2023

Distributed Optimization with Limited Communication in Networks with Adversaries

Iyanuoluwa Emiola
University of Central Florida

DISTRIBUTED OPTIMIZATION WITH LIMITED COMMUNICATION IN NETWORKS WITH ADVERSARIES

by

IYANUOLUWA EMIOLA
M.S. Delaware State University, 2018

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando Florida

Summer Term
2023

Major Professor: Chinwendu Enyioha

ABSTRACT

We all hope for the best but sometimes, one must plan for ways of dealing with the worst-case scenarios, especially in a network with adversaries. This dissertation illustrates a detailed description of distributed optimization algorithms over a network of agents, in which some agents are adversarial. The model considered is such that adversarial agents act to subvert the objective of the network. The algorithms presented in this dissertation are solved via gradient-based distributed optimization algorithm and the effects of the adversarial agents on the convergence of the algorithm to the optimal solution are characterized. The analyses presented establish conditions under which the adversarial agents have enough information to obstruct convergence to the optimal solution by the non-adversarial agents. The adversarial agents act by using up network bandwidth, forcing the communication of the non-adversarial agents to be constrained. A distributed gradient-based optimization algorithm is explored in which the non-adversarial agents exchange quantized information with one another using fixed and adaptive quantization scheme. Additionally, convergence of the solution to a neighborhood of the optimal solution is proved in the communication-constrained environment amidst the presence of adversarial agents.

This dissertation is dedicated to my late mother, Victoria Yetunde Emiola (1958 - 2004) who passed away on August 19, 2004. RIP my sweet mother.

ACKNOWLEDGMENTS

I acknowledge those who have supported me through this doctoral journey. I would like to thank my advisor, Dr. Chinwendu Enyioha for his mentorship and guidance. I started the doctoral program in Electrical Engineering less than four years ago and I have had the opportunity to publish and present my work at selective venues including conferences and a journal. All of these were made possible by my advisor and I appreciate the extra hours he spent with me to ensure I am successful at UCF.

To all of the professors and staffs - including Dr. Andriy Semichaevsky, Dr. John Chikwem, Late Dr. Levi Nwachuku, Mrs Ugochi Nwachuku, Dr. Dawn Lott, Dr. Onur Yavuz and others - who have taught and mentored me at Lincoln University of Pennsylvania, Delaware State University and University of Central Florida, I thank you.

I also want to use the opportunity to appreciate the rest of the dissertation committee members:Dr. Atia, Dr. Rahnavard, Dr. Qu and Dr. Vela for their guidance and constructive feedback. I am grateful to my lab colleague Diego Benalcazar, friends: Robson Adem, John Erhabor, Adesoji Aliu, Philip Bissiwu, Greg Fritjofson, UCF African Graduate Students Association members and others that have encouraged and spurred my career pursuit.

Most importantly, I am grateful to my father, Babafemi Emmanuel Emiola and my sisters Tomilola Emiola and Yejide Emiola for their support and motivation even during some difficult times. You all have been amazing and this accomplishment would not have been possible without you.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Anyone is prone to experiencing adversarial attack whether through phishing, cyber attacks, fraud or in other scenarios. These malicious acts are meant to deceive or prevent an individual from accomplishing his desired goals. In a network of autonomous systems, agents in the network may act in a malicious or adversarial way either because they are faulty, or have been compromised and are being used as stooges for an undesired goal in the network. Optimization problems with malicious attack is applicable in adversarial machine learning where an adversary can attack a machine learning model at the training or testing stage by injecting false data or manipulating data in the dataset. Another application is seen in Wireless Sensor Networks where adversarial nodes disrupt non-adversarial nodes by injecting false data as seen in Figure 1.1. In such attacks, non-adversarial agents in the network sometimes consider preventive mechanisms which leads to solving optimization algorithms. An optimization problem can either be constrained or unconstrained depending on the nature of the attack or problem.

Solutions to unconstrained optimization problems can be applied to multi-agent systems and machine learning problems, especially if the problem is in a decentralized or distributed fashion [1–6]. Sometimes, in adversarial attack applications, malicious agents can be present in a network that will slow down convergence rates to optimal points as seen in [7–9]. To address these lapses, we present and analyze the performance of a distributed optimization problem in a network containing adversarial nodes. To solve an optimization problem over a network of agents in a distributed manner, gradient-based methods alongside an agreement update step are commonly used [10–14]. In the process, agents iteratively update their estimates and exchanges it with neighboring nodes. This well-studied process arrives at the optimal solution depending on certain assumptions made on the cost function being optimized and the choice of the step-size. In typical formulations of

distributed optimization problems, the objective can be considered decomposable as

$$\min_{x \in \mathbb{R}^p} f(x) = \sum_{i=1}^{n} f_i(x_i), \tag{1.1}$$

where n is the total number of agents, $f_i(\cdot)$ is the local objective function of agent i, x_i is the decision variable of each agent and $f(x)$ is the global objective function that is meant to be optimized. Each agent i will also optimize its local objective function $f_i(x_i)$ and iteratively exchanges its decision variable x_i with neighboring agents over a communication network. In the presence of malicious agents, however, the local and consensus computations are altered and sometimes disrupts the performance of these distributed optimization algorithms. When such misbehavior occur in a network with adversaries, one has to either detect these threats or take any needed steps to mitigate their effects.

An illustration of such a preventive strategy is seen in [15–17], where an attempt is made to detect sensitive malicious insider threats. Another application where non-adversarial agents in a network can use preventive mechanisms against adversarial agents is seen in underwater communication systems [18] where the magnitude of information transmitted is very important. Underwater communication systems face challenges due to time variation of communication channels and system complexities. To reduce such bottlenecks, the number of bits of information transmitted has to be taken into huge consideration. Some methods of solving communication-constrained problems as described above include event-triggered control, sparsification and quantization (the crux of this dissertation). In an event-triggered control, communication occurs when a necessary event is prompted and the relevant agents estimates are chosen in the analysis. Sparsification involves either dropping out some agents state or making them zero. Quantization involves limited communications amongst agents using a type of quantizer that depends on parameters such as quantization interval, step and the number of bits. To determine the number of bits of information

needed, we let x_i be the state of each agent i. If $\boldsymbol{x}_i \in \mathbb{R}^p$, agents can transmit up to p floating point numbers and exchange $x_i(k+1) = p(x_i(k), x_j \ldots.)$. Using an inductive approach, the computations needed for 1 time step is initially performed and an iterative approach is employed in a manner where the order on the number of bits is $O(\frac{1}{\epsilon}) * n * p$ for ϵ accuracy and $O(\frac{1}{\epsilon})$ iterations. In the context of the dissertation, it is assumed that adversarial agents use up communication bandwidth and non-adversarial agents have to manage what is left via quantization. Results are corroborated with numerical experiments to show the applications of the proposed methodologies.

Research Impact: This dissertation research will have substantial merit and national importance both theoretically and technologically especially in the field of optimization. Due to the intensive computations involved in large scale distributed optimization algorithms, a major focus of research has been on improving the convergence, coordination and computation costs. Therefore, the goal is to obtain an optimization algorithm that can converge very fast with little cost. This dissertation is also relevant in adversarial distributed optimization as it offers preventive mechanisms in dealing with adversarial attack in a communication-constrained environment.

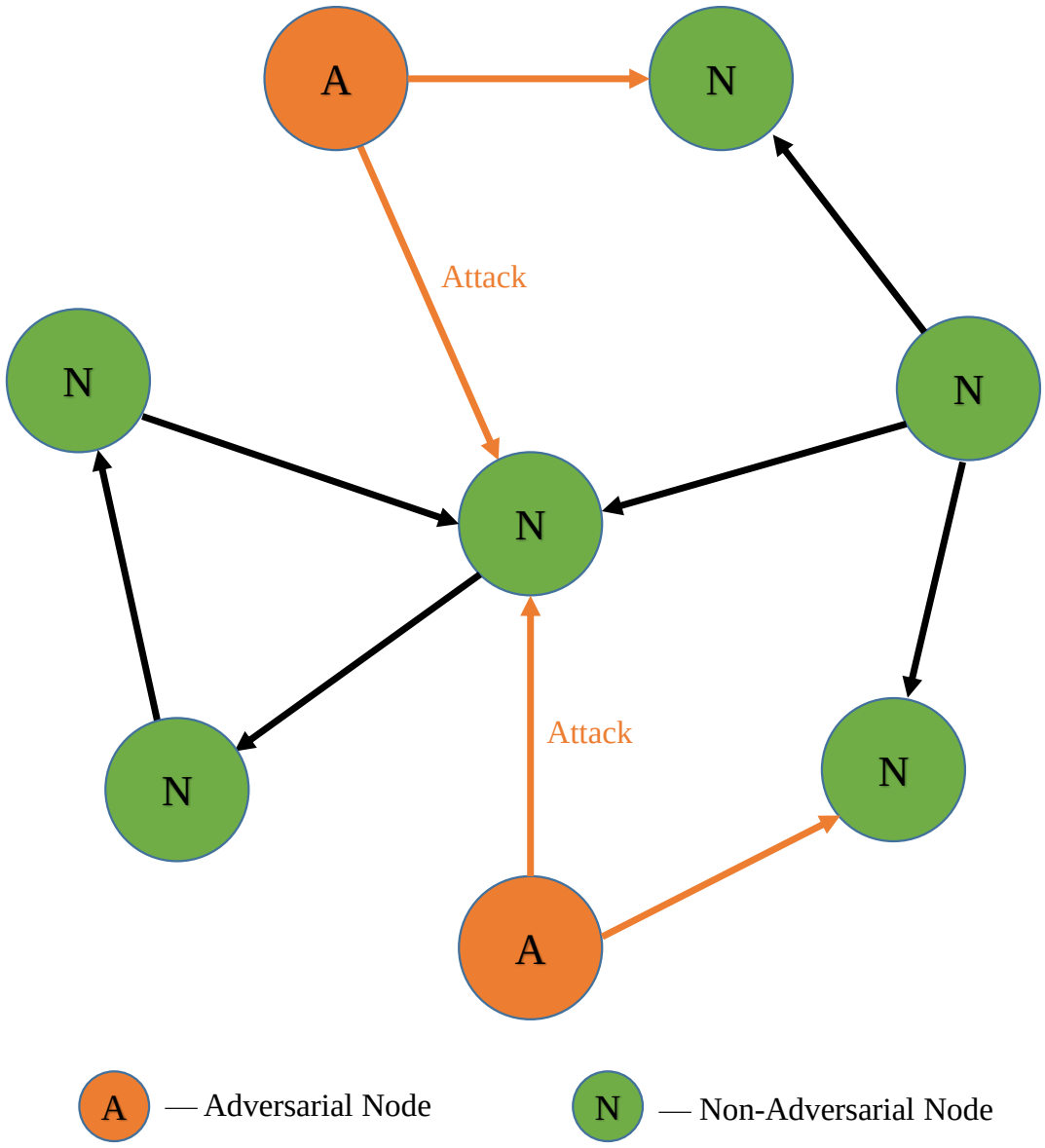

Figure 1.1: In the Network above, adversarial nodes disrupt non-adversarial nodes by injecting attack or false data.

CHAPTER 2: LITERATURE REVIEW

Different approaches have been taken to solve distributed optimization problems when adversarial nodes are present. An example is the approach taken in [9] where the author uses the Fast Row-stochastic Optimization with uncoordinated Step-sizes (FROST) algorithm that does not require the nodes to compute step sizes. The authors in [9] also considers the bounds on a parameter and a gradient bound to show the strength of the attack, though an explicit characterization of the extent to which the perturbed parameter alters and prevents convergence is not presented. Another method is the topological approach in tolerating malicious nodes shown in [19] where the author examines the conditions under which a malicious agent can be identified based on the topology and the size of the network. Some approaches to decision problems in the presence of adversaries often assume certain so-called 'trusted' agents cannot be compromised, and use information being shared by those agents as a benchmark to identify and exclude malicious information; thus, building in resilience to their optimization algorithm [16, 17, 20, 21]. Similarly flavored problems have been studied in the context of state estimation where methods to identify and extract malicious information are proposed [22, 23]. Other related adversarial problems like [24–31] explore the detection of attacks on distributed systems and protection strategies. This dissertation involves exploring the performance of distributed optimization algorithms under adversarial attack in a communication-constrained environment.

An important aspect of this dissertation is to show the conditions under which adversarial agents in a network can obstruct the non-adversarial agents from obtaining their optimal point. When the objective function is strongly convex, with Lipschitz continuous gradients, convergence to the optimal solution using gradient-based methods can be achieved at a linear rate [32–34]. It is also known that noisy communication channels and adversarial nodes in the network can slow down the linear convergence rate [35, 36]. In spatially distributed systems with band-limited communication

channels, where agents may resort to exchanging low-bit (quantized) information at each time-step, the resulting convergence rate is also impacted. Researchers have explored how quantization affects the performance of these distributed optimization algorithms [37–45].

In this dissertation, it is assumed that malicious nodes not only send corrupted information to neighboring agents, but also hog the available communication bandwidth. This results in the need for the non-malicoius nodes to manage the limited bandwidth available by quantizing the iterates broadcast to neighboring nodes. To solve this, a distributed gradient with quantized and adversarial attack method (DISGAQAAM) is proposed where non-adversarial agents send quantized information and adversarial agents send perturbed estimates using an attack vector. The results on adversarial attack obtained in this dissertation focuses on detection of attacks, resilience against attacks and the convergence attributes of distributed optimization models under adversarial attack in a limited communication environment and differ from those in [9,19,22] where the authors focus mostly on detection and preventive strategies in dealing with adversarial agents.

The attack model is such that the adversarial agents consume communication bandwidths available in a bid to obstruct the non-adversarial agents from reaching the optimal solution of the network objective. To cope with hogging of the communication bandwidth, the non-adversarial agents quantize the estimates they broadcast to their neighboring agents via an adaptive quantizer. Researchers have explored different quantization designs and how they affect the behavior and convergence of distributed optimization problems as seen in [42, 46–49]. Nonetheless, these quantization mechanisms usually result in quantization errors and some researchers have also explore techniques on how to compensate for those errors as seen in [50–55]. In [56], a uniform quantizer is used to show how the estimates of non-adversarial agents can converge to the neighborhood of the optimal solution when adversarial agents disrupt the network. This dissertation focuses on the design of a fixed and adaptive quantizer in a communication-constrained environment amidst the presence of adversarial agents. An adaptive quantizer is studied in [57] in a communication-constrained

environment but the authors did not consider the presence of adversarial attacks by malicious actors which is a contribution of this dissertation.

In the cause of this thesis, a novel distributed gradient with adaptive quantization and adversarial attack method (DISGAQAAM) is presented using an adaptive quantizer when non-adversarial agents try to survive the havoc caused by adversarial agents due to the injection of attack by the adversarial agents. The DISGAQAAM method is described in a manner that non-adversarial agents can detect the presence of an attack by using an outlier test according to relationship between the resolution of the quantizer and the attack vector as described. It is shown that under strong convexity of the objective function, convergence to the neighborhood of the optimal solution can still be obtained using a suitable step size. A detection strategy is also proposed as a way to detect adversarial agents in the network.

Solving many distributed optimization algorithms in a network with adversarial agents, especially in large-scale systems fit the paradigm for distributed optimization in which components or agents of the system locally and iteratively solves a part of an optimization problem. The agents including the adversarial and non-adversarial agents exchange information with other (neighboring) agents in the network to arrive at a system-wide solution. Distributed decision problems are common in many areas including power systems, multi-agent systems, wireless sensor networks and have seen a recent surge in distributed machine learning, where server and worker nodes cooperate to solve learning problems as seen in [58–61]. Typically, in a network comprising n agents, each agent (whether good or bad) has a (sometimes private) local objective function $f_i(x)$ and the goal is to optimize an aggregate function comprising the local objective functions of the agents $\sum_{i=1}^{n} f_i(x)$. In much of the literature, the local objective functions are usually assumed to be strongly convex and Lipschitz smooth.

The literature on distributed optimization methods is rich and encompasses a wide range of methods

that have been proposed to solve such problems including Alternating Direction Method of Multipliers (ADMM) dual averaging, gradient-based and Newton-type methods. While the ADMM framework can easily handle a broader class of functions e.g. nondifferentiable functions, can be parallelized and is easy to implement, it has a very poor convergence rate [10]. Dual averaging methods, on the other hand, in which agents keep estimates of the decision variable and exchange them with neighboring agents are known to perform even more poorly than ADMM [62,63]. Many of the work on distributed optimization with adversaries have been done using distributed gradient descent (DGD) methods and their variants, which attempt to combine the merits of the dual averaging and ADMM are appealing and have been studied in the literature [12, 34, 64–69]. Applying the gradient-descent method to minimize strongly convex and smooth objective functions, is known, results in a linear rate of convergence with an appropriately chosen step size [70]. When adversarial agents are present in a network, the algorithms might not converge linearly to be exact but can converge to a neighborhood of an optimal solution as seen in [7] and [56]. To overcome the convergence rate limitations of gradient-based methods, second-order (Newton-type) algorithms have been proposed [71, 72]. Though Newton-type methods have quadratic convergence rates, the computational and storage overhead incurred in inverting the Hessian is significant, particularly for large-scale problems that have a high number of variables especially in large distributed systems with multiple agents. Furthermore, to distribute the computation of Newton-type methods, positive-definiteness of the Hessian of the objective function is needed to ensure methods like matrix splitting are applicable [73, 74].

To ensure great performance of distributed algorithms in the presence of attacks, one has to choose the step size appropriately to speed up convergence. Even though constant or decaying step-sizes are commonly used in gradient-based and Newton-type methods, the *Quasi-Newton* methods that leverage the computation structure of the gradient methods alongside the fast convergence properties of Newton-type methods have been studied, including methods like the Broyden-Fletcher-

Goldfarb-Shanno (BFGS) algorithm [75, 76], the Davidon-Fletcher-Powell [77, 78] algorithm and the Barzilai-Borwein (BB) method first introduced in [79]. To illustrate the performance of Quasi-Newton Methods, we consider solving problem (1.1) in a distributed manner. The gradient descent update is given by:

$$x_i(k+1) = x_i(k) - \alpha_i \nabla f_i(x(k)),$$

where x_i is the estimate of each agent i, α_i is the step size of each agent i and $\nabla f_i(x(k))$ is the gradient of each agent's estimates and the newton update is given by:

$$x_i(k+1) = x_i(k) - F^{-1}(x_i(k))\nabla f_i(x(k)). \tag{2.1}$$

In equation (2.1), computing the inverse of the hessian F^{-1} for large scale problems becomes a bottleneck, hence the relevance of Quasi-Newton Methods. The central idea in the performance of these methods is to speed up convergence by exploiting the information from the inverse hessian without necessarily computing it explicitly. For example, Barzilai-Borwein quasi-Newton method computes step-sizes using the difference of successive iterates and the gradient evaluated at those iterates. One of the appealing properties of the BB method is the simple nature of the computations and updates involved, even in the distributed case as shown in the later part of this thesis.

To improve the convergence rates of distributed gradient descent algorithms in the presence of adversarial agents, Barzilai-Borwein methods can be employed. In this regard, a preliminary work is presented in the later part of this dissertation to show its effectiveness in addressing the challenges of computing the inverse of the hessian. The work on BB method in the last chapter of this dissertation builds on earlier work on Distributed Gradient Descent (DGD) methods [12, 80], [81–83] as well as distributed Barzilai-Borwein methods [84, 85] where the authors analyze two-dimensional convex-quadratic functions. More recent efforts in [86], which took an adapt-then-combine strategy for agreement updates obtained a geometric rate of convergence. As shown in the last chapter of

this dissertation, a fully distributed algorithm that converges Q-linearly to the optimal solution is proposed. The difference between the theoretical results obtained from the later part of this dissertation and the results in [84–86] is the is the direct approach used in achieving Q-Linear convergence which is not the case for the results cited. The approach taken in the BB method is applicable to strongly convex functions and the centralized and distributed cases are analysed where computation of the step-sizes are done in an uncoordinated manner. In this approach, agents locally carry out computations, exchange information with neighboring agents to reach an agreement and use information obtained from other agents to continue the iterative process. The applications of the BB methods to combating adversarial attacks is a future work that will be explored and curious readers are welcome to embark on this research topic.

Optimization problems with models are also applicable in online convex optimization and have been explored in [87] where the author measured the performance of an online problem involving an adversarial loss functions and an adversarial constraint via dynamic regret. In the pursuit of examining the performance of a distributed optimization problem with adversarial attack, an online optimization application of a gradient-based algorithm using the Barzilai-Borwein step sizes is presented in the last chapter of this dissertation. *Online Optimization* involves a process where an online agent makes a decision without knowing whether the decision is correct or not. The objective of the online agent is to make a sequence of accurate decisions given knowledge of the optimal solution to previous decisions. A common notion associated with many optimization problems known as *regret* measures how well the online agent performs after a certain time, based on the the difference between the loss incured and the best decision taken [88]. The problem of online optimization has applications to a number of fields including game theory, the smart grid and classification in machine learning amongst others. Performance of online optimization algorithms is usually measured in terms of the aggregate regret suffered by the online agent compared with the known optimal solution of each problem across the sequence of problems.

Online optimization methods and algorithms have been studied using different methods including gradient-based methods [88–90]. Extensions have been considered on unconstrained problems [91] and online problems with long-term [92]. Problems in dynamic environments have also been analyzed in [87,93–97]. The author in [94] used gradient tracking technique in a static optimization scenario and showed that the regret bounds in the dynamic optimization case is independent of the time horizon. In [95], the authors obtained sublinear regret in a dynamic case for a distributed online problem using the primal-dual descent algorithm. The authors in [96] obtained sublinear regret for a distributed online framework that has time-varying constraints and presented a fit technique to deal with constraint violations. In [87], the authors solves the online optimization problem with an application to adversarial attack. They explored an online constrained problem with adversarial objective functions and constraints and obtained a sublinear regret. The key contribution in the online optimization algorithm proposed in the later part of this dissertation is the introduction of the Quasi-Newton Barzilai-Borwein (BB) method in the online scenario to speed up convergence and avoid the bottleneck of computing the inverse of the hessian that the Newton method demands. Additionally, the results in [88–90] used regular gradient methods in obtaining sublinear regret while the online work proposed in the later part of this dissertation obtains a similar sublinear regret using the BB step sizes. As well-structured as gradient methods are, applying them to large-scale online problems face several challenges and become impractical due to their well-known slow convergence rates in the static settings [98]. The newton method's limitations in storing and inverting the inverse of the hessian also makes them impractical for large-scale online optimization problems. In this regard, an online distributed Barzilai-Borwein method is proposed as a preliminary work in the last chapter of this dissertation.

Problems Studied in the Dissertation: The problems explored are itemized as follows:

- A distributed optimization method is proposed in a scenario where adversarial agents inject

attack into the network by perturbing their estimates by an attack vector. Convergence to a neighborhood of the optimal solution is proved for a complete and general graph despite the presence of adversarial agents in the network.

- Conditions under which adversarial agents can force divergence are shown. Using suitable preventive mechanisms, detection and resilience strategies are shown to deal with the injection of attack performed by adversarial agents. A bound on the attack vector is shown as a detection metric to identify adversarial agents in the network.

- To manage the bandwidth that the adversarial agents use due to their attack, a distributed gradient algorithm that involves non-adversarial agents quantizing their estimates is proposed. Both fixed and adaptive quantizers are used and results show that non-adversarial agents can still approach the neighborhood of the optimal solution.

- To improve the convergence rates of distributed optimization problems with adversaries, a preliminary work on distributed Quasi-Newton Method is proposed as a possible improvement over the Newton method. The applications of Quasi-Newton Method to online optimization is also showed.

The next chapter explores a distributed optimization model in the presence of adversaries.

CHAPTER 3: DISTRIBUTED OPTIMIZATION IN THE PRESENCE OF MALICIOUS AGENTS

This chapter presents an analysis of the effects of malicious agents on the solution of a distributed optimization problem over a network using the gradient descent algorithm. It is shown how adversarial nodes can disrupt convergence to optimal solution of the network with knowledge of the average initial value of the non-malicious agents. When the network structure connecting the agents is a complete graph, it is shown how cooperation enables the agents to prevent convergence to the optimal solution by perturbing their local estimates. However when the network structure is not a complete graph, it is characterized how the malicious agents can cause disruption if they have an initial value of the regular agents estimates. This thesis shows that for the agents solving the distributed optimization problem to converge to a neighborhood of the optimal solution, the distance between their average initial value and the optimal solution has to be less than the magnitude of the attack vector.

Notations

Vectors and matrices are represented by boldface lower and upper case letters, respectively. We denote the set of positive and negative reals as $\mathbb{R}_+$ and $\mathbb{R}_-$, a vector or matrix transpose as $(\cdot)^T$, and the L2-norm of a vector by $||\cdot||$. The gradient of a function $f(\cdot)$ is denoted $\nabla f(\cdot)$ and the Hessian of a function $f(\cdot)$ be $F(\cdot) = \nabla^2 f(\cdot)$. We denote a vector or matrix transpose as $(\cdot)^T$, and the L2-norm of a vector by $||\cdot||$. We also denote the gradient of a function $f(\cdot)$ as $\nabla f(\cdot)$ and an n dimensional vector of ones as 1_n.

The problem on distributed optimization with adversaries is formulated in the next section.

Problem Formulation and Attack Model

We consider a network comprising n agents represented by an undirected graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = 1, 2, ...n$ is the set of nodes (agents) and $\mathcal{E} = (i, j)$ is the set of edges. Let the neighbors of each agent i be denoted by the set $N_i = \{j : (i, j) \in \mathcal{E}\}$. Because the graph is undirected, $(i, j) \in \mathcal{E}$ also implies $(i, j) \in \mathcal{E}$. Let a closed convex set be defined as intersections of closed points in space that are on a side of a hyperplane. The agents collectively solve the unconstrained distributed optimization problem

$$\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) = \sum_{i=1}^{n} f_i(\boldsymbol{x}), \tag{3.1}$$

where each local objective function $f_i(\cdot)$ is convex and smooth and $\mathcal{X}$ is the feasible set. To solve the optimization problem using the gradient descent method, each agent i maintains a local copy $\boldsymbol{x}_i \in \mathbb{R}^p$ of the decision variable $\boldsymbol{x} \in \mathbb{R}^p$ and carries out a local update using their local cost function and broadcast the same to their neighbors:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) - \alpha_i \nabla f_i(\boldsymbol{x}(k)), \tag{3.2}$$

where $\alpha_i \in \mathbb{R}_+$ is an appropriately chosen step size. It is known that if f_i is convex and differentiable, with an appropriately chosen step size α_i the updates in Equation 3.2 will converge to the optimal solution [11]. The problem set-up considers two cases – the complete graph and the non-complete graph case. In the complete graph case, the adversarial agents are assumed to know one other and coordinate the choice of an attack vector.

Similar to adversarial machine learning, the forms of attack can be black box (the attacker has no knowledge of the model), grey box (attacker has partial knowledge of the model) or white box (the attacker has complete knowledge of the model). In the context of this dissertation, it is assumed that the attackers takes the white box attack form and it is assumed that the adversarial agents are

not known to the rest of the network *a priori*. In this chapter, it is assumed that adversarial agents have knowledge of the model needed to cause obstruction. Assumptions on the attack model are shown below:

- Adversarial agents have knowledge of the algorithm and the goal of non-adversarial agents.
- Since adversarial agents know the problem that malicious agents are solving, they can perturb in a manner to result in a deviation of the optimal solution non-adversarial agents are trying to achieve.
- Non-adversarial agents can detect adversarial agents in the network using a detection strategy that will be explored in the later part of the thesis.

The objective of the adversarial or malicious nodes is to deviate the network from reaching the true optimal solution x^* of Problem 3.1. To accomplish the malicious objective, rather than follow the update in Equation 3.2, the adversarial nodes perturb their local estimates with an attack vector $\epsilon \in \mathbb{R}^p$:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) - \alpha_i \nabla f_i(\boldsymbol{x}(k)) + \epsilon(k), \tag{3.3}$$

before broadcasting their estimates to neighboring agents in the network. We note that an alternative formulation is to assume an objective function of the form:

$$\min_x \hat{f}(\boldsymbol{x}) \tag{3.4}$$

for the adversarial agents such that the optimal solution to $\hat{f}(x)$ is $x^a = x^* + \hat{\epsilon}$, where x^* is the optimal solution to the objective function $f(\boldsymbol{x})$. We assume the adversarial agents carefully pick values of ϵ by which to perturb their local estimates.

Next, convergence of the distributed gradient-based method to solve Problem (3.1) using the update

in (3.2) when there are malicious agents is analysed. Before proceeding, however, the assumptions being made on Problem (3.1) is stated below.

Assumption 1. *The cost function $f(\boldsymbol{x})$ in Problems* (3.1) *is strongly convex and twice differentiable. This implies that for any vectors $x, y \in \mathbb{R}^p$, there exists $\mu \in \mathbb{R}_+$ such that:*

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) + \frac{\mu}{2} \|x - y\|^2.$$

Assumption 2. *The gradient of the objective function ∇f is Lipschitz continuous. This implies that for all vectors $x, y \in \mathbb{R}^n$, there exists a constant $L \in \mathbb{R}_+$ such that: $\|\nabla f(\boldsymbol{x}) - \nabla f(y)\| \leq L\|x - y\|$.*

Assumption 3. *The attack vector ϵ is uniformly bounded. This means that a constant $C > 0$ exists such that $\|\epsilon\| \leq C$.*

These assumptions are standard in the distributed optimization literature, as they allow for analysis.

Convergence Analysis

Convergence for the problem and attack model presented in Section 3 will be characterized, based on the distributed gradient descent algorithm and agreement updates. Each agent i updates his local estimate x_i and takes a weighted average of neighboring nodes following

$$\boldsymbol{x}_i (k+1) = \sum_{j \in N_i \cup \{i\}} W_{ij} \boldsymbol{x}_j (k) - \alpha_i \nabla f_i (\boldsymbol{x}_i (k)), \tag{3.5}$$

where W is an n-dimensional square weighting matrix comprising entries W_{ij} that denote the weight attached to agent j's estimate by agent i.

Let $X = [\boldsymbol{x}_1;\ \boldsymbol{x}_2;\ \ldots\ \boldsymbol{x}_n]^T \in \mathbb{R}^{np}$ be the concatenation of the local variables $\boldsymbol{x}_i$, I_p be the identity matrix whose dimension is p, $\otimes$ be the Kronecker operation, 1_n be an n dimensional vector

of ones and let W be doubly stochastic. We can express equation (3.5) more compactly as: $X(k+1) = (W \otimes I_p)X(k) - \alpha_i \nabla f(X(k))$, where $W \otimes I_p \in \mathbb{R}^{np \times np}$, and $\nabla f(X(k)) \in \mathbb{R}^{np}$ is the gradient of the objective $f(\cdot)$ evaluated at $X(k)$. The doubly stochastic matrix W has one eigenvalue $\lambda = 1$ and the other eigenvalues satisfy $0 < \lambda < 1$.

Convergence analysis to a neighborhood of the optimal solution will be examined by using the relationship: $\|x_i(k) - x^*\| = \|x_i(k) - \bar{\boldsymbol{x}}(k)\| + \|\bar{\boldsymbol{x}}(k) - x^*\|$. However, we must keep in mind that the bound on $\|x_i(k) - \bar{\boldsymbol{x}}(k)\|$ in the following Lemma captures the information on the spectral gap.

Lemma 3.0.1. *Let Assumptions 1, 2 and 3 hold with β being the second biggest value of the eigenvalues of the weights W (which has one eigenvalue equal to 1 and others have values less than 1), the distance between the individual estimates and the averaged estimates is bounded by the following:*

$$\|x_i(k) - \bar{\boldsymbol{x}}(k)\| \leq \frac{\alpha \bar{L}_i}{1-\beta}.$$

Proof. See [12] for the proof. □

Convergence Analysis Over a Complete Graph

To characterize convergence in the complete graph case, we introduce some additional notation to be used in the analyses. Let the average of local estimates be $\overline{\boldsymbol{x}}(k)$, the average of local gradients at current estimates be $\overline{g}(k)$ and the average of the attack be $\overline{\epsilon}$; that is, $\overline{\boldsymbol{x}}(k) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i(k)$, $\overline{\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\epsilon}_i(k)$, and $\overline{g}(k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{x}_i(k))$. where $\overline{\boldsymbol{x}}(k) \in \mathbb{R}^p$, $\overline{g}(k) \in \mathbb{R}^p$ and $\overline{\epsilon} \in \mathbb{R}^p$.

