
\((F + 1, F + 1)\)-robustness conditions. Section 4.3 concludes the thesis.
CHAPTER 2
RESILIENT CONSENSUS

2.1 Resilient Distributed Consensus

We model multi-agent systems using a graph or network, $G = (V, E)$. The graph consists of nodes, $V$, which contain some information about its local properties and edges, $E$, which is the way that the nodes are connected. We denote a neighborhood of a node, $N$, which is the set of nodes that shared edges with that node. In a multi-agent system, every node has a value and uses an update model based on the information from itself and its neighbors. We will be focusing on systems with an undirected network, where information can be transferred both ways across an edge. Additionally, we will be focusing on discrete time systems.

In distributed consensus, the goal is to have all of the nodes in the network converge to the same value at steady state. The most common approach to achieve distributed consensus is for a node, $u$, to add the difference between its value and the value of all of its neighbor nodes, $v \in N_u$, to understand which way to change its value. This is modeled by equation (2.1) where $x(k)$ is the state of a node at time $k$ and $w$ is the edge weight between the nodes, which gets each node to essentially calculate the weighted average of its neighborhood.

$$x_u(k + 1) = \sum_{v \in N_u} w_{u,v}(x_u(k) - x_v(k))$$  \hspace{1cm} (2.1)

However, in resilient distributed consensus, we introduce adversarial nodes. As seen in definition (2.1), adversarial nodes are nodes in the network that can have a different update model and can relay different information to its neighbors. This can model different events such as node failure, miscalculation, or even malicious attacks on a network.

Definition 2.1. (Adversarial Node) A node that can follow a different update model and relay incorrect information to its neighbors.
There are two main types of models for adversarial nodes. A malicious model assumes that the malicious node can relay any information to its neighbors, however, the information must be the same across all nodes. A byzantine model assumes that the adversarial node can relay any information to its neighbors and the information given to each neighbor can be different. It is not difficult to see that a byzantine model is more dangerous and more difficult to deal with than an adversarial model. We will be focusing on malicious models in this thesis. We will also denote the number of malicious nodes as $F$.

Malicious nodes can model intelligent attacks on a system level. If someone was trying to disrupt a multi-agent system, they will utilize a set of nodes to prevent the convergence of the network. In addition, multi-agent nodes can model the effect of node failure. For example, in a multi-agent robotic system, if one of the robots fail to move and change their position, they will relay their non-updated position to the remainder of the robots. However, we do not want the rest of the robots to utilize that information, so we design intelligent ways to still perform our main tasks in the presence of the 'bad' or 'broken' nodes.

On a system level, there are two main types of models used. An $F$-total model assumes that the maximum number of adversaries in a network is $F$. On the other hand, an $F$-local model assumes that the maximum number of neighbors of a node that can be adversaries if $F$. We will be focusing on an $F$-total adversarial model. We will be focusing on an $F$-total model in this thesis.

Because malicious nodes can change their values with their own update model, they can potentially move in any way they please. Therefore, when analyzing the consensus problem, it might not be possible for every node to converge at the same value at steady state, because the malicious nodes can contain another value. We will then only be analyzing the steady state location of the normal nodes, which are all the nodes who are not malicious. The malicious nodes can influence the value at which the normal nodes will converge. Because we do not want the system to be influenced too much by the malicious nodes, we will bound
the target consensus value. This target will be within the largest and smallest value of the initial values of the nodes.

**Definition 2.2.** *(Resilient Distributed Consensus)* A network of agents \( G = (V, E) \) achieves consensus if the following conditions are satisfied:

1. *(Safety)* Let \( x_{\text{min}}(0) \) and \( x_{\text{max}}(0) \) denote the minimum and the maximum of the initial states of nodes in \( G \), respectively. Then, \( x_{\text{min}}(0) \leq x_u(k) \leq x_{\text{max}}(0), \ \forall u \in V \), and for all times \( k \).

2. *(Agreement)* As \( k \to \infty \), \( x_u(k) = x_v(k) = x \) for all pairs of nodes \( u, v \in V \).

Definition 2.2 provides a new definition for resilient distributed consensus. The first condition, safety, ensures that the adversarial nodes do not deviate the final values of the normal nodes. We can change the tolerance on where we can allow the final position for the normal nodes, but we focus on when all the nodes to be within the minimum and maximum of the initial positions. For the second condition, agreement, ensures that all of the normal nodes end up with the same values. Again, this can change with some tolerance, but we want to guarantee that the nodes end up in the same value at steady state.

### 2.2 W-MSR Algorithm

The *Weighted-Mean-Subsequence-Reduced* *(W-MSR)* algorithm is a solution to the resilient distributed consensus problem that has gained popularity. The algorithm works by removing the largest and smallest values gained from neighbors to use in the update model. The idea is that if the largest and smallest values are adversarial nodes, they will not converge to the rest of the normal nodes. By removing them from the calculations, it will reduce their impact on the remainder of the normal nodes. The process for implementing the W-MSR algorithm is as follows:
(1) Every normal node, $u$, collects state values from its neighbors at every time step $k$ and sorts them from smallest to largest.

(2) It then removes $F$ of the largest values strictly greater than $x_u(k)$ and $F$ of the smallest values strictly less than $x_u(k)$. If the number of values strictly greater than $x_u(k)$ is less than $F$, then it will remove all of the values strictly greater than $x_u(k)$. Similarly, if the number of values strictly less than $x_u(k)$ is less than $F$, then it will remove all of the values strictly less than $x_u(k)$. The set of nodes in $N_u(k)$ that are removed by this process are denoted by $R_u(k)$.

(3) The node $u$ updates its value according to the following equation 2.2:

$$x_u(k + 1) = \sum_{v \in (N_u \cup U(u)) \setminus R_u(k)} w_{uv} x_v(k)$$

One of the major benefits of using the W-MSR algorithm is that it easily satisfies the safety goal of the resilient distributed consensus problem, $x_{min}(0) \leq x_u(k) \leq x_{max}(0)$. This is because every node will be moving towards the mean of the neighbors. If a node has a maximum or a minimum value, and there exists $F$ total adversaries, then it will not be possible for the adversaries to move that node higher if it is a maximum or lower if it is a minimum. This is because that node removes $F$ extreme adversaries, and therefore, will not move in that direction. Additionally, the W-MSR algorithm allows a guarantee for the normal nodes to converge based on the graphs robustness metric, $(r, s)$-robustness and state relevant conditions on the graph.

2.3 $(r, s)$-Robustness

$(r, s)$-robustness is defined by a set of related notions below.

**Definition 2.3.** $(r$-Reachable Node) Given a graph, $G = (V, E)$, a subset of nodes, $S \subset V$, and a positive integer $r$, a node, $u \in S$, is $r$-reachable if the number of neighbors outside of $S$ is at least $r$, or $|N_u \setminus S| \geq r$. 

10
Definition 2.4. *(r-Reachable Set)* Given a graph, \( G = (V, E) \), a subset of nodes, \( S \subset V \), and a positive integer \( r \), a set of nodes are \( r \)-reachable, \( X^r_S \), if at least one node is \( r \)-reachable in \( S \). In other words:

\[
X^r_S = \{ u \in S : |N_u \setminus S| \geq r \}
\]  
(2.3)

Now, using these notions, we can define \((r, s)\)-robustness.

Definition 2.5. *(\((r, s)\)-Robustness)* \( \text{[LeBlanc et al., 2013]} \) Given a graph \( G = (V, E) \) and positive integers \( r \) and \( s \), \( G \) is considered to be \((r, s)\)-robust if for every pair of disjoint \( S_1, S_2 \subset V \), one of the following holds true:

(i) \( |X^r_{S_1}| = |S_1| \) (i.e., each node in \( S_1 \) have at least \( r \) nodes outside of \( S_1 \)).

(ii) \( |X^r_{S_2}| = |S_2| \) (i.e., each node in \( S_2 \) have at least \( r \) nodes outside of \( S_2 \)).

(iii) \( |X^r_{S_1}| + |X^r_{S_2}| \geq s \) (i.e., the number of nodes in \( S_1 \) having \( r \) nodes outside of \( S_1 \) and \( S_2 \) having \( r \) nodes outside of \( S_2 \) is at least \( s \)).

Figure 2.1 visually shows the three possible cases for a graph to be \((r, s)\)-robust. LeBlanc \( \text{[LeBlanc et al., 2013]} \) provides robustness conditions for convergence under total/local and malicious/byzantine models. The necessary and sufficient condition for convergence of an \( F \)-total malicious model is stated below in Theorem 2.1.

**Theorem 2.1.** Consider a graph, \( G = (V, E) \), with \( |V| = N \) nodes, of which at most \( F \) can be malicious (i.e., \( F \)-total malicious model). If each normal node is implementing the \( W \)-MSR algorithm with parameter \( F \) (remove \( F \) largest and smallest values larger or smaller than current value), then the distributed consensus is achieved if and only if \( G \) is \((F + 1, F + 1)\)-robust.
The W-MSR algorithm guarantees that at each time step, \( k \), there will always be a normal node that will change its position if the graph is \((F + 1, F + 1)\)-robust. For a node to change its value under the W-MSR algorithm, it requires at least \( F + 1 \) nodes with values larger or smaller than its own. If the graph has not converged, then the graph can be separated into a subset of maximum values and minimum values. Since there will exist \( F + 1 \) nodes that are \((F + 1)\)-reachable, then at least one of those nodes has to be a normal node that will update its value. It will move in the direction towards the mean. If that does not exist for all normal nodes, then assume that all extreme values are adversaries. As long as if the edge exists and the edge weight is greater than 0, then if the graph is \((F + 1, F + 1)\)-robust, it will be guaranteed to converge.

However, it is very difficult to determine if a graph is \((r, s)\)-robust. Determining the \((r, s)\)-robustness of a network is an co-NP-complete problem (Zhang et al., 2015) and is currently calculated using a greedy approach. Additionally, the conditions for \((r, s)\)-robustness are stringent. Currently, if the conditions are not satisfied, then we do not know what can be preserved in the network. We will focus on networks that have not achieved \((F + 1, F + 1)\)-robustness, how does the graph behave.
2.4 Robustness of Common Graphs

Since computing \((r, s)\)-robustness is challenging (Zhang et al., 2015), we explore graph structure to determine its robustness. One approach is to find the robustness of a general graph, and then use other algorithms to find the graph type and determine the robustness from that. The benefit of this approach is that there are sophisticated algorithms to determine specific graph types and such graphs can commonly appear in given applications. The four graph types we explored are: complete graphs, complete bipartite graphs, 1-D distance graphs, and circulant graphs.

2.4.1 Complete Graph

Finding the resiliency of a complete graph is of paramount importance due to its fundamental role in understanding and assessing the robustness of complex networks. Complete graphs represent an extreme scenario where each node is directly connected to every other node, making them a relevant model for certain real-world systems and communication networks. Determining the resiliency of such graphs provides critical insights into their ability to maintain connectivity and functionality under various adverse conditions, such as node failures, link disruptions, or targeted attacks. This analysis helps researchers, engineers, and policymakers design more reliable and fault-tolerant network infrastructures, enabling them to anticipate potential vulnerabilities and implement appropriate mitigation strategies to ensure uninterrupted and efficient communication within the network.

Arguably, the most intuitive graph to analyze is a complete graph. A complete graph, \(K_N\), is a graph with \(N\) nodes, where every node has an edge with every other node in the graph. Under distributed consensus, complete graphs are the least controllable graphs. Therefore, it should be difficult for the adversaries to control the other normal nodes in the network. By detecting a complete graph, we can then easily determine the resilience of the graph.
Definition 2.6. **(Complete Graph)** A complete graph, $K_N$, is a graph with $N$ nodes such that every node $u$ is adjacent with every other node $v \subset V \setminus u$ in the graph.

Theorem 2.2. A complete graph, $G = K_N$, with $N$ number of nodes is $(r, r)$-robust given that $N \geq 2r - 1$.

Proof: In a complete graph, every node has an edge with every other node in the graph. Consider two disjoint non-empty subsets of nodes, $S_1$ and $S_2$. For a given subset, say $S_1$, every node in $S_1$ will have $N - |S_1|$ edges outside of $S_1$. If $N - |S_1| \geq r$, then all of the nodes in $S_1$ must have more than $r$ edges outside of $S_1$ and $|X_{S_1}^r| = |S_1|$ condition is satisfied. If $N - |S_1| < r$, then none of the nodes in $S_1$ will have at least $r$ neighbors outside of $S_1$. In this case, $|S_1| > N - r$. By the given assumption, $N \geq 2r - 1$, which means $N - r \geq r - 1$. As a result, $|S_1| > N - r \geq r - 1$, or $|S_1| \geq r$. Since each node in $S_2$ is adjacent to every node in $S_1$, it implies that each node in $S_2$ is adjacent to at least $r$ nodes outside of $S_2$, i.e., $|X_{S_2}^r| = |S_2|$, implying that $K_N$ is $(r, r)$-robust. \qed

### 2.4.2 Complete Bipartite Graph

We will next analyze the complete bipartite graph, a type of graph often seen in neural networks and data science applications, as well as comparisons between groups. These graphs serve as essential models for diverse applications, including social networks, supply chain management, and matching algorithms in various industries. Assessing the resilience of such graphs allows researchers and practitioners to evaluate the network’s ability to withstand failures or attacks on specific nodes or connections. This analysis aids in identifying critical points of vulnerability and devising effective strategies to enhance the overall reliability and performance of the network. Understanding the resilience of complete bipartite graphs is, therefore, vital for designing robust and fault-tolerant systems that can adapt and endure in dynamic and challenging environments.
A complete bipartite graph, denoted by $K_{m,n}$, consists of $m+n$ nodes partitioned into two sets, say $X$ and $Y$, with sizes $m$ and $n$, respectively. All nodes in set $X$ are adjacent to every node in set $Y$, but nodes within the same set are not adjacent to each other. These graphs present compelling use cases for resilient algorithms due to the prevalence of anomalies in data-based applications. By utilizing resilient algorithms, we are better equipped to improve the robustness and reliability of such networks.

