STABILITY OF AGGREGATE COMPUTING

by

Yuanqiu Mo

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering in the Graduate College of The University of Iowa

December 2019

Thesis Committee: Soura Dasgupta, Thesis Supervisor. Jacob Beal, Thesis Supervisor. Er-Wei Bai. Jon Kuhl. Raghuraman Mudumbai. Muthu Krishnamurthy.

Copyright by YUANQIU MO 2019 All Rights Reserved To my family.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to my advisors, Professor Soura Dasgupta and Dr. Jacob Beal, for the continuous support throughout the years of my graduate work. I am highly indebted to them for their patience, encouragement, and immense knowledge.

I would like to thank the rest of my thesis committee members, Prof. Er-wei Bai, Prof. Jon Kuhl, Prof. Raghuraman Mudumbai and Prof. Muthu Krishnamurthy for their insightful comments and advices on this work. I am also thankful to the University of Iowa Professors for knowledge sharing through courses, and department staffs Ms. Cathy Kern and Dina Blanc for their support throughout the program.

I would like to extend my sincere thanks to seniors and lab mates, Dr. Wei Shao, Fahim Anjum, Hui Xie, Yue Pan, Jirong yi, for the frequent conversations and help. I must thank Dr. Giorgio Audrito for simplifying the original proof of the leader election algorithm. I would like to express my gratitude for my seniors at Harbin Institute of Technology, Dr. He Zhu, Dr. Qingbo Guo, Dr. Pengcheng Du, Dr. Mingyi Wang for their constant support. I would also like to thank my long time friends Tianyu Gan and Wenqi Zhang for all the help they extended to me in my stay away from home. Finally, I am profoundly grateful to my parents and grandma for their love and support throughout the years of my PhD study.

ABSTRACT

Open cyber-physical systems like smart cities, tactical information sharing, personal and home area networks, and the Internet of Things (IoT) require seamless, low latency, peer to peer local interactions between devices. Their potential is curtailed by the fact that devices currently interact either through device and application specific protocols that are not reusable, or centralized infrastructures like clouds. The recent proposed aggregate computing approach offers a solution to this bottleneck through a multi-layered architecture. In this thesis, we focus on the middle layer of Aggregate Computing, which consists of three classes of basis blocks that are G-block, C-block and T-block, whose compositions, sometimes in feedback, can be used to realize a wide class of coordination tasks. However, the formal analysis of individual blocks is limited to self-stabilization which only involves eventual convergence and is not endowed with robustness properties. Further, the stability analysis of these compositions, though conjectured, is largely unexplored. In this thesis, we will first investigate the robust stability of the G-block and its variants from a control perspective, then analyze the dynamics and characterize the stability conditions for compositions of those basis blocks.

Characterizing each individual block's behavior is necessary in understanding their stable compositions. Thus, we formulate Lyapunov functions for two special G-block distributed algorithms to prove their global uniform asymptotic stability (GUAS) and global uniform exponential stability (GUES) respectively, as well as find ultimate bounds on states and the time to attain them, under persistent structural perturbations. For the generalized G-block, we prove its GUAS and robustness without using a Lyapunov function. With respect to the compositions, we first study a state estimation algorithm using an open-loop G-C combination by analyzing its error bounds and dynamics. We next present a resilient leader election algorithm using a feedback interconnection of those basis blocks, and prove its GUAS and resilience under transient perturbations.

We will show that these basis block distributed algorithms exhibit unusual and subtle state dependencies that are uncommon in standard stability analysis, which changes both the nature of the Lyapunov functions and the analysis. The ultimate boundedness we derive will open up the prospect of establishing small gain type theorems, which in turn helps to demonstrate closed loop stability. Also, the resilient design and stability analysis of the leader election algorithm will assist in improving algorithms based on basis blocks, and providing conditions for stable composability. Ultimately, those analysis works will help us develop constructs and tools that go well beyond existing approaches and thus will fundamentally impact the standard stability analysis.

PUBLIC ABSTRACT

Open cyber-physical systems have dramatically changed how we relate to computing. These systems allow many computing devices to be involved in provisioning parts of any given service, and each device may simultaneously participate in multiple services. Realization of the full potential of these systems requires that devices interact with others in their locality. Ordinary programming approaches are very device-centric and entangle application design with coordination and communication, leading to lack of modularity and reusability. *Aggregate computing* provides an alternate approach, which simplifies the design, creation, and maintenance of complex distributed systems by using a layered approach. This thesis focuses on the middle layer of *aggregate computing*, which comprises three classes of basis blocks, *G*-block, *C*-block and *T*-block, that facilitate resilient device interactions. Previous work only proved those individual blocks to be self-stabilizing without any implication for robustness, which is an important property in system design. Further, the stability analysis of their compositions remains largely unexplored.

Our work first addresses the robust stability of G-block and its variants, proving their global uniform asymptotic stability (GUAS) or global uniform exponential stability (GUES) and robustness under persistent structural perturbations. Then we analyze the error bounds and dynamics of a commonly used G-C combination. Finally, we design a leader election algorithm via a feedback interconnection of basis blocks, and prove its GUAS and resilience.

TABLE OF CONTENTS

LIST	ГС)F FI	GURES	ix
CHA	٩P	ΓER		
	1	INT	RODUCTION	1
		$1.1 \\ 1.2$	Aggregate computing	2 4
		$\begin{array}{c} 1.3\\ 1.4 \end{array}$	Approach to stability of aggregate computing blocks Organization	$5 \\ 8$
	2	G B	LOCK AND ITS SPECIAL CASES	10
		2.1	Introduction	10
		2.2	The generalized G block \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots	10
		2.3	The adaptive Bellman-Ford algorithm	14
		2.4	The generalized adaptive Bellman-Ford (GABF) algorithm	16
		2.5	The most probable path algorithm	19
		2.6	Other examples	20
		2.7	Conclusion	22
	3	GLC) BAL STABILITY THREE SPECIAL CASES OF THE G -BLOCK	24
		3.1	Introduction	24
		3.2	GUAS and robustness of ABF	25
			3.2.1 A Lyapunov function	25
			3.2.2 Global uniform asymptotic stability	34
			3.2.3 Robustness under perturbations	44
			3.2.4 Simulations	54
		3.3	GUES and robustness of MPP	59
			3.3.1 A Lyapunov function	59
			3.3.2 Tight bound on time to convergence	63
			3.3.3 Robustness under perturbations	68
			3.3.4 Simulations	78
		3.4	GUAS of GABF	80
		0.1	3.4.1 Preliminaries	81
			3.4.2 Global uniform asymptotic stability	83
			3.4.3 Simulations	89
		3.5	Conclusion	92
		0.0		-