Based on the definition of $\overline{X}(K)$, let $\overline{X}(k)$ be such that $\overline{X}(k) = [\overline{\boldsymbol{x}}(k), \ldots..\overline{\boldsymbol{x}}(k)] \in \mathbb{R}^{np}$, then according to Lemma IV.2 in [12], we have the following: $\overline{X}(k) = \frac{1}{n}((1_n 1_n^T) \otimes I)X(k)$. Since W is doubly

stochastic, we have the following relationship:

$$\overline{X}(k+1) = \frac{1}{n}((1_n 1_n^T) \otimes I)X(k). \tag{3.6}$$

From Equation (3.6) as proved in [12], the consensus update can be expressed as $\overline{\boldsymbol{x}}(k+1) = \overline{\boldsymbol{x}}(k) - \alpha\overline{g}(k)$.

For the complete graph case, since the malicious agents are aware of one another and can cooperate, collectively deciding on the degree ϵ to which they want to perturb their local estimates for their adversarial goal. Therefore, all malicious agents choose the same $\epsilon \in \mathbb{R}^p$ which is the average of all attack vectors received from neighbors estimates. In our first result, we derive the condition under which convergence to a neighborhood of the optimal solution may be attained. As we will see, the size of the neighborhood, amongst others depends on the magnitude of the attack vector ϵ.

Lemma 3.0.2. *Suppose Assumptions 1, 2 and 3 hold. If the average initial value of of the agents when malicious agents are present satisfy* $\|\overline{\boldsymbol{x}}(0) - x^*\| < \|\epsilon\|$ *and the step size* α *satisfies the following relationship:* $\alpha < \frac{2}{\mu+L}$*, then the iterates generated converge to a neighborhood of the optimal solution,* x^**; where* μ *and* L *are the strong convexity parameter and Lipschitz constant of the objective function and its gradient respectively with* $\mu \leq L$*.*

Proof. The iterative equation solution for a distributed gradient descent is:

$$X(k) = -\alpha \sum_{s=0}^{k-1} \left(W^{(k-1-s)} \otimes I \right) \nabla f\left(x(s)\right),$$

which accounts for the consensus step as well. The relationship between the optimal solution x^* and the desired malicious solution x^a of the adversarial agents can be expressed as: $x^a = x^* + \epsilon$. When malicious agents are present, we have the following update: $\overline{\boldsymbol{x}}(k+1) - x^a = \overline{\boldsymbol{x}}(k) - \alpha\overline{g}(k) - x^a$. We now have the iterate equation as: $\overline{\boldsymbol{x}}(k+1) - (x^* + \epsilon) = \overline{\boldsymbol{x}}(k) - (x^* + \epsilon) - \alpha\overline{g}(k)$. To analyze

convergence of the iterates to the optimal solution, we will begin by considering the iterative equation. We can express $\|\overline{\boldsymbol{x}}(k+1)-x^*-\epsilon\|^2$ as:

$$\|\overline{\boldsymbol{x}}(k+1)-x^*-\epsilon\|^2=\|\overline{\boldsymbol{x}}(k)-x^*-\epsilon-\alpha\overline{g}(k)\|^2=\|\overline{\boldsymbol{x}}(k)-x^*\|^2+\|\epsilon\|^2+\alpha^2\|\overline{g}(k)\|^2+2\epsilon(\alpha\overline{g}(k)-(\overline{\boldsymbol{x}}(k)-x^*))$$
$$-2(\overline{\boldsymbol{x}}(k)-x^*)^T(\alpha\overline{g}(k)).$$

By using vector norm principle, we know that for vectors a, b, the inequality $2a^Tb \leq \|a\|^2+\|b\|^2$ is satisfied. By similarly applying vector norm principles, we have the following:

$$2\epsilon(\alpha\overline{g}(k)-(\overline{\boldsymbol{x}}(k)-x^*)) \leq \|\epsilon\|^2+\|\alpha\overline{g}(k)-(\overline{\boldsymbol{x}}(k)-x^*)\|^2=\|\epsilon\|^2+\|\alpha\overline{g}(k)\|^2+\|\overline{\boldsymbol{x}}(k)-x^*\|^2-2(\overline{\boldsymbol{x}}(k)-x^*)^T$$
$$\leq \|\epsilon\|^2+\alpha^2\|\overline{g}(k)\|^2+\|\overline{\boldsymbol{x}}(k)-x^*\|^2-\alpha c_1\|\overline{g}(k)\|^2-\alpha c_2\|\overline{\boldsymbol{x}}(k)-x^*\|^2,$$

where the values of c_1 and c_2 are [11]:

$$c_1=\frac{2}{\mu+L} \quad \text{and} \quad c_2=\frac{2\mu L}{\mu+L}.$$

By using strong convexity of the objective function we obtain (Theorem 2.1.12 in [11]):

$$\begin{aligned}\|\overline{\boldsymbol{x}}(k+1)-x^*-\epsilon\|^2&=\|\overline{\boldsymbol{x}}(k)-x^*-\epsilon-\alpha\overline{g}(k)\|^2\leq \|\overline{\boldsymbol{x}}(k)-x^*\|^2+\|\epsilon\|^2+\alpha^2\|\overline{g}(k)\|^2+\|\epsilon\|^2+\alpha^2\|\overline{g}(k)\|^2\\&+\|\overline{\boldsymbol{x}}(k)-x^*\|^2-\alpha c_1\|\overline{g}(k)\|^2-\alpha c_2\|\overline{\boldsymbol{x}}(k)-x^*\|^2-\alpha c_1\|\overline{g}(k)\|^2-\alpha c_2\|\overline{\boldsymbol{x}}(k)-x^*\|^2,\\&=(2-2\alpha c_2)\|\overline{\boldsymbol{x}}(k)-x^*\|^2+(2\alpha^2-2\alpha c_1)\|\overline{g}(k)\|^2+2\|\epsilon\|^2.\end{aligned} \tag{3.7}$$

In what follows we would show that the terms in the right hand side of Equation (3.7) does not grow unbounded and is, in fact, related to the initial iterates and magnitude of the malicious attack. Clearly $\|\epsilon\|^2$ is positive and $(2\alpha^2-2\alpha c_1)$ is negative when $\alpha < c_1$. Now we will show that $(2-2\alpha c_2) > 0$ by equivalently showing that $\alpha c_2 < 1$.

By using the value $c_2=\frac{2\mu L}{\mu+L}$, we obtain the following relationship: $\alpha c_2=\frac{2\alpha\mu L}{\mu+L}$. Since $\alpha<\frac{1}{\mu}$, then we obtain the upper bound αc_2 as follows: $\alpha c_2<\frac{1}{\mu}\frac{2\mu L}{\mu+L}=\frac{2L}{\mu+L}$. We know that both L and μ are

positive and $\mu \leq L$. Therefore if $\mu = L$, then, $2L/(\mu + L) = 1$. So we obtain the fact that $\alpha c_2 < 1$. We have now affirmed that $(2 - 2\alpha c_2) > 0$. Moreover, if $\alpha c_2 > \frac{1}{2}$, then we obtain that $1 - \alpha c_2 < \frac{1}{2}$ and we obtain that $2 - 2\alpha c_2 < 1$. Therefore by using the following condition: $0 < 2 - 2\alpha c_2 < 1$, the left hand side of Equation (3.7) can be upper bounded by

$$\|\overline{\boldsymbol{x}}(k+1) - x^* - \epsilon\|^2 \leq (2 - 2\alpha c_2)\|\overline{\boldsymbol{x}}(k) - x^*\|^2 + 2\|\epsilon\|^2. \tag{3.8}$$

If $\overline{\boldsymbol{x}}(k+1) - x^* < 0 < \epsilon$, then we have the relationship below:

$$\|\overline{\boldsymbol{x}}(k+1) - x^* - \epsilon\|^2 > \|\overline{\boldsymbol{x}}(k+1) - x^*\|^2. \tag{3.9}$$

From equations (3.8) and (3.9), we obtain the following relationship: $\|\overline{\boldsymbol{x}}(k+1) - x^*\|^2 \leq (2 - 2\alpha c_2)\|\overline{\boldsymbol{x}}(k) - x^*\|^2 + 2\|\epsilon\|^2$. By applying recursion we obtain the upper bound of $\|\overline{\boldsymbol{x}}(k) - x^*\|^2$ to be:

$$\|\overline{\boldsymbol{x}}(k) - x^*\|^2 \leq (2 - 2\alpha c_2)^k \|\overline{\boldsymbol{x}}(0) - x^*\|^2 + 2\|\epsilon\|^2, \tag{3.10}$$

from which we conclude that, after taking the square roots of both sides of equation (3.10), the following relationship holds:

$$\|\overline{\boldsymbol{x}}(k) - x^*\| \leq (2 - 2\alpha c_2)^{\frac{k}{2}} \|\overline{\boldsymbol{x}}(0) - x^*\| + \sqrt{2}\|C\|. \tag{3.11}$$

where C is the bound on the attack vector. Hence, the iterates converge to the neighborhood of the optimal solution, x^*. □

The central idea in Lemma 3.0.2 is that the average initial value of the agents need to lie within ϵ of the optimal solution x^* for the agents to converge to a neighborhood of the optimal solution in the presence of malicious agents. Knowledge of the average initial starting value is also critical for

the adversarial nodes, because their choice of ϵ could depend on the initial average value of $\overline{\boldsymbol{x}}$. The compete graph case in Lemma 3.0.2 also allows for adversarial agents who know one another to cooperate in choosing the attack vector or perturbation parameter ϵ. However, the downside is the possibility of jointly picking the wrong attack vector to bolster their malicious objective.

Next, we consider a general case where cooperation is not as easy because of the subset of malicious agents may not be neighbors.

Convergence Analysis over General Graph Structures

We consider the case in which the communication structure is more general, as opposed to being a complete graph. With a general structure, malicious agents do not necessarily have the liberty to cooperate and agree on values for the attack vector ϵ, since they may not be adjacent to one another in the network. In other words, each regular agent independently solves the minimization problem (3.1) with the malicious agents additively perturbing their local estimates by the attack vector ϵ_i. We will now examine the conditions on the attack parameter that enables convergence when non-adversarial and malicious agents are present in a general graph structure. In the analysis that follows, we assume that the largest value of ϵ_i is chosen based on the estimates the agents receive from neighbors.

We will now show conditions on ϵ that enable neighborhood convergence of iterates to the optimal point.

Lemma 3.0.3. *Suppose Assumptions 1, 2 and 3 hold, and let the bounded* $\epsilon \geq 0$. *If* $\|\boldsymbol{x}_i(0) - x^*\| < 0 < \epsilon_i \; \forall i$ *and the step size* α *satisfies* $\alpha < \frac{2}{\mu+L}$, *then the individual iterates generated converge to the neighborhood of the optimal solution,* x^*.

Proof. The proof is similar to the one in Lemma 3.0.2 except that in this scenario, each agent is

individually solving its own problem. In this case, the malicious agents are not cooperating to coordinate the attack vector ϵ. We begin with the iterate equation:

$$\begin{aligned}\|\boldsymbol{x}_i(k+1)-x^*-\epsilon_i\|^2 &= \|\boldsymbol{x}_i(k)-x^*-\epsilon_i-\alpha g(k)\|^2 = \|\boldsymbol{x}_i(k)-x^*\|^2+\|\epsilon_i\|^2+\alpha^2\|g(k)\|^2 \\ &\quad +2\epsilon_i(\alpha g(k)-(\boldsymbol{x}_i(k)-x^*))-2(\boldsymbol{x}_i(k)-x^*)^T(\alpha g(k)).\end{aligned} \tag{3.12}$$

Leveraging the fact that for vectors a, b, the inequality $2a^Tb \leq \|a\|^2+\|b\|^2$ holds, we can further simplify the fourth summand in Equation (3.12) as

$$\begin{aligned}2\epsilon_i(\alpha g(k)-(\boldsymbol{x}_i(k)-x^*)) &\leq \|\epsilon_i\|^2+\|\alpha g(k)-(\boldsymbol{x}_i(k)-x^*)\|^2 = \|\epsilon_i\|^2+\|\alpha g(k)\|^2+\|\boldsymbol{x}_i(k)-x^*\|^2-2(\boldsymbol{x}_i(k)-x^*)^T, \\ &\leq \|\epsilon_i\|^2+\alpha^2\|g(k)\|^2+\|\boldsymbol{x}_i(k)-x^*\|^2-\alpha c_1\|g(k)\|^2-\alpha c_2\|\boldsymbol{x}_i(k)-x^*\|^2.\end{aligned}$$

where the values of c_1 and c_2 are respectively [11]:

$$c_1 = \frac{2}{\mu+L} \quad \text{and} \quad c_2 = \frac{2\mu L}{\mu+L}.$$

Hence, Equation (3.12) can be upper bounded by:

$$\begin{aligned}\|\boldsymbol{x}_i(k+1)-x^*-\epsilon_i\|^2 &= \|\boldsymbol{x}_i(k)-x^*-\epsilon_i-\alpha g(k)\|^2 \leq \|\boldsymbol{x}_i(k)-x^*\|^2+\|\epsilon_i\|^2+\alpha^2\|g(k)\|^2+\|\epsilon_i\|^2+\alpha^2\|g(k)\|^2 \\ &\quad +\|\boldsymbol{x}_i(k)-x^*\|^2-\alpha c_1\|g(k)\|^2-\alpha c_2\|\boldsymbol{x}_i(k)-x^*\|^2-\alpha c_1\|g(k)\|^2-\alpha c_2\|\boldsymbol{x}_i(k)-x^*\|^2, \\ &= (2-2\alpha c_2)\|\boldsymbol{x}_i(k)-x^*\|^2+(2\alpha^2-2\alpha c_1)\|g(k)\|^2+2\|\epsilon_i\|^2.\end{aligned}$$

Since $\|\epsilon_i\|^2$ is positive, the term $(2\alpha^2-2\alpha c_1)$ is negative when $\alpha < c_1$ and using the fact that $0 < (2-2\alpha c_2) < 1$, which we showed in Lemma 3.0.2, we obtain the following:

$$\|\boldsymbol{x}_i(k+1)-x^*-\epsilon_i\|^2 \leq (2-2\alpha c_2)\|\boldsymbol{x}_i(k)-x^*\|^2+2\|\epsilon_i\|^2. \tag{3.13}$$

Since $\boldsymbol{x}_i(k+1) - x^* < 0 < \epsilon_i$, we obtain the relationship below:

$$\|\boldsymbol{x}_i(k+1) - x^* - \epsilon_i\|^2 > \|\boldsymbol{x}_i(k+1) - x^*\|^2. \tag{3.14}$$

From equations (3.13) and (3.14), we obtain: $\|\boldsymbol{x}_i(k+1)-x^*\|^2 \leq (2-2\alpha c_2)\|\boldsymbol{x}_i(k)-x^*\|^2+2\|\epsilon_i\|^2$, and by the recursive relationship, we obtain the bounds: $\|\boldsymbol{x}_i(k)-x^*\|^2 \leq (2-2\alpha c_2)^k\|\boldsymbol{x}_i(0)-x^*\|^2+2\|\epsilon_i\|^2$, from which we conclude that $\|\boldsymbol{x}_i(k)-x^*\|$ can be bounded as the following:

$$\|\boldsymbol{x}_i(k)-x^*\| \leq (2-2\alpha c_2)^{\frac{k}{2}}\|\boldsymbol{x}_i(0)-x^*\|+\sqrt{2}\|\epsilon_i\|.$$

Therefore, the individual iterates converge to the neighborhood of the optimal solution, x^*. □

Lemma 3.0.3 illustrates the deviations of individual agents from the optimal solution and the bound indicates the chosen attack vector affects the neighborhood of convergence. While in Lemma 3.0.2 allows adversarial agents to coordinate and use a uniform attack vector ϵ, the result in Lemma 3.0.3 does not require cooperation or the use of a uniform attack vector.

Remark 1. *The disparities in the convergence analysis of the complete and general graph are due to how adversarial choose their attack vectors. If the attackers jointly choose their attack vector favorably, then the complete graphical structure will be more favorable for the adversarial agents than the general graphical structures as they have more leverage in the obstruction process. However, the general graphical structure has advantages too as there is more privacy and flexibility for the adversarial agents to choose their attack vectors. The results for both the complete and general graphical structures can be improved by making the attack vector as a function of the agents estimates for more interdependence.*

Numerical Experiments

We illustrate our theoretical results of Lemmas 3.0.2 and 3.0.3 in a network of $n = 10$ agents over 100 iterations using the linear regression loss function as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} f(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{1}{2}\|A_i\boldsymbol{x} - b_i\|^2, \tag{3.15}$$

in a distributed way. In equation (3.15), $n = 10$, A_i is a n by n matrix and b_i is an n by 1 matrix. The gradient of the the function in equation (3.15) is $A^T(A_i\boldsymbol{x} - b_i)$. Given that the Lipschitz constant L is the largest eigenvalue of A^TA and the strong convexity constant μ is the smallest eigenvalue of A^TA, we choose a step size such that $\alpha < \frac{2}{\mu+L}$ to satisfy the strong convexity assumptions based on results of Lemmas 3.0.2 and 3.0.3. We will show how the choices of attack vectors of different magnitudes and agents' initial estimates influence convergence to a neighborhood of the optimal solution. In the illustrations to follow, entries of the attack vector was drawn uniform distribution over the interval $(0, 1)$. For the complete and general graph cases below, we use a step size of $\alpha = \frac{1}{\mu+L}$.

Complete Graph Case with Common Attack Vector

We begin with the case when the communication network is a complete graph, the case in which the adversarial agents perturb their local iterates with a common attack vector. We define the error as the distance between the average iterate and the optimal solution and present the error convergence in Figure 3.1 to 3.3. For the plot in Figure 3.3, we assumed the number of non-adversarial nodes was 1 with 9 adversarial nodes.

In Figure 3.2, we illustrate convergence of the error when there are 5 adversarial and 5 non-

adversarial nodes in the 10-node network. Figure 3.1 contains the plot for the scenario with 8 non-adversarial nodes and 2 adversarial nodes, which shows a further reduction in the actual error obtained. In the three figures, we can observe that as the proportion of non-adversarial agents in the network increase, the error convergences faster.

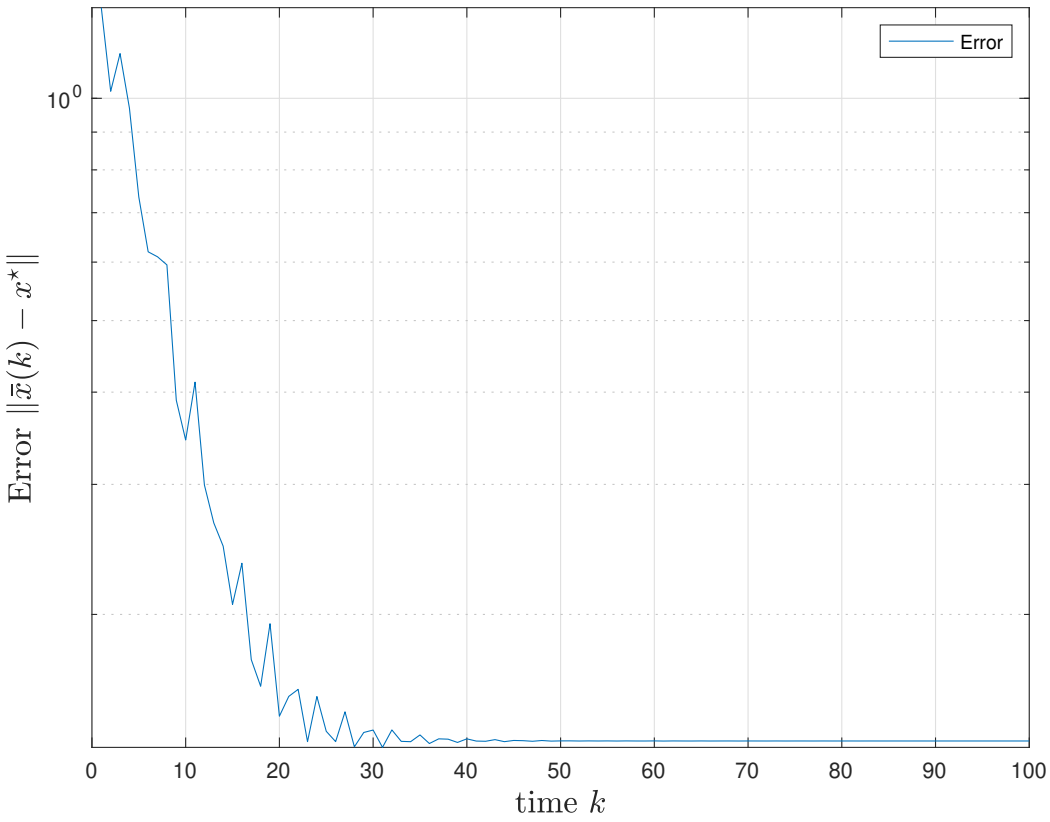

Figure 3.1: Same Attack Simulations for 8 non-adversarial agents and 2 adversarial agents.

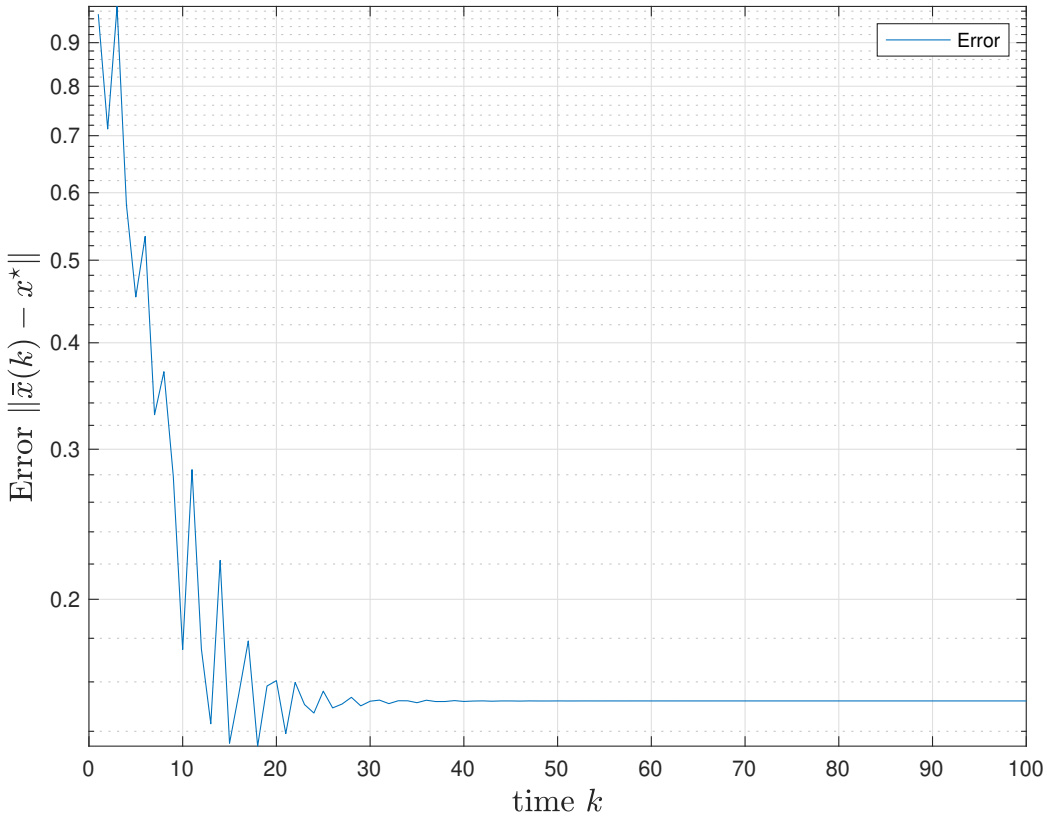

Figure 3.2: Same Attack Simulations for 5 non-adversarial agent and 5 malicious agents.

General Graph Case with Different Attack Vectors

We perform experiments using the same linear regression loss function in a scenario where adversarial agents use a different attack vector. All the parameters used in this scenario are similar to those used in the scenario with same attack vector (section 3). We vary the number of malicious nodes for the general (non-complete) graph case comprising n = 10 agents while solving Problem (3.15). We also show the error evolution for different proportions of malicious to non-malicious nodes. Figure 3.6 shows the case with 3 non-adversarial and 7 adversarial nodes. Figure 3.5 shows the case with 5 non-adversarial and 5 adversarial nodes; and Figure 3.4 shows the case comprising

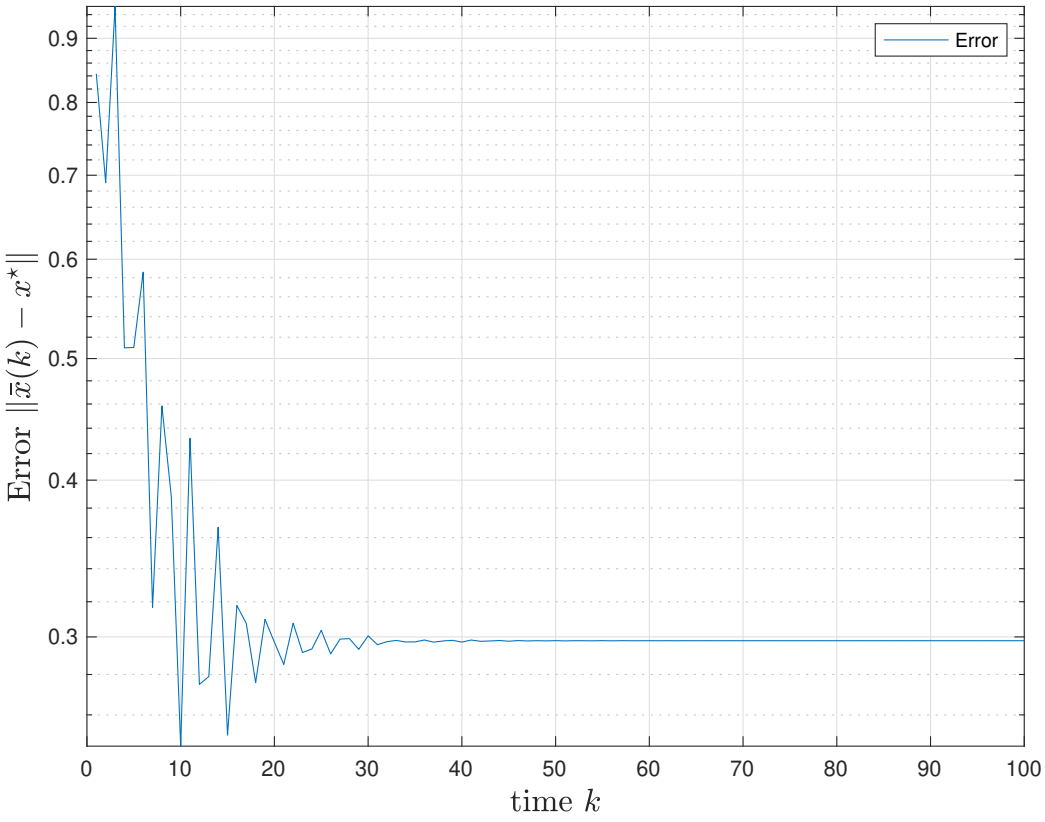

Figure 3.3: Same Attack Simulations for 1 non-adversarial agent and 9 malicious agents.

8 non-adversarial nodes and 2 malicious nodes. From the figures, we can observe that as the ratio of malicious nodes in the network increases, the convergence error increases. This outcome is intuitive and expected, since the presence of more agents causing disruption to the distributed consensus-based gradient algorithm would cause a greater deviation from the optimal solution.

In this chapter, the application of distributed optimization in adversarial attacks is shown. In addition, conditions that guarantee convergence to a neighborhood of the optimal solution despite the presence of malicious nodes in a network is presented. The method used in this chapter is the standard distributed gradient descent method.

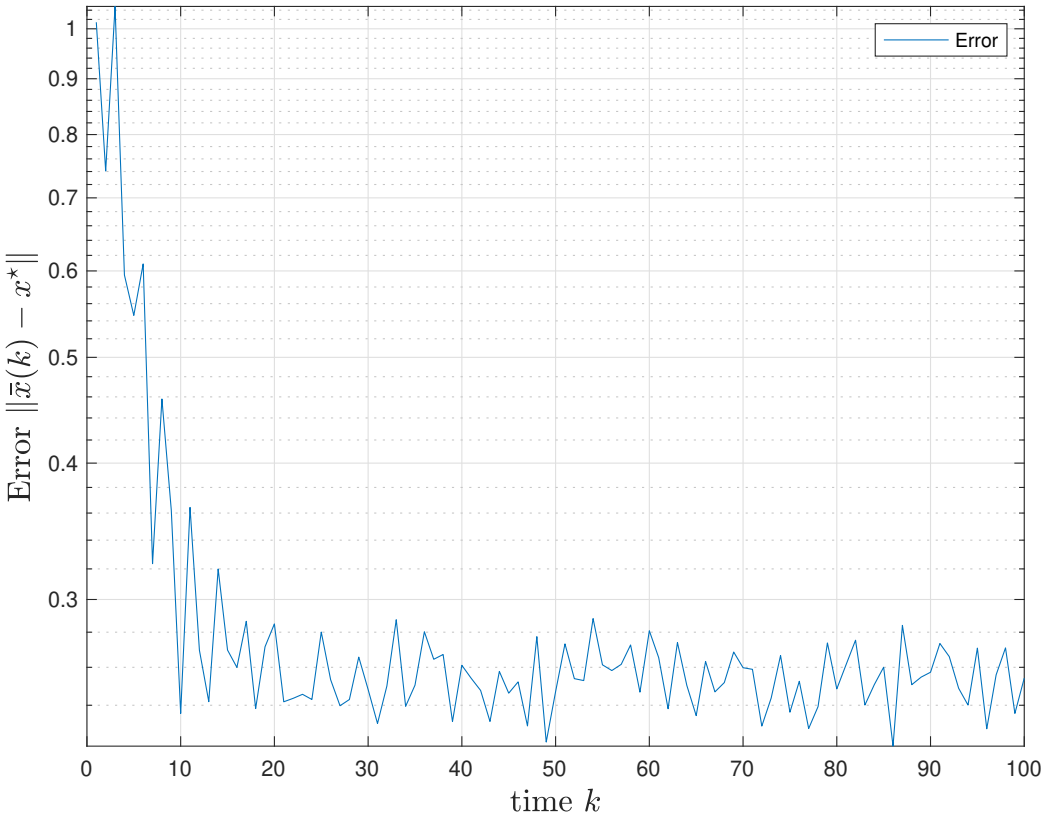

Figure 3.4: Different Attack Simulations for 8 non-adversarial agent and 2 malicious agents.

In the next chapter, an extension of the application of the problem described in this chapter is presented specifically in a scenario where there are constraints from adversarial attack and quantization.

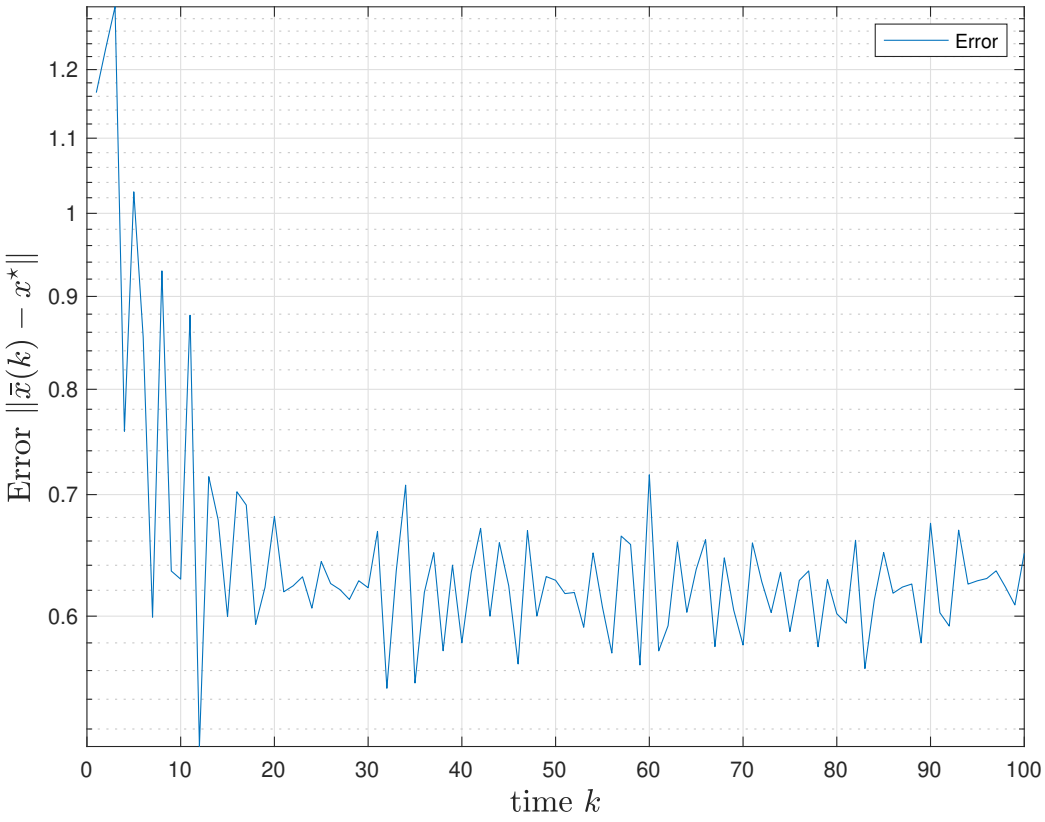

Figure 3.5: Different Attack Simulations for 5 non-adversarial agent and 5 malicious agents.