**Theorem 2.3.** A complete bipartite graph, $G = K_{m,n}$, with $N = m + n$ number of nodes is $(r,s)$-robust if $m, n \geq 2r - 1$ and $s \leq r$.

**Proof:** In a complete bipartite graph, every node in one subset, say $U$, only has edges with every node in the other subset, say $V$. Suppose that $U$ has $m$ nodes and $V$ has $n$ nodes. Partition the nodes in the graph into $S_1$ and $S_2$. Without loss of generality, we will focus on the subset $S_1$. Three cases appear:

(a) $|S_1 \cap U| \leq m - r$ and $|S_1 \cap V| \leq n - r$. In this case, all of the nodes in $|S_1 \cap U|$ have $|V \setminus S_1| \geq 2r - 1 - |S_1 \cap V| \geq r$ and all of the nodes in $|S_1 \cap V|$ have $|U \setminus S_1| \geq 2r - 1 - |S_1 \cap U| \geq r$ neighbors outside of $S_1$. Since every node in $S_1$ has at least $r$ neighbors outside of $S_1$, $|X_{S_1}^r| = |S_1|$ is satisfied.

(b) $|S_1 \cap U| \leq m - r$ and $|S_1 \cap V| > n - r$. In this case, all of the nodes in $|S_1 \cap V|$ will have $|U \setminus S_1| \geq r$ nodes outside of $S_1$, which means that $|X_{S_1}^r| = |S_1 \cap V| > n - r \geq 2r - 1 - r \geq r - 1 \geq s$ and therefore, $|X_{S_1}^r| + |X_{S_2}^r| \geq s$ is satisfied.

(c) $|S_1 \cap U| > m - r$ and $|S_1 \cap V| \leq n - r$. In this case, all of the nodes in $|S_1 \cap U|$ will have $r$-neighbors outside of their subset, since $|V \setminus S_1| \geq r$. Since $|S_1 \cap U| > m - r \geq r$, and $s \leq r$, then $|X_{S_1}^r| \geq s$ and the condition $|X_{S_1}^r| + |X_{S_2}^r| \geq s$ is satisfied.

(d) $|S_1 \cap U| > m - r$ and $|S_1 \cap V| > n - r$. In this case, none of the nodes in $S_1$ will have $r$ neighbors outside of $S_1$. However, $|S_2 \cap U| \leq m - |S_1 \cap U| < m - r$ and
\[|S_2 \cap V| \leq n - |S_2 \cap V| < n - r.\] By following case (a), all of the nodes in \(S_2\) have \(r\) or more neighbors outside of \(S_2\) and \(|X_{S_2}^r| = |S_2|\) is satisfied.

Because all three cases are satisfied, the proof is complete. \(\square\)

### 2.4.3 1-D Distance Graph

Determining the resilience of a 1-D distance graph holds considerable significance in comprehending the robustness and efficiency of linear network structures. 1-dimensional distance graphs, also known as line graphs, represent a simple yet crucial model for various real-world systems, such as transportation networks, linear sensor arrays, and communication channels.

Evaluating the resilience of such graphs is vital to understanding their ability to maintain connectivity and data transmission in the face of node or link failures, signal interference, or environmental disruptions. This analysis enables researchers and engineers to identify weak points along the linear structure and develop strategies to ensure seamless information flow and system operation, even under adverse conditions. Finding the resilience of a 1-dimensional distance graph provides valuable insights into enhancing the reliability and performance of linear network infrastructures, offering practical solutions to challenges faced in industries ranging from telecommunications to transportation and beyond.

We consider a geometric graph, \(G_{n,p,l}^d = (V, E)\), which is an undirected graph which is created by placing \(n\) nodes in a region \(\Omega_d = [0, l]^d\). An edge will exist between node \(u\) and \(v\) if \(||x_u - x_v|| \leq p\), where \(p\) is a threshold which is often taken to be a standard Euclidean norm \((\text{Zhang et al.}, 2015)\).

**Theorem 2.4.** If \(\Omega_1 = [0, l]^1\), given a graph \(G_{n,p,l}^1\), which is a 1-D distance graph, is \(2(r + s - 1)\) connected, then it is \((r, s)\)-robust.

**Proof:** Theorem 4 \((\text{Zhang et al.}, 2015)\) states that if \(\Omega_1 = [0, l]^1\) and a graph, \(G_{n,p,l}^1\), is at least \(r\)-connected, it is \(\frac{r}{2}\)-robust. With our graph being \(2(r + s - 1)\)-connected, we can then find
that our graph is \((r + s - 1)\)-robust. By \cite{LeBlanc2012, Usevitch2017}, then the graph is \((r, s)\)-robust. With this, we can then imply our graph that is \((r + s - 1)\)-robust is \((r, s)\)-robust and the proof is complete.

\[\square\]

### 2.4.4 Circulant Graph

Circulant graphs are characterized by their cyclic nature, making them relevant models for various real-world systems, including communication networks, signal processing, and data distribution. Determining the resiliency of such graphs is crucial to understanding how these repeating patterns impact the overall network’s fault-tolerance and ability to maintain connectivity in the presence of failures or disruptions. This analysis aids researchers and practitioners in identifying vulnerable regions within the cyclic structure and devising effective strategies to fortify the network against potential threats or disturbances. By studying the resiliency of circulant graphs, valuable insights can be gained to optimize their design and performance, ensuring reliable and efficient operation in practical applications, ranging from telecommunications to distributed computing and beyond.

**Definition 2.7.** (Circulant graph) A circulant graph \(C_{N_c}^{1,2,\cdots,M}\) is an undirected graph with \(N_c\) nodes, denoted by \(\{u_0, u_1, \cdots, u_{N_c-1}\}\), where each \(u_i\) is adjacent to \(u_{i \pm j} \pmod{N_c}\) for all \(j \in \{1, \cdots, M\}\).

**Lemma 2.1.** Given a circulant graph, \(G = C_N^{\cdots,x,\cdots} = (V_C, E_C)\), with \(N\) number of nodes arranged in a circle where each node contains an arbitrary adjacency set, denoted as \(\cdots, x, \cdots\). Consider two subsets of nodes, \(S_1, S_2 \subseteq V_C\) and \(S_1 \cap S_2 = \emptyset\), then a partition of \(V_C\) exists that creates two contiguous intervals of nodes, \(I_1\) and \(I_2\), such that \(|I_1 \cap S_1| = |I_2 \cap S_1|\) and \(|I_1 \cap S_2| = |I_2 \cap S_2|\). If \(S_1\) and \(S_2\) partition \(V_C\) (i.e., \(S_1 \cup S_2 = V_C\)) and \(|S_1|, |S_2|\) are even, then \(|I_1| = |I_2|\).
Proof: To prove this, we first show the proof for when each node belongs to a subset and then extend to when nodes that do not belong to a subset are included as well. For notation, assume that $|S_1| = 2a$ and $|S_2| = 2b$, where $a$ and $b$ are integers. Since $|S_1| + |S_2| = N$, then $2a + 2b = N$ as well. Begin by bisecting the graph with two end points, $P_1$ and $P_2$. This induces two contiguous intervals, $I_1 = \{u_{P_1}, u_{P_1+1}, \ldots, u_{P_2-1}\}$ and $I_2 = \{u_{P_2}, u_{P_2+1}, \ldots, u_{P_1-1}\}$, where $|I_1| = |I_2| = \frac{N}{2}$.

Assume that the nodes in $S_1$ have a value of $+1$ and the nodes in $S_2$ have a value of $-1$. If we sum the values of all of the nodes, then $I_1$ will have a value of $\text{sum}(I_1) = |S_1 \cap I_1| - |S_2 \cap I_1|$ and $I_2$ will have a value of $\text{sum}(I_2) = |S_1 \cap I_2| - |S_2 \cap I_2|$. If each interval has half of the nodes from $S_1$ and $S_2$, then $\text{sum}(I_1) = \text{sum}(I_2) = a - b$, because $S_1$ and $S_2$ span the graph. For notation, denote $a - b = m$.

Induce a partition on the graph, such that both $I_1$ and $I_2$ contain at least one node from $S_1$ and $S_2$. Since $S_1$ and $S_2$ span the graph, then this is possible. Without loss of generality, find $\text{sum}(I_1)$. If $\text{sum}(I_1) = m$, then each interval has the required number of nodes from $S_1$ and $S_2$ are in each interval. If $\text{sum}(I_1) > m$ or $\text{sum}(I_2) < m$, then we can rotate the partition. Suppose we denote $\text{sum}(I_1) = m + l$, then $\text{sum}(I_2) = m - l$, where $l$ is an integer, which states how far we are from our desired amount $m$. Say $\text{sum}(I_1) = k + l = |S_1 \cap I_1| - |S_2 \cap I_1|$. Since $|S_1 \cap I_2| = 2a - |S_1 \cap I_1|$ and $|S_2 \cap I_2| = 2b - |S_2 \cap I_2|$, then

$$\text{sum}(I_2) = |S_1 \cap I_2| - |S_2 \cap I_2| = 2a - |S_1 \cap I_1| -(2b - |S_2 \cap I_2|) = 2a + 2b - (m + l)$$

Because $S_1$ and $S_2$ span the entire graph, $2m = 2a + 2b$ and $\text{sum}(I_2) = 2m - m - l = m - l$.

Without loss of generality, rotate the partition clockwise which makes

$I_1(k) = \{u_{P_1+k}, u_{P_1+1+k}, \ldots, u_{P_2-1+k}\}$ and $I_2(k) = \{u_{P_2+k}, u_{P_2+1+k}, \ldots, u_{P_1-1+k}\}$,

where $I_1(k)$ and $I_2(k)$ are the new induced intervals and $k$ is the time step at which the interval was moved, where $k \geq 0$. When this partition moves, there are two possibilities:
Figure 2.2: Eight nodes, four from $S_1$ (red) and four from $S_2$ (blue). By rotating the partition, a partition exists where $S_1$ and $S_2$ are evenly divided.

1. $u_{P_1+k}$ and $u_{P_2+k}$ belong in the same subset. In this case, each interval removes one node from their interval and then adds a node from their interval. In this case, $\text{sum}(I_1(k+1)) = \text{sum}(I_1(k))$ and $\text{sum}(I_2(k+1)) = \text{sum}(I_2(k))$ where the sum does not change.

2. $u_{P_1+k}$ and $u_{P_2+k}$ belong in different subsets. Then, each interval loses a node from $S_1$ and gains a node from $S_2$ or vice versa, so $\text{sum}(I_1(k+1)) = \text{sum}(I_1(k)) \pm 2$ and $\text{sum}(I_2(k+1)) = \text{sum}(I_2(k)) \pm 2$.

Now, assume we repeat this process until $k = \frac{N}{2}$. In this case, $\text{sum}(I_1(\frac{N}{2})) = \text{sum}(I_2(0))$ and $\text{sum}(I_2(\frac{N}{2})) = \text{sum}(I_2(0))$. This is because at $k = \frac{N}{2}$, $P_1$ is in the position where $P_2$ was. Because each rotation step can only change by $\pm 2$, then $l$ must be an even number. In that case, due to the intermediate value theorem, because $\text{sum}(I_1(0)) = m + l$ and $\text{sum}(I_1(\frac{N}{2})) = m - l$, at some time step $0 \leq k \leq \frac{N}{2}$, $\text{sum}(I_1(k)) = m$ and $\text{sum}(I_2(k)) = m$ and the proof is complete.

To extend this proof to an odd number of nodes, where $|S_1|$ or $|S_2|$ might not be even, there will be at least one extra node in either or both of the subsets. We can assign that node as an unassigned node, as it doesn’t matter in which interval it exists. This proof can now be extended to where $S_1$ and $S_2$ do not create a partition, or $|S_1| + |S_2| < N$. In this case, begin
by inducing a random partition as previously, however, now ignore the unassigned nodes. Induce the partition such that each interval contains at least $a + b$ nodes from either $S_1$ or $S_2$. Now, repeat the same process as before, but when we rotate the partition, if the node is an unassigned node, skip it until the node is a node that belongs to $S_1$ or $S_2$. Because the values in each interval will be $m + l$ and $m - l$, and the unassigned values can be ignored, the same rotation can happen where $\text{sum}(I_1(\frac{N}{2})) = \text{sum}(I_2(0))$ and $\text{sum}(I_2(\frac{N}{2})) = \text{sum}(I_2(0))$.

However, now because the unassigned nodes can be ignored, then $|I_1|$ does not have to equal $|I_2|$, since we can freely move the unassigned nodes around. Because $\text{sum}(I_1(0)) = m + l$ and $\text{sum}(I_1(\frac{N}{2})) = m - l$, at some time step $0 \leq k \leq \frac{N}{2}$, $\text{sum}(I_1(k)) = m$ and $\text{sum}(I_2(k)) = m$ and the proof is complete.