4	GLO	DBAL STABILITY OF THE MOST GENERAL G BLOCK	94
	4.1	Introduction	94
	4.2	Preliminaries	94
	4.3	The stationary point	97
	4.4	Global uniform and asymptotic convergence	103
	4.5	Robustness under perturbations	112
	4.6	Simulations	125
	4.7	Conclusion	128
5	ANA	ALYSIS OF COMPOSITIONS OF BASIS BLOCKS	129
	5.1	Introduction	129
	5.2	A G - C combination	130
		5.2.1 Error bounds and dynamics	131
		5.2.2 Empirical comparison	137
	5.3	A resilient leader election algorithm	141
		5.3.1 Preliminaries	145
		5.3.2 Algorithm	146
		5.3.2.1 The <i>MABF</i> block	147
		5.3.2.2 The C block \ldots \ldots \ldots \ldots	150
		5.3.2.3 The B block \ldots \ldots \ldots \ldots \ldots	150
		5.3.3 Global uniform asymptotic stability	151
		5.3.4 Simulations	159
	5.4	Conclusion	162
6	COI	NCLUSIONS AND FUTURE WORK	163
	6.1	Summary	163
	6.2	Open problems	164
	0.2		104
R	EFER	RENCES	166

LIST OF FIGURES

riguit

1.1	Illustration of G-block (figure from [35])	4
1.2	Illustration of C-block (figure from [35])	4
1.3	Illustration of T-block (figure from [35])	4
1.4	Example G - C feedback composition (figure from [58]). A set of distributed services are managed by a controller device, which accepts load information as input and provides a resource allocation plan as its output. The blue subsystems are aggregate computing basis blocks. The two to the left of the controller are composed to implement information collection. The resource allocation plan is disseminated by the block to the right	5
1.5	Aggregate Computing abstraction layers (figure from $[10]$)	6
2.1	Illustration of a tactical wireless network. In this example, circles in red and yellow are high and low speed external link, respectively. Blue circles represent nodes routing to the external links. After 3 rounds, all nodes, including the low-speed link, have converged to route through the high- speed link.	21
3.1	Illustration of ABF (adapted from [50]). Individual distance estimates may go up and down, but the greatest overestimate (Δ^+) and least un- derestimate (Δ^-) are monotonic. This example shows a line network of five nodes (circles, source red, others blue) with unit edges (grey links); distance estimates evolve from initial t = 0 to converge to their correct values at t = 4. The numbers on the edges are the edge lengths. The numbers on the nodes are their current distance estimates	27
3.2	Illustration of the tightness of convergence time (figure from [58]). The subgraph comprising the nodes S and $\{1, \dots, n\}$ is used in the proof of Theorem 3.2.5. The entire graph is used in the proof of Theorem 3.2.9.	35
3.3	Plot from [58] of (a) greatest overestimate $\Delta^+(t)$ and (b) least underesti- mate $\Delta^-(t)$ for 10 runs of 500 nodes randomly distributed in a 4x1 sq. km area, showing that overestimates correct much faster than underestimates.	40

3.4	Examples of effective diameter (figure from [58]), showing edge length as labels on edges. In the left graph \mathcal{G} , $\mathcal{D}(\mathcal{G}) = 4$ and comes from the sequence S , A , B and C , where each node is the true constraining node of its successor. On the other hand in the right graph \mathcal{G}^- , $\mathcal{D}(\mathcal{G}^-) = 5$ and comes from the sequence S , A , B , C and D .	47
3.5	Illustration of the tightness of convergence time with perturbations (figure from [58]).	53
3.6	Comparison between (a) the greatest overestimate $\Delta^+(t)$ and $(\mathcal{D}(\mathcal{G})-1)\epsilon$, and (b) comparison between the least underestimate $\Delta^-(t)$ and $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$ (b), as well as their partial enlarged views (c) and (d). In this example, each node is moving within a circle with a radius of $0.5e_{\min}$ (plot from [58]).	55
3.7	Comparison between (a) the greatest overestimate $\Delta^+(t)$, $(\mathcal{D}(\mathcal{G}) - 1)\epsilon$ and the theoretical bound of $\Delta^+(t)$ for unperturbed system, and (b) compar- ison between the least underestimate $\Delta^-(t)$, $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$ and the the- oretical bound of $\Delta^-(t)$ for unperturbed system. In this example, there are two source nodes in the graph, one is at [3.75, 0.5] and the other is at [0.25, 0.5], switching between source nodes every 50 simulated seconds while all nodes are moving within a circle with a radius of $0.5e_{\min}$ (plot from [58]).	57
3.8	Comparison between (a) the greatest overestimate $\Delta^+(t)$, $(\mathcal{D}(\mathcal{G})-1)\epsilon$ and the theoretical bound of $\Delta^+(t)$ for unperturbed system, and (b) compar- ison between the least underestimate $\Delta^-(t)$, $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$ and the the- oretical bound of $\Delta^-(t)$ for unperturbed system. In this example, there are two source nodes in the graph, one is at [3.75, 0.5] and the other is at [0.25, 0.5], switching between source nodes every 1000 simulated seconds while all nodes are moving within a circle with a radius of $0.5e_{\min}$ (plot from [58]).	58
3.9	Illustration of the tightness of convergence time. The entire graph is used in the proof of Theorem 3.3.6.	66
3.10	Illustration of the tightness of convergence time with perturbations	76
3.11	Trace of (a) greatest overestimate $\Delta^+(t)$ and (b) least underestimate $\Delta^-(t)$ for 10 runs, showing that $\Delta^+(t)$ will converge in $\mathcal{D}(\mathcal{G}) - 1$ rounds while $\Delta^-(t)$ takes longer to converge.	78
3.12	Trace of (a) greatest overestimate $\Delta^+(t)$ and its ultimate bound, and (b) least underestimate $\Delta^-(t)$ and its ultimate bound for 5 runs, as well as their partial enlarged views (c) and (d).	79