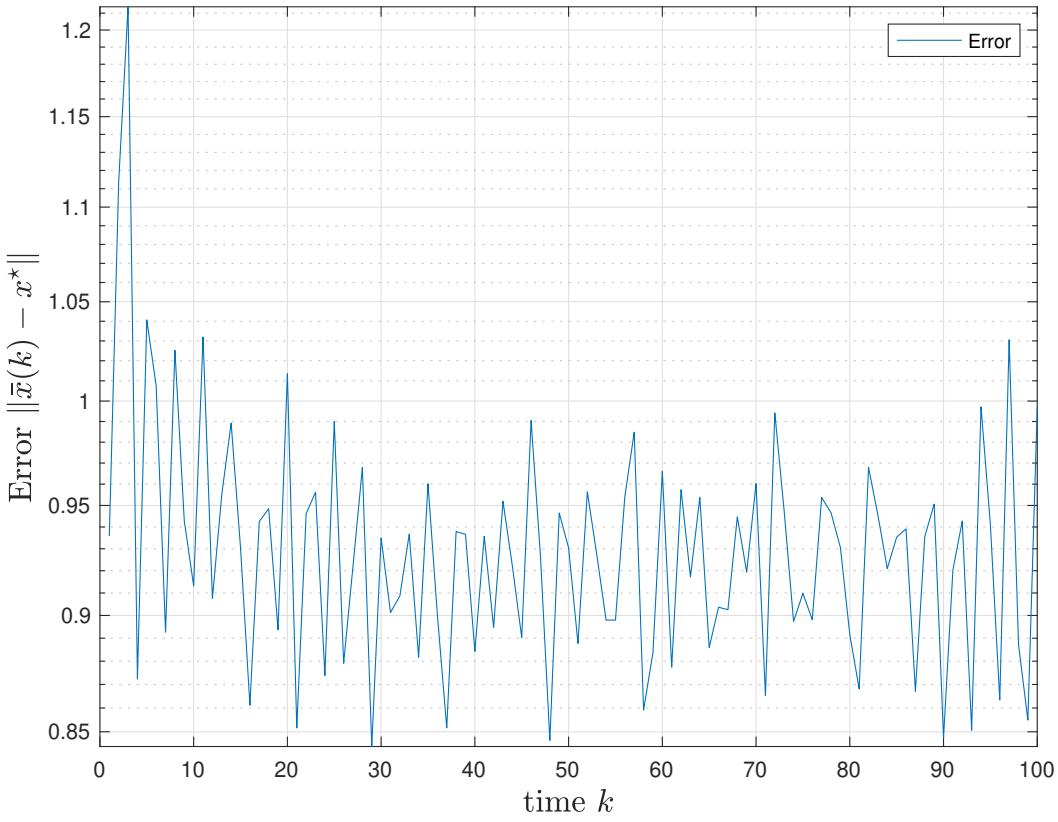

Figure 3.6: Different Attack Simulations for 3 non-adversarial agent and 7 malicious agents.

CHAPTER 4: QUANTIZED AND DISTRIBUTED SUBGRADIENT OPTIMIZATION WITH MALICIOUS ATTACK

There are different methods for solving communication-constrained problems; some of which are event-triggered, sparsification and Quantization (the focus of this dissertation). In an event-triggered strategy, communication is done when necessary and especially when a particular event has occurred. Sparsification entails dropping some terms or making some entries (x_i) zero. Quantization is explored in this chapter where convergence properties of a distributed gradient algorithm in the presence of adversarial agents and limited bandwidth for communication are characterized. The application of a communication-constrained optimization is seen in underwater communication where the communication strategy using the number of bits of information matters due to the complexity of the system. In a communication-constrained environment, agents $\boldsymbol{x}_i \in \mathbb{R}^p$ transmit up to p floating point numbers. For distributed solution, they exchange $x_i(k+1) = p(x_i(k), x_j \ldots.)$ with his neighbors. The calculations on the number of bits needed is initially done by computing the number of bits needed for 1 time step. Subsequently, for ϵ accuracy and $O(\frac{1}{\epsilon})$ iterations, the order on the number of bits is $O(\frac{1}{\epsilon}) * n * p$.

In this chapter, it is shown that the algorithm proposed converges to a neighborhood of the optimal solution and that the closeness to the optimal solution can be expressed in terms of the number of bits from the quantization and the size of the attack vector. The results in this chapter show that if a step size is chosen with respect to the strong convexity and Lipschitz parameters and the subgradient bound is expressed in terms of a suitable step size, then non-adversarial agents are still able to approach the optimal solution despite the presence of adversarial agents. Furthermore, the performance of the algorithm is expressed as a function of the adversarial attack vector and fineness of the quantization.

Problem Formulation

Suppose there is an undirected graph $G = (\mathcal{V}, \mathcal{E})$ comprising n nodes where $\mathcal{V} = 1, 2, \ldots n$ is the set of nodes (agents) and $\mathcal{E} = (i, j)$ is the set of edges, the neighbors of each agent i is defined as the set $N_i = \{j : (i, j) \in \mathcal{E}\}$. Let a closed convex set be defined as intersections of closed points in space that are on a side of a hyperplane. The agents are to jointly solve the problem

$$\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) = \sum_{i=1}^{n} f_i(\boldsymbol{x}), \tag{4.1}$$

where $\boldsymbol{x}$ is the decision variable, $\mathcal{X}$ is the closed, convex, feasible set and each agent i has a component $f_i(\cdot)$ of the strongly convex objective function $f(\cdot)$. We assume that some agents in the network act in an adversarial (malicious) manner by perturbing their estimate at each iteration and overwhelming the communication bandwidth for coordination in the network. To manage the limited bandwidth left, the non-adversarial nodes quantize the information shared with neighboring nodes. The uniform quantizer is chosen to ensure that agents use equal and constant step sizes to broadcast information to their neighbors.

As illustrated in Figure 4.1, the thin communication links are used to denote connection between two non-adversarial agents, while thick pipes are used to depict the connection between an adversarial agent and any other agents (adversarial or non-adversarial). The non-adversarial nodes need to manage the communication bandwidth to approach the optimal solution to Equation (4.1) and to overcome the bottleneck caused by adversarial nodes upscaling their estimates. We solve problem (4.1) using the distributed subgradient method. In this framework, each non-adversarial agent i broadcasts quantized iterates $Q(\boldsymbol{x}_i(k)) \in \mathbb{R}^p$ based on what is received from neighbors and carry

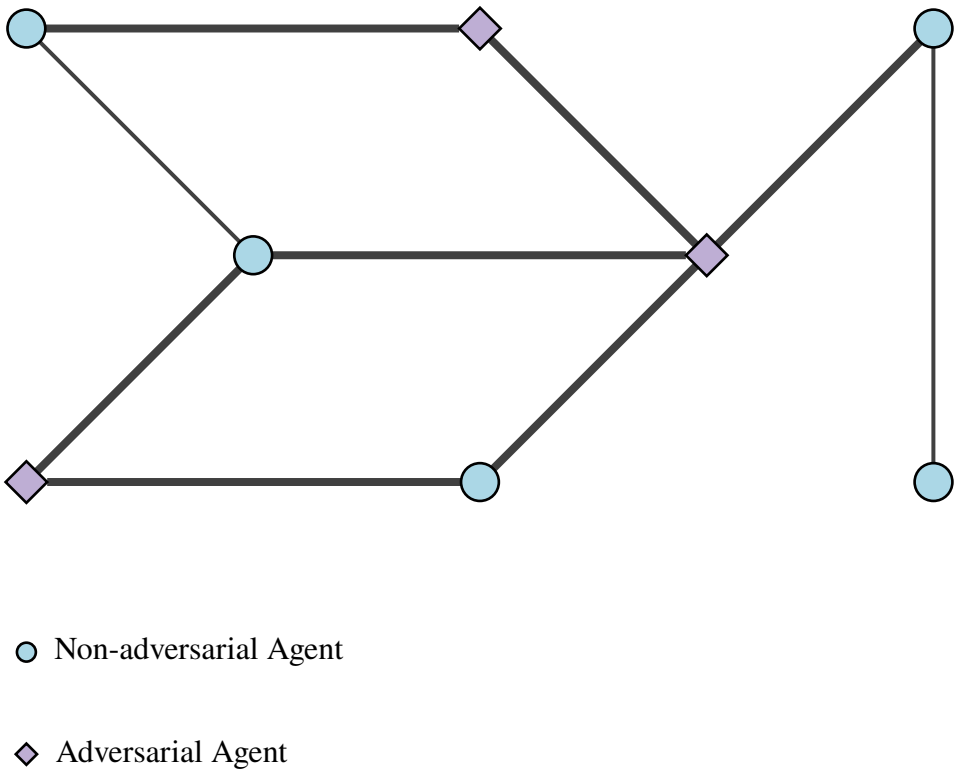

Figure 4.1: Adversarial behavior in WSN

out a local update according to:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) - \boldsymbol{q}_i(k) + \sum_{j \in N_i \cup \{i\}} w_{ij} \boldsymbol{q}_j(k) - \alpha_i \nabla f_i(\boldsymbol{x}_i(k)), \tag{4.2}$$

where $\alpha_i \in \mathbb{R}_+$ is the step size, w_{ij} is the $(i,j)^{th}$ element of the weight matrix, and $q_i(k) = Q_k^i(\boldsymbol{x}_i(k))$ is the quantized value of $\boldsymbol{x}_i(k)$. The attack model is explained in the next section.

Attack Model

The attack model used in this chapter is similar to adversarial machine learning where adversarial agents can attack a machine learning model at the training or testing stage by injecting false data in the dataset. Adversarial agents can attack such a model using different types of attack such as a black box where the attacker has no knowledge of the update, a grey box where the attacker has partial knowledge of the model and a white box where the attacker has complete knowledge of the update. In this chapter, it is assumed that adversarial agents have sufficient knowledge of the model needed to cause obstruction. Assumptions on the attack model are itemized below:

- Adversarial agents have sufficient knowledge of the algorithm and the objective of non-adversarial agents.
- Since adversarial agents are familiar with the problem adversarial agents are solving, they can perturb in a manner to result in a deviation of the optimal solution non-adversarial agents are trying to arrive at.
- Non-adversarial agents can detect adversarial agents in the network using a detection strategy that will be explored in the later part of the thesis.

The aim of the malicious agents is to prevent the network from reaching the optimal solution to Problem (4.1), by perturbing their estimates with either a positive or negative attack vector $e_i(k) \in \mathbb{R}^p$ (with all entries of the vector being positive or negative) according to the following update equation:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) - \boldsymbol{q}_i(k) \quad + \sum_{j \in N_i \cup \{i\}} w_{ij} \boldsymbol{q}_j(k) - \alpha_i \nabla f_i(\boldsymbol{x}_i(k)) + e_i(k). \tag{4.3}$$

After the step in Equation (4.3), the adversarial agents broadcast their estimates to their neighbors.

We note that the adversarial agents can choose either the same attack vector $e(k)$ or different attack vector $e_i(k)$ at every iteration in a general graphical structures where $e(k)$ is average of all the attack vectors $e_i(k)$. We analyze the general graphical structures scenario and details regarding these graphical structures in an adversarial case are seen in [35].

Now we explore the uniform quantization scheme the non-adversarial agents are using to manage the bandwidth used by adversarial agents in the next section.

The Uniform Quantizer

Let $\boldsymbol{x} \in \mathbb{R}^p$. A uniform quantizer with step size δ and mid-value $\boldsymbol{x'}$ is $\boldsymbol{Q}(x) = \boldsymbol{x'} + \text{sign}(\boldsymbol{x} - \boldsymbol{x'})\delta\lfloor \frac{\|\boldsymbol{x}-\boldsymbol{x'}\|}{\delta} + \frac{1}{2} \rfloor$, where $\delta = \frac{\ell}{2^b}$, ℓ is the size of the quantization interval, b is the number of bits, $\lfloor \cdot \rfloor$ is the floor function and sign($\boldsymbol{x}$) is the sign function. Let the quantization interval be set to $[\boldsymbol{x'} - 1/2, \boldsymbol{x'} + 1/2]$, and the uniform quantizer be denoted as $\boldsymbol{Q}^i_k(\boldsymbol{x}_i(k))$ with mid-value expressed as $\boldsymbol{x'}^k_{\Delta,i}$. Let the quantization outcome $\boldsymbol{q}_i(k) = \boldsymbol{Q}^i_k(\boldsymbol{x}_i(k))$; then, the quantization error, $\Delta_i(k)$ is given by $\Delta_i(k) = \boldsymbol{q}_i(k) - \boldsymbol{x}_i(k)$. Suppose ℓ_i is the quantization interval size for each agent i, the quantization error bound of a uniform quantizer is given by $\|\Delta_i(k)\| \leq \frac{\boldsymbol{\ell}_i}{2^{b+1}}$.

The following assumptions are made on Problem (4.1):

Assumption 4. *The cost function $f(\boldsymbol{x})$ in Problems* (4.1) *is strongly convex. This implies that for any vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, there exists a strong convexity parameter $\mu \in \mathbb{R}_+$, with $\mu \leq L$ (where L is the Lipschitz constant) such that: $f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$.*

Assumption 5. *The subgradient g_i of f_i at $\boldsymbol{x}_i$ is uniformly bounded by $\bar{L}_i$ in the feasible set $\mathcal{X}$. This implies that there exists $\bar{L}_i > 0$ such that $\|g_i(\boldsymbol{x}_i)\| \leq \bar{L}_i$, where for all $\boldsymbol{y}$, the relationship $f_i(\boldsymbol{y}) \geq f_i(\boldsymbol{x}) + g_i^T(\boldsymbol{y}_i - \boldsymbol{x}_i)$ holds.*

Assumption 6. *The attack vector $e_i(k)$ is uniformly bounded. This means that a constant $C > 0$ exists such that for all k, $\|e_i(k)\| \leq C$.*

Using the above assumptions, we show below that convergence to a neighborhood of the optimal solution is attained despite the two constraints of malicious attack and quantization described.

Distributed Subgradient Convergence Analysis with Quantization and Attack

We now proceed to the convergence analysis used to solve Problem (4.1). The goal is to analyze convergence to the optimal solution of the minimization problem in equation (4.1) in the presence of quantization and attack as described in section 4. In equation (4.2), each non-adversarial and adversarial agent i achieves consensus with other nodes in the network by taking a weighted average of its estimates and those of its neighbors. This averaged consensus includes non-adversarial and adversarial agents' estimates. Let $X = [\boldsymbol{x}_1;\ \boldsymbol{x}_2;\ \dots\ \boldsymbol{x}_n]^T \in \mathbb{R}^{np}$ be the concatenation of the local variables $\boldsymbol{x}_i$, and let I_p be the identity matrix of dimension p. When quantization occurs among agents during broadcasting of information, the quantized values can result in solutions that are not feasible when subjected to constraints. This results in an error when projected unto the feasible set. Suppose $\boldsymbol{h} \in \mathbb{R}^p$, let $\boldsymbol{\xi}(\boldsymbol{h})$ be the error based on projection of $\boldsymbol{h}$ in the feasible set $\mathcal{X}$, let $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1;\ \boldsymbol{\xi}_2;\ \dots\ \boldsymbol{\xi}_n]^T \in \mathbb{R}^{np}$ be the concatenation of $\boldsymbol{\xi}_i$ and H denotes the concatenation of the local variables of $\boldsymbol{h}_i$ with $\boldsymbol{\xi}(\boldsymbol{h}) = \boldsymbol{h} - [\boldsymbol{h}]_{\mathcal{X}}$. Another representation of Equation (4.2) is given by:

$$\boldsymbol{h}_i(k) = \sum_{j \in \mathcal{N}_i} w_{ij}\boldsymbol{x}_j(k) + \boldsymbol{x}_i(k) - \boldsymbol{q}_i(k) + \sum_{j \in \mathcal{N}_i} w_{ij}(\boldsymbol{q}_j(k) - \boldsymbol{x}_j(k)) - \alpha(k)\boldsymbol{g}_i(\boldsymbol{x}_i(k)),$$

where $\alpha(k)$ is an appropriately chosen step size. The iterative update equation is now given as:

$$\boldsymbol{x}_i(k+1) = [\boldsymbol{h}_i(k)]_{\mathcal{X}} = \boldsymbol{h}_i(k) - \boldsymbol{\xi}_i(\boldsymbol{h}_i(k)). \tag{4.4}$$

We obtain the matrix form of the above update to be the following:

$$\boldsymbol{H}(k) = \boldsymbol{W}\boldsymbol{X}(k)+(\boldsymbol{I}-\boldsymbol{W})(\boldsymbol{X}(k)-\boldsymbol{Q}(k))-\alpha(k)\boldsymbol{G}(\boldsymbol{X}(k)),$$

$$\boldsymbol{X}(k+1) = \boldsymbol{H}(k) - \boldsymbol{\Xi}(\boldsymbol{H}(k)), \tag{4.5}$$

where $\boldsymbol{W}$ is a doubly stochastic weight matrix. Suppose $\bar{\boldsymbol{x}}(k)$ and $\bar{\boldsymbol{\xi}}(k)$ are the mean of $\boldsymbol{x}_i(k)$ and $\boldsymbol{\xi}_i(h_i(k))$ respectively, we obtain $\bar{\boldsymbol{x}}(k) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i(k) = \frac{1}{n}\boldsymbol{X}^T\mathbf{1} \in \mathbb{R}^p$, and

$$\bar{\boldsymbol{\xi}}(k) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}_i(\boldsymbol{h}_i(k)) = \frac{1}{n}\left(\boldsymbol{\Xi}(k)\right)^T\mathbf{1} \in \mathbb{R}^p. \tag{4.6}$$

We can define the quantization error for each agent i as $\Delta_i(k) = \boldsymbol{x}_i(k) - q_i(k)$, the average of the errors as $\Delta(k) = \dfrac{1}{n}\sum_{i=1}^{n}\Delta_i(k)$ and the average of the attack as $\bar{e}(k) = \dfrac{1}{n}\sum_{i=1}^{n}e_i(k)$. We now obtain: $\bar{\boldsymbol{h}}(k) = \bar{\boldsymbol{x}}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))$, and $\bar{\boldsymbol{x}}(k+1) = \bar{\boldsymbol{h}}(k) - \bar{\boldsymbol{\xi}}(k)$. Thus, we obtain the iterative equation:

$$\bar{\boldsymbol{x}}(k+1) = \bar{\boldsymbol{x}}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k)) - \bar{\boldsymbol{\xi}}(k). \tag{4.7}$$

Now we introduce a Lemma that accounts for the bounds of the projection error according to equation (4.6).

Lemma 4.0.1. *Let Assumptions 4, 5 and 6 hold with $\Delta(k)$ being the average of the quantization errors. The error due to projection is bounded given by: $\|\bar{\boldsymbol{\xi}}(k)\| \leq \sqrt{8}\Delta(k) + \sqrt{2}\frac{\bar{L}}{n}\alpha$.*

Proof. See Appendix B. □

Convergence analysis will be examined by using the relationship: $\|x_i(k)-x^*\| = \|x_i(k)-\bar{\boldsymbol{x}}(k)\|+\|\bar{\boldsymbol{x}}(k)-x^*\|$. However, we must examine the bound on $\|x_i(k)-\bar{\boldsymbol{x}}(k)\|$ in the following Lemma and note that

the bound captures the information on the spectral gap.

Lemma 4.0.2. *Let Assumptions 4, 5 and 6 hold with β being the second biggest value of the eigenvalues of the weights W (which has one eigenvalue equal to* 1 *and others have values less than* 1*), the distance between the individual estimates and the averaged estimates is bounded by the following:*

$$\|x_i(k) - \bar{\boldsymbol{x}}(k)\| \leq \frac{\alpha \bar{L}_i}{1-\beta},$$

where α is the step size and $\bar{L}_i$ is the subgradient bound.

Proof. See [12] for the proof. □

Now we proceed to the main result in the next section.

Main Result

This section presents a convergence analysis based on the proposition in Sections 4, which is summarized in Theorem 4.0.3 below.

Theorem 4.0.3. *Let Assumptions 4, 5 and 6 hold, and suppose the step size α satisfy $\alpha < \frac{2}{\mu+L}$. Given that the size of a uniform quantization interval with b bits be upper-bounded by $\boldsymbol{\ell} \leq \frac{2^b}{\sqrt{6}}$, and the subgradient bound be upper-bounded by $\bar{L} \leq 1/\sqrt{6}\alpha$, then the iterates generated when non-adversarial send quantized estimates converge to a neighborhood of the optimal solution, x^* with the neighborhood size given by $\frac{\sqrt{6}(l+2^b\bar{L}\alpha)+2^b\sqrt{3}\|\bar{e}(k)\|}{2^b}$.*

Proof. To begin the proof, we first express the relationship between the optimal solutions of the adversarial and non-adversarial agents. Thereafter, the relationships are substituted in the convergence analysis of the proposed distributed optimization problem. Recall that the adversarial

agents are injecting the attack $e_i(k)$ and the non-adversarial agents have to manage the limited bandwidth left via quantization with parameters shown in Theorem 4.0.3. Let $e(k)$ be the average of $e_i(k)$ received from neighbors and let $\boldsymbol{x}^*$ be the optimal solution of (4.7) and $\boldsymbol{x}^a$ be the adversary according to $\boldsymbol{x}^a = \boldsymbol{x}^* + \bar{e}(k)$, then we obtain the following which will be used for the convergence analysis:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 = \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^* - \bar{e}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - \bar{\boldsymbol{\xi}}(k)\|^2. \tag{4.8}$$

By expanding equation (4.8), we obtain the following relationship:

$$\begin{aligned}
\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 &= \|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|^2 + \|\bar{e}(k)\|^2 + \left\|\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2 \\
&+ \|\bar{\boldsymbol{\xi}}(k)\|^2 + 2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)\right) - 2(\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)^T\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right) \\
&+ 2\bar{\boldsymbol{\xi}}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)\right) + 2\bar{e}(k)\bar{\boldsymbol{\xi}}(\boldsymbol{k}).
\end{aligned}$$

By inspection, the upper bound of the preceding expression $\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2$ has eight terms. In what follows, we bound some of the eight terms, starting with the fifth term.

$$\begin{aligned}
&2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)\right) \\
&\leq \|\bar{e}(k)\|^2 + \left\|\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)\right\|^2 - 2(\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)^T\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right), \\
&\leq \|\bar{e}(k)\|^2 + \frac{\alpha^2}{n^2}\left\|\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2 + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 - \frac{\alpha}{n}c_1\left\|\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2 - \alpha c_2\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2,
\end{aligned}$$

where $c_1 = \dfrac{2}{\mu + L}$ and $c_2 = \dfrac{2\mu L}{\mu + L}$. We proceed by bounding the second term of the derived upper

bound of $2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)$. In this regard, we have the following bound:

$$\left\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\| \leq \sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|. \tag{4.9}$$

By squaring both sides of the equation (4.9), we obtain: $\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|^2 \leq (\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|)^2$. Therefore we have the relationship:

$$\begin{aligned}
&2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)\\
&\quad\leq \|\bar{e}(k)\|^2+\frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2.
\end{aligned}$$

We recall that the expression $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2$ is upper-bounded by eight terms. Now we bound the seventh term, $2\bar{\boldsymbol{\xi}}(k)(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*))$ to obtain:

$$\begin{aligned}
&2\bar{\boldsymbol{\xi}}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)\\
&\quad\leq \|\bar{\boldsymbol{\xi}}(k)\|^2+\frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2.
\end{aligned}$$

Next we bound the expression $2\bar{e}(k)\bar{\boldsymbol{\xi}}(k)$ as $2\bar{e}(k)\bar{\boldsymbol{\xi}}(k)\leq\|\bar{e}(k)\|^2+\|\bar{\boldsymbol{\xi}}(k)\|^2$. Afterwards, we combine

all bounds and obtain the following:

$$
\begin{aligned}
&\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2 \\
&\leq \|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+\|\bar{e}(k)\|^2+\frac{\alpha^2}{n^2}(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|)^2+\|\bar{\boldsymbol{\xi}}(k)\|^2+\|\bar{e}(k)\|^2+\frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2 \\
&\quad-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+\|\bar{\boldsymbol{\xi}}(k)\|^2 \\
&\quad+\frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+\|\bar{e}_i(k)\|^2+\|\bar{\boldsymbol{\xi}}(k)\|^2. \\
&=(3-3\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+3\|\bar{\boldsymbol{\xi}}(\boldsymbol{k})\|^2+\left(\frac{3\alpha^2}{n^2}-\frac{3\alpha}{n}c_1\right)\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2.
\end{aligned}
$$

We will show that $\left(\frac{3\alpha^2}{n^2}-\frac{3\alpha}{n}c_1\right)\leq 0$, when $\alpha\leq c_1$. To do this, it suffices to show that $3\alpha^2-3\alpha nc_1\leq 0$. We know that the root of the equation in α of $\alpha(3\alpha-3nc_1)=0$ is $\alpha=0$ and $\alpha=nc_1$, and we have solution $0\leq\alpha\leq nc_1$. So $\alpha\in[0,nc_1]$. If $\alpha\leq c_1$ and $n\geq 1$, it implies that $\alpha\leq nc_1$. Alternatively, if $\alpha\leq c_1$, then $\alpha^2\leq\alpha c_1$ and consequently $\frac{3\alpha^2}{n^2}-\frac{3\alpha nc_1}{n}\leq 0$ for $n>0$. We now obtain $\left(\frac{3\alpha^2}{n^2}-\frac{3\alpha nc_1}{n}\right)\left(\sum_{i=1}^{n}\|\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|\right)^2\leq 0$. Therefore the bounds on $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}_i(k)\|^2$ is:

$$\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2\leq(3-3\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+3\|\bar{\boldsymbol{\xi}}(\boldsymbol{k})\|^2.$$

To show that $3-3\alpha c_2\geq 0$, we show that $3\alpha c_2\leq 3\Rightarrow\alpha c_2\leq 1$. If $\alpha\leq 1/\mu$, then we have the following: $\alpha c_2\leq\frac{1}{\mu}\frac{2\mu L}{\mu+L}=\frac{2L}{\mu+L}$. If $\mu=L$, then $\mu+L=2L\Rightarrow\alpha c_1\leq 1$. Since $\alpha c_2\leq 1$, then we have that $(3-3\alpha c_2)\geq 0$. For $3-3\alpha c_2$ not to grow unbounded we need $(3-3\alpha c_2)\in(0,1)$. This implies that we need $3(1-\alpha c_2)\leq 1$ or equivalently when $\alpha\geq\frac{2}{3c_2}=\frac{\mu+L}{3\mu L}$. If $\alpha\in\left(\frac{\mu+L}{3\mu L},\frac{1}{\mu}\right)$, then the expression $(3-3\alpha c_2)$ will not grow unbounded. From Lemma 4.0.1, we obtain: $\|\bar{\boldsymbol{\xi}}(k)\|^2\leq 8\|\Delta(k)\|^2+2\bar{L}^2\alpha^2$.

and it leads to the following:

$$\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2 \leq (3-3\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+24\|\Delta(k)\|^2+6\bar{L}^2\alpha^2.$$

This leads to following relationship:

$$\begin{aligned}\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2 &\leq (3-3\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+\frac{24(\boldsymbol{\ell})^2}{2^{2b+2}}+6\bar{L}^2\alpha^2,\\ &\leq (3-3\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+\frac{6(\boldsymbol{\ell})^2}{2^{2b}}+6\bar{L}^2\alpha^2.\end{aligned}$$

From the preceding relationship, the expression $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2$ needs to be bounded to an expression in terms of the expression $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2$. To achieve this goal, we need a condition such that if $\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^* < 0 < \bar{e}(k)$, then $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\| \geq \|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|$. In addition, to ensure that the algorithm also holds for negative attack vector, if $\bar{e}(k) \leq 0$, then $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\| \geq \|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|$ holds. Therefore, we obtain the following relationship:

$$\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2 \leq (3-3\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+\frac{6}{2^{2b}}(\boldsymbol{\ell})^2+6\bar{L}^2\alpha^2.$$

By applying recursion principles, we obtain the following bounds:

$$\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2 \leq (3-3\alpha c_2)^k\|\bar{\boldsymbol{x}}(0)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+\frac{6}{2^{2b}}(\boldsymbol{\ell})^2+6\bar{L}^2\alpha^2,$$

Equivalently, the following relationship holds:

$$\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\| \leq (3-3\alpha c_2)^{k/2}\|\bar{\boldsymbol{x}}(0)-\boldsymbol{x}^*\|+\sqrt{3}\|C\|+\sqrt{\frac{6}{2^{2b}}}\boldsymbol{\ell}+\sqrt{6}\bar{L}\alpha$$

where C is the upper bound on the attack vector. For $\sqrt{6/2^{2b}}\boldsymbol{\ell}$ to be small, we need $\sqrt{6/2^{2b}}\boldsymbol{\ell} \leq 1$, which implies that $\boldsymbol{\ell} \leq \frac{1}{\sqrt{\frac{6}{2^{2b}}}} = \frac{2^b}{\sqrt{6}}$. In addition, if $\sqrt{6}\bar{L}\alpha \leq 1$ or $\bar{L} \leq 1/\sqrt{6}\alpha$, the size of the

neighborhood is given by: $\frac{\sqrt{6}(l+2^b L\alpha)+2^b\sqrt{3}\|\bar{e}(k)\|}{2^b}$, and the non-adversarial agents can converge to the neighborhood of the optimal solution, $\boldsymbol{x}^*$. □

Remark 2. *There are trade-offs in the results obtained by combining adversarial attack and quantization as constraints. This is evident in the proof of Theorem 4.0.3, where we needed the condition* $\|\bar{\boldsymbol{x}}(0)-x^*\|<\|\bar{e}(k)\|$ *when* $\bar{e}(k)>0$ *for convergence. However, such condition is not needed when* $\bar{e}(k)\leq 0$*. Strong convexity assumption ensures that the error does not grow unbounded. Moreover, the gradient bound needs to depend on the step size as well to aid convergence. Although a recent result [35] shows that increasing the number of adversarial agents leads to an increase in the convergence neighborhood, when quantization is added as a constraint, increasing the number of bits leads to reduction of the error bounds. While the authors in [37, 39, 40] show that convergence of distributed gradient methods with quantization depends on the quantization levels and the number of bits, this proposed method adds adversarial attack to the constraint and still guarantee similar convergence attributes.*

Remark 3. *We note that the results presented in Theorem 4.0.3 align with the existing literature; for example, Theorem 4.4 in [8] is relatable in a manner where the authors established that a single adversarial node can cause all nodes to converge to any arbitrary value when they implement algorithm* (4.3)*. The presence of the attack vector can be related to the adversarial agents solving a different objective problem.*

In the next chapter, an adaptive quantization technique used by non-adversarial agents is explored to manage the significant bandwidth used by adversarial agents.

CHAPTER 5: DISTRIBUTED AND ADAPTIVE QUANTIZATION IN A NETWORK WITH ADVERSARIES

This chapter describes a distributed subgradient with adaptive quantization and adversarial attack method (DISGAQAAM) using an adaptive quantizer when non-adversarial agents try to survive the havoc caused by adversarial agents due to the injection of attack by the adversarial agents. A method is proposed in a manner that non-adversarial agents can detect the presence of an attack using a suitable detection mechanism which will be explored in this chapter. Resilience strategies against adversarial attacks are also explored in the course of this chapter. It is proved that under strong convexity of the objective function, convergence to a neighborhood of the optimal solution can still be obtained using a suitable step size.

Problem Formulation

For an undirected graph $G = (\mathcal{V}, \mathcal{E})$ comprising n nodes and $\mathcal{E}$ edges, we let $\mathcal{V} = 1, 2, ...n$ represent the set of nodes, $\mathcal{E} = (i, j)$ to be the set of edges and $N_i = \{j : (i, j) \in \mathcal{E}\}$ to be the neighborhood set. Let a closed convex set be defined as intersections of closed points in space that are on a side of a hyperplane. The agents are represented by nodes and their objective is to jointly solve the following:

$$\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) = \sum_{i=1}^{n} f_i(\boldsymbol{x}). \tag{5.1}$$

In equation (5.1), $\mathcal{X}$ is the closed, convex, feasible set and $\boldsymbol{x}$ is the decision variable. Each agent in the network holds some part $f_i(\cdot)$ of the strongly convex objective function $f(\cdot)$. The goal of the agents is to reach the optimal solution of problem (5.1), despite the injection of attack vectors

by some adversarial agents. We assume implicitly that the adversarial attacks are akin to denial of service attacks in which the resources for communication and coordination are jammed making the non-adversarial agents resort to compressing information being shared across the network to manage the limited bandwidth.