**Theorem 2.5.** Given a graph, $G = C_N^{r \cdots r + \frac{s}{2} - 1}$, which is a circulant graph, where positive integers $r = s$ and the number of nodes $N \geq 4(r + \frac{s}{2} - 1)$, the graph is $(r, s)$-robust.

**Proof:** We are given a graph, $G = C_N^{r \cdots r + \frac{s}{2} - 1}$, which is a circulant graph, with positive integers, $r = s > 0$, and a number of nodes, $N \geq 4(r + \frac{s}{2} - 1)$. Due to the structure of the circulant graph, every node will have $2(r + \frac{s}{2} - 1) = 2r + s - 2 = 3r - 2$ edges connected to the node. By selecting two disjoint subsets, $S_1$ and $S_2$, Without loss of generality, there are two cases that arise:

(a) $|S_1| < 2r$ or $|S_2| < 2r$. In this case, because every node has $3r - 2$ edges, in the worst case, every node has an edge with every other node in the subset. Therefore, the minimum number of edges (including its own value) for each node outside of the subset is $3r - 2 + 1 - (2r - 1) = r$. This satisfies the condition $|X_{S_1}^r| = |S_1|$ or $|X_{S_2}^r| = |S_2|$.

(b) $|S_1| \geq 2r$ and $|S_2| \geq 2r$. By Lemma 2.1, we know that we can partition the graph to create two continuous intervals with at least $r$ nodes each. The induced subgraphs have the same structure as a 1-D distance graph that is $2(r + \frac{s}{2} - 1)$ connected. By Theorem
we find that these subgraphs are \( (r, \frac{s}{2}) \)-robust. Because each interval has at least \( r \) nodes from each subset, then there are two possibilities. If \( |X^r_{S_1}| = |S_1| \) or \( |X^r_{S_2}| = |S_2| \) is the condition that is satisfied to achieve \( (r, s) \)-robustness for either interval, then since \( |S_1| \geq r \) and \( |S_2| \geq r \) in each of the subsets, for the entire circulant graph, at least \( r \) nodes will be \( r \)-reachable, satisfying \( |X^r_{S_1}| + |X^r_{S_2}| \geq s \). If \( |X^r_{S_1}| + |X^r_{S_2}| \geq s \) is the condition that is satisfied in both intervals, then each interval must produce \( \frac{s}{2} \) \( r \)-reachable nodes. Because this is true for 2 intervals, then for the entire circulant graph, \( |X^r_{S_1}| + |X^r_{S_2}| \geq \frac{2s}{2} = s \) is satisfied.

Because all possible conditions are satisfied, the proof is complete.

On each interval, we showed that there must exist at least \( \frac{s}{2} \) nodes that are \( r \)-reachable. However, the main reason that the \( s \) condition is bounded by \( \frac{s}{2} \) is that if, without loss of generality, one of the subsets has all but \( r \) nodes, or \( |S_1| < N - r \), then the subset cannot have any \( r \) reachable nodes. The other subset will only have \( \frac{s}{2} \) \( r \)-reachable nodes on this interval, but if extended with the additional edges provided by the circulant graph, it could have more.

If it can be shown that on those intervals, if \( |S_1| < N - r \) or \( |S_2| < N - r \), then all of the nodes in \( S_1 \) or \( S_2 \) will be \( r \)-reachable and if not, then both subsets will produce 2\( s \) \( r \)-reachable nodes, depending on the \( (r, s) \)-robustness of the interval, then we would be able to show that a circulant graph \( G = C^4_{N, \ldots, r+\frac{r}{2} - 1} \) will be \( (r, s) \)-robust. It can be shown by simulation that this is true for smaller numbers, but it has yet to be proven for all cases of \( r, s, \) and \( N \).

2.5 Limitations of the W-MSR Algorithm

As seen above, the W-MSR algorithm is powerful a powerful strategy to allow convergence under \( F \) malicious nodes, with the condition that the graph is \( (F+1, F+1) \)-robust. However,
if the graph is not \((F + 1, F + 1)\)-robust, then the current state of research currently does not
know the dynamics of the graph, and the graph is presumed to be 'lost'. With the current
state of research, some of the limitations to applying the W-MSR algorithm include:

- **Knowledge of \(F\):** To implement the W-MSR algorithm, we must first know how many
  adversaries will exist in the graph. Because we currently do not know the dynamics of
  the graph when there are more malicious nodes than the graph can handle, we would
  have to create systems that are more resilient than needed, which can be costly to
  implement.

- **Strict Robustness Conditions:** The current robustness conditions are strict and
  can become impractical. Since we need a minimum of \((F + 1, F + 1)\)-robustness, then
  the graphs can become costly to implement as well.

- **Understanding of Resilience:** The W-MSR algorithm and current research depicts
  on what occurs when the graph has met the robustness condition or more. By un-
  derstanding what occurs when the graph has not met those conditions, we can better
  understand how resilience works and design better systems and algorithms accordingly.

These limitations in the current research can make implementation of resilient distributed
networks difficult to implement. In the next chapter, we introduce a new idea of non-
resilience, which precisely studies what happens when the graph has not met the \((F + 1, F +
1)\)-robustness conditions. We will demonstrate that not everything will be lost and in doing
so, we provide a new metric to design better and less costly systems.
CHAPTER 3

QUANTIFYING NON-RESILIENCE

$(r, s)$-robustness and the W-MSR algorithm impose stringent conditions on the graph that might be difficult to achieve. In practice, we may often not have the resources to implement such a graph or we may underestimate the number of adversarial nodes in a graph. This Section proposes a new understanding the dynamics of an $(r, s)$-robust graph, where $r$ or $s \leq F$, or the number of malicious nodes is greater than what the graph can handle. We begin by defining a new notion of non-convergence. Then we demonstrate the properties of non-convergence by finding the upper bound on the number of non-convergent nodes in the graph. Finally, we illustrate the behavior of non-convergent nodes and their correlation with the robustness of the graph.

3.1 Notion of Non-Convergence

In a network with $(r, s)$-robustness, if that network is $(F + 1, F + 1)$-robust, then all of the normal nodes in the graph will converge. If even $r$ or $s$ is less than $F + 1$, then the theory does not guarantee resilient convergence, however, it is still possible for some of the nodes to converge. To be able to quantify the effect of not meeting $(F + 1, F + 1)$-robust conditions, a new metric needs to be defined in order to determine the behavior. This raises the important question:

How can we quantify the performance of a network when the network when the robustness conditions have been failed to be met when facing $F$ total malicious agents?

At a network level, the set of adversaries attempt to disrupt the convergence of the graph. At a node level, an adversary attempts to prevent convergence of a normal node to any other particular normal node in the graph. This Section will focus on determining the node level dynamics, where the adversaries prevent particular normal nodes from converging. An attack
can exist to prevent a node, $u$, to converge with another normal node, $v$. However, an attack might not exist to prevent the node $u$ from converging with another node $w$. We evaluate the adversaries impact at a node level that prevents a normal node, $u$, from converging with any other normal node in the graph. This concept is defined in definition 3.1 as follows:

**Definition 3.1. (Non-Convergent Node)** A node, $u$, is non-convergent (under $F$-total model), if for every $v \in V \setminus \{u\}$, there is a set of at most $F$ malicious nodes from $V \setminus \{u, v\}$ preventing $u$ and $v$ from converging at a common point.

**Definition 3.2. (Local Non-Resilience)** A graph, $G = (V, E)$ has $p$-local non-resilience if the proportion of non-convergent nodes to the total number of nodes in the graph is $p$.

To determine if a node $u$ is non-convergent, then we determine if attacks exists such that they do not converge at with every other normal node in the graph. To highlight the effect of non-convergence on a graph, we construct graphs with certain $r$ and $s$ values to maximize the number of non-convergent nodes. In turn, this will provide a lower bound of resilience for graphs at a specific $(r, s)$-robustness.

If a graph is $(F + 1, F + 1)$-robust, then by definition, there will be no non-convergent nodes. This is because there does not exist an attack that can prevent two normal nodes from converging, therefore, a non-convergent node cannot exist. Whenever the graph is not $(F + 1, F + 1)$-robust, or $r, s < F + 1$, then there can exist some non-convergent nodes, which reflects the graph’s vulnerability to $F$ total adversarial nodes. If the number of non-convergent nodes is high, or the local non-resilience is high, then the graph is more vulnerable. Generally, if the $(r, s)$-robustness increases, then the number of non-convergent nodes or local non-resilience decreases as depicted in Figure 3.1.

### 3.2 Resilience Lower Bound for Graphs

The number of non-convergent nodes or local non-resilience can vary with graphs with the same robustness, which will be described further in later Sections. To be able to understand
Figure 3.1: The number of non-convergent nodes (highlighted red) in (a), (b), and (c) are 6, 3, and 1, respectively.

the extent of the effect that not meeting the robustness conditions, we try to construct graphs that maximize the number of non-convergent nodes. We construct graphs that are not \((F + 1, F + 1)\)-robust and therefore, do not guarantee convergence with \(F\) malicious nodes. We try to construct graphs that maximize the local non-resilience, which will act as the lower bound on resilience. In particular, we construct graphs for \((F + 1, 1)\), \((F, F)\), and \((F + 1, F)\)-robust graphs.

In the construction, we utilize the graph join of complete, circulant, and empty graphs. The definitions for complete and circulant graphs are given above. We now define graph join and empty graphs below.

**Definition 3.3. (Graph Join)** Given two graphs \(G_1 = (V_1, E_1)\) and \(G_2 = (V_2, E_2)\), the join graph, denoted by \(G_1 \oplus G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{(a, b) : a \in V_1, b \in V_2\})\). In other words, each node \(u\) in \(G_1\) is adjacent to all the nodes in \(G_2\).

**Definition 3.4. (Empty Graph)** An empty graph, denoted by \(E_N\), is a graph of \(N\) nodes and an empty edge set, or there are no edges that exist between the nodes in the graph.

Using these graphs, we can construct graphs with the robustness to maximize the local non-resilience. To demonstrate the upper bound of non-convergence on these graphs, we first provide a lemma for the construction of such graph. These constructions will not only
achieve the \((r, s)\)-robustness required, but they will also make sure that the graphs cannot have a larger robustness, or be \((r + 1, s)\)-robust or \((r, s + 1)\)-robust. After the lemma, a Theorem will prove the number of non-convergent nodes for the graph.

3.2.1 \((F + 1, 1)\)-Robust Case

In this subsection, we will explore the case of graphs that are \((F + 1, 1)\)-robust, in which we try to maximize the number of non-convergent nodes. This type of graph is considerably less robust than the \((F + 1, F + 1)\)-robust graph, which guarantees convergence under \(F\)-total adversarial model. The interest for exploring an \((F + 1, 1)\)-robust graph is that even though the \(r\) condition is satisfied, the \(s\) condition is completely relaxed. That means, that every pair of subsets will be able to have \(F + 1\)-reachable nodes, but it will only take one adversary to be able to disrupt two subsets from converging. A construction of such a graph is shown below.

**Lemma 3.1.** For given integers \(F \geq 2\) and \(N_c \geq 2F + 1\), the graph \(G = \mathcal{E}_2 \oplus \mathcal{C}_{N_c}^{1, \cdots, F-1}\) is \((F + 1, 1)\)-robust.

**Proof:** Let \(U\) and \(V\) denote the set of nodes in \(\mathcal{C}_{N_c}^{1, \cdots, F-1}\) and \(\mathcal{E}_2\), respectively. Let \(S_1\) and \(S_2\) be two disjoint non-empty sets of nodes in \(G\). We show that at least one of these subsets is \((F + 1)\)-reachable. There are three cases.

(a) At least one of the subsets contains nodes from \(V\) only. Without the loss of generality (w.l.o.g.), assume \(S_1 \subseteq V\). Since \(|U| = N_c \geq 2F + 1\) and each \(v \in S_1\) is adjacent to all nodes in \(U\), we get \(X_{S_1}^{F+1} = S_1\).

(b) At least one of the subsets, say \(S_1\), contains nodes from \(U\) only, i.e., \(S_1 \subseteq U\). There are two choices for \(S_2\).
(b-1) $S_2 \cap V = \emptyset$: In this case, at least one of the subsets $S_1$ and $S_2$ have at most $F$ nodes as $|U| \geq 2F + 1$. Without loss of generality, assume $|S_1| \leq F$. Each node in $U$, and hence in $S_1$, has at least $2F - 2$ neighbors in $U$. Thus, each $u \in S_1$ has at least $(2F - 2) - (F - 1) = F - 1$ neighbors in $U \setminus S_1$. Also, each $u \in S_1$ is adjacent to both nodes in $V$. Thus, $u \in S_1$ has at least $F + 1 = F$ neighbors outside of $S_1$, thus, the subset $S_1$ is $(F + 1)$-reachable.

(b-2) $S_2 \cap V \neq \emptyset$: If $|S_1| \leq F$, then $S_1$ is $(F + 1)$-reachable by the above case (b-1). So, assume $|S_1| \geq F + 1$. Since $V \cap S_2 \neq \emptyset$, let $v \in (S_2 \cap V)$. Note that $v$ is adjacent to all nodes in $S_1$, which means the subset $S_2$ is $(F + 1)$-reachable.