3.13	Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations	90
3.14	Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = \delta = 2\sqrt{17}$ and $D = 0$	91
3.15	Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = \delta = 2\sqrt{17}$ and $D = 0.06$.	91
3.16	Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = \delta = 0.1$ and $D = 0$.	92
3.17	Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = 2\sqrt{17}, \delta = 0.1$ and $D = 0.$	93
4.1	Illustration of sets $\mathcal{A}(t)$, $\mathcal{E}(t)$, $\mathcal{R}(t)$ and $\mathcal{U}(t)$. Each edge length in the graph is 1, $M = 4, \delta = 1, D = 0, s_i = 0$ for $i = 1, 5$ and $s_i = \infty$ for $i = 2, 3, 4$. In this case, $\mathcal{R}(1) = \mathcal{A}(1) = \{1, 2, 5\}, \mathcal{U}(1) = \mathcal{E}(1) = \{3, 4\}, \ldots, \ldots$	104
4.2	Comparison of time to convergence between GABF and ABF for 100 runs of 500 nodes randomly distributed in a $4 \times 1 \text{ km}^2$ area, showing that GABF converges faster than ABF.	125
4.3	Comparison of robustness between GABF and ABF for 100 runs of 500 nodes randomly distributed in a $4 \times 1 \text{ km}^2$ area, showing that both GABF and ABF have the same ultimate bounds.	127
5.1	Example of achieving the maximum source value (adapted from $[56]$)	134
5.2	Examples of typical estimate variation over time (figure from [56]), showing excerpts from a individual simulation run on a network dispersed through a space (a) 4 units in width and (b) 20 units in width. Notice that most transients are relatively short, but they are much more frequent for hop- count than for Adaptive Bellman-Ford or Flex-Gradient, and they tend to be underestimates rather than overestimates, particularly for hop-count.	138
5.3	The more hops spanned by the network, the higher the relative error in estimates (a) and the smaller amount of time that the estimate is correct. Furthermore, the smoother the input potential, the better estimation performs (figure from [56]).	140

5.4	Plot from [56] of typical histogram of error values, showing that smooth adaptation of potential prevents transient value loss.	141
5.5	Block diagram of the leader election algorithm	148
5.6	Comparison of convergence time between our algorithm and Datta's algorithm without perturbations	160
5.7	Comparison of convergence time between our algorithm and Datta's algorithm with perturbations	161

CHAPTER 1 INTRODUCTION

This thesis develops a framework of studying the stability of compositions of certain basis blocks used to realize resilient interactions at the core of open cyberphysical systems. Each of these blocks is itself a distributed graph algorithm and together they form the nucleus of a recently proposed paradigm known as Aggregate Computing, [10].

As our planet becomes increasingly interdependent and interconnected, the last few decades have witnessed a proliferation of complex networked distributed systems involving compositions of numerous physical and logical systems, which may themselves be distributed. Understanding their dynamics, stability, and reliability is of critical importance. Robustness and stability have been addressed for limited classes of large-scale distributed systems in the controls literature for decades [70], using a mature set of tools from stability theory [45]. In recent years this line of research has been dominated by the control of multiagent systems, exemplified by consensus theory [59] and formation control [9, 49, 23, 33, 68, 67]. Classical stability theory tools like Lyapunov theory, passivity theory [5, 40], center manifold theory [49, 68], and the Perron-Frobenius Theorem [59], are often leveraged in the analysis.

The ongoing dispersion of services into local devices, manifest in the domains of open cyber-physical systems like smart cities, tactical information sharing, personal and home area networks, intelligent transportation, and the Internet of Things (IoT) [10], however, poses new and challenging problems for analysis and design. All require devices to interact *safely and seamlessly* with others in their vicinity through low latency peer to peer communication, and to share tasks. What separates them from more traditional infrastructure reliant cyber-physical systems is that they are *open*, and cannot effectively meet user needs unless they support frequent and noncentralized changes in the applications and services that are being hosted.

Current modes of device interactions restrict the potential of these systems as they are typically either highly constrained and inflexible (e.g., single-use devices) or else rely on remote infrastructure like cloud services. The former lacks modurality and impairs reusability. The latter is centralized with high latency and lacks the agility to exploit local communication, services, and devices [10]. Section 1.1 describes aggregate computing that offers a solution to these problems. Section 1.2 describes the layered approach to aggregate computing. Section 1.3 describes the core issues animating this thesis.

1.1 Aggregate computing

Aggregate computing offers a potential approach to meeting this challenge by permitting seamless coordination between complex distributed services. It has been considered in varying ways by different communities who have generated many programming paradigms that are specific to them, [15, 17]. It views the basic computing unit as a physical region comprising a collection of interacting computing devices, rather than an individual physical device [10]. In particular, [10] introduces a separation of concerns into multiple abstraction layers which are agnostic to each other, much like the OSI model [73] does for communication between individual devices, factoring the overall task of distributed system design for device interactions into sub-tasks of device-level communication and discovery, coherence between collective and local operations, resilience, and programmability.

Among the lowest layers, there are fundamental device interactions and a small universal calculus of aggregate-level field calculus constructs that implement services such as neighborhood discovery and distributed scoping of shared information. The next layer facilitates resilient device interactions and can be described by three types of basis blocks, that are themselves distributed algorithms:

- G Block: This is a spreading block that disseminates information through a network of devices. For example it can be used to compute generalized distances of devices in a network from a set of sources. Alternatively, it can be used by a distinguished set of devices to broadcast information they hold to the devices that are nearest to them (see Figure 1.1).
- C blocks: This collects information to the source down the gradient of the network to be used by interacting units. For example it may inform a coordinating unit the net resources available in a network (see Figure 1.2).
- **T** blocks: This is performs temporal operations, such as delaying the dissemination of information, or by acting as timers or as time limited memories (see Figure 1.3).

Introduced in [16] it has been shown that a broad class of coordination services can be described by various compositions of these three blocks. Figure 1.4 illustrates a

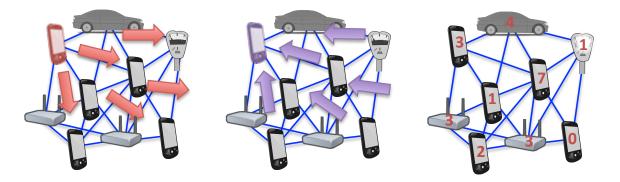


Figure 1.1: Illustration of Figure 1.2: Illustration of C-block (figure from [35]) G-block (figure from [35])

Figure 1.3: Illustration of T-block (figure from [35])

typical example of such services, in this case making use of G and C blocks to realize a resource allocation plan. A key function of these blocks is to coordinate interacting devices using packet-based message passing in a highly distributed environment.