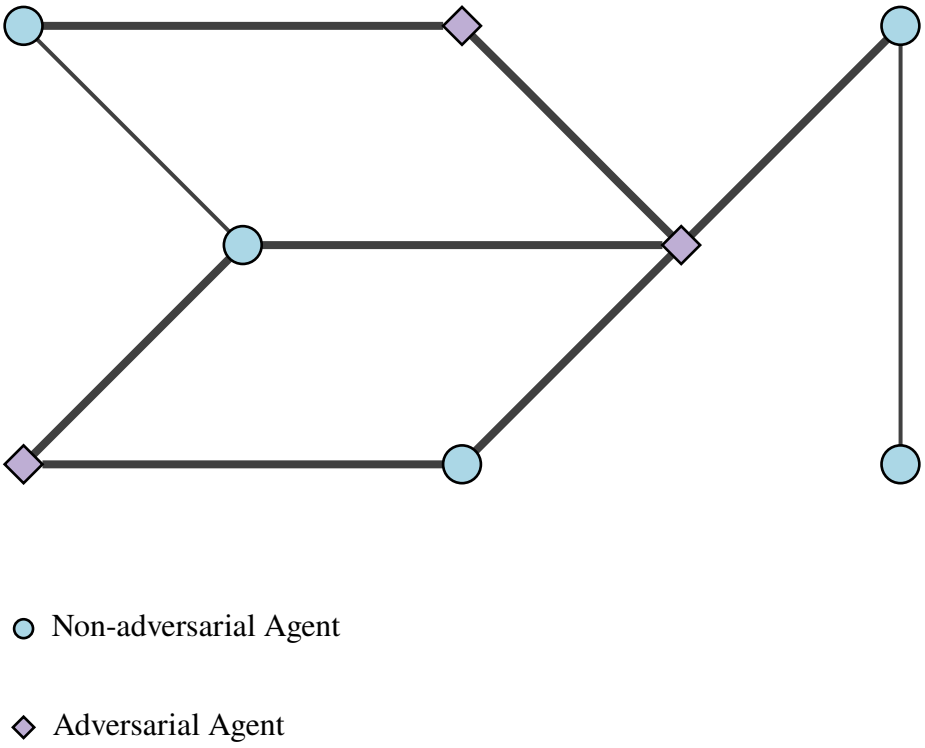

Figure 5.1: Adversarial behavior in WSN

A recent result [56] solves the illustrated problem in Figure 5.1 using a fixed quantizer but the crux of this paper is to solve the problem using an adaptive quantizer. As seen in Figure 5.1, the thin communication links represent the connection between non-adversarial agents, while thick pipes can either represent the connection between a malicious agent and a non-adversarial agent or the connection between a malicious agent and any other malicious agent in the network. The goal of the non-adversarial is to utilize the limited communication bandwidth left in order to reach

the desired optimal solution to problem (5.1). An adaptive quantization framework is used by the non-adversarial agents to manage the limited bandwidths due to the injected attack to the network caused by malicious agents.

Suppose each non-adversarial agent broadcast quantized information $Q(\boldsymbol{x}_i(k)) \in \mathbb{R}^p$ to their neighbors, the local update equation results in the following:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) - \boldsymbol{q}_i(k) + \sum_{j \in N_i \cup \{i\}} w_{ij} \boldsymbol{q}_j(k) - \alpha_i \nabla f_i(\boldsymbol{x}_i(k)). \tag{5.2}$$

In Equation (5.2), $\boldsymbol{x}_i \in \mathbb{R}^p$ is the estimate of agent i, $q_i(k) = Q_k^i(\boldsymbol{x}_i(k))$ is the quantized estimates non-adversarial agents broadcast to neighbors, w_{ij} is the $(i,j)^{th}$ element of the doubly stochastic weight matrix, and $\alpha_i \in \mathbb{R}_+$ is the step size.

The Adaptive Quantizer

In contrast to the fixed quantizer studied in chapter 4 section 4, the adaptive quantizer described in this chapter serves as a preventive mechanism for the bandwidth used up by adversarial agents due to their attack. The adaptive quantizer addressed in this chapter is a variation of the uniform quantizer except that it is made adaptive to allow the non-adversarial nodes to compensate for the adversarial nodes utilizing too much bandwidth.

We start by explaining the idea behind the uniform quantizer and we build up on that to derive the adaptive version of it.

Let $\boldsymbol{x} \in \mathbb{R}^p$, we define a uniform quantizer with resolution δ and mid-value $\boldsymbol{x'}$ as

$$\boldsymbol{Q}(x) = \boldsymbol{x'} + \text{sign}(\boldsymbol{x} - \boldsymbol{x'})\delta \lfloor \frac{\|\boldsymbol{x} - \boldsymbol{x'}\|}{\delta} + \frac{1}{2} \rfloor,$$

where sign($\boldsymbol{x}$) and $\lfloor\cdot\rfloor$ are the sign and floor function respectively. Let b be the number of bits sent by the non-adversarial agents. Then the resolution is expressed by the relationship $\delta = \frac{\ell}{2^b}$, where ℓ is the size of the quantization interval; and the quantization interval is set to $[\boldsymbol{x'} - 1/2, \boldsymbol{x'} + 1/2]$ with mid value $\boldsymbol{x'}^k_{\Delta,i}$. The quantization error obtained for each non-adversarial agent i of a uniform quantizer is: $\Delta_i(k) = \boldsymbol{q}_i(k) - \boldsymbol{x}_i(k)$. For clarity, if ℓ_i is the quantization interval for each agent i, the error bounds of a uniform quantizer is given by $\|\Delta_i(k)\| \leq \frac{\boldsymbol{\ell}_i}{2^{b+1}}$.

We make the quantizer adaptive to manage the bandwidth used by the adversarial agents by using a variation between the attack vector and the parameters of the quantizer. Let the variable $z_i(k)$ denote the magnitude of the attack vector (or correspondingly the amount of bandwidth used). If $z_i(k)$ is large, we make δ a low resolution and if $z_i(k)$ is small (or low bandwidth is used), we make δ high resolution. Consequently, agent i can detect the presence of $z_i(k)$ by using an outlier test according to $\delta \propto \frac{1}{z_i(k)}$.

Attack Model

The main objective of the adversarial agents is to inject positive and negative attack vector values, $e_i(k) \in \mathbb{R}^p$ to the network to prevent the non-adversarial agents from reaching the optimal solution of equation (5.1). In this chapter, it is assumed that adversarial agents have complete knowledge of what the non-adversarial agents are trying to accomplish which in this case is obtaining the optimal solution of problem (5.1). Assumptions on the attack model are indicated below:

- Adversarial agents have complete knowledge of the algorithm and the objective of non-adversarial agents.
- Consequently adversarial agents can perturb to result in a deviation of the optimal solution non-adversarial agents are trying to obtain.

- Non-adversarial agents can detect adversarial agents in the network using a detection strategy that will be explored later in the chapter.

When adversarial agents inject attack, they perturb and broadcast their estimates by the relationship $x_j + e_i(k)$ where $e_i(k)$ is the attack vector. Consequently, this also leads to the perturbation in the optimal solution. This attack framework holds for complete and general graph structures and as described, the non-adversarial agents receives the update and broadcast quantized information to neighbors according to the following update:

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) - \boldsymbol{q}_i(k) + \sum_{j \in N_i \cup \{i\}} w_{ij} \boldsymbol{q}_j(k) - \alpha_i \nabla f_i(\boldsymbol{x}_i(k)) + \boldsymbol{e}_i(k). \tag{5.3}$$

We note that equation (5.3) is the update equation for all agents and that only the adversarial agents inject attack in the network while the non-adversarial agents manage the bandwidth used by adversarial agents via quantization. After the step in Equation (5.3), an iterative exchange of information occurs amongst the agents until convergence is attained.

Now, we state the assumptions needed to prove the convergence of the optimization problem (5.1).

Assumption 7. *The cost function $f(\boldsymbol{x})$ in equation* (5.1) *is strongly convex, meaning that for any vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, there exists a strong convexity constant, $\mu \in \mathbb{R}_+$ (where $\mu \leq L$), such that: $f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^T(\boldsymbol{x} - \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$. where L represents the Lipschitz constant.*

Assumption 8. *The gradient $\boldsymbol{g}_i(\boldsymbol{x}_i(k)) = \nabla f_i(\boldsymbol{x}_i(k))$ of the objective f_i at $\boldsymbol{x}_i$ is uniformly bounded by $\bar{L}_i$ in the feasible set $\mathcal{X}$. This means that there exists $\bar{L}_i > 0$ such that $\|g_i(\boldsymbol{x}_i)\| \leq \bar{L}_i$.*

Assumption 9. *The attack vector $e_i(k)$ is uniformly bounded. This means that a constant $C > 0$ exists such that for all k, $\|e_i(k)\| \leq C$.*

Consensus Update Equation for Non-Adversarial Agents

We re-write the update equation in (5.2) by ensuring feasibility of the error due to projection is taken into account. Let $X = [\boldsymbol{x}_1;\ \boldsymbol{x}_2;\ \dots\ \boldsymbol{x}_n]^T \in \mathbb{R}^{np}$, $\boldsymbol{G} = [\boldsymbol{g}_1;\ \boldsymbol{g}_2;\ \dots\ \boldsymbol{g}_n]^T \in \mathbb{R}^{np}$ and Δ are the concatenation of $\boldsymbol{x}_i$, $\boldsymbol{g}_i$ and Δ_i respectively, and I_p is the identity matrix of dimension p. As described, quantization of estimates by non-adversarial agents can lead to solutions not feasible especially when subject to constraints according to problem (5.1). Hence, an error due to projection unto the feasible set $\mathcal{X}$, is taken into consideration in the analysis. Let $\boldsymbol{h} \in \mathbb{R}^p$ and $\boldsymbol{\xi}(\boldsymbol{h})$ be the error due to projection of $\boldsymbol{h}$ in $\mathcal{X}$. Suppose $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1;\ \boldsymbol{\xi}_2;\ \dots\ \boldsymbol{\xi}_n]^T \in \mathbb{R}^{np}$ and $\boldsymbol{H}$ denote the concatenation of $\boldsymbol{\xi}_i$ and $\boldsymbol{h}_i$ respectively with $\boldsymbol{\xi}(\boldsymbol{h}) = \boldsymbol{h} - [\boldsymbol{h}]_{\mathcal{X}}$, we can re-formulate Equation (5.2) as:

$$\boldsymbol{x}_i(k+1) = [\boldsymbol{h}_i(k)]_{\mathcal{X}} = \boldsymbol{h}_i(k) - \boldsymbol{\xi}_i(\boldsymbol{h}_i(k)). \tag{5.4}$$

The matrix form of the above update equation is:

$$\boldsymbol{H}(k) = \boldsymbol{W}\boldsymbol{X}(k) + (\boldsymbol{I} - \boldsymbol{W})(\boldsymbol{X}(k) - \boldsymbol{Q}(k)) - \alpha(k)\boldsymbol{G}(\boldsymbol{X}(k)),$$

$$\boldsymbol{X}(k+1) = \boldsymbol{H}(k) - \boldsymbol{\Xi}(\boldsymbol{H}(k)), \tag{5.5}$$

provided $\boldsymbol{W}$ is a doubly stochastic weight matrix.

We introduce some variables to denote the average of the estimates of $\boldsymbol{x}_i(k)$, $\boldsymbol{\xi}_i(\boldsymbol{h}_i(k))$, $\Delta_i(k)$ and $e_i(k)$ as follows:

$$\bar{\boldsymbol{x}}(k) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i(k) = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{1} \in \mathbb{R}^p,$$

$$\bar{\boldsymbol{\xi}}(k) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\xi}_i(\boldsymbol{h}_i(k)) = \frac{1}{n}\left(\boldsymbol{\Xi}(k)\right)^T \mathbf{1} \in \mathbb{R}^p, \tag{5.6}$$

$$\bar{\Delta}(k) = \frac{1}{n}\sum_{i=1}^{n} \Delta_i(k).$$

$$\bar{e}(k) = \frac{1}{n}\sum_{i=1}^{n} e_i(k).$$

This leads to the relationship: $\bar{\boldsymbol{h}}(k) = \bar{\boldsymbol{x}}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))$, and $\bar{\boldsymbol{x}}(k+1) = \bar{\boldsymbol{h}}(k) - \bar{\boldsymbol{\xi}}(k)$. Consequently, we obtain the following update:

$$\bar{\boldsymbol{x}}(k+1) = \bar{\boldsymbol{x}}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - \bar{\boldsymbol{\xi}}(k). \tag{5.7}$$

The following Lemmas are essential to prove algorithm (5.1). First, we begin with the error due to projection bounds.

Lemma 5.0.1. *Suppose Assumptions 7, 8 and 9 hold, the error due to projection satisfy the bounds:*

$$\|\bar{\boldsymbol{\xi}}(k)\| \leq \sqrt{8}\bar{\Delta}(k) + \sqrt{2}\frac{\bar{L}}{n}\alpha(k).$$

Proof. Refer to [56] for the proof. □

Lemma 5.0.2. *Let Assumptions 7, 8 and 9 hold with the resolution of a uniform quantizer satisfying* $\delta = \frac{p}{z_i(k)}$ *(adaptive quantizer) where p is the constant of proportionality and* $\delta = \frac{l_i}{2^b}$*, the quantization error is bounded by:* $\Delta_i(k) \leq \frac{p}{2z_i(k)}$.

Proof. There are two different expressions for the resolution of the uniform quantizer; one from the adaptive quantizer in terms of the attack vector and the other from the general resolution formula of a uniform quantizer. By setting the two relationships of the resolution to each other, we obtain

the relationship:

$$\frac{p}{z_i(k)} = \frac{l_i}{2^b}. \tag{5.8}$$

By multiplying both sides of equation (5.8) by $\frac{1}{2}$, we obtain the relationship $\frac{p}{2z_i(k)} = \frac{l_i}{2^{b+1}}$. By using the fact that the upper bound of a uniform quantizer is $\Delta_i \leq \frac{\delta}{2} = \frac{l_i}{2^{b+1}}$, we obtain the result $\Delta_i(k) \leq \frac{p}{2z_i(k)}$. □

Convergence to a neighborhood of the optimal solution will be proved by using the relationship: $\|x_i(k) - x^*\| = \|x_i(k) - \bar{\boldsymbol{x}}(k)\| + \|\bar{\boldsymbol{x}}(k) - x^*\|$. First, we examine the bound on $\|x_i(k) - \bar{\boldsymbol{x}}(k)\|$ and we note that the bound captures the information on the spectral gap.

Lemma 5.0.3. *Let Assumptions 7, 8 and 9 hold with β being the second biggest value of the eigenvalues of the weights W (which has one eigenvalue equal to* 1 *and others have values less than* 1*), the distance between the individual estimates and the averaged estimates is bounded by the following:*

$$\|x_i(k) - \bar{\boldsymbol{x}}(k)\| \leq \frac{\alpha_i(k)\bar{L}_i}{1-\beta}. \tag{5.9}$$

Proof. See [12] for the proof. □

Now we analyze convergence of our proposed method (DISGAQAAM) in the next section.

Main Result

We examine the convergence to the neighborhood of the optimal solution of problem (5.1) and update (5.3) using the distributed gradient descent algorithm. In the following analysis, we account for the impact of non-adversarial agents (via the attack vector) on the quantized update in equation (5.7). The analysis is in two parts: First, each agent i sends its estimate (quantized or malicious)

to his neighbors after which a consensus is reached. For the purpose of this analysis, we assume that consensus is reached when the distance between each individual agent's estimates and the average estimates is bounded according to Lemma 5.0.3. Second, we prove convergence to the neighborhood of the optimal solution, x^* by bounding the distance between the averaged consensus value and the optimal solution x^*.

The main result of the claims in Section 5 are formally stated in Theorem 5.0.4.

Theorem 5.0.4. *Suppose Assumptions 7, 8 and 9 are satisfied, with α satisfying $\alpha \leq \frac{2}{\mu+L}$. If the gradient bound is upper-bounded by $\bar{L} \leq 1/\sqrt{6}\alpha$, the iterates converge to a neighborhood of the optimal solution, x^* when $\|\bar{\boldsymbol{x}}(0) - \boldsymbol{x}^*\| \leq \bar{e}(k)$. The size of the neighborhood is given by: $\sqrt{3}\|\bar{e}(k)\| + \sqrt{\frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2} + \sqrt{6}\bar{L}\alpha$. If however $\|\bar{\boldsymbol{x}}(0) - \boldsymbol{x}^*\| > \bar{e}(k)$, then adversarial agents can force divergence or prevent convergence to the optimal solution.*

Proof. We begin by noting that the adversarial agents solve the objective function $\min_x \hat{f}(\boldsymbol{x})$, such that the optimal solution to $\hat{f}(x)$ is $x^a = x^* + \bar{e}(k)$. We show convergence to a neighborhood of the optimal solution by establishing the relationship $\|x_i(k) - x^*\| = \|x_i(k) - \bar{\boldsymbol{x}}(k)\| + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$. The analysis of $\|x_i(k) - \bar{\boldsymbol{x}}(k)\|$ is seen in Lemma 5.0.3. Now we show the analysis of $\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$ below.

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 = \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^* - \bar{e}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - \bar{\boldsymbol{\xi}}(k)\|^2.$$

Alternatively, we can rewrite $\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2$ as:

$$\begin{aligned}
&\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 \\
&= \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + \|\bar{e}(k)\|^2 + \left\|\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2 + \|\bar{\boldsymbol{\xi}}(k)\|^2 + 2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)\right) \\
&\quad - 2(\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)^T\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k))\right) + 2\bar{\boldsymbol{\xi}}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)\right) + 2\bar{e}(k)\bar{\boldsymbol{\xi}}(\boldsymbol{k}).
\end{aligned}$$

We will bound the last four terms of the preceding expression starting with the fifth term as follows.

$$
\begin{aligned}
&2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)\\
&\quad\leq\|\bar{e}(k)\|^2+\left\|\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right\|^2-2(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)^T\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right),\\
&\quad\leq\|\bar{e}(k)\|^2+\frac{\alpha^2}{n^2}\left\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2,
\end{aligned}
$$

with $c_1=\dfrac{2}{\mu+L}$ and $c_2=\dfrac{2\mu L}{\mu+L}$. We now bound the second term of the upper bound of $2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)$. However, we need to first bound $\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|$ as the following:

$$
\left\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|\leq\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|. \tag{5.10}
$$

Equation (5.10) can be re-written as: $\|\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|^2\leq(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|)^2$ which leads to the expression:

$$
\begin{aligned}
&2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)\\
&\quad\leq\|\bar{e}(k)\|^2+\frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2.
\end{aligned}
$$

We note that $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2$ is upper-bounded by eight terms. Now, the seventh term, $2\bar{\boldsymbol{\xi}}(k)(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*))$ can be bounded as:

$$
\begin{aligned}
&2\bar{\boldsymbol{\xi}}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)\\
&\quad\leq\|\bar{\boldsymbol{\xi}}(k)\|^2+\frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2+\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2-\frac{\alpha}{n}c_1\left(\sum_{i=1}^{n}\|g_i(\boldsymbol{x}_i(k))\|\right)^2-\alpha c_2\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2.
\end{aligned}
$$

Similarly, we bound $2\bar{e}(k)\bar{\boldsymbol{\xi}}(k)$ as $2\bar{e}(k)\bar{\boldsymbol{\xi}}(k) \leq \|\bar{e}(k)\|^2 + \|\bar{\boldsymbol{\xi}}(k)\|^2$. Combining all bounded expressions leads to:

$$
\begin{aligned}
&\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 \\
&\leq \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + \|\bar{e}(k)\|^2 + \frac{\alpha^2}{n^2}\Big(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\Big)^2 + \|\bar{\boldsymbol{\xi}}(k)\|^2 + \|\bar{e}(k)\|^2 + \frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\right)^2 + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 \\
&\quad - \frac{\alpha}{n} c_1 \left(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\right)^2 - \alpha c_2 \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 - \frac{\alpha}{n} c_1 \left(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\right)^2 - \alpha c_2 \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + \|\bar{\boldsymbol{\xi}}(k)\|^2 \\
&\quad + \frac{\alpha^2}{n^2}\left(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\right)^2 + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 - \frac{\alpha}{n} c_1 \left(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\right)^2 - \alpha c_2 \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + \|\bar{e}(k)\|^2 + \|\bar{\boldsymbol{\xi}}(k)\|^2, \\
&= (3 - 3\alpha c_2)\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + 3\|\bar{e}(k)\|^2 + 3\|\bar{\boldsymbol{\xi}}(\boldsymbol{k})\|^2 + \left(\frac{3\alpha^2}{n^2} - \frac{3\alpha}{n} c_1\right)\left(\sum_{i=1}^{n} \|g_i(\boldsymbol{x}_i(k))\|\right)^2.
\end{aligned}
$$

It can be easily verified that $\left(\frac{3\alpha^2}{n^2} - \frac{3\alpha}{n} c_1\right) \leq 0$, provided that $\alpha \leq c_1$. We do this by equivalently showing that $3\alpha^2 - 3\alpha n c_1 \leq 0$. Since $\alpha \leq c_1$ and $n \geq 1$, then α can be upper-bounded according to $\alpha \leq n c_1$. Equivalently, $\alpha^2 \leq \alpha c_1$ and $\frac{3\alpha^2}{n^2} - \frac{3\alpha n c_1}{n} \leq 0$, for $n > 0$. This leads to the following affirmation:

$$
\left(\frac{3\alpha^2}{n^2} - \frac{3\alpha n c_1}{n}\right)\left(\sum_{i=1}^{n} \|\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|\right)^2 \leq 0,
$$

from which we then obtain the bounds on $\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \boldsymbol{e}_i(k)\|^2$ as the relationship below:

$$
\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 \leq (3 - 3\alpha c_2)\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + 3\|\bar{e}(k)\|^2 + 3\|\bar{\boldsymbol{\xi}}(\boldsymbol{k})\|^2.
$$

We need conditions such that $3 - 3\alpha c_2 \geq 0$ by proving that $3\alpha c_2 \leq 3 \Rightarrow \alpha c_2 \leq 1$. If $\alpha \leq 1/\mu$, $\alpha c_2 \leq \frac{1}{\mu}\frac{2\mu L}{\mu + L} = \frac{2L}{\mu + L}$. If $\mu = L$, it implies that $\mu + L = 2L \Rightarrow \alpha c_1 \leq 1$. Since it has been established that $\alpha c_2 \leq 1$, then $(3 - 3\alpha c_2) \geq 0$. We need to ensure that $3 - 3\alpha c_2$ does not grow unbounded by choosing $3 - 3\alpha c_2$ in the domain $(3 - 3\alpha c_2) \in (0, 1)$, implying that $3(1 - \alpha c_2) \leq 1$ or $\alpha \geq \frac{2}{3c_2} = \frac{\mu + L}{3\mu L}$. If $\alpha \in \left(\frac{\mu + L}{3\mu L}, \frac{1}{\mu}\right)$, $(3 - 3\alpha c_2)$ will be bounded. By using the results from Lemma 5.0.1, we have:

$\|\bar{\boldsymbol{\xi}}(k)\|^2 \leq 8\|\bar{\Delta}(k)\|^2 + 2\bar{L}^2\alpha^2$. which yields:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 \leq (3 - 3\alpha c_2)\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + 3\|\boldsymbol{e}_i(k)\|^2 + 24\|\bar{\Delta}(k)\|^2 + 6\bar{L}^2\alpha^2.$$

By using the results from Lemma 5.0.2, specifically with $\Delta_i(k) = \frac{p}{2z_i(k)}$, we obtain the following relationship:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 \leq (3 - 3\alpha c_2)\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + 3\|\bar{e}(k)\|^2 + \frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2 + 6\bar{L}^2\alpha^2.$$

We note that in equation (5.8), $z_i(k)$ can be expressed in terms of the number of bits b by using the relationship $z_i(k) = \frac{2^b}{l_i}$. If $\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* < 0 < \bar{e}(k)$, we obtain the following relationship:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\| \geq \|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|.$$

Furthermore, $\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\| \geq \|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|$ if $\boldsymbol{e}_i(k) \leq 0$. We now obtain the relationship:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|^2 \leq (3 - 3\alpha c_2)\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + 3\|\bar{e}(k)\|^2 + \frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2 + 6\bar{L}^2\alpha^2.$$

After applying recursion, we obtain the following inequality bound:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\| \leq (3 - 3\alpha c_2)^{\frac{k}{2}}\|\bar{\boldsymbol{x}}(0) - \boldsymbol{x}^*\| + \sqrt{3}\|C\| + \sqrt{\frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2} + \sqrt{6}\bar{L}\alpha.$$

where C is the bound on the attack vector. If $\sqrt{6}\bar{L}\alpha \leq 1$ or $\bar{L} \leq 1/\sqrt{6}\alpha$, the neighborhood size is given by the following:

$$\sqrt{3}\|\bar{e}(k)\| + \sqrt{\frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2} + \sqrt{6}\bar{L}\alpha.$$

Using the results from 5.0.3 and 5, we obtain that the non-adversarial agents converge to the neighborhood of the optimal solution, $\boldsymbol{x}^*$ according to $\|x_i(k) - x^*\| = \|x_i(k) - \bar{\boldsymbol{x}}(k)\| + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$.

If $\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* > \bar{e}(k)$, we obtain the relationship:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\| \leq \|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|.$$

which leads to the relationship:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|^2 \geq (3 - 3\alpha c_2)\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + 3\|\bar{e}(k)\|^2 + \frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2 + 6\bar{L}^2\alpha^2.$$

After applying recursion, we obtain the following inequality bound:

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\| \geq (3 - 3\alpha c_2)^{\frac{k}{2}}\|\bar{\boldsymbol{x}}(0) - \boldsymbol{x}^*\| + \sqrt{3}\|C\| + \sqrt{\frac{24}{n}\|\sum_{i=1}^{n} \frac{p}{2z_i(k)}\|^2} + \sqrt{6}\bar{L}\alpha.$$

Consequently, adversarial agents can force divergence and prevent convergence to the optimal solution x^*. □

Numerical experiments to bolster the proposed claims are shown in the next section.

Numerical Experiments

We validate our theoretical claims via numerical experiments over a network of $n = 10$ agents (some adversarial) and 100 iterations using the linear regression loss function as follows:

$$\min_{\boldsymbol{x} \in \mathbb{R}^p} f(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{1}{2}\|A_i\boldsymbol{x} - b_i\|^2, \tag{5.11}$$

solved in a distributed manner. In equation (5.11), the number of agents, $n = 10$, A_i is a n by n matrix and b_i is an n by 1 matrix. The gradient of the the function in equation (5.11) is $A^T(A_i\boldsymbol{x} - b_i)$. It is established that the Lipschitz constant L is the largest eigenvalue of A^TA and the strong convexity

constant μ is the smallest eigenvalue of $A^T A$. In this regard, we choose a step size such that $\alpha < \frac{2}{\mu+L}$ to satisfy the strong convexity assumptions.

We will show convergence to the neighborhood of the optimal solution despite the presence of adversaries and an adaptive quantizer using the error. Different proportions of adversarial agents will be used in the simulations to affirm the influence of the adversaries on the algorithm in the communication-constrained network. The resolution of the adaptive quantizer used in the simulations varies inversely as the size of the attack according to Lemma 5.0.2

Remark 4. *As seen in Figures 5.2, 5.3 and 5.4, the error is least when there are less adversarial agents and most when there are more adversarial agents in the network. Clearly, the communication-constrained algorithm still converges to a neighborhood of the optimal solutions even amidst the presence of adversarial agents. The relationship between the resolution of the quantizer and the size of the attack vector ensures that the algorithm does not significantly diverge even when the network is flooded with adversarial attack.*

Remark 5. *Clearly, the size of the neighborhood of the optimal solution depends on the attack vector, the gradient bound, number of bits and the step size. The results in this paper still holds for both positive and negative attack vector even with the adaptive quantization added as constraints. It is seen that the adversarial agents greatly influence the convergence of the proposed algorithm. If there were no misbehaviour by the adversarial agents, then a faster rate of convergence (be it linear or better) would have been accomplished. Additionally, the results in this paper holds when the objective function is strongly convex quadratic.*

So far in this dissertation, it has been examined how non-adversarial agents can approach a neighborhood of the optimal solution despite the presence of adversarial agents in a limited communication niche. Next, a proposition to detect adversarial agents is shown in the following section.

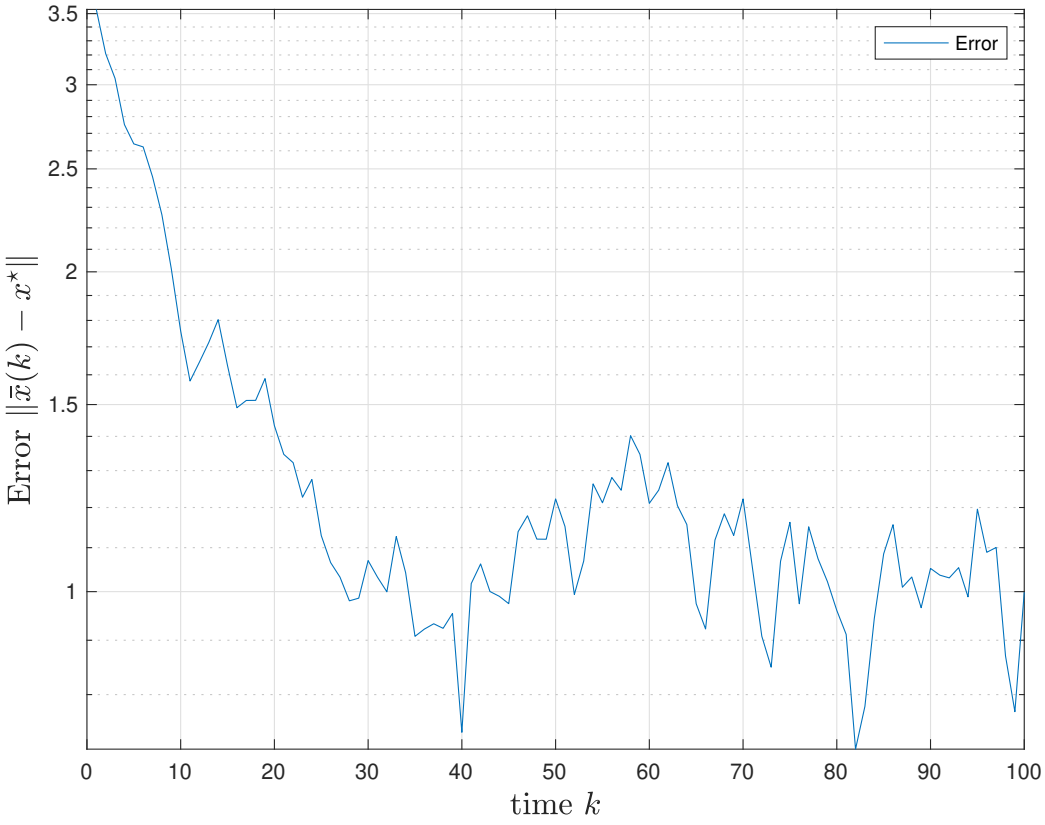

Figure 5.2: Adaptive Simulations for 9 non-adversarial agents and 1 malicious agent.

Proposition to Detect Malicious Agents

We examine how adversarial agents can be detected in the network by considering two different update equation. If the neighbor of agent i is a non-adversarial agent, we use the following update equation:

$$\boldsymbol{x}_i(k+1) = \sum_{j \in N_i \cup \{i\}} w_{ij} \boldsymbol{x}_j(k) - \alpha_i \nabla f_i(\boldsymbol{x}_i(k)) \tag{5.12}$$

If the neighbor of agent i is an adversarial agent, the update in equation (5.3) holds.