(c) $S_1$ and $S_2$ contain nodes from both $U$ and $V$. Let $v_1 \in (S_1 \cap V)$ and $v_2 \in (S_2 \cap V)$. Since $|U| \geq 2F + 1$, at least one of the subsets $S_1 \cap U$ and $S_2 \cap U$ has at most $F$ nodes. Assume without loss of generality that $|S_1 \cap U| \leq F$. Then, $v_1$ has at least $F + 1$ neighbors outside of $S_1$ (as $v_1$ is adjacent to all the nodes in $U$). As a result, $S_1$ is $(F + 1)$-reachable. This completes the proof.

Figure 3.2 demonstrates a graph, $G = E_2 \oplus C_1^{1,2}$ which produces a graph that is $(4,1)$-robust. In this graph, any subset, $S_1$ or $S_2$, that includes $v_1$ or $v_2$, one of those two nodes will always be $r$-reachable, which makes the other nodes the ones that are non-convergent. On the other hand, if those two nodes are not included in $S_1$ or $S_2$, then one of the nodes, \( \{u_0, \ldots, u_6\} \) will be $r$-reachable. Next, we show how many non-convergent nodes there are in this construction.

**Theorem 3.1.** For given integers $F \geq 2$ and $N \geq 2F + 3$, there exists an $(F + 1, 1)$-robust graph with $N$ nodes, such that the number of non-convergent nodes is $N - 2$.

**Proof:** Let $N_c = N - 2$, and consider the graph $G = E_2 \oplus C_1^{1,2,F-1}$, which is $(F + 1, 1)$-robust by Lemma 3.1. Let $U = \{u_0, \ldots, u_{N_c-1}\}$ denote the set of nodes in $C_1^{1,2,F-1}$ and
$V = \{v_1, v_2\}$ denote the two nodes in $E_2$. Note that $|U| + |V| = N$. We will show that each $u_i \in U$ is a non-convergent node under the $F$-total attack model. First, we show that $G$ is not $(F+1, F+1)$-robust.

Consider two disjoint subsets $S_1$ and $S_2$, where $S_1 = \{u_0, \cdots, u_{F-1}\} \cup \{v_1\}$, and $S_2 = (U \setminus S_1) \cup \{v_2\}$. Note that each $u_j \in U$ has $2(F-1)$ neighbors in $U$. Also, each $u_i \in S_1 \cap U$ has at most $F-1$ neighbors in $U \setminus S_1$ and only one neighbor in $V \setminus S_1$. Thus, each node in $S_1 \cap U$ has at most $F$ neighbors outside of $S_1$. Since $v_1$ is adjacent to all nodes in $U \setminus S_1$ and $|U \setminus S_1| \geq F+1$, we have $X_{S_1}^{F+1} = \{v_1\}$. Similarly, each $u_j \in S_2 \cap U$ has at most $F$ neighbors outside of $S_2$. Also, $v_2$ has at most $F$ neighbors outside of $S_2$ (as $|S_1| = F+1$ and $v_2$ is not adjacent to $v_1 \in S_1$). Thus, $X_{S_2}^{F+1} = \emptyset$, which means $|X_{S_1}^{F+1} \cup X_{S_2}^{F+1}| = 1 < F+1$, and $G$ is not $(F+1, F+1)$-robust.

Now, assign some value $a \in \mathbb{R}$ to all the nodes in $S_1$, and some value $b > a$ to all the nodes in $S_2$. Let $v_1 \in S_1$, which is the only node having $F+1$ neighbors outside of $S_1$, be the malicious node, and all the remaining nodes in $S_1$ and $S_2$ are normal. Then, each normal node has at most $F$ neighbors outside of its respective subset (i.e., $S_1$ and $S_2$). It means each normal node in $S_1$ has at most $F$ neighbors with values strictly greater than the node’s value. Similarly, each normal node in $S_2$ has at most $F$ neighbors with values strictly smaller than the node’s value. By implementing the W-MSR algorithm, each normal node
in $S_1 \cup S_2$ removes values from all of its neighbors that are outside of its respective subset, and hence, never updates its value. This means normal nodes in $S_1$ and $S_2$ maintain the values $a$ and $b$, respectively, and do not converge at a common value.

In particular, consider $u_0 \in S_1$, and observe that it does not converge to any of the nodes in $S_2 = \{u_F, \ldots, u_{N_c-1}, v_2\}$. Now, we select again two disjoint nonempty subsets, $S'_1$ and $S'_2$, as following:

Let $S'_1 = \{u_0, u_{N_c-F+1}, \ldots, u_{N_c-1}\} \cup \{v_2\}$ (i.e., in $S'_1$, include the nodes in $U$ that are on the ‘left’ of $u_0$ compared to the previous case of $S_1$, where nodes to the ‘right’ of $u_0$ were included). Note that $|S'_1| = F + 1$. Moreover, let $S'_2 = (U \setminus S'_1) \cup \{v_1\}$, and assume $v_2 \in S'_1$ to be the malicious node. Then, by the same argument used above (i.e., in the case of $S_1$ and $S_2$), we can ensure that $u_0$ does not converge to any of the nodes in $S'_2$. Since $S_2 \cup S'_2 = (U \cup V) \setminus \{u_0\}$, we ensure that for every node pair $(u_0, x)$, where $x \in (U \cup V) \setminus \{u_0\}$, there is an attack of at most $F$ nodes such that $u_0$ and $x$ do not converge. It means that $u_0$ is a non-convergent node. By the symmetry of the graph and applying the same arguments as above to other nodes in $U$ implies that all the nodes in $U$, where $|U| = N - 2$, are non-convergent, which completes the proof.

Thus, an upper bound on the number of non-convergent nodes is $N - 2$. As $N \to \infty$, then the local non-resilience, or the ratio of non-convergent nodes to $N$, approaches 1. In this case, we show that if we just relax the $s$ condition, it is possible to lose the guarantee of convergence for almost every node in the graph.

3.2.2 $(F,F)$-Robust Case

Next, we explore the case of $(F,F)$-robust graphs. These graphs only relax the $r$ and $s$ condition by 1, which would guarantee convergence, thus making them converge under $F - 1$ adversaries. The importance of these graphs is that we are only increasing the number of adversaries by one. This examination will demonstrate the upper bound of non-convergence on such graphs.
Before the graph construction, we state the following observation 3.1 about circulant graphs.

**Observation 3.1.** Consider a circulant graph $C_{N_c}^{1,2,\cdots,\lceil \frac{F}{2} \rceil -2}$, where $F \geq 5$ and $N_c \geq F + 1$. Let $i$ be some positive integer, where $3 \leq i \leq \frac{F+2}{2}$. If $S$ is a subset of nodes in the circulant graph, where $F - i \leq |S| \leq N_c - (1+i)$, then, at least one of the following is true.

(i) The number of nodes in $S$ that are adjacent to at least $(i-2)$ nodes outside of $S$ is at least $F + 2 - 2i$, i.e., $|X_{i-2}^S| \geq F + 2 - 2i$.

(ii) All nodes in $S$ are adjacent to at least $i - 2$ nodes outside of $S$, i.e., $|X_{i-2}^S| = |S|$.

Figure 3.3 illustrates the observation through examples. Consider a circulant graph $C_{10}^{1,2}$ with $F = 8$ and $N_c = 10$. For $i = 4$, Figure 3.3(a) shows a set $S$ of size 4. There are two $(F + 2 - 2i = 2)$ nodes in $S$, shown in red, such that each of them has two $(i - 2 = 2)$ neighbors outside of $S$. Similarly, in Figure 3.3(b), we consider $i = 3$ and a set of nodes $S$ of size 5. By Observation 3.1 there exist four $(F + 2 - 2i = 4)$ nodes in $S$ (red colored), each of which has at least $i - 2 = 1$ neighbor outside of $S$.

![Figure 3.3](image-url)

**Figure 3.3:** (a) A set $S$ of four nodes contains two nodes (red), each of which has two neighbors outside of $S$. (b) A set $S$ contains five nodes, of which four nodes (red) have at least one neighbor outside of $S$.

Now, we present construction of such $(F, F)$-graph.
Lemma 3.2. For integers $F > 4$ and $N \geq 2F + 3$, the graph $G = K_{F+2} \oplus C_1^{N-(F+2)}$, which is the join of complete graph $K_{F+2}$ and circulant graph $C_1^{N-(F+2)}$, is $(F, F)$-robust.

Proof: Let $U$ and $V$ denote the set of nodes in $C_1^{N-(F+2)}$ and $K_{F+2}$, respectively. Let $S_1$ and $S_2$ be two disjoint non-empty sets of nodes in the given $G$. There are three cases:

(a) One of the subsets contains nodes from $V$ only. W.l.o.g, assume $S_1 \subseteq V$. Since $|U| \geq F + 1$ and each $v \in S_1$ is adjacent to all nodes in $U$, we get $X^F_{S_1} = S_1$.

(b) One of the subsets contains nodes from $U$ only. W.l.o.g, assume $S_1 \subseteq U$: Since $|V| \geq F + 2$ and each $u \in S_1$ is adjacent to all nodes in $V$, we get $X^F_{S_1} = S_1$.

(c) Both $S_1$ and $S_2$ contain nodes from $U$ and $V$. We have further two cases.

(c-1) One of the subsets, say $S_1$ has at most $(F - 1)$ nodes: In this case, consider $|S_1 \cap V| = \nu_1$, then $|S_1 \cap U| \leq F - 1 - \nu_1$. Let $v \in S_1 \cap V$. The number of neighbors of $v$ outside of $S_1$ are:

$$= ((F + 2) - \nu_1) + (|U| - |S_1 \cap U|)$$

$$\geq (F + 2 - \nu_1) + ((F + 1) - (F - 1 - \nu_1))$$

$$= F + 4.$$

Similarly, let $u \in S_1 \cap U$. Note that $u$ has $2([F/2] - 2)$ neighbors in $U$. Then, the number of neighbors of $u$ outside of $S_1$ are:

$$\geq ((F + 2) - \nu_1) + (2([F/2] - 2) - ((S_1 \cap U) - 1))$$

$$\geq (F + 2 - \nu_1) + (F - 4 - (F - 1 - \nu_1 - 1))$$

$$= F.$$

Thus, each node in $S_1$ has at least $F$ neighbors outside of $S_1$, i.e., $X^F_{S_1} = S_1$. 

31
(c-2) Both subsets $S_1$ and $S_2$ have at least $F$ nodes:

In this case, if at least one of the subsets, say $S_1$, has at most two nodes from $V$. Then, since $|V \setminus S_1| \geq F$ and each node in $S_1$ is adjacent to all nodes in $V$, we have $X^F_{S_1} = S_1$. So, we consider that both $S_1$ and $S_2$ contain at least three nodes from $V$. We will next compute $|X^F_{S_1}|$ and $|X^F_{S_2}|$, and show that $|X^F_{S_1}| + |X^F_{S_2}| \geq F$.

For this, let $|S_1 \cap V| = \nu_1$ and $|S_2 \cap V| = \nu_2$.

Since each node in $S_1 \cap V$ (resp. $S_2 \cap V$) is adjacent to all the nodes in $S_2$ (resp. $S_1$), where $|S_2| \geq F$, we have $|X^F_{S_1}| \geq \nu_1$. Similarly, $|X^F_{S_2}| \geq \nu_2$. So, if $\nu_1 + \nu_2 \geq F$, we have $|X^F_{S_1}| + |X^F_{S_2}| \geq F$, and we are done. Thus, we assume,

$$\nu_1 + \nu_2 \leq F - 1.$$  \hfill (3.1)

Also, note that since $|V| = F + 2$, one of the subsets, say $S_1$, must contain at most $\frac{F+2}{2}$ nodes from $V$, i.e., $|S_1 \cap V| \leq \frac{F+2}{2}$. So, we get

$$3 \leq \nu_1 \leq \frac{F + 2}{2}. \hfill (3.2)$$

Using the above details and (3.1), we also get

$$3 \leq \nu_2 \leq F - 1 - \nu_1. \hfill (3.3)$$

Next, we consider $|S_1 \cap U| = \mu_1$, and $|S_2 \cap U| = \mu_2$.

Observe that $F - \nu_1 \leq \mu_1$ (as $|S_1| \geq F$). Similarly, $F - \nu_2 \leq \mu_2$. Consequently, we get an upper bound on $\mu_1$, i.e., $\mu_1 \leq |U| - (F - \nu_2)$. Using (3.1),

$$|U| - (F - \nu_2) \leq |U| - (1 + \nu_1),$$

thus, $\mu_1 \leq |U| - (1 + \nu_1)$. We write the upper and lower bounds on $\mu_1$ again,

$$F - \nu_1 \leq \mu_1 \leq |U| - (1 + \nu_1). \hfill (3.4)$$
Similarly, the bounds on $\mu_2$ are,

$$F - \nu_2 \leq \mu_2 \leq |U| - (F - \nu_1).$$

(3.5)

Next, we compute $X_{S_1}^F$ and $X_{S_2}^F$.