Aggregate computing layers 1.2

Figure 1.5 depicts the abstraction layers of aggregate computing. At the base is field calculus [26], [25], that models device behavior and interaction. The G, C and T basis blocks combine to provide resilient coordination and produce APIs for applications like sensing, decision, and action. These can be invoked in a transparent way to produce API to implement complex services [10].

From these one can realize and describe a system, potentially complex in several pieces: First specifying resilient coordination blocks, then specifying how these impart resilience, then how to realize these blocks and how devices in a network use them.

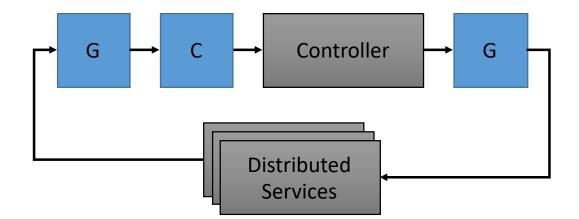


Figure 1.4: Example G-C feedback composition (figure from [58]). A set of distributed services are managed by a controller device, which accepts load information as input and provides a resource allocation plan as its output. The blue subsystems are aggregate computing basis blocks. The two to the left of the controller are composed to implement information collection. The resource allocation plan is disseminated by the block to the right.

1.3 Approach to stability of aggregate computing blocks

Empirically the dynamics of these basis set systems appear amenable to effective composition [50, 10, 11], but, barring [27, 58, 56, 57, 7], to date formal analysis of individual blocks has been limited to self-stabilization [32], which does not ensure the stability of compositions as it does not guarantee robustness to perturbations that among other things, e.g., feedback injects perturbations. Accordingly, this thesis focuses on the robust stability analysis of individual basis block, the resilient design and stability analysis of their compositions, which include feedback.

Specifically, our thesis presents three interrelated sets of works: (i) stability

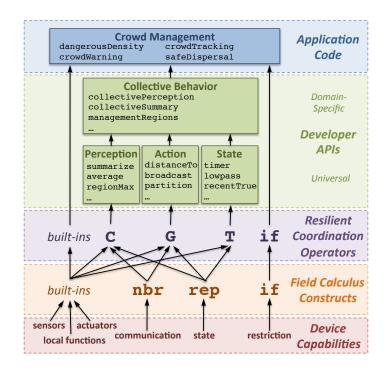


Figure 1.5: Aggregate Computing abstraction layers (figure from [10]).

analysis of the G block and its variants, (ii) stability analysis of an open loop G-C combination and (iii) stability analysis of a distinguished feedback interconnection of G and C blocks for *leader election*. As will be explained in a later chapter, this algorithm outperforms and is a more resilient to transient leader loss and appearance of false leaders than the state of the art, [29], [30]. As a first step in understanding the stability of arbitrary compositions of blocks, it is important to characterize how these individual systems behave under persistent perturbations. Are their stability properties robust to perturbations? Does stability in the ideal unperturbed setting translate to acceptable behavior in the face of perturbations? Such robust behavior is face of a symptotic stability. Rather,

as well understood in adaptive control [3], one should instead show global uniform asymptotic stability (GUAS) of the unperturbed system, as it guarantees total stability [39], an ability to withstand modest departures from idealizing assumptions. Thus our approach is to develop Lyapunov based tools and use of them to show both GUAS and *ultimate boundedness* under *persistent perturbations*. Ultimate bounds refer to those achieved following the lapse of initial transients. Such bounds are critical to addressing issues of composition in open and closed loop. Specifically, ultimate boundedness can be used in variants of the small gain theorem [45] or equivalent theorems [4], like the passivity theorem and its offshoots [37, 36] to demonstrate closed loop stability. While ultimate boundedness by itself is not enough to invoke the classical small gain theorem, there are more sophisticated variants of this theorem that use ultimate bounds [41, 42] to demonstrate closed loop stability. That a Lyapunov framework is possible is shown in Chapter 3, where a Lyapunov function is formulated for two variants of G block, and is used to demonstrate their GUAS and GUES respectively, as well as show their ultimate boundedness under a limited class of structural perturbations.

Most aggregate computing blocks are graph based algorithms that update states at nodes. Analyses in [58], along with other insights we have obtained, reveal certain challenges not confronted in classical stability theory. First, in most discrete time nonlinear systems, particularly those obtained by discretizing a continuous time system, the current state element is an increment on its previous value, making it natural to compare the two values for the same node. What makes the analysis of the G block non-intuitive is that the current state of a node bears no direct relationship to its previous value, but must be compared with previous states at certain distinguished neighbors. This introduces curious dependencies that fundamentally alter the nature of the Lyapunov functions one uses. Our investigation in [56] suggests similar interdependencies characterize the C block. Second, it is also important to tightly bound both the time to converge and the time to attain ultimate bounds. Third, the perturbation analysis in [58] and [57] reveals the need for new techniques involving *bounding graphs*. Fourth, to ensure GUAS and resilience in feedback algorithms like *leader election*, moderate modification for individual blocks is needed.

We conclude by distinguishing this approach to that used in the distributed algorithms literature. The self self stabilization techniques are by their very nature ill-equipped to cope with the type of persistent perturbations that feedback induces. Even the literature of other distributed algorithms like those that find the shortest path between two nodes, [66], [47], [43], [6], [60] and [72], analyze assuming that perturbation eventually subside. By contrast in this thesis we have developed tools that permit meaningful analysis even when the perturbations, though bounded, are perpetual.

1.4 Organization

The rest of the thesis is organized as follows. Chapter 2 introduces the generalized G block explains its significance, giving examples of its applications. It particularly focuses on three special cases: the adaptive Bellman-Ford algorithm (ABF) algorithm, the most probable path algorithm (MPP) and the generalized adaptive Bellman-Ford algorithm (GABF). Chapter 3 provides the robust stability analysis of these three special cases and shows that the first two can be analyzed using the same Lyapunov function. Specifically, it proves the GUAS of ABF and GABF, as well as the GUES of MPP. In Chapter 4, we prove the GUAS of the generalized G block. Chapter 5 presents the stability analysis of compositions of those basis blocks, including an open loop G-C combination and a feedback interconnection of basis blocks. Finally, Chapter 6 concludes the work in this thesis and points out avenues of future research.