We will use the error difference between the neighborhood of the update equations (5.12) and (5.3)

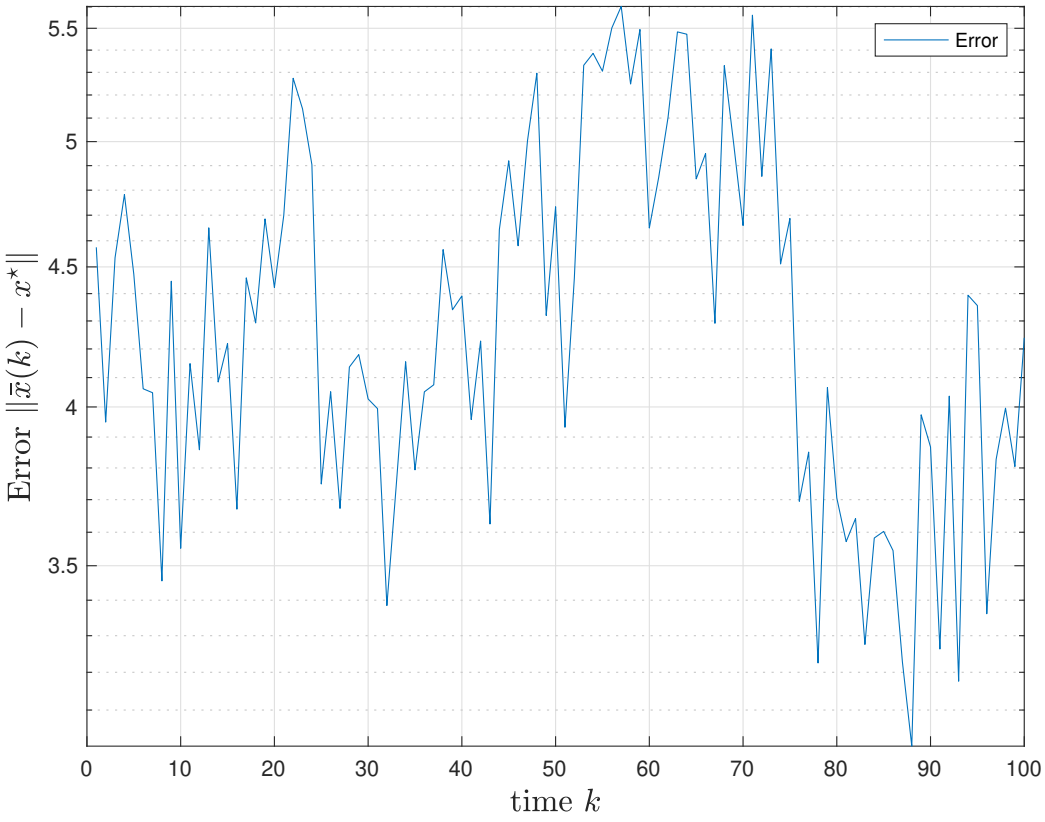

Figure 5.3: Adaptive Simulations for 5 non-adversarial agents and 5 malicious agents.

to derive a bound on the attack vector which will be used as a metric for detection. The analyses for a scenario when the neighbor of agent i is a non-adversarial agent is shown in the following theorem.

Theorem 5.0.5. *Let the neighbor of agent i be a non-adversarial agent and suppose Assumptions 7, 8 and 9 are satisfied, with α satisfying $\alpha \leq \frac{2}{\mu+L}$. If the gradient bound is bounded by $\bar{L} \leq 1/\sqrt{6}\alpha$, the iterates converge to a neighborhood of the optimal solution, x^* and the size of the neighborhood is given by: $\sqrt{24}\|\bar{\Delta}(k)\|^2+\sqrt{6}\bar{L}\alpha$.*

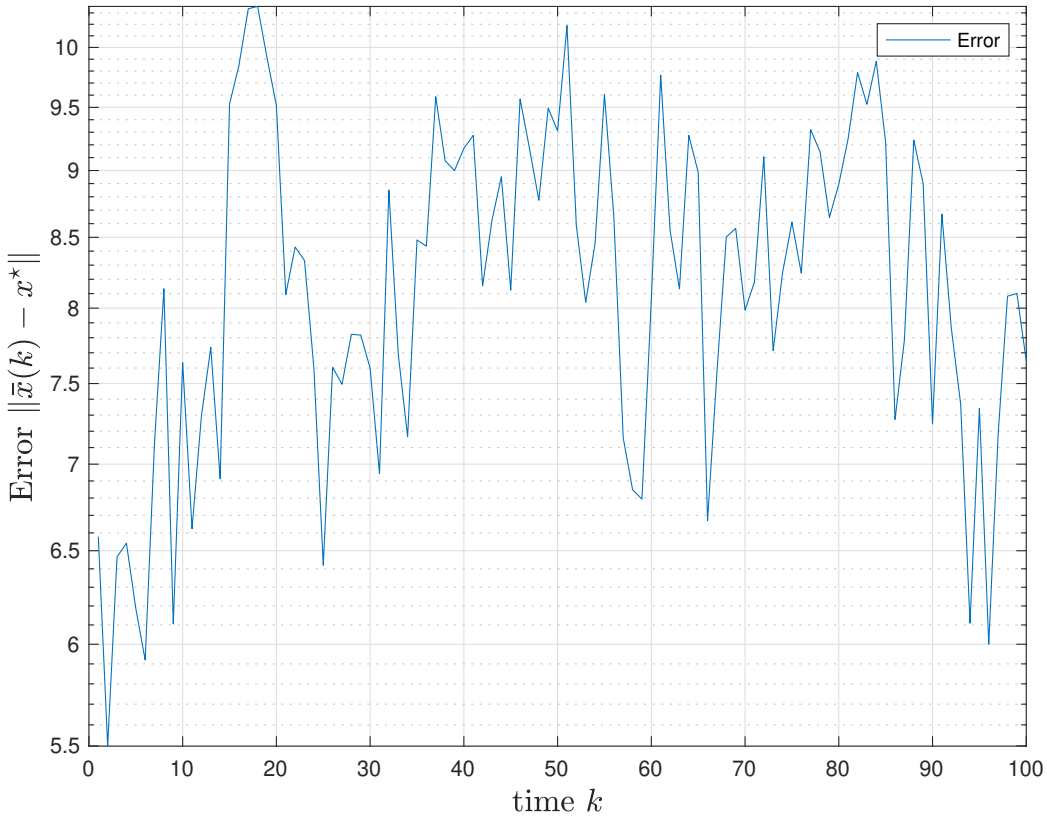

Figure 5.4: Adaptive Simulations for 1 non-adversarial agent and 9 malicious agents.

Proof. We show convergence to a neighborhood of the optimal solution by establishing the relationship $\|x_i(k) - x^*\| = \|x_i(k) - \bar{\boldsymbol{x}}(k)\| + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$. The analysis of $\|x_i(k) - \bar{\boldsymbol{x}}(k)\|$ is already shown in Lemma 5.0.3. Now, we analyze $\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$ below.

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^*\|^2 = \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^* - \frac{\alpha(k)}{n} \sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - \bar{\boldsymbol{\xi}}(k)\|^2.$$

Alternatively, we can rewrite $\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2$ as:

$$\begin{aligned}
&\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2 \\
&= \|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2+\left\|\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2+\|\bar{\boldsymbol{\xi}}(k)\|^2-2(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)^T\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right) \\
&\quad -2(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)^T\bar{\boldsymbol{\xi}}(\boldsymbol{k})+2\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right)\bar{\boldsymbol{\xi}}(\boldsymbol{k}).
\end{aligned}$$

By using strong convexity and simplification of like terms, we obtain the following bounds:

$$\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2\leq(2-\alpha c_2)\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+(2\alpha^2-\alpha c_1)\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|^2+3\|\bar{\boldsymbol{\xi}}(k)\|^2$$

with $c_1=\dfrac{2}{\mu+L}$ and $c_2=\dfrac{2\mu L}{\mu+L}$.

It has already been verified in Theorem 5.0.4 that $2\alpha^2-\alpha c_1\leq 0$, provided that $\alpha\leq c_1$. Therefore, we can deduce that $(2\alpha^2-\alpha c_1)\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\|^2\leq 0$. The bounds reduces to the relationship:

$$\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2\leq(2-\frac{2\alpha\mu L}{\mu+L})\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+24\|\bar{\Delta}(k)\|^2+6\bar{L}^2\alpha^2.$$

We need to ensure that $(2-\frac{2\alpha\mu L}{\mu+L})$ does not grow unbounded by choosing $(2-\frac{2\alpha\mu L}{\mu+L})$ in the domain $(2-\frac{2\alpha\mu L}{\mu+L})\in(0,1)$, implying that $(2-\frac{2\alpha\mu L}{\mu+L})\leq 1$.

After applying recursion, we obtain the following inequality bound:

$$\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|\leq(2-\frac{2\alpha\mu L}{\mu+L})^{\frac{k}{2}}\|\bar{\boldsymbol{x}}(0)-\boldsymbol{x}^*\|+\sqrt{24}\|\bar{\Delta}(k)\|+\sqrt{6}\bar{L}\alpha.$$

If $\sqrt{6}\bar{L}\alpha\leq 1$ or $\bar{L}\leq 1/\sqrt{6}\alpha$, the neighborhood size is given by $\sqrt{24}\|\bar{\Delta}(k)\|+\sqrt{6}\bar{L}\alpha$.

□

To achieve the proposed detection aim, we will use the convergence neighborhood in Theorem 5.0.4 and the neighborhood of Theorem 5.0.5 to obtain the error. Clearly the expressions $\sqrt{24}\|\bar{\Delta}(k)\|$ and $\sqrt{6}\bar{L}\alpha$ are both common in the neighborhood bounds of Theorems 5.0.4 and 5.0.5. Therefore, we can obtain the deviations of the two neighborhoods to be the expression $(1-\frac{\alpha\mu L}{\mu+L})+\sqrt{3}\|e_i(k)\|$.

Now, we let the deviation of the two neighborhoods be bounded by an expected convergence rate B as follows:

$$(1-\frac{\alpha\mu L}{\mu+L})+\sqrt{3}\|e_i(k)\|\leq B. \tag{5.13}$$

By solving for $\|e_i(k)\|$, we obtain the following relationship:

$$\|e_i(k)\|\leq \frac{B(\mu+L)}{\sqrt{3}(\mu+L-\alpha\mu L)}. \tag{5.14}$$

Based on the bound in equation (5.14), if an attack vector is chosen outside of the bound shown, it can be detected.

Proposition for Resilience against Adversarial Attacks

Now that a detection method has been proposed, resilience methods against attacks are also shown to mitigate the effects of adversarial agents in the network. In the update equation (5.3), a doubly stochastic weight matrix w_{ij} was used but a variation of equation (5.3) will be used in the problem formulation in this section. The new update equation is:

$$\boldsymbol{x}_i(k+1)=\boldsymbol{x}_i(k)-\boldsymbol{q}_i(k)+\sum_{j\in N_i\cup\{i\}} G_{ij}\boldsymbol{q}_j(k)-\alpha_i\nabla f_i(\boldsymbol{x}_i(k))+\boldsymbol{e}_i(k). \tag{5.15}$$

where G_{ij} is the doubly stochastic weight matrix that is a piecewise variation of the weights w_{ij} depending on whether the neighbor of each agent i is adversarial or not. We vary the weights depending on who the neighbor of agent i is. If the neighbor of agent i is non-adversarial or if the previous reading of agent i is similar to current reading, we use the original weight w_{ij}. Otherwise if the neighbor of agent i is malicious or if the previous reading of agent i is far away from current reading, we use a different weight matrix.

Consider a metric such that the approximation of the estimates of each agent to the average of the estimates of the neighborhood as follows:

$$P_{ij}(k) = \|x_i(k) - x_a\| \tag{5.16}$$

where x_a is the average of the estimates of the neighborhood of agent i. Let the threshold of attack detection as seen in equation (5.13) be the following:

$$\|x_i(k) - x_a\| \leq (1 - \frac{\alpha\mu L}{\mu + L}) + \sqrt{3}\|e_i(k)\|. \tag{5.17}$$

After the weights are determined depending on the neighborhood of each agent i in the network, adversarial agents are excluded in the network. To quantify how this is done, we let agent i be connected to neighbors in a manner where the weight $G_{ij} = w_{ij}$ in equation (5.15) is assigned when the neighbor is a regular or good agent where w_{ij} is the weight used in the convergence analysis according to the result of Theorem 5.0.4. Let $G_{ij} = w_b$ be the weight when the neighbor is a bad agent. If the estimates obtained by using the update equation (5.15) with $G_{ij} = w_b$ falls outside of the threshold as seen in equation (5.17), then the weights of adversarial agents are excluded and $G_{ij} = w_{ij} - w_b$ in equation (5.15). We also take into account that non-adversarial agents do not always make the correct decision during the detection process. The results for the proposition for

resilience against attacks are shown in the Theorem below:

Theorem 5.0.6. *Suppose Assumptions 7, 8 and 9 are satisfied, with α satisfying $\alpha \leq \frac{2}{\mu+L}$. If the gradient bound is upper-bounded by $\bar{L} \leq 1/\sqrt{6}\alpha$ and the weights of adversarial attackers are removed from the estimates according to update equation* (5.15)*, then the iterates converge to a neighborhood of the optimal solution, x^*.*

Proof. Analyses for the results of Theorem 5.0.6 follows the same pattern as that of Theorem 5.0.4 except that the weight in equation (5.15) is a concatenation of $w_{ij} - w_b$ instead of the regular w_{ij} that is used in equation (5.3).

Let the probability that non-adversarial agents make a correct decision in detecting adversarial agents be p. Then the probability that non-adversarial agents make an incorrect decision in detecting adversarial agents is $1-p$. Let $p \in [0.5, 1]$ be the set of probabilities that non-adversarial agents make a correct decision. Similarly according to Theorem 5.0.4, we show convergence to a neighborhood of optimal solution by using the relationship $\|x_i(k) - x^*\| = \|x_i(k) - \bar{\boldsymbol{x}}(k)\| + \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$. The bound of $\|x_i(k) - \bar{\boldsymbol{x}}(k)\|$ is seen in Lemma 5.0.3. Now we proceed to the analysis of $\|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|$ below.

$$\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 = \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^* - \bar{e}(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - \bar{\boldsymbol{\xi}}(k)\|^2.$$

Alternatively, one can rewrite $\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2$ as:

$$\begin{aligned}
&\|\bar{\boldsymbol{x}}(k+1) - \boldsymbol{x}^* - \bar{e}(k)\|^2 \\
&= \|\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*\|^2 + \|\bar{e}(k)\|^2 + \left\| \frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) \right\|^2 + \|\bar{\boldsymbol{\xi}}(k)\|^2 + 2\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*) \right) \\
&\quad - 2(\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*)^T \left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) \right) + 2\bar{\boldsymbol{\xi}}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{x}_i(k)) - (\bar{\boldsymbol{x}}(k) - \boldsymbol{x}^*) \right) + 2\bar{e}(k)\bar{\boldsymbol{\xi}}(\boldsymbol{k}).
\end{aligned}$$

By taking expectation of both sides, we obtain the relationship:

$$
\begin{aligned}
&E\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2 \\
&\quad = E\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+E\|\bar{e}(k)\|^2+E\left\|\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right\|^2+E\|\bar{\boldsymbol{\xi}}(k)\|^2 \\
&\quad +2E\bar{e}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)-2E(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)^T\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))\right) \\
&\quad +2E\bar{\boldsymbol{\xi}}(k)\left(\frac{\alpha(k)}{n}\sum_{i=1}^{n}\boldsymbol{g}_i(\boldsymbol{x}_i(k))-(\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*)\right)+2E\bar{e}(k)\bar{\boldsymbol{\xi}}(\boldsymbol{k}).
\end{aligned}
$$

By adapting the results from Theorem 5.0.4 and Lemma 5.0.2, specifically with $\Delta_i(k) = \frac{1}{2z_i(k)}$, we obtain the following relationship:

$$
E\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*-\bar{e}(k)\|^2 \leq (3-3\alpha c_2)E\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3E\|\bar{e}(k)\|^2+\frac{24}{n}\|\sum_{i=1}^{n}\frac{1}{2z_i(k)}\|^2+6\bar{L}^2\alpha^2.
$$

Equivalently, we obtain the following relationship:

$$
E\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2 \leq (3-3\alpha c_2)E\|\bar{\boldsymbol{x}}(k)-\boldsymbol{x}^*\|^2+3E\|\bar{e}(k)\|^2+\frac{24}{n}\|\sum_{i=1}^{n}\frac{1}{2z_i(k)}\|^2+6\bar{L}^2\alpha^2.
$$

where $c_2 = \frac{2\mu L}{\mu+L}$. By using the definition of p being the probability of a correct decision made in the detection process and also using the fact that $E(x) = \sum_{i=1}^{n} xp(x)$, we obtain the relationship:

$$
\begin{aligned}
E\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\|^2 &\leq p(3-3\alpha c_2)\|\bar{\boldsymbol{x}}\boldsymbol{c}(k)-\boldsymbol{x}^*\|^2 \\
&\quad +(1-p)(3-3\alpha c_2)\|\bar{\boldsymbol{x}}\boldsymbol{i}(k)-\boldsymbol{x}^*\|^2+3\|\bar{e}(k)\|^2+\frac{24}{n}\|\sum_{i=1}^{n}\frac{1}{2z_i(k)}\|^2+6\bar{L}^2\alpha^2.
\end{aligned}
$$

where xc denotes the averaged estimates when a correct decision is made and xi denotes the averaged estimates when an incorrect decision is made. By using $p = 0.5$ as the threshold of correct/incorrect decision and also taking the square root of both sides, we obtain the following

bounds:

$$E\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\| \leq \sqrt{0.5(3-3\alpha c_2)}\|\bar{\boldsymbol{x}}\boldsymbol{c}(k)-\boldsymbol{x}^*\| + \sqrt{0.5(3-3\alpha c_2)}\|\bar{\boldsymbol{x}}\boldsymbol{i}(k)-\boldsymbol{x}^*\|+3\|\bar{e}(k)\|+\sqrt{\frac{24}{n}}\|\sum_{i=1}^{n}\frac{1}{2z_i(k)}\|+\sqrt{6}\bar{L}\alpha.$$

By applying recursion, we obtain the following:

$$E\|\bar{\boldsymbol{x}}(k+1)-\boldsymbol{x}^*\| \leq 0.5(3-3\alpha c_2)^{\frac{k}{2}}\|\bar{\boldsymbol{x}}_c(0)-\boldsymbol{x}^*\| + 0.5(3-3\alpha c_2)^{\frac{k}{2}}\|\bar{\boldsymbol{x}}\boldsymbol{i}(0)-\boldsymbol{x}^*\|+3\|\bar{e}(k)\|+\sqrt{\frac{24}{n}}\|\sum_{i=1}^{n}\frac{1}{2z_i(k)}\|+\sqrt{6}\bar{L}\alpha.$$

Similarly, it follows that convergence to a neighborhood of the optimal solution is obtained and will be corroborated via numerical simulations in the next section: □

Numerical Experiments

We illustrate our claims via numerical experiments over a network of $n = 10$ agents including some malicious agents with 100 iterations using the linear regression loss function as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} f(\boldsymbol{x}) = \sum_{i=1}^{n}\frac{1}{2}\|A_i\boldsymbol{x}-b_i\|^2, \tag{5.18}$$

solved in a distributed manner. We will show how varying or excluding weights G_{ij} in equation (5.15) performs especially when $G_{ij} = w_{ij}$ performs compared to $G_{ij} = w_{ij} - w_b$. We consider a scenario with 70% adversarial agents in the network for the two cases $G_{ij} = w_{ij}$ and $G_{ij} = w_{ij} - w_b = \frac{nw_{ij}-1}{n^2}$ where n is the number of agents. To verify that $G_{ij} = w_{ij} - w_b = \frac{nw_{ij}-1}{n^2}$ is indeed less than

$G_{ij} = w_{ij}$, it suffices to show that $w_{ij} - \frac{nw_{ij}-1}{n^2} \geq 0$. This leads to the following relationship:

$$w_{ij} - \frac{nw_{ij} - 1}{n^2} = w_{ij} - \frac{w_{ij}}{n} + \frac{1}{n^2}. \tag{5.19}$$

Clearly, the first two terms of the expressions in equation (5.19) is positive because $w_{ij} - \frac{w_{ij}}{n} = \frac{w_{ij}(n-1)}{n}$ is positive when $n \geq 1$.

Simulations are shown below:

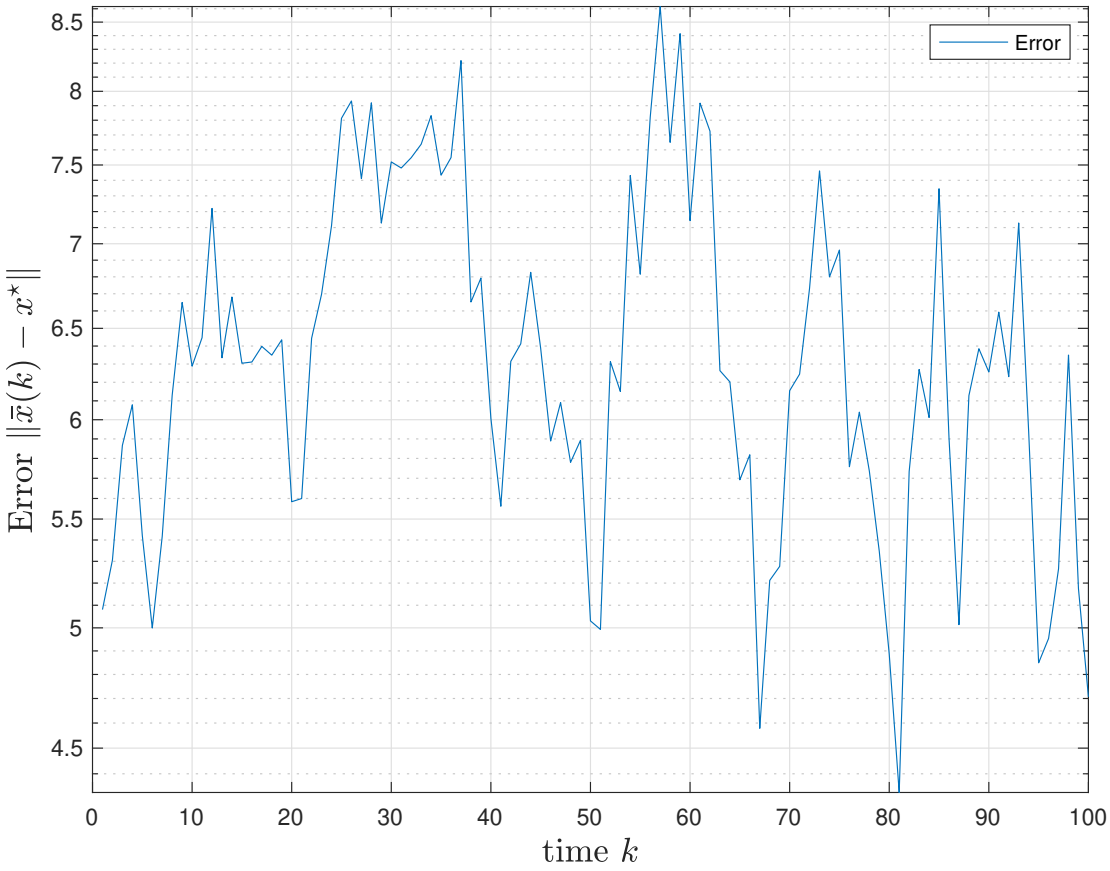

Figure 5.5: Simulations for regular adversarial weights $G_{ij} = w_{ij}$ for more adversarial agents

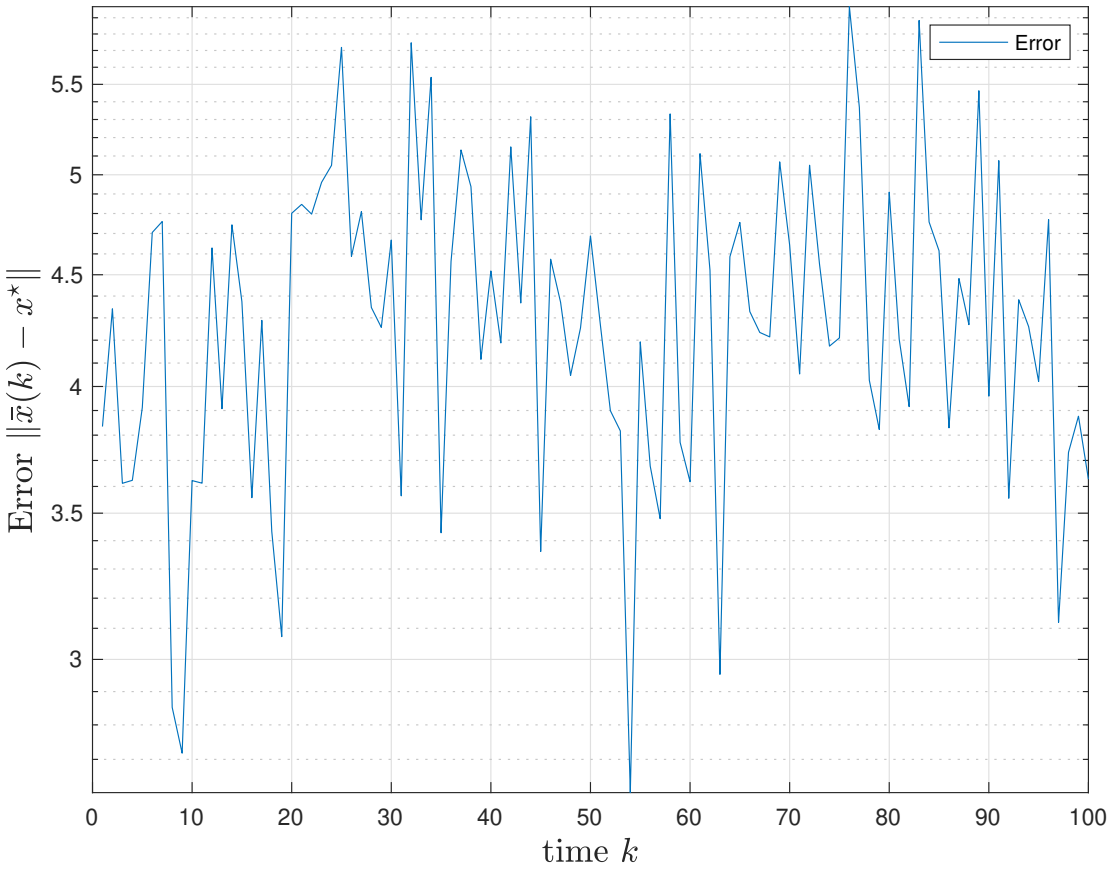

Figure 5.6: Simulations with weights of Adversarial agents removed $G_{ij} = w_{ij} - w_b = \frac{nw_{ij}-1}{n^2}$ for more adversarial agents

Remark 6. *As seen in Figures 5.5 and 5.6 when there are more adversarial agents in the network, the error in Figure 5.6 is smaller than that of Figure 5.5 even when the weights are reduced. Moreover, convergence to a neighborhood of optimal solution is still guaranteed. Similar conclusions can be made when there are less adversarial agents in the network as seen in Figures 5.7 and 5.8 as the error in Figure 5.8 is smaller than that of Figure 5.7 when the weights are smaller. The errors are also lower when there are less adversarial agents in the network as seen in Figures 5.7 and 5.8 compared to Figures 5.5 and 5.6. We can conclude that excluding or removing weights*

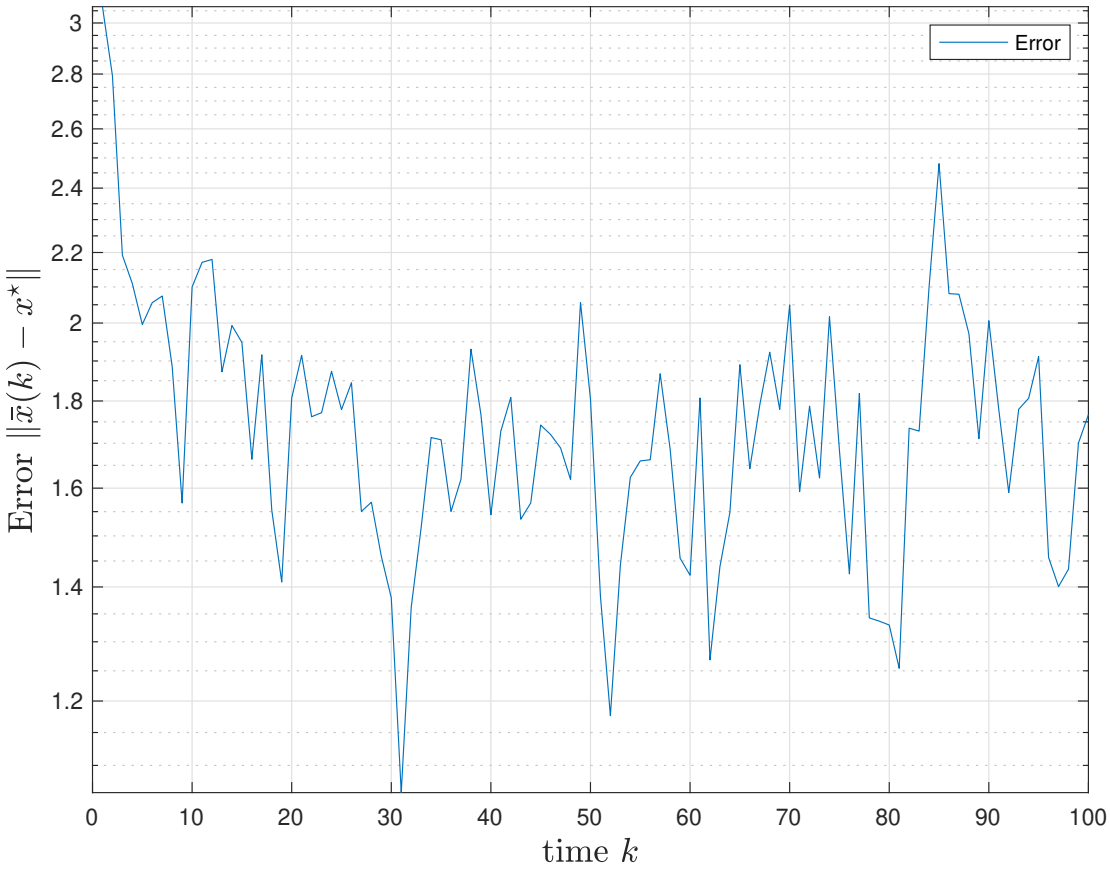

Figure 5.7: Simulations for regular adversarial weights $G_{ij} = w_{ij}$ for less adversarial agents

of adversarial agents result in a better performance. Although a better result is obtained when the weights of adversarial agents are removed, it should be noted that convergence to the exact optimal solution is still not guaranteed because the decisions on non-adversarial agents detecting or suspecting adversarial agents are not always accurate.

So far, distributed gradient descent methods have been used to explore distributed optimization methods with adversarial attack and quantization. In the pursuit of obtaining better performance of distributed optimization algorithms under adversarial attack, some preliminary work using Quasi-

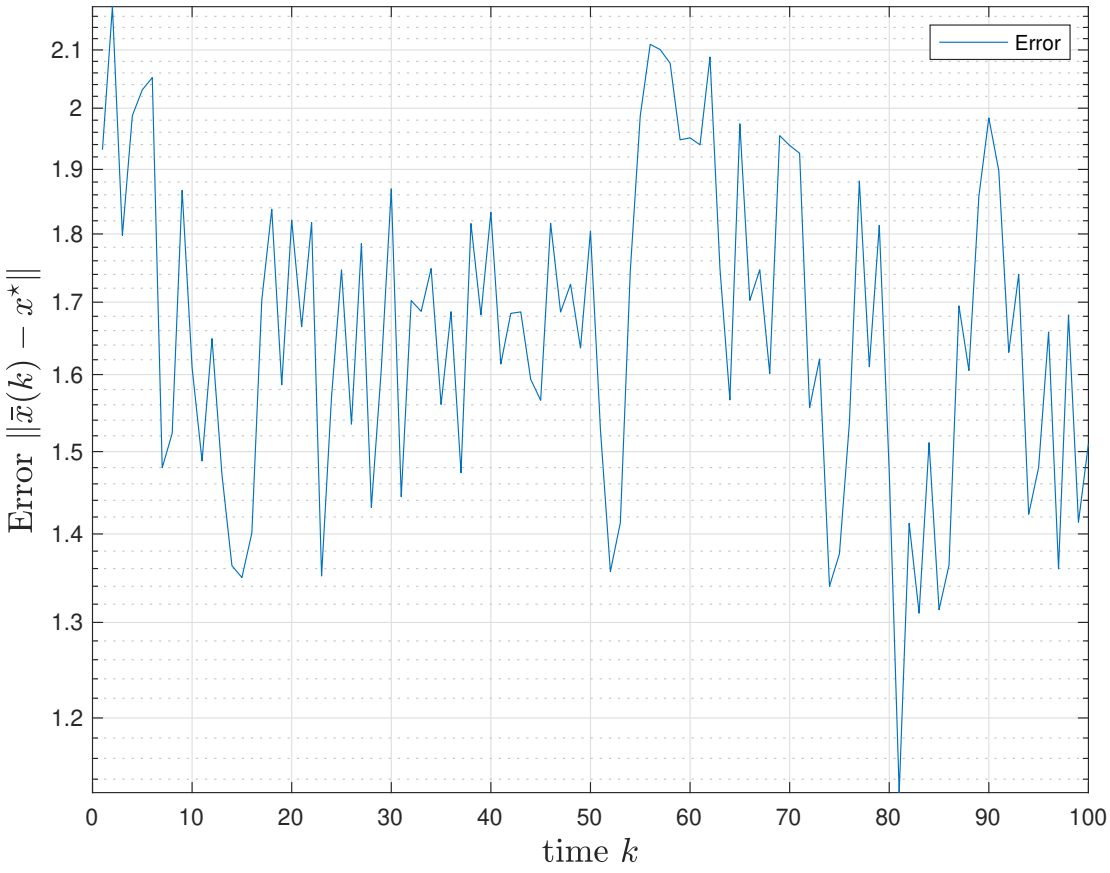

Figure 5.8: Simulations with weights of Adversarial agents removed $G_{ij} = w_{ij} - w_b = \frac{nw_{ij}-1}{n^2}$ for less adversarial agents

Newton Barzilai-Borwein methods and their applications are explored in the next chapter. The ease of computing the inverse of the hessian and fast convergence properties make the BB method a suitable fit to perform well in a distributed optimization with adversarial attack and quantization. We leave this topic to curious researchers to explore.