Since $(S_1 \cap V) \subseteq X_{S_1}^F$, we have $|X_{S_1}^F| \geq |X_{S_1}^F \cap V| = \nu_1$. Note that each $u \in (S_1 \cap U)$ is adjacent to at least $(F + 2) - \nu_1$ nodes in $V \setminus (S_1 \cap V)$. So, if $u \in (S_1 \cap U)$ is adjacent to at least $\nu_1 - 2$ nodes in $U \setminus S_1$, then $u \in X_{S_1}^F$ (as $u$ will have at least $F$ neighbors outside of $S_1$). Now consider (3.2), (3.4), and use Observation 3.1 (plugging $i = \nu_1$), we deduce that the number of nodes in $S_1 \cap U$, each of which is adjacent to at least $\nu_1 - 2$ nodes in $U \setminus S_1$ is at least $F - 2(\nu_1 - 1)$. This gives

$$|X_{S_1}^F| = |X_{S_1}^F \cap V| + |X_{S_1}^F \cap U| \geq \nu_1 + (F - 2(\nu_1 - 1))$$

$$= F + 2 - \nu_1. \tag{3.6}$$

Similarly, considering (3.3), (3.5), and applying a similar argument as for $X_{S_1}^F$, we obtain

$$|X_{S_2}^F| = |X_{S_2}^F \cap V| + |X_{S_2}^F \cap U| \geq F + 2 - \nu_2. \tag{3.7}$$

Now, from (3.6) and (3.7), we get

$$|X_{S_1}^F| + |X_{S_2}^F| \geq (F + 2 - \nu_1) + (F + 2 - \nu_2)$$

$$= 2F + 4 - (\nu_1 + \nu_2) \tag{3.8}$$

Using (3.1),

$$|X_{S_1}^F| + |X_{S_2}^F| \geq 2F + 4 - (F - 1) = F + 3, \tag{3.9}$$

which is the desired result. This completes the proof. \qed
Figure 3.4: $G = K_7 \oplus C_7^1$ is $(F, F)$-robust graph for $F = 5$.

Figure 3.4 demonstrates a graph, $G = K_7 \oplus C_7^1$, which produces a graph that is $(5, 5)$-robust. With any subsets, $S_1$ or $S_2$, if the nodes contain any nodes from $K_7$, then the other nodes will be the nodes that are non-convergent. Next, we show how many non-convergent nodes there are in this construction.

**Theorem 3.2.** For given integers $F > 4$ and $N \geq 2F + 3$, there exists an $(F, F)$-robust graph with a total of $N$ nodes of which $N - (F + 2)$ are non-convergent under an $F$-total attack.

**Proof:** Consider $G = K_{F+2} \oplus C_{N-(F+2)}^{1,\cdots,\lfloor F \rfloor -2}$, which is $(F, F)$-robust by Lemma 3.2. Let $V = \{v_1, \cdots, v_{F+2}\}$ denote the set of nodes in $K_{F+2}$, and $U = \{u_1, \cdots, u_{N-(F+2)}\}$ be the set of nodes in $C_{N-(F+2)}^{1,2,\cdots,\lfloor F \rfloor -2}$. Note that each $u_i \in U$ has $2(\lceil \frac{F}{2} \rceil - 2)$ neighbors in $U$.

For the non-convergent nodes, first, we show that the graph is not $(F+1, F+1)$-robust. Let $S_1$ be a set consisting of a single node from $U$, say $u \in U$, and $F-1$ nodes from $V$. Also, let $S_2$ be the set of remaining nodes, i.e., $S_2 = (U \cup V) \setminus S_1$. Note that $|S_1| = F$, so $X_{S_2}^{F+1} = \emptyset$. Also, $u \in S_1$ has $3 + 2(\lfloor \frac{F}{2} \rfloor - 2) = 2\lceil \frac{F}{2} \rceil - 1 \leq F$ neighbors outside of $S_1$. At the same time, each $v \in (S_1 \cap V)$ has at least $F + 1$ neighbors outside of $S_1$. Thus, $X_{S_1}^{F+1} = S_1 \setminus \{u\}$, and $|X_{S_1}^{F+1}| = F - 1 < |S_1|$. As a result, none of the three conditions for $(F+1, F+1)$-robustness are satisfied by sets $S_1$ and $S_2$, the considered graph is not $(F+1, F+1)$-robust.
Now, assume that the set of malicious nodes contains \( X_{S_1}^{F+1} \cup X_{S_2}^{F+1} \). Note that \( |X_{S_1}^{F+1} \cup X_{S_2}^{F+1}| \leq F \). Also, \( u \in S_1 \) is the only normal node in \( S_1 \) as \( u \notin X_{S_1}^{F+1} \). Now, assign value \( a \) to all nodes in \( S_1 \), and value \( b > a \) to nodes in \( S_2 \). Note that all normal nodes in \( S_1 \) and \( S_2 \) have at most \( F \) neighbors outside of their respective sets, and as per the W-MSR algorithm, each normal node in \( S_1 \) and \( S_2 \) removes \( F \) values outside of its respective set. Thus, \( u \in S_1 \), and other normal nodes in \( S_2 \) never update their values. Thus, \( u \) never converges to another normal node and is a non-convergent node. This scenario can be replicated for every node in \( U \) while applying the same arguments; thus, the number of non-convergent nodes in the graph is \( |U| = N - (F + 2) \). This completes the proof.

Thus, an upper bound on the number of non-convergent nodes is \( N - (F + 2) \). As \( N \to \infty \), then the local non-resilience approaches 1 as well. Just like the previous case, just by relaxing some of the conditions, we show that locally, it is possible to lose guarantee of most of the nodes in the graph. However, this bound is lower than the bound for the \((F + 1, 1)\)-case. As \( F \) gets larger, we show that the upper bound decreases.

### 3.2.3 \((F + 1, F)\)-Robust Case

The final graph that will be explored is the \((F + 1, F)\)-robust case. In this case, like the \((F, F)\)-case, the graph is guaranteed to converge under \( F - 1 \) adversaries. However, like the \((F + 1, 1)\)-case, the \( r \) condition is satisfied and only the \( s \) condition is relaxed slightly. As before, we demonstrate a graph that maximizes the number of non-convergent nodes to demonstrate the lower bound on the resilience of the graph. For this, we start with the following construction.

**Lemma 3.3.** For given integers \( F \geq 3 \) and \( N \geq 3F \), the graph \( G = \mathcal{E}_{N-2F} \oplus \mathcal{C}_{2F}^{1,\ldots,F-1} \) is \((F + 1, F)\)-robust.

**Proof:** First, we will show the result for \( N = 3F \), and then extend the result to \( N > 3F \).
Assume $N = 3F$. Let $U$ and $V$ denote the set of nodes in $C_{2F}^{1,\ldots,F-1}$ and $E_F$, respectively. Let $S_1$ and $S_2$ be two disjoint non-empty sets of nodes in the given $G$. There are following cases for the choices of $S_1$ and $S_2$.

(a) At least one of $S_1$ and $S_2$ contains nodes from $V$ only: Without loss of generality, let $S_1 \cap U = \emptyset$ (i.e., $S_1 \subseteq V$). Then, each node in $S_1$ has $2F$ neighbors outside of $S_1$, and $X_{S_1}^{F+1} = S_1$.

(b) Both $S_1$ and $S_2$ contain nodes from $U$:

In this case, $S_1 \cap U \neq \emptyset$ and $S_2 \cap U \neq \emptyset$. Let $|S_1 \cap U| = \nu$.

Since each node in $U$ has a degree $2F - 2$, each $u \in (S_1 \cap U)$ has $(2F - 2) - (\nu - 1) = 2F - 1 - \nu$ neighbors in $U \setminus S_1$. Based on $\nu$, we have the following sub-cases.

(b-1) $\nu \leq F - 2$: In this case, for each $u \in S_1 \cap U$, the number of neighbors in $U \setminus S_1$ is:

$$2F - 1 - \nu \geq 2F - 1 - (F - 2) = F + 1,$$

which means $X_{S_1}^{F+1} \cap U = S_1 \cap U$. Similarly, since each $v \in S_1 \cap V$ is adjacent to all nodes in $U$, and $|U \setminus S_1| \geq F + 1$, we have $X_{S_1}^{F+1} \cap V = S_1 \cap V$. Thus, $X_{S_1}^{F+1} = (S_1 \cap V) \cup (S_1 \cap U) = S_1$.

(b-2) $\nu \geq F + 2$: This implies that $|S_2 \cap U| \leq F - 2$. As a result, we apply the sub-case (b-1) on $S_2$ and get $X_{S_2}^{F+1} = S_2$.

(b-3) $\nu = F - 1$: In this case, note that $|U \setminus S_1| = F + 1$. We have two scenarios: First, if $S_1 \cap V = V$ (i.e., $S_1$ contains all nodes in $V$), then each $v \in (S_1 \cap V)$ is adjacent to $F + 1$ nodes in $U \setminus S_1$. Thus, $(S_1 \cap V) \subseteq X_{S_1}^{F+1}$. Since $|S_1 \cap V| = |V| = F$, we have $X_{S_1}^{F+1} \geq F$. Second, if $(S_1 \cap V) \neq V$, then there is at least one node
By the same argument, we add $E_2$ circulant graph. This gives the graph $E$ add a new node $v$ which is $(S \cap U)$ non-convergent nodes there are in this construction. Thus, each $u \in (S \cap U)$ hast at least $F + 1$ neighbors outside $S_1$, implying $(S \cap U) \subseteq X_{S_1}^{F+1}$. Moreover, each $v \in (S \cap V)$ is adjacent to all nodes $U \setminus S_1$ (where $|U \setminus S_1| = F + 1$), thus $(S \cap V) \subseteq X_{S_1}^{F+1}$. As a result, we get $X_{S_1}^{F+1} = S_1$.

(b-4) $v = F$: Since $|U| = 2F$, we have $|S_2 \cap U| \leq F$. If $|S_2 \cap U| \leq F - 1$, we apply the argument in sub-case (b-3) above on $S_2$. So, consider $|S_2 \cap U| = F$. Now, since $|V| = F$, at least one of the following is true: (i) $|V \setminus S_1| \geq \lceil \frac{F}{2} \rceil$, (ii) $|V \setminus S_2| \geq \lceil \frac{F}{2} \rceil$. Without loss of generality, we assume (i) is true. It means that each $u \in (S_1 \cap U)$ has at least $\lceil \frac{F}{2} \rceil$ neighbors in $V \setminus S_1$. Also, each $u \in (S_1 \cap U)$ has $2F - 1 - F = F - 1$ neighbors in $U \setminus S_1$. Noting that $F \geq 3$, we deduce that each $u \in (S_1 \cap U)$ has at least $F + 1$ neighbors outside $S_1$. Since $|S_1 \cap U| = F$, we have $|X_{S_1}^{F+1}| \geq F$.

(b-5) $v = F + 1$: In this case $|S_2 \cap U| \leq F - 1$, thus, we apply the sub-case (b-3) on $S_2$.

All the above cases establish that the graph $E_F \oplus C_{2F}^{1,\ldots,F-1}$ is $(F + 1, F)$-robust. Now, we add a new node $v$ to $E_F \oplus C_{2F}^{1,\ldots,F-1}$ such that $v$ is adjacent to all nodes in $U$ (i.e., nodes in the circulant graph). This gives the graph $E_{F+1} \oplus C_{2F}^{1,\ldots,F-1}$. Since the new node $v$ is adjacent to $2F$ nodes in the existing graph, the $(F + 1, F)$-robustness of $E_F \oplus C_{2F}^{1,\ldots,F-1}$ implies that the new graph $E_{F+1} \oplus C_{2F}^{1,\ldots,F-1}$ is also $(F + 1, F)$-robust (by [LeBlanc et al. 2013, Theorem 5]).

By the same argument, we add $N - 3F$ vertices to $E_F \oplus C_{2F}^{1,\ldots,F-1}$ to get $G = E_{N-2F} \oplus C_{2F}^{1,\ldots,F-1}$, which is $(F + 1, F)$-robust.

Figure 3.5 demonstrates a graph, $G = E_6 \oplus C_6^{1,2}$, that is $(4,3)$-robust. In this graph, with any subsets, $S_1$ or $S_2$, if the nodes contain any nodes from $C_6^{1,2}$, then those nodes will be $r$-reachable, which makes the other nodes non-convergent. Next, we show how many non-convergent nodes there are in this construction.
Theorem 3.3. For given integers $F \geq 3$ and $N \geq 2F + 3$, there exists an $(F + 1, F)$-robust graph with $N$ nodes, of which $N - 2F$ are non-convergent under an $F$-total attack.

Proof: Consider $G = E_6 \oplus C_6^{1,2}$, which is $(F + 1, F)$-robust by Lemma 3.3. Let $V = \{v_1, \ldots, v_{F+2}\}$ be the set of nodes in $E_{N-2F}$, and $U = \{u_1, \ldots, u_{2F}\}$ be the set of nodes in $C_{2F}^{1,2\cdot\cdot\cdot,F-1}$. We will show that each $v_i \in V$ is a non-convergent node.