CHAPTER 2 G BLOCK AND ITS SPECIAL CASES

2.1 Introduction

The basis blocks for *aggregate computing* are formally described in [71] using the language of field calculus. In the chapter, we translate the G block that is used to spread information like distances of devices from a source set through a network, for those who are unfamiliar with field calculus. The rest of the chapter is organized as the following: Section 2.2 introduces the generalized G block, which is the primitive G block defined in [71]. The Adaptive Bellman-Ford (ABF) algorithm and its generalized version the Generalized Adaptive Bellman-Ford (GABF) algorithm, which are two special cases of G block, are presented in Section 2.3 and Section 2.4 respectively. Another variant of G block, the most probable path algorithm is introduced in Section 2.5. Section 2.6 provides some other applications.

2.2 The generalized G block

We begin with the definition of the generalized G block that spreads information across a spatial network of devices, potentially further organizing and computing as it proceeds. This operator is a generalization covering two of the most commonly used self-stabilizing distributed algorithms: distance estimation and broadcast, as well as a number of other applications, such as forecasting along paths.

Consider an undirected graph $\mathcal{G} = (V, E)$ with $V = \{1, 2, \dots, N\}$ the node set and E the set of edges. Nodes i and k are *neighbors* if they share an edge and can communicate with each other. Denote $\mathcal{N}(i)$ as the set of neighbors of i. With $\hat{x}_i(t)$ the state estimate of i in the *t*-th iteration. Then at its core of the most general Gblock described later in this section, is the *special case* where $\hat{x}_i(t+1) \geq 0$ obeys

$$\hat{x}_{i}(t+1) = \min\left\{\min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{x}_{k}(t), e_{ik}\right) \right\}, s_{i} \right\}, \forall t \ge t_{0}.$$
(2.1)

The e_{ik} define the structural aspects of \mathcal{G} , e.g., they may be the edge lengths between neighbors; s_i is the maximum value that $\hat{x}_i(t)$ can acquire, which may be either finite bound or infinite, though the intention is that at least one s_i is finite.

The function $f(\cdot, \cdot)$ is progressive i.e., for some $\sigma > 0$,

$$f(a,b) > a + \sigma \tag{2.2}$$

and monotonic in the first variable, i.e., $f(a_1, b) \ge f(a_2, b)$, if $a_1 \ge a_2$. and is finite for finite a and b.

The goal is that with fixed s_i and e_{ij} , the state estimates $\hat{x}_i(t)$ converge to their stationary values

$$x_{i} = \min\left\{\min_{k \in \mathcal{N}(i)} \left\{ f\left(x_{k}(t), e_{ik}\right) \right\}, s_{i} \right\}.$$
 (2.3)

We will show later that these values are unique. For example in the ABF described in Section 2.3, $x_i = d_i$ represents the distance of node *i* from a source set *S*. In this case from Bellman's principle of optimality these distances obey

$$d_i = \begin{cases} \min_{j \in \mathcal{N}(i)} \{ d_j + e_{ij} \} & i \notin S \\ 0 & i \in S \end{cases}$$

Thus in this case f(a, b) = a + b and $s_i = 0$ when $i \in S$ and infinity otherwise. As described in Section 2.5, a more general $f(\cdot, \cdot)$ accommodates non-Euclidean distance metrics. For ABF, the stationary values of nodes with finite s_i equal s_i . As described in Section 2.6 it may be desirable for this to not always be the case. Indeed (2.1) accommodates such settings.

As will be shown in this thesis, a drawback of (2.1) is that underestimates can rise very slowly to their stationary values. The most general *G*-block adds certain frills that speeds up this rise. Proposed in [71], for reasons to be described at the end of this section, we use a modified version where the state estimates are updated as

$$\hat{x}_i(t+1) = F\left(\tilde{x}_i(t+1), \hat{x}_i(t), v_i\right)$$
(2.4)

with $\tilde{x}_i(t+1)$ obeying

$$\tilde{x}_{i}(t+1) = \min\left\{\min_{k\in\mathcal{N}(i)}\left\{f\left(\hat{x}_{k}(t), e_{ik}\right)\right\}, s_{i}\right\}, \forall t \ge t_{0},$$
(2.5)

where v_i are certain environmental variables. While $f(\cdot, \cdot)$ continues to satisfy the progressive and monotonic properties, the three variable function $F(\ell_1, \ell_2, v)$ is raising in that for some M it obeys for a *dead zone variable*, $D \ge 0$,

$$F(\ell_1, \ell_2, v) = \begin{cases} \ell_1 & \ell_2 > M \text{ or } |\ell_2 - \ell_1| \le D \\ g(\ell_2) & \text{otherwise} \end{cases},$$
(2.6)

where $M \ge 0$ and the *strictly increasing* g(x) takes finite values for finite x and obeys for some $\delta > 0$,

$$g(x) \ge x + \delta \tag{2.7}$$

In (2.4) and (2.6), $\hat{x}_i(t) = \ell_2$. The second bullet in (2.6) permits a faster initial ascent of $\hat{x}_i(t)$. The first bullet yields a state estimate *identical to* (2.1). We will later show that eventually the algorithm always uses the first bullet. Observe that the stationary points of this generalized block is identical to (2.3) as the second bullet of (2.6) cannot be invoked at a stationary point. As such M, δ and D are additional parameters that guide the rate of ascent. Observe (2.1) is a special case with $D = \infty$, as then the second bullet of (2.6) can never be invoked.

There are differences both algorithmic and analytical from the original block given in [71]. First, the dead zone D is zero, in [71]. While this will perform well in the absence of structural perturbations like those in e_{ij} . In the presence of persistent perturbations $\ell_2 = \ell_1$ cannot be generically sustained. Thus the second bullet of (2.6) will almost always be invoked when the state estimate is less than M. This will cause the state estimates to repeatedly rise to the artificial ceiling of M.

There is also a subtle difference between the assumptions made in [71], and those we make here. In [71], ℓ_i are assumed to lie in a Noetherian ring with M the maximal element. Translated to our settings this means that one *a priori* assumes the boundedness of the states of the algorithm with M, serving as an upper bound. For distance estimation, this implicitly assumes that one knows the largest possible distance in the network and that M exceeds that value. For open networks that can grow unpredictably this is an unappealing assumption.

In contrast we do not assume that the distance estimates are *a priori* bounded. Nor do we assume that M exceeds the largest distance estimate. Thus the proof of self-stabilization given in [71] does not apply to our setting. Rather we prove in Chapter 4 that the algorithm is GUAS without assuming an *a priori* bound on the state estimates.