CHAPTER 6: IMPROVING CONVERGENCE RATES OF DISTRIBUTED OPTIMIZATION ALGORITHMS UNDER ADVERSARIES: ONLINE PERFORMANCE AND BARZILAI-BORWEIN QUASI-NEWTON METHODS

Solving a distributed optimization problem with adversaries usually require a model with suitable convergence properties. In this regard, a fully distributed algorithm for solving an unconstrained optimization problem is presented in this chapter using uncoordinated BB step-sizes and *Q-linear* convergence is obtained when the cost function is strongly convex. The Barzilai-Borwein Quasi-Newton Method is explored in the following section.

Problem Formulation

Consider the unconstrained problem over a network of agents. The decision set of the agents is $\mathcal{X}$, and their objective is to

$$\min_{x\in\mathcal{X}} f(x) = \sum_{i=1}^{n} f_i(x), \tag{6.1}$$

where $\mathcal{X}$ is a convex, feasible set. In Problem (6.1), f is strongly convex and smooth. Each agent i in the network has access to a f_i a component of f and the agents collectively seek to optimize $f(x)$ by locally optimizing $f_i(x)$ iteratively.

The communication graph of the multi-agent network is represented by an undirected weighted Graph $G = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V} = 1, 2, ...n$ is the set of nodes (agents) and $\mathcal{E} = (i, j)$ is the set of edges such that agents i, j are connected in the edge set, where $j \neq i$. The neighbors of agent i is represented by the set $N_i = \{j : (i, j) \in \mathcal{E}\}$. Symmetry of the underlying graph implies that agents

i and j for which $(i, j) \in \mathcal{E}$ means that information flows in both directions between both agents.

The common approach to solve Problem (6.1) is to use first-order methods, which involves updating the variable $x(k)$ iteratively using the gradient of the cost function with the following equation:

$$x(k+1) = x(k) - \alpha \nabla f(x(k)). \tag{6.2}$$

It is well known that with an appropriate choice of the step size α, the sequence $\{x(k)\}$ generated from Equation (6.2) converges to x^*.

The Newton method, which leverages curvature information of the cost function in addition to direction; and are known to speed up the convergence in the neighborhood of the optimal solution. The Newton-type methods have an update of the form:

$$x(k+1) = x(k) - \nabla f(x)(\nabla^2 f(x))^{-1}. \tag{6.3}$$

Though they have good convergence properties, there are computational costs associated with building and computing the inverse hessian. In addition, some modification are needed if the hessian is not positive definite [99].

Barzilai-Borwein Quasi-Newton Method

The Barzilai-Borwein method differs from other quasi-Newton methods because it only uses one step size for the iteration as opposed to other quasi-Newton method that need approximations for the inverse of the hessian, thus, increasing the computation overhead. Problem (6.1) is solved using the iterative scheme, where a step-size $\alpha(k)$ is computed in the gradient descent method (6.2) so that $\alpha(k)\nabla f(x(k))$ approximates the $(\nabla^2 f(x(k)))^{-1}\nabla f(x(k))$ term in the Newton update (6.3).

Let $s(k-1) \triangleq x(k) - x(k-1)$ and $y(k-1) = \nabla f(x(k)) - \nabla f(x(k-1))$. The first BB step size is given by:

$$\alpha_1(k) = \frac{s(k-1)^T s(k-1)}{s(k-1)^T y(k-1)}. \tag{6.4}$$

Similarly, the second step size, $\alpha_2(k)$ is given by:

$$\alpha_2(k) = \frac{s(k-1)^T y(k-1)}{y(k-1)^T y(k-1)}. \tag{6.5}$$

In general, there is flexibility in the choice to use $\alpha_1(k)$ or $\alpha_2(k)$ [79]. In addition, both step sizes can be alternated within the same algorithm after a considerable amount of iterations to facilitate convergence. The procedure is summarized in Algorithm 1 below.

Algorithm 1 Algorithm for Centralized BB

Initialize: $\alpha_1(0), x(0), \nabla f(x(0)), \varepsilon$.

1: **while** $\|\nabla f(x(k))\| \geq \varepsilon$ **do**

2: Compute

$$\alpha_1(k) = \frac{s(k-1))^T s(k-1)}{s(k-1)^T y(k-1)}$$

▷ α_2 in Equation (6.5) may also be used

3: Update $x(k+1) = x(k) - \alpha_1(k)\nabla f(x(k))$

4: **end while**

Before proceeding with the distributed BB algorithm and its convergence analyses, we first present a convergence analysis of the centralized case, where the following assumptions are made about Problem (6.1) and Algorithm 1.

Assumption 10. *The decision set $\mathcal{X}$ is bounded.*

Assumption 11. *The objective function $f(x)$ in Problem* (6.1) *is strongly convex and twice differentiable. This implies that for $x, y \in \mathbb{R}^{np}$, there exists $\mu > 0$ such that: $f_i(x) \geq f_i(y) + \nabla f_i(y)^T(x - y) + \frac{\mu}{2}\|x - y\|^2$.*

Assumption 12. *The inner product between the iterates deviations, s, and the gradient deviations, y, is strictly positive for all time step k. We make this assumption in both the centralized and distributed case.*

Assumption 13. *The gradient of the objective function ∇f is Lipschitz continuous. This implies that for all x and y, there exists $L > 0$ such that:* $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

The assumptions above are typical in the literature for distributed optimization problems; and in fact, Assumption 12 was made in [86] as well.

Convergence Analysis of Centralized BB

We present a convergence analysis using the two BB step sizes for Problem (6.1) where strong convexity of the cost function is assumed.

Convergence Analysis with step size α_1

Lemma 6.0.1. *Consider Algorithm 1 for Problem* (6.1) *and let Assumptions 10, 11, 12 and 13 hold. If $\alpha_1(k)$ in Equation* (6.4) *is such that $1/L \leq \alpha_1(k) \leq 2/(\mu + L)$, the iterates generated from Algorithm 1 converge Q-Linearly to the optimal point x^*.*

Proof. From Equation (6.2), we first consider $\|x(k+1) - x^*\|^2$ to obtain bounds for convergence. First, we let $g(k) = \nabla f(x(k))$ and we obtain that: $\|x(k+1) - x^*\| = \|x(k) - x^* - \alpha_1(k)g(k)\|$. By squaring both sides, we have:

$$\|x(k) - x^* - \alpha_1(k)g(k)\|^2 = \|x(k) - x^*\|^2 + \alpha_1^2(k)\|g(k)\|^2 - 2\,(x(k) - x^*)^T\,(\alpha_1(k)g(k))\,. \tag{6.6}$$

Using the fact that for all vectors $a,\ b$, $2a^T b \leq \|a\|^2 + \|b\|^2$. So we obtain the relationship:

$$2\left(x(k) - x^*\right)^T \left(g(k)\right) \leq \|g(k)\|^2 + \|x(k) - x^*\|^2.$$

By using strong convexity, where μ and L are strong convexity and Lipschitz parameters respectively and c_1, c_2 are given by $c_1 = 2/(\mu + L)$ and $c_2 = (2\mu L)/(\mu + L)$, we obtain:

$$\begin{aligned}
\|x(k) - x^* - \alpha_1(k) g(k)\|^2 &\leq \|x(k) - x^*\|^2 + \alpha_1^2(k)\|g(k)\|^2 - \alpha_1(k) c_1 \|g(k)\|^2 - \alpha_1(k) c_2 \|x(k) - x^*\|^2, \\
&\leq (1 - \alpha_1(k) c_2)\|x(k) - x^*\|^2 + (\alpha_1^2(k) - \alpha_1(k) c_1)\|g(k)\|^2, \\
&\leq (1 - \alpha_1(k) c_2)\|x(k) - x^*\|^2.
\end{aligned} \tag{6.7}$$

and the last inequality is due to Theorem 2.1.12 from chapter 2 of [70]. We also note that in the previous inequality, the term $(\alpha_1^2(k) - \alpha_1(k) c_2)\|g(k)\|^2 \leq 0$ provided $\alpha_1(k) \leq c_1$. We establish that the step size $\alpha_1(k) = c_1$ indeed is within the range of the BB step size bounds such that: $\frac{1}{L} \leq c_1 \leq \frac{1}{\mu}$, and refer readers to Appendix A for details.

Therefore the Barzilai-Borwein convergence can thus be analysed as:

$$\|x(k+1) - x^*\|^2 \leq (1 - \alpha_1(k) c_2)\, \|x(k) - x^*\|^2. \tag{6.8}$$

By dividing both sides of equation (6.8) by $\|x(k) - x^*\|^2$, we obtain the following:

$$\frac{\|x(k+1) - x^*\|^2}{\|x(k) - x^*\|^2} \leq 1 - \alpha_1(k) c_2. \tag{6.9}$$

By taking the square roots of both sides of equation (6.9), we obtain the following relationship:

$$\frac{\|x(k+1) - x^*\|}{\|x(k) - x^*\|} \leq (1 - \alpha_1(k) c_2)^{\frac{1}{2}}.$$

We will now analyze the right hand side of the above equation by bounding $(1-\alpha_1(k)c_2)^{\frac{1}{2}}$. The first Barzilai-Borwein step size $\alpha_1(k)$ is given by:

$$\alpha(k) = \frac{\|s(k-1)\|^2}{[x(k)-x(k-1)]^T\left[\nabla f\left(x(k)\right)-\nabla f\left(x(k-1)\right)\right]}.$$

By using Lipschitz continuity of $\nabla f(\cdot)$, with L as the Lipschitz constant, we obtain the lower bound of the first BB step size as:

$$\alpha_1(k) > \frac{\|x(k)-x(k-1)\|^2}{L\|x(k)-x(k-1)\|^2} = \frac{1}{L}.$$

If $\alpha_1(k)$ and c_2 are positive and $\alpha_1(k) > 1/L$, then $-\alpha_1(k)c_2 < -c_2/L$. Since $\alpha_1(k) > 1/L$, it implies that $0 < 1-\alpha_1(k)c_2 < 1-c_2/L$. So we obtain the bound: $0 < (1-\alpha_1(k)c_2)^{\frac{1}{2}} < \left(1-\frac{c_2}{L}\right)^{\frac{1}{2}}$. We will now show that $c_2/L < 1$. If $c_2 = 2\mu L/(\mu+L)$, then it implies that $c_2/L = 2\mu/(\mu+L)$. If $\mu < L$, then it implies that $\mu+\mu < L+\mu$ and we obtain the fact that $2\mu/(\mu+L) < 1$.

Therefore we obtain the relationship:

$$\lim_{k\to\infty} \quad \frac{\|x(k+1)-x^*\|}{\|x(k)-x^*\|} < \left(1-\frac{c_2}{L}\right)^{\frac{1}{2}} < 1,$$

from which we conclude that the iterates $x(k)$ converge Q-linearly to the optimal point, x^*. □

Remark 7. *When strong convexity is assumed, it is shown that the algorithm converges Q-Linearly to the optimal point x^*.*

Convergence Analysis of Centralized BB with Second Step Size

Here, we state a similar result to Lemma 6.0.1 using the second BB step size in Equation (6.5).

Lemma 6.0.2. *Consider Algorithm 1 for Problem* (6.1) *and let Assumptions 10, 11, 12 and 13*

hold. If $\alpha_2(k)$ *in Equation* (6.5) *is such that* $1/L \leq \alpha_2(k) \leq 2/(\mu + L)$*, the iterates generated from Algorithm 1 converge Q-Linearly to the optimal point* x^**.*

Proof. See Chapter A, Appendix A. □

Distributed Barzilai-Borwein Quasi-Newton Method

We present a distributed solution to Problem (6.1), where Assumptions 10, 11, 12 and 13 hold. In our proposed distributed algorithm, each agent in the network keeps a local copy of the decision variable $x_i(k)$ and a local gradient $\nabla f_i(x_i(k))$ and updates them at each time-step using locally computed step sizes $\alpha_i(k)$. The step size computation is similar to the centralized case. Using the local variables $x_i(k)$ and local gradient variables $\nabla f_i(x_i(k))$, each agent computes

$$s_i(k-1) = x_i(k) - x_i(k-1), \tag{6.10}$$

$$y_i(k-1) = \nabla f_i(x_i(k)) - \nabla f_i(x_i(k-1)), \tag{6.11}$$

and computes $\alpha_i(k)$ in a manner that ensures

$$(\alpha_i(k)^{-1} I) s_i(k-1) \approx y_i(k-1). \tag{6.12}$$

Using the expressions in (6.10) and (6.11), we obtain the distributed form of the step size for each agent i, which is given by:

$$\alpha_{i1}(k) = \frac{(s_i(k-1))^T s_i(k-1)}{(s_i(k-1))^T y_i(k-1)}. \tag{6.13}$$

and

$$\alpha_{i2}(k) = \frac{(s_i(k-1))^T y_i(k-1)}{(y_i(k-1))^T y_i(k-1)}. \tag{6.14}$$

To distribute the computations locally at each time step k, each agent i uses the following update scheme $x_i \in \mathbb{R}^n$:

$$x_i(k+1) = x_i(k) - \alpha_i(k)\nabla f_i(x_i(k)). \tag{6.15}$$

To ensure all agents converge to the optimal solution, each agent carries an iterative local computation step and the interaction with neighbors lead to a consensus step. Each agent takes a weighted average of the information received from its neighbors to compute its next update. With this protocol, the local update at each agent i is given by:

$$x_i(k+1) = \sum_{j \in N_i \cup i} (w_{ij}x_j(k) - \alpha_i(k)\nabla f_i(x_i(k))). \tag{6.16}$$

where w_{ij} are weights attached by agent i to agent j's estimate.

Given that $W = [w_{ij}]$, and if we let $X = [x_1, \ldots, x_n] \in \mathbb{R}^{np}$ be the concatenation of local variables x_i, I_p is the identity matrix of dimension p, $\otimes$ represents the Kronecker operation of matrix product and $W = [w_{ij}]$ is the doubly stochastic weight matrix that satisfies: $W \otimes I_p \in \mathbb{R}^{np \times np}$. We can re-write Equation (6.16) as follows:

$$X(k+1) = (W \otimes I_p)X(k) - \alpha_i \nabla f(X(k)). \tag{6.17}$$

where X is the concatenation of local x_i, and $\nabla f(X(k)) \in \mathbb{R}^{np}$ is the concatenated gradients. Similarly we can denote the average of local estimates to be $\overline{x}(k)$, the average of local estimates to be $g(x(k))$, the average of the Lipschitz constants for agents to be L, and the average of the strong convexity parameters for agents to be μ. We also denote the averages of the two step sizes of agents (α_{i1} and α_{i1}) to be $\overline{\alpha_{i1}}$ and $\overline{\alpha_{i2}}$ respectively. Because W is doubly stochastic, it has one eigenvalue $\lambda = 1$ and the other eigenvalues satisfy $0 < \lambda < 1$.

In the distributed implementation of the BB algorithm, neighbors compute their local step sizes us-

ing local information and exchange decision estimates with their neighbors over the communication network. The process is summarized in Algorithm 2.

Algorithm for Distributed BB

Algorithm 2 Algorithm for Distributed BB

Initialize: $\alpha_i(0), x_i(0), \nabla f_i(x_i(0))$

1: **while** $\|g(x(k))\| \geq \varepsilon$ **do**

2: Compute

$$s_i(k-1) \text{ using (6.10)},$$
$$y_i(k-1) \text{ using (6.11)},$$
$$\alpha_i(k+1) \text{ using (6.13)}.$$

3: Local update in equation (6.16)

4: Communicate updates $x_i(k+1)$ with neighbors.

5: **end while**

Convergence Analysis of Distributed BB

We examine convergence of Algorithm 2 to the optimal point based on the local estimates.

Distributed BB Convergence Analysis with the First Step-Size

We present the main result of this section in Theorem 6.0.3.

Theorem 6.0.3. *Consider Algorithm 2 for Problem* (6.1) *and let Assumptions 10, 11, 12 and 13 hold. If* $\alpha_{i1}(k)$ *in Equation* (6.13) *is such that* $1/L_i \leq \alpha_{i1}(k) \leq 2/(\mu_i + L_i)$*, the iterates of each agent* i *generated from Algorithm 3 converge Q-Linearly to the optimal point* x^**; that is* $\|x_i(k) - x^*\| \leq \|x_i(k) - \overline{x}(k)\| + \|\overline{x}(k) - x^*\|$*. Moreover, each local estimate* $x_i(k)$ *converges to the neighborhood of the optimal solution,* x^* *based on the two step sizes* α_{i1} *and* α_{i2}*.*

To prove the main result in Theorem 6.0.3 we will take a two-step approach. First, we upper bound the norm of the difference between the individual agent iterates and the average of the agents' iterates in Lemma 6.0.4. Next, we show that the average of the agents' iterates converges Q-linearly to the optimal solution in Lemma 6.0.5.

Lemma 6.0.4. *Consider Algorithm 2 with BB step size α_{i1} in Equation* (6.13) *for Problem 6.1 and suppose Assumptions 10, 11, 12 and 13 hold; and G be the upper bound of the gradients, then the norm of the difference between each local agent's estimate and the average agents' estimate is bounded by the following:*

$$\|x_i(k)-\overline{x}(k)\|\leq G\left(\sum_{m=0}^{k-1}\alpha_{i1}^2(m)\right)^{\frac{1}{2}}\left(\sum_{m=0}^{k-1}\lambda^{2(k-1-m)}\right)^{\frac{1}{2}},$$

where $\left(\sum_{m=0}^{k}\alpha_i^2(m)\right)^{\frac{1}{2}}\leq\frac{\sqrt{k}}{\mu}$, *and* $\left(\sum_{m=0}^{k-1}\lambda^{2(k-1-m)}\right)^{\frac{1}{2}}\leq\left(\frac{1}{1-\lambda^2}\right)^{\frac{1}{2}}=\frac{1}{\sqrt{1-\lambda^2}}\triangleq Q_3$. *Moreover if the following holds:*

$$\sum_{m=0}^{k-1}\alpha_{i1}^2(m)\leq\frac{1}{G^2\sum_{m=0}^{k-1}\lambda^{2(k-1-m)}},$$

then each local agent's estimates converges Q-linearly to its average; that is $\|x_i(k)-\overline{x}(k)\|\leq 1$.

Proof. The proof is presented in Chapter A, Appendix A □

After obtaining the norm of the difference between local estimates and the consensus average estimates, we now examine the convergence attribute of the average estimates $\bar{x}(k)$ to the optimal solution x^*. Before proceeding, we state an important lemma that leads to the convergence behavior of average estimates to the optimal point.

Lemma 6.0.5. *Consider Algorithm 2 for Problem* (6.1) *and let Assumptions 10, 11, 12 and 13 hold. If the average of the first distributed BB step size $\overline{\alpha_{i1}}$ is such that $1/L\leq\overline{\alpha_{i1}}\leq 2/(\mu+L)$, For finite*

values of i and k, and bounded gradients, the consensus average estimate converges Q-Linearly to the optimal point. Moreover, the local agent estimates converge to the neighborhood of the optimal point, x^*.

Proof. Let $\overline{x}(k) = \frac{1}{n}\sum_{i=1}^{n} x_i(k)$, $g(k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i\left(x_i(k)\right)$, and $\overline{\alpha_{i1}}(k) = \frac{1}{n}\sum_{i=1}^{n} \alpha_{i1}(k)$.

From equation 6.15, we first consider $\|\overline{x}(k+1) - x^*\|^2$ to obtain bounds for convergence. First, we let $g(k)$ be the average of gradient at local estimates and we let $\overline{\alpha_{i1}}$ be the average of the agents step sizes corresponding to the average of the iterates.

$$\|\overline{x}(k+1) - x^*\| = \|\overline{x}(k) - x^* - \overline{\alpha_{i1}} g(k)\|.$$

By squaring both sides and evaluating the right hand side, we have:

$$\|x(k) - x^* - \overline{\alpha_{i1}}(k) g(k)\|^2 = \|x(k) - x^*\|^2 + \overline{\alpha_{i1}}^2(k)\|g(k)\|^2 - 2\left(x(k) - x^*\right)^T \left(\overline{\alpha_{i1}}(k) g(k)\right). \tag{6.18}$$

Using the fact that for all vectors $a,\ b,\ 2a^T b \leq \|a\|^2 + \|b\|^2$, we obtain the relationship:

$$2\left(\overline{x}(k) - x^*\right)^T \left(g(k)\right) \leq \|g(k)\|^2 + \|\overline{x}(k) - x^*\|^2.$$

Just as we did for the centralized case, μ and L are strong convexity and Lipschitz parameters respectively and c_1, c_2 are given by $c_1 = 2/(\mu + L)$ and $c_2 = 2\mu L/(\mu + L)$. We now obtain:

$$\begin{aligned}
\|\overline{x}(k+1) - x^* - \overline{\alpha_{i1}}(k) g(k)\|^2 &\leq \|\overline{x}(k) - x^*\|^2 + \overline{\alpha_{i1}}^2(k)\|g(k)\|^2 - \overline{\alpha_{i1}}(k) c_1 \|g(k)\|^2 - \overline{\alpha_{i1}}(k) c_2 \|\overline{x}(k) - x^*\|^2, \\
&\leq (1 - \overline{\alpha_{i1}}(k) c_2)\|\overline{x}(k) - x^*\|^2 + (\overline{\alpha_{i1}}^2(k) - \overline{\alpha_{i1}}(k) c_1)\|g(k)\|^2, \\
&\leq (1 - \overline{\alpha_{i1}}(k) c_2)\|\overline{x}(k) - x^*\|^2.
\end{aligned} \tag{6.19}$$

We note that the last inequality is due to Theorem 2.1.12 from chapter 2 of [70]. We also

note that in the previous inequality, the term that contains the distributed form of the step size, $(\overline{\alpha_{i1}}^2(k) - \overline{\alpha_{i1}}(k)c_1)\|g(k)\|^2 \leq 0$ provided $\overline{\alpha_{i1}}(k) \leq c_1$. We show that the step size $\overline{\alpha_{i1}}(k) = c_1$ is within the range of the BB step size bounds below:

Corollary 1. *Let L and μ be the Lipschitz and strong convexity parameters respectively with $\mu \leq L$. The range of the average of the distributed BB step size $\overline{\alpha_{i1}}(k)$ is given by:*

$$\frac{1}{L} \leq \overline{\alpha_{i1}}(k) \leq \frac{2}{\mu + L} \leq \frac{1}{\mu}. \tag{6.20}$$

where the condition $\overline{\alpha_{i1}}(k) \leq \frac{1}{\mu}$ is assumed. See Chapter A, Appendix A for details.

Therefore the distributed BB convergence using the first BB step size can be analysed as the following:

$$\|\overline{x}(k+1) - x^*\|^2 \leq (1 - \overline{\alpha_{i1}}(k)c_2)\,\|\overline{x}(k) - x^*\|^2. \tag{6.21}$$

Dividing both sides of equation (6.21) by $\|\overline{x}(k) - x^*\|^2$ yields:

$$\frac{\|\overline{x}(k+1) - x^*\|^2}{\|\overline{x}(k) - x^*\|^2} \leq 1 - \overline{\alpha_{i1}}(k)c_2. \tag{6.22}$$

Taking the square root of both sides of equation (6.22) yields the following:

$$\frac{\|\overline{x}(k+1) - x^*\|}{\|\overline{x}(k) - x^*\|} \leq (1 - \overline{\alpha_{i1}}(k)c_2)^{\frac{1}{2}}.$$

We will now bound: $(1 - \overline{\alpha_{i1}}(k)c_2)^{\frac{1}{2}}$. The distributed form of the first Barzilai-Borwein step size $\alpha_{i1}(k)$ is given by:

$$\alpha_{i1}(k) = \frac{\|s_i(k-1)\|^2}{[x_i(k) - x_i(k-1)]^T\,[\nabla f_i\,(x_i(k)) - \nabla f_i\,(x_i(k-1))]}$$

By using Lipschitz continuity of $\nabla f(\cdot)$ with L as the Lipschitz constant, we obtain the lower bound of distributed form of the first BB step size as:

$$\alpha_{i1}(k) > \frac{\|x_i(k) - x_i(k-1)\|^2}{L\|x_i(k) - x_i(k-1)\|^2} = \frac{1}{L}.$$

We know that $\overline{\alpha_{i1}}(k) = \frac{1}{n}\sum_{i=1}^{n} \alpha_{i1}(k)$, and it follows that $n\overline{\alpha_{i1}}(k) = \sum_{i=1}^{n} \alpha_{i1}(k)$. But we know that $\alpha_{i1}(k) > \frac{1}{L}$, and as a fact, $\alpha_{i1}(k) < \sum_{i=1}^{n} \alpha_{i1}(k)$. Therefore we obtain the relationship:

$$\frac{1}{L} < \alpha_{i1}(k) < \sum_{i=1}^{n} \alpha_{i1}(k). \tag{6.23}$$

From equation (6.23), $n\overline{\alpha_{i1}}(k) = \sum_{i=1}^{n} \alpha_{i1}(k) > \frac{1}{L}$ and we obtain the fact that $\overline{\alpha_{i1}}(k) > \frac{1}{nL}$.

If $\overline{\alpha_{i1}}(k)$ and c_2 are positive and $\overline{\alpha_{i1}}(k) > 1/nL$, then $-\overline{\alpha_{i1}}(k)c_2 < -c_2/nL$. Since $\overline{\alpha_{i1}}(k) > 1/nL$. it implies that $0 < 1 - \overline{\alpha_{i1}}(k)c_2 < 1 - c_2/nL$, Therefore we obtain the bound $0 < (1 - \overline{\alpha_{i1}}(k)c_2)^{\frac{1}{2}} < \left(1 - \frac{c_2}{nL}\right)^{\frac{1}{2}}$. We will now show that $c_2/nL < 1$. If $c_2 = 2\mu L/(\mu + L)$, then it implies that $c_2/nL = 2\mu/n(\mu + L)$. If $\mu \leq L$, then we have $\mu + \mu \leq L + \mu$ and we obtain that $2\mu/n(\mu + L) \leq 1$ for all positive values of n. We obtain the convergence bounds as the following:

$$\lim_{k\to\infty} \quad \frac{\|\overline{x}(k+1) - x^*\|}{\|\overline{x}(k) - x^*\|} \leq \left(1 - \frac{c_2}{nL}\right)^{\frac{1}{2}} \leq 1,$$

and we conclude that the average of the estimates converges Q-linearly to the optimal point, x^*. □

Distributed BB with Second Step-Size

Lemma 6.0.6. *Suppose Assumptions 10, 11, 12 and 13 hold. Let the second BB step size be given by:*

$$\alpha_{i2} = \frac{s_i(k-1)^T y_i(k-1)}{y_i(k-1)^T y_i(k-1)}.$$

For finite values of i and k, and bounded gradients, where G is the upper bound of the gradients, the norm of the difference between the local agents estimate and the consensus average estimate is bounded and given by the following:

$$\|x_i(k) - \overline{x}(k)\| \leq G \left(\sum_{m=0}^{k-1} \alpha_{i2}^2(m) \right)^{\frac{1}{2}} \left(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)} \right)^{\frac{1}{2}}. \tag{6.24}$$

Where: $\left(\sum_{m=0}^{k} \alpha_i^2(m)\right)^{\frac{1}{2}} \leq \frac{\sqrt{k}}{\mu}$. *and:* $\left(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)}\right)^{\frac{1}{2}} \leq \left(\frac{1}{1-\lambda^2}\right)^{\frac{1}{2}} = \frac{1}{\sqrt{1-\lambda^2}} \triangleq Q_3$.

Proof. The proof is similar to the proof of Lemma 6.0.4; see Chapter A, Appendix A. □

Now, we bound the convergence of average of local estimates to the optimal point using the second step size.

First let us consider $\|\overline{x}(k+1) - x^*\|^2$: We will state a lemma before we proceed to the convergence analysis.

Lemma 6.0.7. *Consider Algorithm 2 for Problem* (6.1) *and let Assumptions 10, 11, 12 and 13 hold. If the average of the second distributed BB step size* $\overline{\alpha_{i2}}$ *is such that* $1/L \leq \overline{\alpha_{i2}} \leq 2/(\mu + L)$*, For finite values of i and k, and bounded gradients, the consensus average estimate converges Q-Linearly to the optimal point. Moreover, the local agent estimates converge to the neighborhood of the optimal point,* x^*.

Proof. See Chapter A, Appendix A. □

Numerical Experiments

We show some simulation for results in Theorem 6.0.3 (the distributed case). We consider the following objective function, which is separable per agent:

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} x^T A_i x + b_i^T x, \tag{6.25}$$

where $A_i \in \mathbb{R}^{p\times p}$, $b_i \in \mathbb{R}^p$ are used by each agent i for their own computation, and n is the number of agents in the network and its dimension is $m = 10$. In equation (6.25), the gradient function is given by:

$$\nabla f(x) = \frac{1}{2}(A_i + A_i^T)x + b_i. \tag{6.26}$$

We note that the function (6.25) is strongly convex and its gradient function in (6.26) is Lipschitz continuous for an appropriate value of A. We verify this through its strong convexity parameters μ_i and Lipschitz parameter values L_i, where $\mu_i \leq L_i$ for each agent i in the network. We also note that μ_i is the maximum of all the eigenvalues of matrix $(A_i + A_i^T)$ and L_i is the spectral norm of matrix $(A_i + A_i^T)$. We will use a scenario where there are 100 nodes in the network and the matrix W is a positive, symmetric, random doubly stochastic matrix. Our simulations aim to compare different step sizes with the distributed Barzilai-Bowein in equations (6.13) and (6.14). Specifically, we use the following step sizes of $\alpha_i = \frac{1}{L_i}$, $\alpha_i = \frac{1}{\mu_i}$, and $\alpha_i = \frac{2}{L_i+\mu_i}$ according to Lemmas 6.0.5 and 6.0.7, and the BB step size as seen in equation (6.13). In Figure 6.1, the label $c_1 = \frac{2}{\mu+L}$ bound curve (the circular curve) is the step size according to convergence result in Lemma 6.0.5. The label BB step size in figure 6.1 (the curve beneath all other curves) is the actual BB step size in equation (6.13) and the labels BB-Upperbound step size curve (the one in asterisk) and BB-Lowerbound

step size curve (triangular) are the lower and upper bounds of the BB step sizes ($\alpha_i = \frac{1}{L_i}$, $\alpha_i = \frac{1}{\mu_i}$) respectively. Our simulations affirm that the step size $\alpha_i = \frac{2}{L_i+\mu_i}$ lies in between the lower and upper bounds of the BB step size and also agrees with the theoretical result in Corollary 1. We apply these step sizes to the iteration shown in equation (6.16) to compare the rates at which each step size converges to the optimal point. By setting the gradient function in equation (6.26), to zero, we obtain the optimal solution x^* given by the following equation:

$$x^* = -2(A_i + A_i^T)^{-1}b. \tag{6.27}$$

The optimal solution in equation (6.27) is then obtained for the three different step sizes we used for simulation. Though convergence is attained for the three step sizes used, we run the simulations for 10 iterations to compare convergence speeds. We compare convergence rates by first initializing x, A and b as zeros between time step $k = 1$ to the total number of iterations $T = 10$. We plot the error curve for the three step sizes indicated by the expression $\|\overline{x}(k+1) - x^*\|$ and compare the results. It should be noted that the Barzilai-Borwein method is not necessarily monotone decreasing at each time-step [100] as seen in Figure 6.1 where the curve increases from the seventh to ninth time step. Our Numerical results are shown graphically below in Figure 6.1:

Now that the distributed representation using the Barzilai-Borwein Quasi-Newton method has been illustrated, the next section applies the Barzilai-Borwein Quasi-Newton method in online optimization. This work on online optimization is the preliminary result obtained and interested researchers are welcome to explore adversarial problems in online optimization scenarios.