For this, first, we show that $G$ is not $(F+1, F+1)$-robust. Consider two nonempty disjoint subset of nodes in $G$. Let $S_1 = \{v_1\} \cup \{u_1, u_2, \ldots, u_F\}$, and $S_2$ be the set of remaining nodes, i.e., $S_2 = (V \cup U) \setminus S_1$. We now compute $X^{F+1}_{S_1}$ and $X^{F+1}_{S_2}$. Note that each $u_i \in S_1$ is adjacent to at least $2(F - 1) - (F - 1) = F - 1$ nodes in $U \setminus S_1$. Also, each $u_i \in S_1$ is adjacent to $(N - 2F) - 1 \geq F - 1$ nodes in $V \setminus S_1$. Thus, $u_i \in S_1$ is adjacent to at least $2(F - 1)$ nodes outside $S_1$. Since $F \geq 3$, we have $2(F - 1) \geq F + 1$, and $(S_1 \cap U) \subseteq X^{F+1}_{S_1}$. Moreover, $v_1 \in (S_1 \cap V)$ is adjacent to exactly $F$ nodes outside of $S_1$. Thus, $X^{F+1}_{S_1} = S_1 \cap U$, i.e., $|X^{F+1}_{S_1}| = F$. As for $S_2$, each $v_i \in (S_2 \cap V)$ is adjacent to only $F$ nodes outside $S_2$ (which are the nodes in $S_1 \cap U$). Further, each $u_j \in (S_2 \cap U)$ is adjacent to $2(F - 1)$ nodes in $U$, of which $F - 1$ nodes are in $S_2 \cap U$. Thus, each $u_j \in (S_2 \cap U)$ is adjacent to $F - 1$ nodes in $U \setminus S_2$. Also, each such $u_j$ is adjacent to one node in $V \setminus S_2$. Thus, each $u_j \in (S_2 \cap U)$ is adjacent to $(F - 1) + 1 = F$ nodes outside $S_2$, which means $X^{F+1}_{S_2} = \emptyset$. In other words, $|X^{F+1}_{S_1}| + |X^{F+1}_{S_2}| = F$, and $G$ is not $(F+1, F+1)$-robust.
Now, assign some real value $a \in \mathbb{R}$ to all nodes in $S_1$, and some value $b > a$ to nodes in $S_2$. Moreover, assume that nodes in $X^{F+1}_{S_1} = S_1 \cap U$ are malicious. Since $|S_1 \cap U| = F$, the number of malicious nodes is $F$. Note that all normal nodes in $S_1$ and $S_2$ have at most $F$ neighbors outside of their respective sets. Thus, following the W-MSR algorithm, each normal node in $S_1$ and $S_2$ removes $F$ values outside of its respective set, and hence never updates its value. In particular, $v_1 \in S_1$ does not converge to any normal node in $S_2 = (V \setminus \{v_1\}) \cup (U \setminus \{u_1, \ldots, u_F\})$. Now, by selecting $S_1 = \{v_1\} \cup \{u_{F+1}, \ldots, u_{2F}\}$ and $S_2 = (V \cup U) \setminus S_1$, we can ensure, by the same arguments as above, that there is an attack of $F$ nodes (i.e., $\{u_{F+1}, \ldots, u_{2F}\}$) preventing $v_1$ to converge to any of the (normal) nodes in $\{u_1, \ldots, u_F\}$. As a result, for each node $x \in (V \cup U) \setminus \{v_1\}$, there is an attack of $F$ nodes guaranteeing that $v_1$ and $x$ do not converge, implying that $v_1$ is a non-convergent node. Finally, noting the symmetry of nodes in $G$, we can replicate the same arguments as above to show that each $v_i \in V$ is non-convergent. Since $|V| = N - 2F$, we get the desired result.

This shows that the upper bound on the number of non-convergent nodes is $N - 2F$. Just like the previous two cases, as $N \to \infty$, then the local non-resilience approaches 1. As $F$ increases, then the number of non-convergent nodes decreases as well. When comparing this case to the $(F + 1, 1)$-robust case, the only thing that is different is the $s$ condition. However, we find that the upper bound on the non-convergent nodes is independent of $F$ in that case and in the $(F + 1, F)$-case it is dependent on $F$.

### 3.3 Illustrations and Simulations

Based on the previous examples, we can now state a sufficient condition for a node to be a non-convergent node. In Theorems 3.1, 3.2, and 3.3, the main idea to show that an $F$-total attack exists such that it prevents a normal node, $u$, from converging with another normal node, $v$, is as follows: First identify two disjoint subsets, $S_1$ and $S_2$, such that $u \in S_1$
and \( v \in S_2 \) or vice versa. Moreover, the two subsets do not satisfy any of the conditions as proposed in 2.5 such that an attack exists to prevent the two subsets from converging. Finally, we ensure that the neither \( u \) or \( v \) are \( F + 1 \)-reachable, such that those two nodes will not update their values and do need to be adversaries. Therefore, the \((r, s)\)-robustness conditions are not satisfied, there will exist less than \( F F + 1 \)-reachable nodes, or \( |X^{F+1}_{S_1}| + |X^{F+1}_{S_2}| \leq F \).

The nodes that are \( F + 1 \)-reachable can be set as adversaries and the two subsets will not converge. By leveraging this strategy, we can efficiently state a sufficient condition for a node to be a non-convergent node.

**Proposition 3.1.** A node \( u \in V \) in a graph \( G = (V, E) \) is non-convergent (under the \( F \)-total malicious attack) if for every \( v \in V \setminus \{u\} \), there exist a pair of non-empty disjoint subsets \( S_1, S_2 \subset V \) such that

1. \( |X^{F+1}_{S_1}| < |S_1| \), and \( |X^{F+1}_{S_2}| < |S_2| \), and \( |X^{F+1}_{S_1}| + |X^{F+1}_{S_2}| < F + 1 \), (i.e., \( S_1 \) and \( S_2 \) do not satisfy the \((F + 1, F + 1)\)-robustness criteria in Definition 2.5.)

2. \( u \) and \( v \) belong to distinct subsets, i.e., if \( u \in S_1 \), then \( v \in S_2 \) and vice versa.

3. Neither of \( u \) and \( v \) have \( F + 1 \) neighbors outside of their respective subsets.

We demonstrate this proposition through an example:

**Example:** Consider the graph in Figure 3.6 which is \((3, 3)\)-robust. Under the \( F \)-total malicious nodes model, where \( F = 3 \), nodes in \( \{v_2, v_3, v_6, v_7\} \) are non-convergent as they satisfy the conditions in Proposition 3.1. We explain the non-convergence of \( v_6 \). Consider a pair of subsets, \( S_1 = \{v_4, v_5, v_6\} \) and \( S_2 = \{v_1, v_2, v_3, v_7, v_8\} \) (as highlighted in Figure 3.6(a)). These subsets meet the first condition in Proposition 3.1. Notably, node \( v_5 \in S_1 \) is the only node with four \((F + 1 = 4)\) neighbors outside its subset \( S_1 \). Consequently, there exists an attack (involving \( v_5 \)) that prevents \( v_6 \) from converging to any of the nodes in \( S_2 \). For the non-convergence of \( v_6 \), we further need to show that there is also an attack that
prevents convergence of $v_6$ with nodes $v_4, v_5 \in S_1$. For this, consider subsets $S_1$ and $S_2$ in Figure 3.6(b), where $S_1 = \{v_1, v_6, v_7, v_8\}$ and $S_2 = \{v_2, v_3, v_4, v_5\}$. Note that $v_6$ is in a different subset than in $v_4$ and $v_5$, and none of these nodes have four neighbors outside their respective sets, thereby satisfying the conditions in the aforementioned proposition. As a result, we can guarantee that node $v_6$ does not converge to $v_4$ and $v_5$, thus confirming its non-convergence.

![Figure 3.6: G is (3,3)-robust and $v_6$ is non-convergent for $F = 3$.](image)

We now want to illustrate two main concepts about non-convergent nodes: 1) illustrate the notion of a non-convergent node and 2) demonstrate how the number of non-convergent nodes changes as we alter the graph’s robustness.

For an illustration of a non-convergent node, consider $G = (V, E)$ in Figure 3.7 which is (4,3)-robust (but not (4,4)-robust). Assuming $F = 3$ and $F$-total malicious attack, $G$ has four non-convergent nodes, $\{v_7, v_8, v_9, v_{10}\}$. For instance, considering $v_7$, we show that for every other $v_i \in V$, there is an attack consisting of $F = 3$ malicious nodes ensuring that $v_7$ and $v_i$ do not converge. In Figure 3.7(a), we design an attack involving $v_1, v_2$ and $v_3$. Their state trajectories are shown in red in Figure 3.7(c). The state of $v_7$ is in green, and the states of the remaining nodes are in blue. As a result of this attack, none of the nodes in $\{v_4, v_5, v_6, v_8, v_9, v_{10}\}$ and $v_7$ converge at the same state. Next, we need to show that there is an attack that can prevent $v_7$ from converging to any of the nodes in $\{v_1, v_2, v_3\}$. 

Figure 3.7(d) demonstrates such a situation. Hence, for every node pair \((v_7, v_i)\), we have an attack guaranteeing that \(v_i\) and \(v_7\) do not converge, establishing that \(v_7\) is a non-convergent node.

![Diagram of non-convergent nodes](image)

Figure 3.7: \(v_7\) (green) is a non-convergent node. For every node \(v_i \neq v_7\), there is an attack consisting of \(F = 3\) nodes (as in (c) and (d)) preventing \(v_7\) and \(v_i\) from converging to a common point.

Now we want to demonstrate how the number of non-convergent nodes changes as we alter the graph’s robustness. As per Theorems 3.1, 3.2, and 3.3, we found that the upper bound on the local non-resilience approaches 1. This means that almost all guarantees are lost when the graph is no longer \((F + 1, F + 1)\)-robust. However, what we now want to show that the number of non-convergent nodes does not only depend on the robustness of the graph, but also by the graph structure.
Figure 3.8 presents 3 sets of graphs, all of which are not (4, 4)-robust, with the assumption that $F = 3$. Consequentially, none of the graphs will be able to guarantee consensus under $F = 3$ total adversarial nodes. What we find is that every graph has a different amount of non-convergent nodes. In fact, even nodes with the same robustness have different amounts of non-convergent nodes.

For instance, Figure 3.8(a) shows three (3, 3)-robust graphs, each having a different number of non-convergent nodes (red) and hence, a varying level of non-resilience. Similarly, Figures 3.8(b) and 3.8(c) present examples of (4, 1)- and (4, 3)-robust graphs, respectively. Each of these graphs contains a distinct number of non-convergent nodes. This implies that just relying on the $(r, s)$-robustness metric is not enough to fully describe how a graph will behave under an $F$-total adversarial attack.

Figure 3.8: Non-convergent nodes (colored red) in (3, 3)-robust, (4, 1)-robust, and (4, 3)-robust graphs.
In addition to these specific examples, we generated fifty instances of random graphs with $N = 10$ and $14$ for three different levels of robustness: $(3,3)$-robust, $(4,1)$-robust, and $(4,3)$-robust. We calculated the expected number of non-convergent nodes for each level of robustness under the $F$-total malicious model with $F = 3$. The results are in Table 3.1.

Table 3.1: The number of non-convergent nodes in graphs with various robustness considering $F = 3$ malicious nodes.

<table>
<thead>
<tr>
<th>$N$</th>
<th># of non-convergent nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>(3,3)-robust 6.5 (4,1)-robust 3.8 (4,3)-robust 0.1</td>
</tr>
<tr>
<td>14</td>
<td>9.5 6.3 1</td>
</tr>
</tbody>
</table>

In general, we find the parameter, $r$, takes precedence in determining the relative robustness of the graph, where larger values of $r$ result in larger relative resilience. Additionally, we see that larger values of $s$ will also result in a larger relative resilience for the graph. Graphs with generally high robustness generally have fewer number of non-convergent nodes. These graphs were tested with $N = 10$ and $N = 14$. When the number of nodes increases, we find that the percentage difference in the graphs is also different. As the general robustness increases, the percentage increase with higher nodes seems to be larger as well. This might demonstrate that even though the graph is quite resilient at low numbers of nodes, it can increases with larger number of nodes.

Figure 3.9(a) shows a $(2,2)$-robust graph consisting of $N = 12$ nodes. The $(2,2)$-robustness implies that under a single malicious node, the graph will be guaranteed to converge. However, as we increase the number of malicious nodes, or $F$, the number of non-convergent nodes will also increase, as Figure 3.9(b) illustrates. This small example demonstrates the increasing non-resilience as the number of adversaries increases in a graph.

---

1Here, the robustness of each graph is maximal. We use maximal to mean the highest level of robustness that the graph can achieve without moving to the next level of robustness. For example, a $(3,3)$-robust graph considered is not $(4,1)$-robust. Similarly, the $(4,1)$-robust graphs considered are maximally robust in that they are not $(4,2)$-robust.
This new notion of non-convergence allows us to now explore and quantify the behavior of graphs when the \((F+1, F+1)\)-conditions have not been met. Graphs with the same robustness can have different amount of local non-resilience, and therefore, just \((r, s)\)-robustness is not accurate enough to identify the resilience of the graph. However, although we have identified and demonstrated that graphs can have different amount of non-converging nodes depending on the edge set, the cause for the non-converging nodes opens the door for new research. By understanding the causes for non-convergent nodes and their behavior under certain attacks, we can better understand how to utilize the notion of non-convergent nodes to build better systems, as will be explored in the next chapter.
CHAPTER 4
DISCUSSION AND CONCLUSION

We explored the implications of non-ideal robustness conditions on resilient consensus. By quantifying the impact of sub-par network robustness in terms of non-resilience, as measured by non-convergent nodes, our results indicated that even when the conditions are not fully satisfied, it remains possible to guarantee the convergence of specific graph portions. Thus, relying solely on robustness conditions becomes inadequate in describing the behavior of resilient consensus.

Our investigation has laid the foundation for further exploring this problem, revealing new research opportunities. This section delves into applying the concept of non-convergence and its potential for deeper exploration. Section 3.1 explores additional uses for understanding non-convergence, which reveals what we cannot guarantee. Additionally, Section 4.2 illustrates how the notion of non-convergence can be leveraged to ensure that some subgraph(s) converge, even when the overall graph fails to meet robustness conditions.