2.3 The adaptive Bellman-Ford algorithm

We turn to an archetypal and commonly used version of the aggregate computing G block, the Adaptive Bellman-Ford algorithm, a globally asymptotically stable variant of the classical Bellman-Ford algorithm [18], [34], which estimates the shortest distances of nodes in an undirected graph from the nearest source in a distributed fashion. ABF eliminates the constraint that all initial distance estimates should be larger than the actual distances, which is needed for the classical Bellman-Ford algorithm, indicating that the *classical Bellman-Ford algorithm is not globally uniformly asymptotically stable*. In particular, under persistent topological perturbations (e.g. from interaction with other components in a feedback system), these stringent initial condition requirements cannot be met.

Consider an undirected graph $\mathcal{G} = (V, E)$ with $V = \{1, 2, \dots, N\}$ the node set and E the set of edges. We define the edge length between node i and node j as e_{ij} , and assume that there exists an e_{\min} such that:

$$e_{ij} > e_{\min} > 0, \ \forall i \in V \text{ and } j \in \mathcal{N}(i),$$

$$(2.8)$$

i.e., edge lengths between neighbors are all positive. Define distance d_{ij} between two nodes as the shortest walk from i to j. The principle of optimality specifies the recursion:

$$d_{ij} = \min_{k \in \mathcal{N}(i)} \{ e_{ik} + d_{jk} \}.$$
 (2.9)

This is also in effect a statement of the triangle inequality. A subset S of the nodes

in the graph will form a source set. the goal of ABF is to find the shortest distance between each node and the source set S. More precisely we must find:

$$d_i = \min_{k \in S} \{ d_{ik} \}.$$
 (2.10)

In view of (2.9), d_i obeys the recursion:

$$d_i = \begin{cases} 0, & i \in S \\ & & \\ \min_{k \in \mathcal{N}(i)} \{e_{ik} + d_k\} & i \notin S \end{cases}$$
(2.11)

In the classical Bellman-Ford algorithm [18, 34] distance from every node in an arbitrary graph to a designated source node is estimated by the relaxation of a triangle inequality constraint across weighted graph edges. However, the classical algorithm only works if the initial distance estimates are all overestimates, i.e. with t_0 the initial time, for all i

$$\hat{d}_i(t_0) \ge d_i. \tag{2.12}$$

In an interconnected environment, the input to the algorithm may be graph topology or the source set, which may change over time: At a given instant the current estimate may well fall below the true current distance. Classical Bellman-Ford cannot survive such perturbations, prompting the adaptive variant.

ABF is based closely on the classical Bellman-Ford algorithm, but unlike that algorithm, computes distances to the nearest member of a set of source nodes rather than just a single node. Moreover, we wish to support the case where the set of sources and/or the graph may change. Thus ABF is an *adaptive algorithm* that a) sets the distance estimate of every source node to zero, and b) for all other nodes, rather than starting at infinity and always decreasing, recomputes distance estimates periodically, ignoring the current estimate at a node and using only the minimum of the triangle inequality constraints of its neighbors. In particular, suppose $\hat{d}_i(t)$ is the current estimated distance of *i* from the source set. Then the algorithm is:

$$\hat{d}_{i}(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \left\{ \hat{d}_{j}(t) + e_{ij} \right\} & i \notin S \\ 0 & i \in S \end{cases}, \ \forall t \ge t_{0}. \tag{2.13}$$

The behavior of this algorithm reduces to something very close to the classical Bellman-Ford in the case where there is precisely one source node and neither the graph nor the source ever change.

ABF is a special case of (2.4) with $M = -\infty$. It can be readily verified that (2.13) satisfies the progressive and monotonic properties required by the generalized G block. In this case, the maximum value $s_i = \infty$ if i is a non-source node and $s_i = 0$ if i is a source.

2.4 The generalized adaptive Bellman-Ford (GABF) algorithm

Though ABF is GUAS and robust to perturbations, underestimates in ABF are slow to converge. In this section, we present another variant of the generalized G block, GABF, as an alternative to the ABF. In fact while ABF is a direct special case of (2.1), GABF is a special case of (2.15, 2.6). Thus, this algorithm has additional parameters that can be tuned to achieve faster convergence while maintaining the identical ultimate bounds as ABF under same structural perturbations.

Define $\tilde{d}_i(t+1)$ as the following: With S the set of sources and e_{ij} obeying

(2.8), for all $t \ge t_0$,

$$\tilde{d}_{i}(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \{ \hat{d}_{j}(t) + e_{ij} \}, & i \notin S \\ 0, & i \in S \end{cases}$$
(2.14)

Then for suitable $D \ge 0$ and M > 0, distance estimates $\hat{d}_i(t)$ evolves as:

$$\hat{d}_{i}(t+1) = \begin{cases} \tilde{d}_{i}(t+1), & |\tilde{d}_{i}(t+1) - \hat{d}_{i}(t)| \leq D \text{ or } \hat{d}_{i}(t) > M \\ g(\hat{d}_{i}(t)), & \text{otherwise} \end{cases}$$
(2.15)

where $M \ge 0$ is a finite number and the *strictly increasing* g(x) takes finite values for finite x and obeys for some $\delta > 0$,

$$g(x) \ge x + \delta. \tag{2.16}$$

Further,

$$\widetilde{d}_i(t_0) \ge 0 \text{ and } \widehat{d}_i(t_0) \ge 0, \ \forall i \in V.$$

$$(2.17)$$

Thus unlike the ABF, (2.14-2.16) does not assume that distance estimates of the sources are anchored to zero.

Evidently, GABF is also an instance of (2.4). Initially, most estimates update according to the the second bullet of (2.15) and for large δ rise rapidly until exceeding M, then the first bullet is invoked. While the second may again be invoked, we show in Chapter 3 that there comes a time after which using the first bullet yields immediate convergence.

Our proof in Chapter 3 will show that underestimates are eventually eliminated. Though not explicitly quantified in our proof, in generic networks and large M and δ this elimination is rapid, especially if M exceeds the graph diameter. In such cases the *rising value problem* [14] that slows the convergence of underestimates in ABF is obviated. As our proof is for all M > 0 and $\delta > 0$, it provides a conservative estimate of the convergence time. However, simulations confirm rapid convergence when M and δ are large.