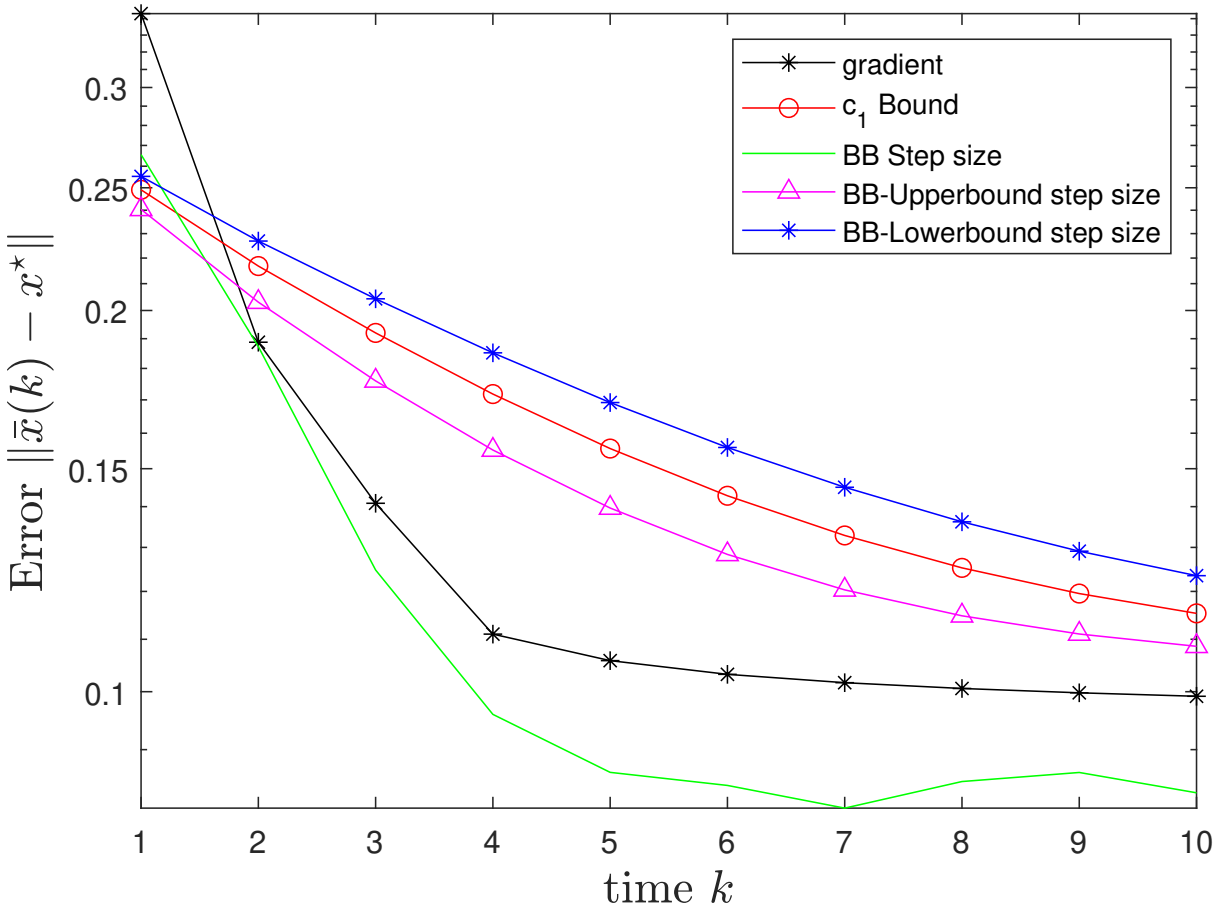

Figure 6.1: Distributed Simulations for 10 iterations

Sublinear Regret with Barzilai-Borwein Step Sizes

In this section, a regret analysis using the Barzilai-Borwein Quasi-Newton method in an online optimization scenario is presented. Due to the fast convergence property of the Newton methods, the work [97] is an improvement on existing online optimizations application problems in [94], [95], [96], and [87]. However, the Quasi-Newton method using the BB step sizes presented in this paper is better than Newton methods in dealing with convergence speeds and computing the inverse of the

hessian. Even though the author in [90] also obtained a similar sublinear regret result, BB Quasi-Newton algorithm is known to be suitable for dealing with large-scale optimization bottleneck that the Newton method is not appropriate for. Additionally, strong convexity assumption is not needed in this paper to establish sublinear regret.

Problem Formulation

Consider an online optimization problem below

$$\min_{x(k)\in\mathcal{X}} f_k(x(k)), \tag{6.28}$$

in which the feasible decision set $\mathcal{X} \in \mathbb{R}^n$ is known, assumed to be convex quadratic, non-empty, bounded, closed and fixed for all time $k = 1\ldots, K$. We assume the number of iterations during which the online players make choices, K, is unknown to the player. By convexity of the cost function $f_k(\cdot)$ and $\mathcal{X}$, Problem (6.28) has an optimal solution x^*, which is the best possible choice or decision agents can make at each time k. A player (an online agent) at time k uses some algorithm to choose a point $x(k) \in \mathcal{X}$, after which the player receives a loss function $f_k(\cdot)$. The loss incurred by the player is $f_k(x(k))$. These problems are common in contexts such as real time resource allocation, online classification [88]. The goal of the online agent is to minimize the aggregate loss by determining a sequence of feasible online solutions $x(k)$ at each time-step of the algorithm.

Let the aggregate loss incurred by the online algorithm that solves Problem (6.28) at time K be given by: $f(K) = \sum_{k=1}^{K} f_k(x(k))$. To measure performance of the online player, we use the regret framework. The *static regret* is a measure of the difference between the loss of the online player and the loss from the static case $\min_{x\in\mathcal{X}} f_k(x)$, where the single best decision x^* is chosen with the

benefit of hindsight. Let the aggregate loss up to time K incurred by the single best decision be given by $f_x(K) = \sum_{k=1}^{K} f_k(x)$. Then the static regret at time K is defined as [88]:

$$R(K) = f(K) - \min_{x \in X} f_x(K). \tag{6.29}$$

Algorithms for Online Optimization Problem

A commonly used algorithm for solving the static case of Problem (6.28) is the gradient descent method, which involves updating the variable $x(k)$ iteratively using the gradient of the cost function with the following equation:

$$x(k+1) = x(k) - \alpha \nabla f(x(k)). \tag{6.30}$$

It is known that with an appropriate choice of the step size α, the sequence $\{x(k)\}$ converges to x^* in $O(1/k)$; that is, an ε-optimal solution is attained in about $O(\frac{1}{\varepsilon})$ iterations [70]. Moreover, when the cost function is strongly convex, the update equation in (6.30) reaches an ε-optimal solution in about $O(1/\varepsilon^2)$ iterations. Even though the update scheme of gradient method are easily implementable in a distributed architecture as seen in [70] and [7], there have been a need for an improvement in convergence rates of gradient methods as seen in [101]. Nonetheless techniques to accelerate convergence lag behind the Newton and quasi-Newton methods [101].

To improve convergence rates in static optimization problems, algorithms that use second order information (hessian of the cost function) have been introduced. These methods leverage curvature information of the cost function in addition to direction; and are known to speed up the convergence in the neighborhood of the optimal solution. The Newton-type method is an example used as an improvement in enabling faster convergence rates than the regular gradient method. In fact, when the cost function is quadratic, the Newton algorithm is known to converge in one time-step. For non-quadratic, the Newton method still converges in just a few time steps [102]. Though

they have good convergence properties, there are computational costs associated with building and computing the inverse hessian. In addition, some modification are needed if the hessian is not positive definite [99]. To avoid the computation burden of second-order methods while maintaining the structure of first-order methods, the BB Quasi-Newton methods will be used.

The BB Quasi-Newton Method

The Barzilai-Borwen (BB) Quasi-Newton method is an iterative technique suitable for solving optimization problems that can yield superlinear convergence rates when the objective functions are strongly convex and quadratic [79, 85]. It differs from other quasi-Newton methods because it only uses one step size for the iteration as opposed to other quasi-Newton method that have more computation overhead. The Barzilai-Borwein method solves Problem (6.28) iteratively using the update in (6.30); however, the step-size $\alpha(k)$ is computed so that $\alpha(k)\nabla f(x(k))$ approximates the the inverse Hessian. We briefly introduce the two forms of the BB step-sizes used in Algorithm 3.

Consider the update $x(k+1) = x(k) - \alpha(k)\nabla f(x(k))$. The two forms of the BB step sizes [79] $\alpha_1(k)$ and $\alpha_2(k)$ are given by:

$$\alpha_1(k) = \frac{s(k-1)^T s(k-1)}{s(k-1)^T y(k-1)}. \tag{6.31}$$

$$\alpha_2(k) = \frac{s(k-1)^T y(k-1)}{y(k-1)^T y(k-1)}. \tag{6.32}$$

and $s(k)$ and $y(k)$ are such that $s(k-1) \triangleq x(k) - x(k-1)$ and $y(k-1) = \nabla f(x(k)) - \nabla f(x(k-1)$.

In general, there is flexibility in the choice to use $\alpha_1(k)$ or $\alpha_2(k)$ [79], and both step sizes can be alternated within the same algorithm after a considerable amount of iterations to facilitate convergence. The rest of this work will characterize performance of the online Algorithm 3 using

the step sizes in Equations (6.31) and (6.32), which as we will show has a regret that is sublinear in time with the average regret approaching zero.

Before stating the main result, we state some assumptions about Problem (6.28) and Algorithm 3.

Assumption 14. *The decision set $\mathcal{X}$ is bounded.*

Assumption 15. *The decision set $\mathcal{X}$ is closed.*

Assumption 16. *For all decision iterates $x(k)$, the cost function $f(x(k))$ is differentiable and the gradient of the objective function ∇f is Lipschitz continuous. This means that for all x and y, there exists $L > 1$ such that: $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.*

Algorithm 3 Online Barzilai-Borwein Quasi-Newton Alg.

Given: Feasible set $\mathcal{X}$ and time horizon K
Initialize: $x(0)$ and $\nabla f_0(x(0))$ arbitrarily
1: **for** $k = 1$ to K **do**
2: Agents predicts $x(k)$ and observes $f_k(\cdot)$
3: Update $x(k+1) = x(k) - \alpha(k)\nabla f_k(x(k))$
4: **end for**

Regret Bounds

Before the results are presented (in Theorems 6.0.10 and 6.0.11), we first present two lemmas that will be used in its proof. The first is a result in [90], which will be used in the definition of regret and the other is the Sedrakyan's inequality.

Lemma 6.0.8. *([90]) Without loss of generality, for all iterates k, there exists gradient $g(k) \in \mathbb{R}^n$ such that for all x, $g_k.x = f_k(x)$, where $g_k = \nabla f_k(x(k))$.*

Proof. The proof can be seen in [90]. □

Lemma 6.0.9. *(**The Sedrakyan's Inequality**) For all positive reals $a_1, a_2, ...a_n$ and $b_1, b_2, ...b_n$, the following inequality holds:*

$$\sum_{i=1}^{n} \frac{a_i^2}{b_i} \geq \frac{(\sum_{i=1}^{n} a_i)^2}{\sum_{i=1}^{n} b_i}.$$

Proof. We refer readers to [103] for a proof. □

Another result we will use is the static regret bounds for $R(K)$ which is shown in [90]:

$$R(K) \leq \|D\|^2 \frac{1}{2\alpha(K)} + \frac{\|\nabla f_m\|^2}{2} \sum_{k=1}^{K} \alpha(k), \tag{6.33}$$

As seen in [90], D denotes the maximum value of the diameter of $\mathcal{X}$ and $\|\nabla f_m\| = \max_{x \in \mathcal{X}} \|\nabla f_k(x)\|$.

We will now proceed to characterize the regret obtained from Algorithm 3 for Problem (6.28) with the two BB step sizes.

Theorem 6.0.10. *Consider Problem* (6.28) *and let:*

$$\alpha(k) = \frac{s(k-1)^T s(k-1)}{s(k-1)^T y(k-1)}$$

in Algorithm 3. If

$$\frac{e-d}{c-b} \leq \frac{d}{b},$$

where $b = (\|(x(1)-x(0)\|+\|(x(2)-x(1)\|)^2$, $c = 2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)$, $d = \sum_{k=1}^{K}(x(k) - x(k-1))^T(\nabla f(x(k)) - \nabla f(x(k-1)))$, *and* $e = \sum_{k=1}^{K} L\|x(k) - x(k-1)\|^2$.

Also if $P = \min(P, Z)$ *where:* $P = \sum_{k=1}^{K} \alpha(k)$ *and*

$$Z = \frac{2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)}{L\sum_{k=1}^{K}(\|x(k)\|^2+\|x(k-1)\|^2)},$$

then the average regret is bounded by the following relationship:

$$\frac{R(K)}{K} \leq \|D\|^2 \frac{1}{2K\alpha(K)} + \frac{\|\nabla f_m\|^2}{2K}\ \Psi,$$

$$\textit{where}\ \ \Psi = \frac{2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)}{L\sum_{k=1}^{K}\|x(k)\|^2+L\sum_{k=1}^{K}\|x(k-1)\|^2},$$

$L = \max_k L_k$, L_k *is the Lipschitz parameter of* $\nabla f_k(x(k)$, *in Problem* (6.28) *and* $\lim_{K\to\infty} \frac{R(K)}{K}$ *approaches* 0.

Proof. First, by using the results of Lemma 6.0.8, the regret of Algorithm 3 can be expressed as: $R(K) = \sum_{k=1}^{K}(x(k) - x^*)g(k)$. Then from Equation (6.30), the regret $R(K) = \sum_{k=1}^{K}(x(k-1) - \alpha(k-1)\nabla f(x(k-1)) - x^*)g(k)$, where $\alpha(k)$ is as expressed in (6.31) . To prove Theorem 6.0.10, the approach will be to upper-bound the aggregate sum of the step size $\alpha(k)$ and use the generalized bound for online gradient descent in Equation (6.33). This approach is possible since the gradient of the cost function at each time in the sequence of problems is bounded (Assumption 16). Proceeding, the running sum of the step sizes $\alpha(k)$ up to time K is expressed as

$$\sum_{k=1}^{K}\alpha(k) = \sum_{k=1}^{K}\frac{s(k-1)^T s(k-1)}{s(k-1)^T y(k-1)} = \sum_{k=1}^{K}\frac{\|x(k)-x(k-1)\|^2}{(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))}.$$

By applying the result in Lemma 6.0.9 to the right hand side of the preceding inequality, we obtain that:

$$\sum_{k=1}^{K}\alpha(k) \geq \frac{(\sum_{k=1}^{K}\|(x(k)-x(k-1))\|)^2}{\sum_{k=1}^{K}(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))} \tag{6.34}$$

By inspection, if write the first few terms of the numerator of equation (6.34), it is evident that equation (6.34) can be further lower bounded according to the following:

$$\sum_{k=1}^{K}\alpha(k) \geq \frac{(\|(x(1)-x(0)\|+\|(x(2)-x(1)\|)^2}{\sum_{k=1}^{K}(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))} \tag{6.35}$$

Clearly because the terms $\|(x(1)-x(0)\|$ and $\|(x(2)-x(1)\|$ are positive, the numerator of equation (6.35) can be upper-bounded according to the following:

$$(\|(x(1)-x(0)\|+\|(x(2)-x(1)\|)^2 \leq 2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)$$

To bound the denominator of Equation (6.35), we use the Lipschitz continuity of the gradients of $f(\cdot)$ with parameter $L > 1$. Therefore,

$$\sum_{k=1}^{K}(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1))) \leq \sum_{k=1}^{K} L\|x(k)-x(k-1)\|^2.$$

If we represent the bounds in the numerator and denominator of equation (6.35) by the following variables such that: $b = (\|(x(1)-x(0)\|+\|(x(2)-x(1)\|)^2$, $c = 2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)$, $d = \sum_{k=1}^{K}(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))$, and $e = \sum_{k=1}^{K} L\|x(k)-x(k-1)\|^2$. It has been shown that $b \leq c$ and $d \leq e$. Therefore to find an upper bound for equation (6.35), we use the condition that if $\frac{e-d}{c-b} \leq \frac{d}{b}$, then we obtain: $\frac{b}{d} \leq \frac{c}{e}$ So we obtain the bounds of the right hand side of (6.35) as:

$$\frac{(\|(x(1)-x(0)\|+\|(x(2)-x(1)\|)^2}{\sum_{k=1}^{K}(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))} \leq \frac{2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)}{L\sum_{k=1}^{K}(\|x(k)\|^2+\|x(k-1)\|^2)}$$

If we let the left hand side of equation (6.35) be represented by $P = \sum_{k=1}^{K}\alpha(k)$ and we let the right hand side of equation (6.35) be denoted as the following:

$$Q = \frac{(\|(x(1)-x(0)\|+\|(x(2)-x(1)\|)^2}{\sum_{k=1}^{K}(x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))}$$

Similarly if we let the derived upper bound of Q be given by:

$$Z = \frac{2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)}{L\sum_{k=1}^{K}(\|x(k)\|^2+\|x(k-1)\|^2)}$$

From the above analysis, we observe that $P \geq Q$ and $Q \leq Z$. Therefore, if $P = \min(P, Z)$, then we can deduce that $P \leq Z$.

By the established relationship between P and Z and also using the triangle inequality, we obtain the bound for using the first BB step size as:

$$\sum_{k=1}^{K} \alpha(k) \leq \frac{2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)}{L\sum_{k=1}^{K}\|x(k)\|^2+L\sum_{k=1}^{K}\|x(k-1)\|^2}$$

By using the regret bound equation in (6.33), we obtain: $R(K) \leq \|D\|^2 \frac{1}{2\alpha(K)} + \frac{\|\nabla f_m\|^2}{2}\Psi$,

$$\text{where } \Psi = \frac{2(\|(x(1)-x(0)\|^2+\|(x(2)-x(1)\|^2)}{L\sum_{k=1}^{K}\|x(k)\|^2+L\sum_{k=1}^{K}\|x(k-1)\|^2}.$$

The average regret over K time steps can then be expressed as

$$\frac{R(K)}{K} \leq \|D\|^2 \frac{1}{2K\alpha(K)} + \frac{\|\nabla f_m\|^2}{2K}\Psi.$$

Since $\|D\|$ is constant based on its value in (6.33), and $\|\nabla f_m\|^2$ is also constant, we conclude that that the average regret $\lim_{K\to\infty} \frac{R(K)}{K}$ approaches 0. □

Next, we consider the performance of Algorithm 3 using the second BB step-size in Equation (6.32).

Theorem 6.0.11. *Consider Problem* (6.28) *and let Algorithm 3 be used to solve Problem* (6.28) *where*

$$\alpha(k) = \frac{s(k-1)^T y(k-1)}{y(k-1)^T y(k-1)};$$

and L is the maximum of all Lipschitz continuity parameters of all gradients of the cost function in

Problem (6.28)*, then, the regret is upper bounded by*

$$R(K) \leq \|D\|^2 \frac{1}{2\alpha(K)} + \frac{\|\nabla f_m\|^2}{2}\zeta,$$

where $\zeta = (\sum_{k=1}^{K}(((A(k)^T)^2)^{\frac{1}{2}}(\sum_{k=1}^{K}((B(k))^2)^{\frac{1}{2}}(\sum_{k=1}^{K}((C(k))^2)^{\frac{1}{2}}$*. and the average regret* $\lim_{K\to\infty}\frac{R(K)}{K}$ *approaches* 0*.*

Proof. The approach to proving Theorem 6.0.11 will be similar to that of Theorem 6.0.10, where we will obtain bounds for the aggregate sum of the step sizes in $R(K)$ and use the generalized bound for online gradient descent algorithm. In this case, the sum of the aggregate step sizes is expressed as

$$\sum_{k=1}^{K}\alpha(k) = \sum_{k=1}^{K}\frac{s(k-1)^T y(k-1)}{y(k-1)^T y(k-1)}.$$

By using the relationship $s(k-1) \triangleq x(k) - x(k-1)$ and $y(k-1) = \nabla f(x(k)) - \nabla f(x(k-1))$. By noting that $y(k-1)^T y(k-1) = \|y(k-1)\|^2$, and also expressing as a product of three different functions, we obtain the following relationship:

$$\sum_{k=1}^{K}\alpha(k) = \sum_{k=1}^{K}((x(k)-x(k-1))^T(\nabla f(x(k))-\nabla f(x(k-1)))\|\nabla f(x(k))-\nabla f(x(k-1)\|^{-2}) \tag{6.36}$$

For the purpose of clarity, let $A(k) = ((x(k)-x(k-1))$, $B(k) = (\nabla f(x(k))-\nabla f(x(k-1)))$ and $C(k) = \|\nabla f(x(k))-\nabla f(x(k-1)\|^{-2}$. Applying the Cauchy-Schwarz inequality to the right hand side of Equation (6.36), yields:

$$\sum_{k=1}^{K}\alpha(k) = \sum_{k=1}^{K}(A(k)^T B(k))C(k) \leq (\sum_{k=1}^{K}(((A(k)^T)^2)^{\frac{1}{2}}(\sum_{k=1}^{K}((B(k))^2)^{\frac{1}{2}}(\sum_{k=1}^{K}((C(k))^2)^{\frac{1}{2}}.$$

Applying the generalized regret bound as seen in Equation (6.33), we obtain the regret $R(K)$ as:

$$R(K) \leq \|D\|^2 \frac{1}{2\alpha(K)} + \frac{\|\nabla f_m\|^2}{2}\zeta,$$

where the value of ζ is the upper bound of $\sum_{k=1}^{K} \alpha(k)$ obtained above after applying Cauchy-Schwarz inequality and it is given by:

$$\zeta = (\sum_{k=1}^{K}(((A(k)^T)^2)^{\frac{1}{2}}(\sum_{k=1}^{K}((B(k))^2)^{\frac{1}{2}}(\sum_{k=1}^{K}((C(k))^2)^{\frac{1}{2}}.$$

Therefore the average regret is given by the bounds: $\frac{R(K)}{K} \leq \|D\|^2 \frac{1}{2K\alpha(K)} + \frac{\|\nabla f_m\|^2}{2K}\zeta$ Furthermore, since $\|D\|$ is constant based on its value in (6.33), and the terms $A(k)$, $B(k)$ and $C(k)$ are also positive, we conclude that the average regret $\lim_{K\to\infty} \frac{R(K)}{K}$ approaches 0.

□

The Barzilai-Borwein step size in the gradient-based Algorithm 3 results in a regret that grows sublinearly in time and yields an average regret of zero as time K goes to infinity. This result differs from the results obtained in [90,104,105] because the authors use online convex programming either via gradient descent or distributed primal dual algorithms to obtain sublinear regret. However, the result of this chapter uses the Barzilai-Borwein method to obtain sublinear regret.

Having explored distributed optimization algorithms and performance under adversarial attack, we give the concluding remarks of this dissertation in the next chapter.

CHAPTER 7: CONCLUSION AND OPEN PROBLEMS

The performance of a distributed gradient algorithm with adversarial attack in a communication-constrained environment is explored as a preventive measure that non-adversarial agents employ to manage the communication resources used up by the adversarial agents. The communication constraints are explored using both fixed and adaptive quantization scheme amidst the presence of adversarial agents. The thesis is presented in a way that adversarial agents can choose to perturb their iterates by either using either the same attack vector or different attack vector at each iteration without affecting the convergence to a neighborhood of the optimal solution. Conditions needed for convergence are established and corroborated via simulations. Detection strategies to identify adversarial attack and resilience against those attacks are also explored in the course of the dissertation.

The adaptive quantizer used by the non-adversarial agents performs well when the objective function is strongly convex. Results show that the agents can detect the presence of adversarial agents using an inverse relationship between resolution and the attack vector. Using a suitable step size, it is shown that convergence to a neighborhood of the optimal solution of the distributed optimization problem is attained despite quantization errors and attack vectors. Scenarios where adversarial agents upscale their estimates using different vector at each iteration are explored and it is shown that convergence to a neighborhood of the optimal solution is unaffected even with one bit of information being exchanged by non-adversarial agents in a network.

Numerical experiments affirm that when adversarial agents send an attack vector at each iteration, increasing the number of bits mostly leads to reduction of errors in approaching the optimal solution despite the presence of attack values. However, as the number of adversarial agents increase in the network, the convergence error increases. All the results on adversarial attack

and quantization presented in this dissertation assumes strong convexity and curious researchers should consider exploring the problems described for convex functions. Finally, in addressing the need to have models with suitable convergence properties that many distributed optimization problems - including adversarial optimization- demands, this dissertation examines performance of Quasi-Newton methods specifically the Barzilai-Borwein (BB) gradient method. The convergence analysis explored in this dissertation regarding the BB method holds for strongly convex quadratic functions and interested readers and researchers can explore the BB algorithm for strongly convex functions and convex functions in general as research topics. Additionally, the analysis of the BB method can be helpful in improving the convergence and performance of adversarial attack problems, and interested researchers can also explore the BB method in such applications.

APPENDIX A: RANGE OF THE BARZILAI-BORWEIN STEP SIZE BOUNDS

Proof of Corollary 1

Proof. We establish equation (6.20) in a manner that the range of step size bounds below holds:

$$\frac{1}{L} \leq \alpha_{i1}(k) \leq \frac{2}{\mu + L} \leq \frac{1}{\mu}.$$

We also note that the step size range also applied to both the centralized and distributed form of the step sizes. First, according to [86], we first start by noting that the BB step size can be upper and lower bounded according to: $\frac{1}{L} \leq \alpha_{i1}(k) \leq \frac{1}{\mu}$. To include $2/(\mu + L)$ between the first distributed step size, $\alpha_{i1}(k)$ and $1/\mu$, we prove that $2/(\mu + L) \leq 1/\mu$ and $2/(\mu + L) \geq 1/L$.

To prove that $2/(\mu + L) \leq 1/\mu$, we show that $\frac{1}{\mu} - \frac{2}{\mu + L} > 0$. To prove this, we need to solve :

$$\frac{1}{\mu} - \frac{2}{\mu + L} = \frac{L - \mu}{\mu(\mu + L)}.$$

We know that $L \geq \mu$ and L and μ are positive, then we obtain that $1/\mu - 2/(\mu + L) > 0$.

Now we will prove that $2/(\mu + L) \geq 1/L$. In doing this, it suffices to show that $\frac{2}{\mu + L} - \frac{1}{L}$ is positive. So we solve:

$$\frac{2}{\mu + L} - \frac{1}{L} = \frac{L - \mu}{L(L + \mu)}.$$

Since $L \geq \mu$ and L and μ are both positive, we have that $2/(\mu + L) \geq 1/L$. We obtain the upper and lower bounds of $\alpha_{i1}(k)$ as

$$\frac{1}{L} \leq \alpha_{i1}(k) \leq \frac{2}{\mu + L} \leq \frac{1}{\mu}.$$

So we conclude that the step size $\alpha_{i1}(k) = 2/(\mu + L)$ lies between the lower $(1/L)$ and lower bounds $(1/\mu)$ of the BB step sizes. The fact that $\overline{\alpha_{i1}}(k) = \frac{1}{n} \sum_{i=1}^{n} \alpha_{i1}(k)$ and $\overline{\alpha_{i1}}(k) \leq \frac{1}{\mu}$ hold completes the proof. □

Proof of Lemma 6.0.2

Proof. From Equation (6.2), we first consider $\|x(k+1)-x^*\|^2$ to obtain bounds for convergence. First, we let $g(k) = \nabla f(x(k))$ and we obtain that: $\|x(k+1)-x^*\| = \|x(k)-x^*-\alpha_2(k)g(k)\|$. By squaring both sides of the preceeding equation, we have:

$$\|x(k)-x^*-\alpha_2(k)g(k)\|^2 = \|x(k)-x^*\|^2+\alpha_2^2(k)\|g(k)\|^2-2\left(x(k)-x^*\right)^T\left(\alpha_2(k)g(k)\right). \tag{A.1}$$

For all vectors a, b, $2a^Tb \leq \|a\|^2+\|b\|^2$. We now have the relationship:$2\left(x(k)-x^*\right)^T\left(g(k)\right) \leq \|g(k)\|^2+\|x(k)-x^*\|^2$. By strong convexity assumption with parameter μ, Lipschitz constant L, and constants c_1, c_2 expressed as $c_1 = 2/(\mu+L)$ and $c_2 = 2\mu L/(\mu+L)$, we obtain:

$$\begin{aligned}\|x(k)-x^*-\alpha_2(k)g(k)\|^2 &\leq \|x(k)-x^*\|^2+\alpha_2^2(k)\|g(k)\|^2-\alpha_2(k)c_1\|g(k)\|^2-\alpha_2(k)c_2\|x(k)-x^*\|^2,\\ &\leq (1-\alpha_2(k)c_2)\|x(k)-x^*\|^2.\end{aligned} \tag{A.2}$$

where the last inequality is due to Theorem 2.1.12 from chapter 2 of [70] and the term $(\alpha_2^2(k)-\alpha_2(k)c_2)\|g(k)\|^2 \leq 0$ provided $\alpha_2(k) \leq c_1$. According to the verification as seen in a similar manner in the proof of corollary A, the second step size bounds satisfies the bound: $\frac{1}{L} \leq \alpha_2(k) \leq \frac{2}{\mu+L} \leq \frac{1}{\mu}$. Therefore we have the bounds:

$$\|x(k+1)-x^*\|^2 \leq (1-\alpha_2(k)c_2)\,\|x(k)-x^*\|^2. \tag{A.3}$$

By dividing both sides of equation (A.3) by $\|x(k)-x^*\|^2$, the following holds:

$$\frac{\|x(k+1)-x^*\|^2}{\|x(k)-x^*\|^2} \leq 1-\alpha_2(k)c_2. \tag{A.4}$$

Taking the square roots of both sides of equation (A.4) yields the following bounds:

$$\frac{\|x(k+1)-x^*\|}{\|x(k)-x^*\|} \leq (1-\alpha_2(k)c_2)^{\frac{1}{2}}.$$

Now we analyse the right hand side of the above equation by bounding $(1-\alpha_2(k)c_2)^{\frac{1}{2}}$. The second Barzilai-Borwein step size $\alpha_2(k)$ is given by:

$$\alpha_2(k) = \frac{s(k-1)^T y(k-1)}{y(k-1)^T y(k-1)}. \tag{A.5}$$

By using Lipschitz continuity of $\nabla f(\cdot)$, with L as the Lipschitz constant, we obtain according to [86] that the second BB step size is lower bounded by $1/L$. If $\alpha_2(k)$ and c_2 are positive and $\alpha_2(k) > 1/L$, then $-\alpha_2(k)c_2 < -c_2/L$. Since $\alpha_2(k) > 1/L$, and

$$0 < 1-\alpha_2(k)c_2 < 1-\frac{c_2}{L},$$

This implies that

$$0 < (1-\alpha_2(k)c_2)^{\frac{1}{2}} < \left(1-\frac{c_2}{L}\right)^{\frac{1}{2}}.$$

To prove that $c_2/L < 1$, if $c_2 = 2\mu L/(\mu+L)$, then we obtain the fact that $c_2/L = 2\mu/(\mu+L)$. If $\mu < L$, it implies that $\mu+\mu < L+\mu$ and we obtain $2\mu/(\mu+L) < 1$. Therefore we have the following relationship:

$$\lim_{k\to\infty} \quad \frac{\|x(k+1)-x^*\|}{\|x(k)-x^*\|} < \left(1-\frac{c_2}{L}\right)^{\frac{1}{2}} < 1,$$

from which we conclude that the iterates $x(k)$ converge Q-linearly to the optimal point, x^*. □

Proof of Lemma 6.0.4

Proof. We first consider the first step size α_{i1} as expressed in equation (6.13), and for notation simplicity, we denote $Z = W \otimes I_p \in \mathbb{R}^{np \times np}$. Then the distributed iteration at time step k is:

$$X(k+1) = ZX(k) - \alpha_{i1}(k)\nabla f(X(k)). \tag{A.6}$$

where $\otimes$ is the kronecker product. From the definitions of $\overline{x}(k)$ as the average of local estimates, we have the expression: $\overline{x}(k) = \frac{1}{n}\sum_{i=1}^{n} x_i(k)$. Likewise, from the definition of $g(x(k))$ as the average of local gradients, we obtain the relationship: $g(x(k)) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i\left(x_i(k)\right)$. If we solve for $X(k)$ in equation (A.6) where α_{i1} is the BB step size, we obtain the expression:

$$X(k) = -\sum_{m=0}^{k-1} \alpha_1(m)\left(W^{(k-1-m)} \otimes I\right)\nabla f\left(x(m)\right). \tag{A.7}$$

Suppose $\overline{X}(k)$ is the average of all concatenated $x_i(k)$, then we obtain the expression below:

$$\overline{X}(k) = \frac{1}{n}\left(\left(1_n 1_n^T\right) \otimes I\right)x(k). \tag{A.8}$$

Also, we know that the following relationship holds:

$$\|x_i(k) - \overline{x}(k)\| \leq \|X(k) - \overline{X}(k)\|. \tag{A.9}$$

where the distributed form of the first BB step size is given in α_{i1} as expressed in equation (6.13). From equations (A.7) and (A.8), we obtain:

$$\begin{aligned}\|x_i(k)-\overline{x}(k)\| \leq \|X(k)-\overline{X}(k)\| &= \|X(k)-\frac{1}{n}\left(\left(1_n 1_n^T\right)\otimes I\right)X(k)\| = \|-\sum_{m=0}^{k-1}\alpha_{i1}(m)(W^{(k-1-m)}\otimes I)\nabla f(x(m)) \\ &+\frac{1}{n}((1_n 1_n^T)\otimes I)\sum_{m=0}^{k-1}(\alpha_{i1}(m)(W^{(k-1-m)}\otimes I)\nabla f(x(m)))\|, \\ &= \|-\sum_{m=0}^{k-1}(\alpha_{i1}(m)(W^{(k-1-m)}\otimes I)\nabla f(x(m))) + \sum_{m=0}^{k-1}\frac{\alpha_{i1}(m)}{n}((1_n 1_n^T)\otimes I)\nabla f(x(m))\|.\end{aligned} \tag{A.10}$$

Because W is doubly stochastic, then we have the relationship:

$$\|x_i(k)-\overline{x}(k)\| \leq -\sum_{m=0}^{k-1}\|\alpha_{i1}(m)((W^{(k-1-m)}-\frac{1}{n}1_n 1_n^T)\otimes I)\nabla f(x(m)).\|$$

If

$$\alpha_{i1}(k) = \frac{s_i(k-1)^T s_i(k-1)}{s_i(k-1)^T y_i(k-1)},$$

then

$$\alpha_{i1}(m) = \frac{s_i(m-1)^T s_i(m-1)}{s_i(m-1)^T y_i(m-1)}.$$

Therefore we have the expression:

$$\|x_i(k)-\overline{x}(k)\| \quad \leq \quad -\sum_{m=0}^{k-1}\frac{\|s_i(m-1)\|^2}{s_i(m-1)^T y_i(m-1)}\|W^{(k-1-m)}-\frac{1}{n}1_n 1_n^T\|\|\nabla f(x(m))\|.$$

So we obtain that the following expression:

$$\|x_i(k)-\overline{x}(k)\| \leq \sum_{m=0}^{k-1}\frac{\|s_i(m-1)\|^2}{s_i(m-1)^T y_i(m-1)}\lambda^{(k-1-m)}\|\nabla f(x(m)).$$

If $\nabla f(x(m))$ is bounded meaning that $\|\nabla f(x(m)\| \leq G$ where G is positive, then we have:

$$\|x_i(k) - \overline{x}(k)\| \leq \sum_{m=0}^{k-1} \frac{G\|s_i(m-1)\|^2}{s_i(m-1)^T y_i(m-1)} \lambda^{(k-1-m)}. \tag{A.11}$$

The eigenvalues λ of the weight matrix W satisfies the bounds, $0 < \lambda \leq 1$. From equation (A.11) and by Cauchy-Schwarz on sums, we obtain:

$$\|x_i(k) - \overline{x}(k)\| \leq G \left(\sum_{m=0}^{k-1} \alpha_{i1}^2(m) \right)^{\frac{1}{2}} \left(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)} \right)^{\frac{1}{2}}. \tag{A.12}$$

We know that the BB step size is upper bounded such that $\alpha_i < 1/\mu$. By squaring both sides, $\alpha_i^2 \leq \frac{1}{\mu^2}$ and we obtain the relationship

$$\sum_{m=0}^{k} \alpha_i^2(m) \leq \frac{k}{\mu^2}.$$

Equivalently, we obtain the result

$$\left(\sum_{m=0}^{k} \alpha_i^2(m) \right)^{\frac{1}{2}} \leq \frac{\sqrt{k}}{\mu}.$$

In equation (A.12),

$$\|x_i(k) - \overline{x}(k)\| \leq G \left(\sum_{m=0}^{k-1} \alpha_{i1}^2(m) \right)^{\frac{1}{2}} \left(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)} \right)^{\frac{1}{2}}. \tag{A.13}$$

where

$$(\sum_{m=0}^{k} \alpha_i^2(m))^{\frac{1}{2}} \leq \frac{\sqrt{k}}{\mu},$$

and

$$(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)})^{\frac{1}{2}} \leq (\frac{1}{1-\lambda^2})^{\frac{1}{2}} = \frac{1}{\sqrt{1-\lambda^2}} \triangleq \mathrm{Q}_3.$$

Moreover if

$$\sum_{m=0}^{k-1} \alpha_{i1}^2(m) \leq \frac{1}{G^2 \sum_{m=0}^{k-1} \lambda^{2(k-1-m)}}$$

holds, then each local agent's estimates converges Q-linearly to its average; that is $\|x_i(k) - \overline{x}(k)\| \leq 1$.