4.1 Expanding on Non-Convergence

Non-convergence is a novel concept that assesses the extent to which adversaries can disrupt a graph at a local scale. Each graph structure exhibits a distinct count of non-convergent nodes. While graphs with higher robustness generally tend to have a lower number of non-convergent nodes, it is worth noting that even within graphs with comparable robustness levels, the count of non-convergent nodes can vary.

In exploring the notion of non-convergent nodes, we made the assumption that nodes $u$ and $v$ will not converge if both belong to subsets $S_1$ and $S_2$ (that are not $(r, s)$-reachable for appropriate $r$ and $s$), respectively, and there exists an attack that prevents the convergence of these subsets, while neither $u$ nor $v$ are adversaries. Such an attack can be referred to as a ‘blockage attack’, as formally defined below:
Definition 4.1. (Blockage Attack) A blockage attack between two normal nodes, \( u \) and \( v \), occurs when \( u \) belongs to a subset of nodes \( S_1 \), and \( v \) belongs to a subset of nodes \( S_2 \), where \( S_1 \cap S_2 = \emptyset \), satisfying the condition \( |X_{S_1}^{F+1}| + |X_{S_2}^{F+1}| \leq F \). Additionally, all nodes in \( S_1 \) must have values strictly larger than those in \( S_2 \). For such a blockage attack to be valid, it must hold that \( u \) and \( v \) are not \( F+1 \)-reachable. As a result of this blockage attack, nodes in subsets \( S_1 \) and \( S_2 \) will fail to converge with each other, leading to the non-convergence of \( u \) and \( v \).

This type of attack has been explored throughout this thesis. To illustrate the functionality of the attack, assume such subsets exist. Without the loss of generality, assume all the nodes in \( S_1 \) have a value of \( a \) and all of the nodes in \( S_2 \) have a value of \( b \), where \( a > b \). Additionally, assume that all of the \( F+1 \)-reachable nodes are adversaries. Because the number of \( F+1 \)-reachable nodes is at most \( F \), then all of those nodes can become adversaries. Therefore, none of the normal nodes will update their values, and since \( a \neq b \), then node \( u \) and \( v \) will not converge.

However, there are other types of attacks that can prevent two nodes from converging. One such attack can exist such that both nodes \( u \) and \( v \) do not belong in \( S_1 \) or \( S_2 \), or \( u, v \in V \setminus (S_1 \cup S_2) \). Such attack can be called a separation attack as defined below:

Definition 4.2. (Separation Attack) A separation attack between two normal nodes, \( u \) and \( v \), is an attack where a subset of nodes, \( S_1 \) and \( S_2 \), where \( S_1 \cap S_2 = \emptyset \) exist, such that \( |X_{S_1}^{F+1}| + |X_{S_2}^{F+1}| \leq F \). Additionally, all nodes in \( S_1 \) have values strictly larger than \( S_2 \). However, \( u \) and \( v \) exist such that they have more than \( F+1 \) neighbors in both \( S_1 \) and \( S_2 \), and the steady state values of the \( N_u \) and \( N_v \) are different. In this case, the average that \( u \) and \( v \) converge to is different. This can be achieved where \( |S_1 \cap N_u| \neq |S_1 \cap N_v| \) and \( |S_2 \cap N_u| \neq |S_2 \cap N_v| \). This can be extended further by analyzing steady state values of other nodes affected by the separation attack, and making sure the steady state values of the neighborhoods of each node are different.
In a separation attack, both nodes \( u \) and \( v \) are able to converge, but because their neighborhood might not converge to a common point, they will converge to the mean of their neighborhood, which may be different for both \( u \) and \( v \). This type of attack can be demonstrated in Figure 4.1a and 4.1b later in this section. In the Figure, we see a graph with one adversary, \( F = 1 \). Node 5 is shown to not converge with the remainder of the nodes because nodes 1-4 do not converge with nodes 6 and 7, and node 5 has enough edges to both subsets to utilize in its calculation. In this scenario, it can be shown how an attack can be imposed to separate a normal node from another normal node, even if one of the nodes does not belong in one of the subsets, \( S_1 \) or \( S_2 \).

The main difference between a blockage attack and a separation attack is that in a blockage attack, both nodes, \( u \) and \( v \), need to belong in \( S_1 \) and \( S_2 \) respectively. In a separation attack, neither node needs to be in \( S_1 \) or \( S_2 \), but can exist in undefined nodes. Although this is a useful notion, it relies on the understanding of what the steady state values of the neighbors are for each node. Because an attack exists that can disrupt \( S_1 \) and \( S_2 \), then the nodes in the middle, even if they all satisfy \( F + 1 \)-reachable conditions with other subsets, might not converge because the trimmed averages of their neighbors may be different. The notion of non-convergence can help understand where those values end up and which nodes can be most non-resilient.

This notion of different types of attacks can be expanded to get a more complete picture of non-convergence. By understanding these new types of attacks, it may be possible to discover new methods for preventing nodes from converging, which can change a graph’s non-resilience.

In terms of defining non-convergence, like the previous notion of robustness, it is a ‘binary’ notion, i.e., a node is either non-convergent or not. This does not take into account the convergence with other nodes. The notion of non-convergence can be refined even further. Some points in this direction include:
Figure 4.1: Graph, $G$, which is not $(2, 2)$-robust, with $F = 1$ adversaries, but subgraph $\{u_1, \cdots, u_4\}$ which is subgraph $(2, 2)$-robust with adversary at node 6. (a) depicts the graph and (b) plots the convergence in time.

1. Change the value of non-convergence on a scale from 0 to 1. This can provide valuable insights into the graph, as some nodes can be more non-resilient than others. Additionally, this can allow for deeper insight into how the graphical structure and parameters can affect the non-convergence of a node. Although this can be a useful parameter for analyzing graphs, the computational complexity will significantly increase.
2. Apply non-convergence to various attack types. Since the graph is no longer \((F+1, F+1)\)-robust, new types of attacks can prevent nodes from converging. We explored one type of attack, where node \(u\) and \(v\) belong to the non-converging subsets. Providing an understanding of non-convergence through different types of attacks will allow us to understand the most vulnerable parts of the system.

3. Exploring the number of simultaneous non-convergence. In the current state, when looking at non-convergence, we analyze the prevention of only a single node \(u\) from converging with another node in the graph. However, often, an adversary will try to prevent multiple nodes from converging at the same time. This poses a new question: \textit{How many nodes can be non-convergent simultaneously?}. This can show the relationship between the resilience of nodes and how much can be lost from a global perspective.

4. Understanding relation of structure with non-convergence. Non-convergent nodes depend on the graphical structure. By understanding what structures make a node less resilient, we can develop methods for creating more resilient graphs.

Additionally, we studied the non-convergence primarily in the context of an \(F\)-total adversarial model, where each node implements the W-MSR algorithm. This process can now be applied to different types of adversarial models and different algorithms to update their parameters. This can also be applied to other resilient distributed problems other than consensus. By exploring this notion of non-convergent nodes, we can better understand our resilient algorithms.

\subsection*{4.2 Subgraph Convergence}

Previously, we focused on the negative result of non-convergence, or \textit{what is lost when resilience conditions are not met?}. Another direction that remains to be explored is \textit{what can still be preserved when resilience conditions are not met?}
Instead of focusing on normal nodes that cannot guarantee convergence with any other
normal node in the graph, we can now focus on nodes that are guaranteed to converge with
other nodes in a graph. If two nodes are guaranteed to converge, they can belong in what's
called a converging subgraph.

When the graph, \( G \), is \((F + 1, F + 1)\)-robust, then we know that every normal node in
\( G \) will be guaranteed to converge when implementing the W-MSR algorithm. However, if
\( G \) is not \((F + 1, F + 1)\)-robust, but an induced subgraph, \( H \), meets a certain robustness
parameter, then we could say that the subgraph will converge. This is important as it can
be used to analyze what can be preserved under an \( F \)-total attack model.

**Definition 4.3.** A converging subgraph, \( H \subseteq G \), is a converging subgraph if the nodes
converge under resilient distributed consensus, as in Definition 2.2.

To analyze a converging subgraph, we need to be able to prevent attacks on the subgraph.
The two types of attacks to look at include the blockage attack (as in Definition 4.1) and
the separation attack (as in Definition 4.2). To prevent a blockage attack, we need to ensure
that all of the disjoint subsets \( S_1 \) and \( S_2 \) within the subgraph are \((F + 1, F + 1)\)-robust. To
prevent separation attacks, we need to ensure we do not have as much resilience with the
nodes outside of the converging subset, as their dynamics are still unknown. This creates an
interesting issue—we need to maximize the resilience within the subgraph and minimize the
resilience with nodes outside of the subgraph. In order to do so, new definitions of robustness
involving other subsets are needed.

**Definition 4.4. (Subset r-Reachable Node)** Given a graph, \( G = (V, E) \), two disjoint
subsets of nodes, \( S_1, S_2 \subseteq V \), and a positive integer \( r \), a node, \( u \in S_1 \), is subset r-reachable
if the number of neighbors \( u \) has inside of \( S_2 \) is at least \( r \), or \( |N_u \cap S_2| \geq r \).

**Definition 4.5. (r-Reachable Set)** Given a graph, \( G = (V, E) \), and two disjoint subsets
of nodes, \( S_1, S_2 \subseteq V \), and a positive integer \( r \), a set of nodes of nodes are subset r-reachable,
\(X_{S_1,S_2}^r\), if at least one node is subset \(r\)-reachable in \(S_1\) to \(S_2\). In other words:

\[
X_{S_1,S_2}^r = u \in S_1 : |\mathcal{N}_u \cap S_2| \geq r
\]  

(4.1)

The next thing that needs to be addressed is the dynamics of the nodes outside of the subgraph. The most important nodes are the ones that are adjacent to the nodes within the subgraph, which we call \textit{boundary nodes}. Assuming that a node within the converging subgraph has \(b\) boundary nodes, then it requires \(F + 1 - b\) adversaries to make the node want to converge to outside of the subgraph. Now because \(b\) can be greater than 1, that means that \(F + 1 - b \leq F + 1\). Therefore, there can exist enough malicious nodes to want to converge to the boundary nodes. In order to analyze further, boundary nodes, sets, and degree are defined below.

**Definition 4.6. (Boundary Node)** A node, \(u\), is a boundary node of a subset \(S \cap H\) if it has a neighbor with a node in \(S \cap H\).

**Definition 4.7. (Boundary Set)** A boundary set, \(B\), of a subset, \(S\), and subgraph \(H\), is a set of nodes that exist in a converging subset, where one of the nodes is a boundary node of \(S \cap H\). The set of boundary sets, \(\mathcal{B}\), for a subset, \(S \cap H\), is the set of unique boundary sets of \(S \cap H\), or \(\mathcal{B} = \{B_1, B_2, \cdots, B_3\}\).

**Definition 4.8. (Boundary Degree)** The boundary degree, \(b\), of a subset, \(S\), is the maximum reachability of all nodes in \(S \cap H\), or \(b = \max_u |\mathcal{N}_u \setminus (S \cap H)|\)

It is possible that nodes outside of the converging subgraph do not converge, and if two nodes have boundary nodes to other nodes that do not converge, then those nodes will not converge as well. To illustrate this, suppose \(H\) is a converging subgraph, with nodes \(u\) and \(v\), such that \(u, v \in V_H\). Additionally, assume that \(u\) has \(b_u\) boundary nodes and \(v\) has \(b_v\) boundary nodes. Suppose the boundary nodes \(\mathcal{N}_u \cap (G \setminus H)\) and \(\mathcal{N}_v \cap (G \setminus H)\) are non-convergent, such that an attack exists to prevent them from converging. To implement such
attack, assume that $F - f$ nodes are required, therefore only $f$ adversarial nodes remain. If $b_u + b_v + f \leq 2(F + 1)$, then there can exist a separation attack on the nodes $u$ and $v$.

One way to potentially guarantee the convergence of the subgraph, $H$, is to analyze the robustness of the subgraph. In order for $H$ to converge, we must prevent blockage attacks and separation attacks. As stated before, to prevent blockage attacks, we can approach this by making sure that $H$ is $(F + 1, F + 1)$-robust. To prevent separation attacks, we need to analyze the graph in entirety. By utilizing the $(F + 1, s_0)$-robustness of $G$, we can see that some attack can exist such that it prevents subsets $S_1$ and $S_2$ from converging since it provides the minimum number of adversaries required to prevent two disjoint subsets from having $F + 1$-reachable normal nodes. The amount of adversaries required to implement such an attack is $s_0$. This leaves a remainder of $F - s_0$ to be able to disrupt the $H$ in the worst case scenario. To be able to prevent an attack from disrupting the convergence of $H$, we can use these ideas to formulate a definition for the convergence of a subgraph:

**Definition 4.9. (Subgraph $(r,s)$-Robust)** Given a graph, $G = (V, E)$, of $(r_0, s_0)$-robustness, a subgraph $H = (V_H, E_H)$, where $V_H \subset V$ and $E_H \subset E$, positive integers $r$ and $s$, and $f = \max(0, s - s_0 - 1)$, $H$ is considered to be subgraph $(r,s)$-robust if for every pair of disjoint $S_1, S_2 \subset V_H$ and their respective boundary sets, $B_1$ and $B_2$, and respective boundary degrees, $b_1$ and $b_2$ one of the following holds true:

(i) $|X_{S_1 \cap H \setminus S_1}^r| = |S_1 \cap H|$ and $b_1 \leq f + 1$ (i.e., all nodes in $S_1 \cap H$ have at least $r$ neighbors in $H \setminus S_1$ and none have more than $f + 1$ edges outside of $H$).