Yet we argue that (2.15) and indeed (2.4) are fundamentally nonrobust to perturbations in the edge lengths. This is so as when $M > \hat{d}_i$, update using the first bullet requires the precise satisfaction of $\tilde{d}_i(t + 1) = \hat{d}_i(t)$. Because of (2.14) this precise equality cannot be sustained under perturbations in the e_{ij} . This results in the repeated use of the second bullet and the persistent rise of estimates to M. In other words ultimate bounds under structural perturbations become M. Thus instead of (2.15) we use for some $D \ge 0$

$$\hat{d}_{i}(t+1) = \begin{cases} \tilde{d}_{i}(t+1), \ |\tilde{d}_{i}(t+1) - \hat{d}_{i}(t)| \le D \text{ or } \hat{d}_{i}(t) > M \\ g(\hat{d}_{i}(t)), \text{ otherwise} \end{cases}$$
(2.18)

Indeed (2.14,2.16,2.18) is GABF. Of course (2.15) is a special case of (2.18), with $D = \infty$. A larger D tolerates larger perturbations. At the same time it brings GABF closer to ABF slowing convergence. This of course accords with the behavior of most algorithms. Faster systems are high pass filters that amplify effects of noise. As such M, D and δ are additional parameters that guide convergence rate and response to perturbations.

2.5 The most probable path algorithm

The most probable path algorithm (MPP) is also a special case of G block. While ABF and GABF focus on the shortest path problem, in MPP, conventional geographic distance between nodes is replaced by a probabilistic value, and instead of finding a path between two nodes in a graph such that the total sum of edge weights is minimized, MPP aims to find a path such that the multiplication of weights of its constituent edges is maximized. The MPP problem has been studied as a routing approach in various applications, e.g., channel cognitive radio networks, networkdriven contagion phenomena and time-ordered graphs (TOGs), [44] and so on. The most common solution to this problem is to apply a – log operation on the edge weight represented by a probabilistic value and then run a Dijkstra-like algorithm [31, 44, 21], or dynamically search the maximum probability using maximization recursively [64].

Consider the undirected graph $\mathcal{G} = (V, E)$. In MPP, e_{ij} between node *i* and *j* obeys

$$0 < e_{\min} \le e_{ij} \le e_{\max} < 1, \ \forall i \in V \text{ and } j \in \mathcal{N}(i), \tag{2.19}$$

and reflects the success rate of delivery between the connected two nodes. A source set $S \subset V$ comprises nodes called a source. The MPP algorithm aims to enable each non-source node in the graph find a path with the maximum success rate or equivalently the minimum failure rate of delivery from the source set.

Suppose $\hat{p}_i(t)$ is the estimated failure rate of delivery of *i* from the source set

at time t. Then at time t + 1, $\hat{p}_i(t + 1)$ will update as

$$\hat{p}_{i}(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \{1 - (1 - \hat{p}_{j}(t))e_{ij}\} & i \notin S \\ 0 & i \in S \end{cases}$$
(2.20)

In our MPP algorithm, the estimated failure rate for a source is fixed at 0. While for a non-source node, it seeks to find the minimum estimated failure rate at each iteration while ignoring its previous estimated probability. Specifically, in the second bullet of (2.20), $1 - \hat{p}_j(t)$ represents the estimated success rate of delivery of node j, $(1 - \hat{p}_j(t))e_{ij}$ represents the estimated success rate of i by communicating with j, and consequently $1 - (1 - \hat{p}_j(t))e_{ij}$ stands for the estimated failure rate of i through j.

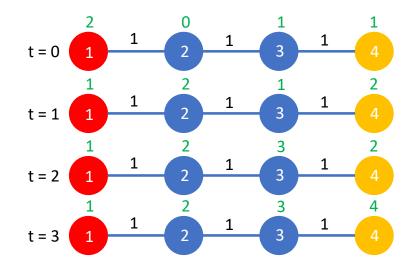
According to the principle of optimality, by using the metric of (2.20), p_i the true failure rate of delivery of *i* from the source set follows:

$$p_{i} = \begin{cases} \min_{j \in \mathcal{N}(i)} \{1 - (1 - p_{j}) \cdot e_{ij}\} & i \notin S \\ 0 & i \in S \end{cases}$$
(2.21)

2.6 Other examples

The foregoing is by no means an exhaustive set of useful specializations of the general G-block. One important algorithm is *broadcast*, where each node in a network receives a distinguished value from the source nearest to it.

To see how this can be achieved call the node j that minimizes the first bullet in the right hand side of (2.13) the current constraining node of i. Then each node receives the value held by its current constraining node. Though this thesis exclusively assumes that $\hat{x}_i(t)$ is a scalar, one can conceive an added state attached to each node



as an alternative implementation of broadcast.

Figure 2.1: Illustration of a tactical wireless network. In this example, circles in red and yellow are high and low speed external link, respectively. Blue circles represent nodes routing to the external links. After 3 rounds, all nodes, including the low-speed link, have converged to route through the high-speed link.

One of the hallmarks of ABF is that nodes with finite maximum states s_i converge to s_i . In some applications this may not be desirable. Thus consider (2.1) with

$$f(\hat{x}_k(t), e_{ik}) = \hat{x}_k(t) + e_{ik}.$$
(2.22)

Further, the maximum value s_i obeys

$$s_i = \begin{cases} 1 & i = 1 \\ 5 & i = 4 \\ \infty & \text{otherwise} \end{cases}$$
(2.23)

This algorithm here is the same as ABF except that finite s_i are not necessary zero, are unequal and $\hat{x}_1(0), \hat{x}_4(0)$ are not fixed at their maximum values. This setting is depicted in Figure 2.1. Node 1 (red) and 4 (yellow) represent high speed and low speed links respectively, to external networks in a tactical wireless network, and the numbers in green refer to the state estimates. By execution of (2.1), all nodes try to route to the external network through the shortest effective path. After 3 rounds, all nodes, including the low-speed link, converge to route the traffic through the highspeed link, where in this case the stationary state of low-speed link need not equal to its maximum value even though that value is finite. On the other hand, should node 1 the gateway to the high speed link disappear, then all route through node 4, and its final state equals its maximum value 4.

2.7 Conclusion

In this chapter, we present a modified version of the generalized G block and its special cases especially ABF, GABF and MPP. We show that the G block accommodates both Euclidean and non-Euclidean metrics, and can be instantiated to fit various applications. In the next chapter, GUAS or GUES and ultimate boundedness under certain structural perturbations of ABF, GABF and MPP algorithms are proved. While the analysis for GABF is not Lyapunov based, that of ABF and MPP are and intriguingly use identical Lyapunov functions.