□

Proof of Lemma 6.0.6

Proof. We consider the second step size α_{i2} as expressed in equation (6.14), and for notation simplicity, we denote $Z = W \otimes I_p \in \mathbb{R}^{np \times np}$. Then the distributed iteration at time step k is:

$$X(k+1) = ZX(k) - \alpha_{i2}(k) \nabla f(X(k)). \tag{A.14}$$

Similarly based on the definition of $\overline{x}(k)$ as the average of local estimates, then we have $\overline{x}(k) = \frac{1}{n} \sum_{i=1}^{n} x_i(k)$. Because $g(x(k))$ to be the average of local gradients, So we obtain the expression $g(x(k)) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x_i(k))$. If we solve equation (A.14) for $X(k)$ where α_{i2} is the second BB step size, we obtain the expression:

$$X(k) = -\sum_{m=0}^{k-1} \alpha_2(m) \left(W^{(k-1-m)} \otimes I \right) \nabla f\left(x(m)\right). \tag{A.15}$$

Let $\overline{X}(k)$ be the average of all concatenated $x_i(k)$, then we obtain:

$$\overline{X}(k) = \frac{1}{n} \left(\left(1_n 1_n^T \right) \otimes I \right) x(k). \tag{A.16}$$

Also, we know that the following relationship holds.

$$\|x_i(k) - \overline{x}(k)\| \leq \|X(k) - \overline{X}(k)\|. \tag{A.17}$$

We note that in equation (A.15), α_{i2} is expressed in equation (6.14). From equations (A.16) and (A.17), we obtain the result:

$$\begin{aligned}\|x_i(k)-\overline{x}(k)\| \leq \|X(k)-\overline{X}(k)\| &= \|X(k)-\frac{1}{n}\left(\left(1_n 1_n^T\right)\otimes I\right)X(k)\| = \|-\sum_{m=0}^{k-1}\alpha_{i2}(m)(W^{(k-1-m)}\otimes I)\nabla f(x(m)) \\ &+\frac{1}{n}((1_n 1_n^T)\otimes I)\sum_{m=0}^{k-1}(\alpha_{i2}(m)(W^{(k-1-m)}\otimes I)\nabla f(x(m)))\|, \\ &= \|-\sum_{m=0}^{k-1}(\alpha_{i2}(m)(W^{(k-1-m)}\otimes I)\nabla f(x(m))) + \sum_{m=0}^{k-1}\frac{\alpha_{i2}(m)}{n}((1_n 1_n^T)\otimes I)\nabla f(x(m))\|.\end{aligned} \tag{A.18}$$

Because W is doubly stochastic, then we obtain the following bounds:

$$\|x_i(k)-\overline{x}(k)\| \leq -\sum_{m=0}^{k-1}\|\alpha_{i2}(m)((W^{(k-1-m)}-\frac{1}{n}1_n 1_n^T)\otimes I)\nabla f(x(m))\|.$$

If the second BB step size is given by:

$$\alpha_{i2}(k) = \frac{s_i(k-1)^T y_i(k-1)}{y_i(k-1)^T y_i(k-1)},$$

then we have the relationship:

$$\alpha_{i2}(m) = \frac{s_i(m-1)^T y_i(m-1)}{y_i(m-1)^T y_i(m-1)}.$$

$$\|x_i(k)-\overline{x}(k)\| \quad \leq \quad -\sum_{m=0}^{k-1}\frac{s_i(m-1)^T y_i(m-1)}{\|y_i(m-1)\|^2}\|W^{(k-1-m)}-\frac{1}{n}1_n 1_n^T\|\|\nabla f(x(m))\|.$$

We now obtain the bound:

$$\|x_i(k)-\overline{x}(k)\| \leq \sum_{m=0}^{k-1}\frac{s_i(m-1)^T y_i(m-1)}{\|y_i(m-1)\|^2}\lambda^{(k-1-m)}\|\nabla f(x(m))\|.$$

If $\nabla f(x(m))$ is bounded meaning that $\|\nabla f(x(m))\| \leq G$ where G is positive, then we obtain:

$$\|x_i(k) - \overline{x}(k)\| \leq \sum_{m=0}^{k-1} \frac{s_i(m-1)^T y_i(m-1)}{\|y_i(m-1)\|^2} \lambda^{(k-1-m)} G. \tag{A.19}$$

and λ is the second largest eigenvalue of W. We also note that the weight matrix W satisfies the inequality, $0 < \lambda \leq 1$. From equation (A.19) and by Cauchy-Schwarz on sums, we obtain:

$$\|x_i(k) - \overline{x}(k)\| \leq G \left(\sum_{m=0}^{k-1} \alpha_{i2}^2(m) \right)^{\frac{1}{2}} \left(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)} \right)^{\frac{1}{2}}. \tag{A.20}$$

We know that the second BB step size is upper bounded such that $\alpha_i < \frac{1}{\mu}$. By squaring both sides, $\alpha_{i2}^2 \leq \frac{1}{\mu^2}$ and we obtain

$$\sum_{m=0}^{k} \alpha_i^2(m) \leq \frac{k}{\mu^2}.$$

Equivalently, we obtain the result

$$\left(\sum_{m=0}^{k} \alpha_{i2}^2(m) \right)^{\frac{1}{2}} \leq \frac{\sqrt{k}}{\mu}.$$

In equation (A.12), the following expression holds:

$$\|x_i(k) - \overline{x}(k)\| \leq G \left(\sum_{m=0}^{k-1} \alpha_{i2}^2(m) \right)^{\frac{1}{2}} \left(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)} \right)^{\frac{1}{2}}, \tag{A.21}$$

where

$$(\sum_{m=0}^{k} \alpha_{i2}^2(m))^{\frac{1}{2}} \leq \frac{\sqrt{k}}{\mu},$$

and

$$(\sum_{m=0}^{k-1} \lambda^{2(k-1-m)})^{\frac{1}{2}} \leq \left(\frac{1}{1-\lambda^2} \right)^{\frac{1}{2}} = \frac{1}{\sqrt{1-\lambda^2}} \triangleq \mathrm{Q}_3.$$

□

Proof of Lemma 6.0.7

Proof. Let the variables $\overline{x}(k)$, $g(k)$, and $\overline{\alpha_{i2}}(k)$ be defined just as they were in Lemma 6.0.5 except that $\overline{\alpha_{i2}}(k)$ is now the average of the second distributed BB step size. We consider $\|\overline{x}(k+1)-x^*\|^2$ to obtain bounds for convergence. So we start with the iterate: $\|\overline{x}(k+1)-x^*\| = \|\overline{x}(k)-x^*-\overline{\alpha_{i2}}g(k)\|$. Squaring both sides and simplifying the right hand side of the above equation yields:

$$\|x(k)-x^*-\overline{\alpha_{i2}}(k)g(k)\|^2=\|x(k)-x^*\|^2+\overline{\alpha_{i2}}^2(k)\|g(k)\|^2-2\left(x(k)-x^*\right)^T\left(\overline{\alpha_{i2}}(k)g(k)\right). \tag{A.22}$$

By using vector norm principles, for vectors a, b, $2a^Tb \leq \|a\|^2+\|b\|^2$, therefore we obtain: $2\left(\overline{x}(k)-x^*\right)^T\left(g(k)\right) \leq \|g(k)\|^2+\|\overline{x}(k)-x^*\|^2$. Just as was defined in Lemma 6.0.5, μ and L are strong convexity and Lipschitz parameters respectively and c_1, c_2 are given by $c_1 = 2/(\mu+L)$ and $c_2 = 2\mu L/(\mu+L)$. Therefore we have the relationship:

$$\begin{aligned}\|\overline{x}(k+1)-x^*-\overline{\alpha_{i2}}(k)g(k)\|^2 &\leq \|\overline{x}(k)-x^*\|^2+\overline{\alpha_{i2}}^2(k)\|g(k)\|^2-\overline{\alpha_{i2}}(k)c_1\|g(k)\|^2-\overline{\alpha_{i2}}(k)c_2\|\overline{x}(k)-x^*\|^2,\\ &\leq (1-\overline{\alpha_{i2}}(k)c_2)\|\overline{x}(k)-x^*\|^2+(\overline{\alpha_{i2}}^2(k)-\overline{\alpha_{i2}}(k)c_1)\|g(k)\|^2,\\ &\leq (1-\overline{\alpha_{i2}}(k)c_2)\|\overline{x}(k)-x^*\|^2.\end{aligned} \tag{A.23}$$

We note that the last inequality is due to Theorem 2.1.12 from chapter 2 of [70]. We also note that $(\overline{\alpha_{i2}}^2(k)-\overline{\alpha_{i2}}(k)c_1)\|g(k)\|^2 \leq 0$ provided $\overline{\alpha_{i2}}(k) \leq c_1$. We also note that $\overline{\alpha_{i2}}(k) = c_1$ is within the range of the BB step size bounds below and the details are shown in Chapter A, Appendix A: Therefore the distributed BB convergence using the second BB step size can be finalized as:

$$\|\overline{x}(k+1)-x^*\|^2 \leq \left(1-\overline{\alpha_{i2}}(k)c_2\right)\|\overline{x}(k)-x^*\|^2. \tag{A.24}$$

Dividing both sides of equation (A.24) by $\|\overline{x}(k)-x^*\|^2$ yields the following bounds:

$$\frac{\|\overline{x}(k+1)-x^*\|^2}{\|\overline{x}(k)-x^*\|^2} \leq 1-\overline{\alpha_{i2}}(k)c_2. \tag{A.25}$$

By taking the square roots of both sides of equation (A.25), we obtain the relationship:

$$\frac{\|\overline{x}(k+1)-x^*\|}{\|\overline{x}(k)-x^*\|} \leq (1-\overline{\alpha_{i2}}(k)c_2)^{\frac{1}{2}}.$$

We will now bound: $(1-\overline{\alpha_{i2}}(k)c_2)^{\frac{1}{2}}$ where $\alpha_{i2}(k)$ is expressed in equation (6.14) and $\overline{\alpha_{i2}}(k) = \frac{1}{n}\sum_{i=1}^{n}\alpha_{i2}(k)$. By using Lipschitz continuity of $\nabla f(\cdot)$ with L as the Lipschitz constant, the distributed form of the second BB step size is lower bounded by $\frac{1}{L}$. Now, $\overline{\alpha_{i2}}(k) = \frac{1}{n}\sum_{i=1}^{n}\alpha_{i2}(k)$, and it results to

$$n\overline{\alpha_{i2}}(k) = \sum_{i=1}^{n}\alpha_{i2}(k).$$

Also, $\alpha_{i2}(k) > \frac{1}{L}$, $\alpha_{i2}(k) < \sum_{i=1}^{n}\alpha_{i2}(k)$ and we obtain the result:

$$\frac{1}{L} < \alpha_{i2}(k) < \sum_{i=1}^{n}\alpha_{i2}(k). \tag{A.26}$$

From equation (A.26),

$$n\overline{\alpha_{i2}}(k) = \sum_{i=1}^{n}\alpha_{i2}(k) > \frac{1}{L},$$

and we have $\overline{\alpha_{i2}}(k) > \frac{1}{nL}$. If $\overline{\alpha_{i2}}(k)$ and c_2 are positive and $\overline{\alpha_{i2}}(k) > 1/nL$, then $-\overline{\alpha_{i2}}(k)c_2 < -c_2/nL$. Therefore, $\overline{\alpha_{i2}}(k) > 1/nL$. it implies that $0 < 1-\overline{\alpha_{i2}}(k)c_2 < 1-c_2/nL$, So we obtain the bounds:

$$0 < (1-\overline{\alpha_{i2}}(k)c_2)^{\frac{1}{2}} < \left(1-\frac{c_2}{nL}\right)^{\frac{1}{2}}.$$

It has been established in Lemma 6.0.5 that $c_2/nL \leq 1$ for all positive values of n. The convergence analysis is finalized according to the following relationship:

$$\lim_{k\to\infty} \quad \frac{\|\overline{x}(k+1)-x^*\|}{\|\overline{x}(k)-x^*\|} \leq \left(1-\frac{c_2}{nL}\right)^{\frac{1}{2}} \leq 1,$$

Therefore we conclude that the average of the estimates converges Q-linearly to the optimal point, x^*. □

APPENDIX B: ERROR DUE TO PROJECTION BOUNDS PROOF

Proof of Lemma 1

Proof. We begin with the relationship:

$$\bar{\boldsymbol{\xi}}((k) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}_i(\boldsymbol{h}_i(k)). \tag{B.1}$$

By squaring both sides of equation (B.1), and using the bound of $\sum_{i=1}^{n}\|\boldsymbol{\xi}_i(\boldsymbol{h}_i(k))\|^2$ shown in [40], we obtain

$$\sum_{i=1}^{n}\|\boldsymbol{\xi}_i(\boldsymbol{h}_i(k))\|^2 \leq 8\sum_{i=1}^{n}\|\Delta_i(k)\|^2+2\bar{L}^2\alpha^2(k). \tag{B.2}$$

Equivalently, the expression below holds:

$$\|\bar{\boldsymbol{\xi}}((k)\| \leq \frac{\sqrt{8}}{n}\left(\sum_{i=1}^{n}\|\Delta_i(k)\|^2\right)^{1/2} + \frac{\sqrt{2}\bar{L}\alpha}{n}. \tag{B.3}$$

If $\ell \leq 1$, then $\|\Delta_i(k)\| \leq 1$ and $\|\Delta_i(k)\|^2 \leq 1$. So we obtain the bounds below:

$$\sum_{i=1}^{n}\|\Delta_i(k)\|^2 \leq \sum_{i=1}^{n}1 = n.$$

From equation (B.3),

$$\frac{\sqrt{8}}{n}\left(\sum_{i=1}^{n}\|\Delta_i(k)\|^2\right)^{1/2} \leq \frac{\sqrt{8}\sqrt{n}}{n} = \frac{\sqrt{8}}{\sqrt{n}} = \sqrt{\frac{8}{n}}.$$

When $n \geq 1$, we obtain the following bound:

$$\sqrt{8}\left(\sum_{i=1}^{n}\|\Delta_i(k)\|^2\right)^{1/2} \leq \sqrt{8}\sum_{i=1}^{n}\Delta_i(k). \tag{B.4}$$

By dividing both sides of equation (B.4) by n, we obtain the bounds:

$$\frac{\sqrt{8}}{n}\left(\sum_{i=1}^{n}\|\Delta_i(k)\|^2\right)^{1/2} \leq \frac{\sqrt{8}}{n}\sum_{i=1}^{n}\Delta_i(k) = \sqrt{8}\Delta(k),$$

and the norm of the error due to projection is bounded as:

$$\|\bar{\boldsymbol{\xi}}(k)\| \leq \sqrt{8}\Delta(k) + \sqrt{2}\frac{\bar{L}}{n}\alpha. \tag{B.5}$$

By squaring both sides of equation (B.5), we obtain

$$\|\bar{\boldsymbol{\xi}}(k)\|^2 \leq 8\|\Delta(k)\|^2 + 2\bar{L}^2\alpha^2(k),$$

which consequently proves Lemma 5.0.1. □

LIST OF REFERENCES

[1] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[2] S. Yang, Q. Liu, and J. Wang, "Distributed optimization based on a multiagent system in the presence of communication delays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 5, pp. 717–728, 2016.

[3] E. Montijano and A. R. Mosteo, "Efficient multi-robot formations using distributed optimization," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6167–6172.

[4] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *2012 50th annual allerton conference on communication, control, and computing (allerton)*. IEEE, 2012, pp. 1543–1550.

[5] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[6] Y. Arjevani and O. Shamir, "Communication complexity of distributed convex learning and optimization," in *Advances in neural information processing systems*, 2015, pp. 1756–1764.

[7] I. Emiola, L. Njilla, and C. Enyioha, "On distributed optimization in the presence of malicious agents," 2021.

[8] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

[9] N. Ravi, A. Scaglione, and A. Nedić, "A case of distributed optimization in adversarial environment," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5252–5256.

[10] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[11] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[12] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3141–3155, 2018.

[13] X. Zhang, J. Liu, Z. Zhu, and E. S. Bentley, "Compressed distributed gradient descent: Communication-efficient consensus over networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2431–2439.

[14] H. Li, H. Zhang, Z. Wang, Y. Zhu, and Q. Han, "Distributed consensus-based multi-agent convex optimization via gradient tracking technique," *Journal of the Franklin Institute*, vol. 356, no. 6, pp. 3733–3761, 2019.

[15] M. Maybury, "Detecting malicious insiders in military networks," MITRE CORP BEDFORD MA, Tech. Rep., 2006.

[16] M. Yemini, A. Nedić, A. J. Goldsmith, and S. Gil, "Characterizing trust and resilience in distributed consensus for cyberphysical systems," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 71–91, 2021.

[17] Y. Chen, S. Kar, and J. M. Moura, "Resilient distributed estimation through adversary detection," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2455–2469, 2018.

[18] C. Zou, F. Yang, J. Song, and Z. Han, "Underwater wireless optical communication with one-bit quantization: A hybrid autoencoder and generative adversarial network approach," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[19] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2010.

[20] J. S. Baras and X. Liu, "Trust is the cure to distributed consensus with adversaries," in *2019 27th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2019, pp. 195–202.

[21] C. Zhao, J. He, and Q.-G. Wang, "Resilient distributed optimization algorithm against adversary attacks," in *2017 13th IEEE International Conference on Control & Automation (ICCA)*. IEEE, 2017, pp. 473–478.

[22] S. Sundaram and B. Gharesifard, "Secure local filtering algorithms for distributed optimization," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 1871–1876.

[23] A.-Y. Lu and G.-H. Yang, "Distributed secure state estimation in the presence of malicious agents," *IEEE Transactions on Automatic Control*, 2020.

[24] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 16–29, 2008.

[25] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. T. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 900–908.

[26] J. Zhang, P. Jaipuria, A. Chakrabortty, and A. Hussain, "A distributed optimization algorithm for attack-resilient wide-area monitoring of power systems: Theoretical and experimental methods," in *International Conference on Decision and Game Theory for Security*. Springer, 2014, pp. 350–359.

[27] B. Kailkhura, S. Brahma, and P. K. Varshney, "Consensus based detection in the presence of data falsification attacks," *arXiv preprint arXiv:1504.03413*, 2015.

[28] R. Duo, X. Nie, N. Yang, C. Yue, and Y. Wang, "Anomaly detection and attack classification for train real-time ethernet," *IEEE Access*, vol. 9, pp. 22 528–22 541, 2021.

[29] L.-N. Liu and G.-H. Yang, "A resilient distributed optimization strategy against false data injection attacks," *Optimal Control Applications and Methods*, 2022.

[30] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, and E. S. Alomari, "A new proactive feature selection model based on the enhanced optimization algorithms to detect drdos attacks," *Int. J. Electr. Comput. Eng*, vol. 12, no. 2, pp. 1869–1880, 2022.

[31] J. Yang, P. Ning, X. S. Wang, and S. Jajodia, "Cards: A distributed system for detecting coordinated attacks," in *Information Security for Global Information Infrastructures: IFIP TC11 Sixteenth Annual Working Conference on Information Security August 22–24, 2000, Beijing, China 15*. Springer, 2000, pp. 171–180.

[32] S. Magnússon, C. Enyioha, N. Li, and C. Fischione, "Practical coding schemes for bandwidth limited one-way communication resource allocation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 221–226.

[33] J. Lei, H.-F. Chen, and H.-T. Fang, "Primal–dual algorithm for distributed constrained optimization," *Systems & Control Letters*, vol. 96, pp. 110–117, 2016.

[34] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[35] I. Emiola, L. Njilla, and C. Enyioha, "On distributed optimization in the presence of malicious agents," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, 2021, pp. 1–6.

[36] N. Ravi, A. Scaglione, and A. Nedić, "A case of distributed optimization in adversarial environment," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5252–5256.

[37] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 4177–4184.

[38] M. Rabbat and R. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[39] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization design for distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2107–2120, 2017.

[40] T. T. Doan, S. T. Maguluri, and J. Romberg, "Fast convergence rates of distributed subgradient methods with adaptive quantization," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2191–2205, 2021.

[41] S. Zhu, M. Hong, and B. Chen, "Quantized consensus admm for multi-agent distributed optimization," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4134–4138.

[42] C.-S. Lee, N. Michelusi, and G. Scutari, "Finite rate quantized distributed optimization with geometric convergence," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1876–1880.

[43] B. Wang, M. Safaryan, and P. Richtárik, "Theoretically better and numerically faster distributed optimization with smoothness-aware quantization techniques," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9841–9852, 2022.

[44] F. Alimisis, P. Davies, and D. Alistarh, "Communication-efficient distributed optimization with quantized preconditioners," in *International Conference on Machine Learning*. PMLR, 2021, pp. 196–206.

[45] J. Zhang, K. You, and L. Xie, "Innovation compression for communication-efficient distributed optimization with linear convergence," *IEEE Transactions on Automatic Control*, 2023.

[46] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 380–392, 2014.

[47] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[48] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization design for distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2107–2120, 2016.

[49] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "Quantized decentralized consensus optimization," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 5838–5843.

[50] H. Peng, J. Wu, Z. Zhang, S. Chen, and H.-T. Zhang, "Deep network quantization via error compensation," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[51] S. Khirirat, S. Magnússon, and M. Johansson, "Compressed gradient methods with hessian-aided error compensation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 998–1011, 2020.

[52] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized sgd and its applications to large-scale distributed optimization," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5325–5333.

[53] M. Danilova and E. Gorbunov, "Distributed methods with absolute compression and error compensation," in *Mathematical Optimization Theory and Operations Research: Recent Trends: 21st International Conference, MOTOR 2022, Petrozavodsk, Russia, July 2–6, 2022, Revised Selected Papers*. Springer, 2022, pp. 163–177.

[54] X. Qian, P. Richtárik, and T. Zhang, "Error compensated distributed sgd can be accelerated," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 401–30 413, 2021.

[55] H. Tang, Y. Li, J. Liu, and M. Yan, "Errorcompensatedx: error compensation for variance reduced algorithms," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 102–18 113, 2021.

[56] I. Emiola and C. Enyioha, "Quantized and distributed subgradient optimization method with malicious attack," *IEEE Control Systems Letters*, vol. 7, pp. 181–186, 2023.

[57] T. T. Doan, S. T. Maguluri, and J. Romberg, "Fast convergence rates of distributed subgradient methods with adaptive quantization," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2191–2205, 2020.

[58] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, 2014, pp. 583–598.

[59] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[60] B. Li, S. Cen, Y. Chen, and Y. Chi, "Communication-efficient distributed optimization in networks with gradient tracking and variance reduction," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7331–7381, 2020.

[61] Y. Yu, J. Wu, and L. Huang, "Double quantization for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[62] A. Agarwal, M. J. Wainwright, and J. C. Duchi, "Distributed dual averaging in networks," in *Advances in Neural Information Processing Systems*, 2010, pp. 550–558.

[63] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.

[64] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[65] A. Nedich, D. P. Bertsekas, and V. S. Borkar, "Distributed asynchronous incremental subgradient methods," *Studies in Computational Mathematics*, vol. 8, no. C, pp. 381–407, 2001.

[66] D. Mateos-Núnez and J. Cortés, "Distributed subgradient methods for saddle-point problems," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5462–5467.

[67] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4177–4184.

[68] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *International Conference on Machine Learning*, 2019, pp. 344–353.

[69] D. Das, S. Avancha, D. Mudigere, K. Vaidynathan, S. Sridharan, D. Kalamkar, B. Kaul, and P. Dubey, "Distributed deep learning using synchronous stochastic gradient descent," *arXiv preprint arXiv:1602.06709*, 2016.

[70] Y. Nesterov, "Introductory lectures on convex programming volume i: Basic course," *Lecture notes*, vol. 3, no. 4, p. 5, 1998.

[71] A. Fischer, "A special newton-type optimization method," *Optimization*, vol. 24, no. 3-4, pp. 269–284, 1992.

[72] B. T. Polyak, "Newton's method and its use in optimization," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1086–1096, 2007.

[73] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed newton method for network optimization," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 2736–2741.

[74] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization–i: Algorithm," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.

[75] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-newton methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2613–2628, 2017.

[76] Y. Li, Y. Gong, N. M. Freris, P. Voulgaris, and D. Stipanović, "Bfgs-admm for large-scale distributed optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1689–1694.

[77] D. Pu and W. Tian, "The revised dfp algorithm without exact line search," *Journal of computational and applied mathematics*, vol. 154, no. 2, pp. 319–339, 2003.

[78] Q. Liu, R. Sang, and Q. Zhang, "Fpga-based acceleration of davidon-fletcher-powell quasi-newton optimization method," *Transactions of Tianjin University*, vol. 22, no. 5, pp. 381–387, 2016.

[79] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA journal of numerical analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[80] D. Jakovetić, J. M. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented lagrangian algorithms," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, 2014.

[81] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[82] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[83] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, "Geometrically convergent distributed optimization with uncoordinated step-sizes," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3950–3955.

[84] Y.-H. Dai and L.-Z. Liao, "R-linear convergence of the barzilai and borwein gradient method," *IMA Journal of Numerical Analysis*, vol. 22, no. 1, pp. 1–10, 2002.

[85] Y.-H. Dai, "A new analysis on the barzilai-borwein gradient method," *Journal of the operations Research Society of China*, vol. 1, no. 2, pp. 187–198, 2013.

[86] J. Gao, X. Liu, Y.-H. Dai, Y. Huang, and P. Yang, "Geometric convergence for distributed optimization with barzilai-borwein step sizes," *arXiv preprint arXiv:1907.07852*, 2019.

[87] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6350–6364, 2017.

[88] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[89] S. Shalev-Shwartz and S. M. Kakade, "Mind the duality gap: Logarithmic regret algorithms for online optimization," in *Advances in Neural Information Processing Systems*, 2009, pp. 1457–1464.

[90] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 928–936.

[91] B. Mcmahan and M. Streeter, "No-regret algorithms for unconstrained online convex optimization," in *Advances in neural information processing systems*, 2012, pp. 2402–2410.

[92] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2503–2528, 2012.

[93] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 7195–7201.

[94] Y. Zhang, R. J. Ravier, M. M. Zavlanos, and V. Tarokh, "A distributed online convex optimization algorithm with improved dynamic regret," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 2449–2454.

[95] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Transactions on Signal Processing*, vol. 68, pp. 731–746, 2020.

[96] P. Sharma, P. Khanduri, L. Shen, D. J. Bucci Jr, and P. K. Varshney, "On distributed online convex optimization with sublinear dynamic regret and fit," *arXiv preprint arXiv:2001.03166*, 2020.

[97] A. Lesage-Landry, J. A. Taylor, and I. Shames, "Second-order online nonconvex optimization," *IEEE Transactions on Automatic Control*, 2020.

[98] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[99] P. E. Gill and W. Murray, "Quasi-newton methods for unconstrained optimization," *IMA Journal of Applied Mathematics*, vol. 9, no. 1, pp. 91–108, 1972.

[100] Y. Huang, Y.-H. Dai, X.-W. Liu, and H. Zhang, "On the acceleration of the barzilai–borwein method," *Computational Optimization and Applications*, vol. 81, no. 3, pp. 717–740, 2022.

[101] W. Su, S. Boyd, and E. Candes, "A differential equation for modeling nesterov's accelerated gradient method: Theory and insights," in *Advances in Neural Information Processing Systems*, 2014, pp. 2510–2518.

[102] I. Emiola and R. Adem, "Comparison of optimization methods with application to a network containing malicious agents," *arXiv preprint arXiv:2101.10546*, 2021.

[103] H. Sedrakyan and N. Sedrakyan, *Algebraic inequalities*. Springer, 2018.

[104] S. Lee and M. M. Zavlanos, "On the sublinear regret of distributed primal-dual algorithms for online constrained optimization," *arXiv preprint arXiv:1705.11128*, 2017.

[105] P. Sharma, P. Khanduri, L. Shen, D. J. Bucci, and P. K. Varshney, "On distributed online convex optimization with sublinear dynamic regret and fit," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2021, pp. 1013–1017.