(ii) $|X_{S_2 \cap H \setminus S_2}^r| = |S_2 \cap H|$ and $b_2 \leq f + 1$ (i.e., all nodes in $S_2 \cap H$ have at least $r$ neighbors in $H \setminus S_1$ and none have more than $f + 1$ edges outside of $H$).

(iii) $|X_{S_1 \cap H \setminus S_1}^r| + |X_{S_2 \cap H \setminus S_2}^r| \geq s$, $|X_{S_1 \cap H \setminus (H \setminus S_1) \cap B_2}^r| \geq f + 1 - b_2$, $|X_{S_2 \cap H \setminus (H \setminus S_2) \cap B_1}^r| \geq f + 1 - b_1$, and $b_1 + b_2 + f \leq 2r$ (i.e., the number of nodes between $S_1 \cap H$ and $S_2 \cap H$...
that have r-neighbors outside of their respective subset in \( H \) is at least \( s \) and outside of respective subset in \( H \cap B_n \) is \( f + 1 - b_n \).

This definition of subgraph robustness not only creates a way for us to analyze the robustness within the subgraph, but it also allows us to analyze the relationship with the other nodes outside of the subgraph. These conditions are still pretty restrictive, but by understanding the dynamics of boundary nodes, they can be reduced significantly. Although we have not proven that the subgraph will be guaranteed to converge under these conditions, we can state the following: Given a graph, \( G = (V, E) \), with \((F + 1, s_0)\) robustness and a subgraph, \( H = (V_H, E_H) \), where \( V_H \subset V \) and \( E_H \subset E \), an attack can exist to prevent \( H \) from converging if \( H \) is not subgraph \((F + 1, F + 1)\)-robust.

One of the major insights this construction gives is that even when the induced subgraph is \((F + 1, F + 1)\)-robust, it still does not guarantee that the graph will converge. This can be demonstrated by the following example:

Example: This example demonstrates how the induced subgraph can be \((F + 1, F + 1)\)-robust, but not subgraph \((F + 1, F + 1)\)-robust. In Figure 4.1a previously shown, if we take the graph induced by nodes 1-6, that induced subgraph is \((2, 2)\)-robust. This means that it can handle one adversary and still be guaranteed to converge. Now, we add the addition of node 7 to the graph, which has a single edge with node 5. In this case, because every node is updating their values by removing one largest node and one smallest node, node 7 will not update its value, and the graph is no longer \((2, 2)\)-robust. Now, to demonstrate how the subgraph is no longer \((2, 2)\)-robust, assume node 6 is our adversary. Give the nodes 1-5 a value of \( a \) and nodes 6 and 7 a value of \( b \), where \( a > b \). Node 6 and 7 will not update their values, however, node 5 will begin to decrease its value because it has more than one extreme neighbor. As it decreases its value, nodes 1-4 will not update their value as they each only have one neighbor with an extreme value. Therefore, node 5 will no longer be converged with nodes 1-4. Even though the subgraph, with nodes 1-6 is not subgraph \((2, 2)\)-robust,
the subgraph with nodes 1-4 is subgraph (2, 2)-robust, and will always converge under one adversary. If the adversary is in any other location, the nodes will be guaranteed to converge, regardless of the convergence of the remainder of the nodes. Figure 4.1b shows how when the malicious node is present, node 5 does not converge with nodes 1-4.

By understanding the dynamics of non-convergent nodes outside of the converging subgraphs, we can design distributed consensus algorithms with improved resilience. One promising approach involves adopting a heterogeneous strategy, wherein nodes implement different algorithms for updating their states based on the specific structures they are part of. For example, if a set of nodes belongs to a highly resilient converging subgraph (with \((r, s)\)-robustness such that \(r, s > F + 1\)), we can optimize their update step in the W-MSR algorithm 2.2 by increasing the number of values they retain from their neighborhoods. This ensures node convergence while mitigating vulnerability to separation attacks. The heterogeneous approach’s adaptability and robustness make it well-suited for various distributed optimization setups, extending its benefits beyond the consensus problem.

4.3 Conclusion

The findings of this research contribute to our understanding of the effects of network structures with sub-par robustness on the performance of resilient distributed algorithms, particularly the resilient distributed consensus. Our results demonstrate a crucial insight that graphs with identical robustness can exhibit varying levels of non-resilience, as measured by the number of non-convergent nodes. Hence, relying solely on network robustness to quantify the performance of resilient algorithms in the face of adversarial and misbehaving agents is inadequate.

Some of the proposed framework’s limitations include the computation of non-convergent nodes that demands further investigation and development. Overcoming this limitation presents an opportunity to enhance the efficacy and efficiency of resilient algorithms, leading
to more robust and adaptable systems that can thrive even in adverse conditions. Furthermore, understanding the dynamics of non-convergent nodes at a system level can provide invaluable insights into the behavior of resilient algorithms when they encounter non-ideal conditions. This knowledge can be instrumental in predicting the performance of algorithms and systems in real-world scenarios where resilience may be compromised due to various factors. By gaining this understanding, we can make informed decisions to improve the performance of distributed systems even in challenging environments.

Addressing the identified limitations and further exploring the dynamics of non-convergent nodes becomes a promising avenue for future research. By tackling these challenges, we can refine and advance the state-of-the-art in resilient algorithms, leading to more adaptable and reliable systems that can better withstand the complexities of distributed environments. It is worth emphasizing that the methods proposed in this study are primarily tailored to the W-MSR algorithm. However, the implications of this research extend far beyond the confines of a single algorithm and consensus problem. The conceptual framework established here has the potential to be adapted and applied to a broader set of algorithms and distributed systems, making it a valuable resource for researchers and practitioners in the field.

The practical applications of this work are far-reaching, particularly concerning engineering constraints in network systems. Engineers can make informed decisions and design more resilient and dependable systems by analyzing network performance under sub-optimal conditions. This can positively impact various industries, including telecommunications, transportation systems, and smart grids. Moreover, thoroughly evaluating distributed structures and algorithms this research facilitates can significantly contribute to advancing distributed system design. By understanding the intricacies of algorithmic behavior, designers can tailor systems to suit specific use cases better, optimizing efficiency and reliability. This research deepens our understanding of non-convergent nodes and resilient algorithms and sets the stage for future investigations that will push the boundaries of this field. By addressing the
identified challenges and expanding the scope of application, we can unlock the full potential of resilient algorithms, empowering our networks and systems to operate with improved efficiency and adaptability in the face of adversity.
REFERENCES


BIOGRAPHICAL SKETCH

Leon Khalyavin is a master’s degree student at the Systems Engineering Department at UT Dallas. Before that, he started his education at UT Dallas with a Bachelor’s in Electrical Engineering in Fall 2019. During his degree, he began his academic work with Dr. Jae Mo Park in signal processing for MRI applications. His major contributions include the restoration process for time domain corrupted FID signals, resulting in abstract publications in the *International Society for Magnetic Resonance in Medicine Journal* in 2022 and presentations during the conference. He later continued that work as a research assistant at UT Southwestern Medical Center. Additionally, during his studies, he has worked as a test development engineering intern and research intern at Ericsson. One of the projects he had worked on was a project for the Department of Defense on radar detection.

He also was a part of the UT Dallas IEEE chapter. During his time there, he acted as the Head of Forge, in which he organized semester-long engineering projects on campus, with a focus on electrical engineering. In addition, he worked as a tutor for international high-school students, where he tutored subjects such as college level calculus, statistics, physics, and chemistry.

During his bachelor’s degree, he got admitted into a fast track program at UT Dallas for a master’s in electrical engineering, with a concentration in signals and systems. He focused his education in control systems, signal processing, and machine learning. After completion of his master’s in electrical engineering in Spring 2022, he continued his education and receive a master’s in systems engineering from UT Dallas as well. While in the department, he had the opportunity to work as a research assistant at UT Dallas. His research work has focused on the resilience of network systems, which was funded by Amazon Robotics. He was also a teaching assistant for the graduate Linear Systems course, taught by Dr. Waseem Abbas.
After his master’s degree, he has been admitted and will attend Imperial College London for a PhD in Electrical and Electronics Engineering with Dr. Thomas Parisini. His research interests lie in control systems, resilience, distributed systems, and optimization. After his PhD, he hopes to continue a career in academia with research in control systems.
CURRICULUM VITAE

Leon Khalyavin
Lsk180000@utdallas.edu | lsk223@ic.ac.uk

Education
The University of Texas at Dallas
Master of Science in Systems Engineering; GPA: 3.9/4.0 Expected Aug. 2023
- Master’s Thesis: Non-Resilience in Distributed Networks under Adversarial Model (August 2023)
Master of Science in Electrical Engineering; GPA: 3.9/4.0 May 2023
Bachelor of Science in Electrical Engineering; GPA: 3.88/4.0; Magna Cum Laude May 2022

Academic Experience
CIRENs Lab – The University of Texas at Dallas
Graduate Research Assistant/Teaching Assistant June 2022 – Present
- Characterizing the non-resilience of robust distributed consensus algorithms under adversarial models, with focus on WMSR
- Developing novel methods for characterizing resilience in distributed networks, with a focus on network topology
- Submitted research on non-resilience for publication to IEEE Conference on Decision and Control
- Teaching assistant for graduate linear systems course, marking papers and creating course content and lecturing, with focus on exam preparation
- Participated in UT Dallas outreach program by presenting about engineering to underprivileged students in secondary school in the Dallas area

Park Lab – The University of Texas Southwestern Medical Center
Research Assistant May 2021 - Present
- Developed MRI data pipeline for image reconstruction using novel algorithms
- Designed FID restoration algorithm for signals corrupted with signal overflow and am currently finishing paper on algorithm for publication
- Primary author and co-author on two abstracts accepted into The International Society for Magnetic Resonance in Medicine (ISMRM) 2022 and had a virtual poster presentation at hybrid conference

Erik Johnson School of Engineering – The University of Texas at Dallas
- Created lesson plans and led lessons in the course Introduction to Engineering and Computer Science
- Graded weekly assignments and papers submitted by students and wrote constructive feedback for each assignment

Professional Experience


Ericsson 5G Smart Factory  
**Lewisville, TX**  
*Engineering Intern*  
May 2021 – Aug. 2021; March 2022 – Aug. 2022  
**Tester Fault Detection Automation**  
- Improved tester reliability by reducing variance over 85% with performing Gage R&R analysis, and fixing the faulty test points and reduced test time to 5 minutes by predicting component failure  
- Created performance report using Minitab for analysis and data visualization and presented to factory leadership team  

**MATLAB Interface for Test Equipment**  
- Developed MATLAB library, creating a user friendly and intuitive interaction with test equipment  
- Standardized MATLAB interaction with testers by creating a GUI application and creating documentation for library  
- Created and taught courses on MATLAB and the library developed for a team of 20 senior engineers  

**Operator Performance Data Analysis**  
- Created questionnaire to collect data on operators to produce data driven insights on individual performance  
- Analyzed performance data Azure and SQL to increase operator efficiency by about 30%  

Ericsson Research/Department of Defense  
**Santa Clara, CA**  
*Research Intern*  
Aug 2021 – March 2022  
- Researched radar detection algorithms for military applications, particularly the efficacy of the DFS algorithm  
- Modeled algorithm and created a simulation in MATLAB, to study performance properties (false alarm, SNR, etc.)  
- Co-authored an internal paper over the performance of the algorithm for military applications  

The University of Texas at Dallas IEEE Chapter  
**Richardson, TX**  
*Officer/Head of Forge*  
Nov. 2020 – Dec. 2021  
- Managed Forge division of IEEE by helping create student-led projects: including robotics outreach, facial recognition research, Discord bot, and more.  
- Increased funding for division by $30,000 by utilizing effective communication skills with donors  
- Expanded Forge division over 230% using outreach, showcases, and quality performance  
- Led guest workshop on project management and system level approach  

**Technical Skills**  
- **Technology/Applications:** MATLAB, Simulink, Python (TensorFlow, PyTorch, NumPy, Pandas, Scikit, NetworkX), C, C++, C#, Git, Linux, LaTeX  
- **Theory:** Linear/Nonlinear Systems, Stochastic Systems, Distributed Systems, Hybrid Systems, Feedback Control, Optimal Control, Dynamic Programming, Convex Optimization, Bayesian Optimization, Deep Neural Networks  

**Publications and Presentations**  
- Leon Khalyavin and Waseem Abbas. “On the Non-resilience of Resilient Distributed Consensus in Multiagent Networks”; IEEE Transactions on Control of Network Systems; (submitted)
• **Leon Khalyavin** and Jae Mo Park. “Reconstruction of 13C Spectroscopy with Signal Overflow”; The International Society for Magnetic Resonance in Medicine (ISMRM) 2022; Virtual poster presentation at hybrid conference

• Sung-Han Lin, Junjie Ma, **Leon Khalyavin**, and Jae Mo Park. “Flexible FOV image reconstruction using multi-echo 13C imaging with rotating spiral arms”; The International Society for Magnetic Resonance in Medicine (ISMRM) 2022

• Ericsson Internal Classified Publication; 2022

**Awards and Recognitions**

• Bachelor of Electrical Engineering Magna Cum Laude – 2022

• The International Society of Magnetic Resonance in Medicine (ISMRM) Educational Stipend Program – 2022

• 2nd Place UT Dallas Senior Design Capstone – 2021

• Academic Excellence Scholarship - 2019