CHAPTER 3 GLOBAL STABILITY THREE SPECIAL CASES OF THE G-BLOCK

3.1 Introduction

Rather than first addressing the stability of the most general $G_b lock$, in this chapter, we prove the robust stability of three distinguished special cases described in Chapter 2, namely the Adaptive Bellman-Ford (ABF), the Most Probable Path (MPP) and the Generalized Adaptive Bellman-Ford (GABF) algorithms. The next chapter addresses the most general G-block.

This is so as while the stability of GABF and the general *G*-block is established without a Lyapunov analysis, the same attractive Lyapunov function is used to prove the stability of ABF and MPP. Moreover, while in the general case we provide an upper bound on the time to convergence, for ABF and MPP this bound is tight. Further all three are of particular importance and the proof of GABF provides insights into the proof of the general block.

We also analyze robustness to persistent perturbations due to perceived or real changes in edge length caused by noise and mobility. The perturbations are assumed to be bounded but persistent in every iteration. They never settle down to permit convergence. In contrast analysis in distantly related algorithms including various search and path planning algorithms ([66, 46, 47, 43, 6, 60]), assumes that the changes are sufficiently slow to permit convergence between successive instances of structural changes. In all cases we provide ultimate bounds on the perturbations in the state estimates and the time to attain them. Here again the analyses of ABF and MPP are different from the general G-block, as for these the ultimate bounds are *tight* as are the bounds on the time to attain them. As shown in Chapter 4, the corresponding bounds on the general G-block are not tight and are obtained under an additional Lipschitz condition on $f(\cdot, \cdot)$ in (2.1). Sections 3.2-3.4 respectively deal with ABF, MPP and GUAS.

3.2 GUAS and robustness of ABF

In this section, we provide a Lyapunov analysis of ABF described in (2.13), including ultimate bounding under a specific class of persistent structural perturbations, [58], [27]. The following assumption holds for ABF.

Assumption 1. The graph $\mathcal{G} = (V, E)$ is connected, undirected, with edge lengths $e_{ij} \in E$ obeying (2.8) and the distance d_i of node *i* from the source set $S \neq V$ obeying (2.11).

3.2.1 A Lyapunov function

We now provide a Lyapunov function that will be used to prove the GUAS of (2.13). We first need a few definitions. Recall the true distance d_i for $i \in V$ with Vthe set of nodes defined in (2.11), we define the following definition.

Definition 3.1. A *j* that minimizes the right hand side of (2.11) is a true constraining node of *i* of a connected graph \mathcal{G} . As there may be two nodes *k* and *l* such that $d_l + e_{il} = d_k + e_{ik}$, a node may have multiple true constraining nodes. The set of true constraining nodes of a node $i \in V \setminus S$ is denoted as $\mathcal{C}(i)$. Consider any sequence of nodes such that the predecessor of each node is one of its true constraining node. We define $\mathcal{D}(\mathcal{G})$, the effective diameter of \mathcal{G} , as the longest length such a sequence can have in \mathcal{G} .

In view of (2.11) the following holds:

$$d_k < d_i, \ \forall k \in \mathcal{C}(i). \tag{3.1}$$

Correspondingly, for ABF defined in (2.13), we have the following definition,

Definition 3.2. A minimizing j in the first equation of (2.13) will be called a current constraining node of i at time t.

We now show that the effective diameter is finite.

Lemma 3.2.1. Under Assumption 1, $\mathcal{D}(\mathcal{G})$ defined above is finite.

Proof. As defined in Definition 3.1, consider a sequence of nodes k_i in \mathcal{G} such that, for all k_{i-1} is a true constraining node of k_i . Since there are only a finite number of nodes in the graph, the only way that $\mathcal{D}(\mathcal{G})$ can be infinite is if for some i > j, $k_i = k_j$. From (3.1) this leads to the contradiction:

$$d_{k_i} > d_{k_j} = d_{k_i}.\tag{3.2}$$

The goal of this section is to postulate a discrete time Lyapunov function and demonstrate that it is non-decreasing. On the face of it, distance estimation errors,

$$\Delta_i(t) = \hat{d}_i(t) - d_i, \tag{3.3}$$

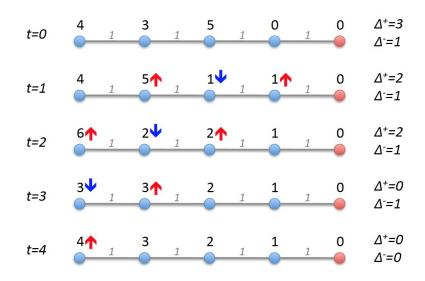


Figure 3.1: Illustration of ABF (adapted from [50]). Individual distance estimates may go up and down, but the greatest overestimate (Δ^+) and least underestimate (Δ^-) are monotonic. This example shows a line network of five nodes (circles, source red, others blue) with unit edges (grey links); distance estimates evolve from initial t = 0 to converge to their correct values at t = 4. The numbers on the edges are the edge lengths. The numbers on the nodes are their current distance estimates.

appear to form a natural measure of the algorithm's performance. However, as seen in Figure 3.1, $\Delta_i(t)$ may well increase in magnitude for individual nodes. This stems from the nature of ABF given in (2.13): $\hat{d}_i(t+1)$ does not explicitly depend on $\hat{d}_i(t)$. Instead, as will be evident in the sequel, depending on its sign, $\Delta_i(t+1)$ bears a natural comparison with $\Delta_j(t)$ where j is among one of two distinguished neighbors of i: either a true constraining node of i or a current constraining node at time t. This subtlety constitutes a key distinction between the analysis here and typical discrete time Lyapunov analyses.

One requires a more global point of comparison from one iteration to the next. As empirically studied in [50], the greatest overestimate of the error $\Delta^+(t)$ and the least underestimate of the error $\Delta^-(t)$ below collectively provide such a comparison:

$$\Delta^{+}(t) = \max\left[0, \max_{i} \Delta_{i}(t)\right]$$
(3.4)

$$\Delta^{-}(t) = \max\left[0, -\min_{i} \Delta_{i}(t)\right].$$
(3.5)

Should, as empirically suggested by [50], each of these be non-increasing then their sum forms a natural Lyapunov function:

$$L(t) = \Delta^+(t) + \Delta^-(t). \tag{3.6}$$

Indeed in Figure 3.1, while individual Δ_i may increase in magnitude, Δ^+ and Δ^- never do. The rest of this section verifies the validity of (3.6) as a Lyapunov function.

We begin by noting that this function clearly meets the non-negativity requirement for a Lyapunov function as

$$L(t) \ge 0, \tag{3.7}$$

with equality holding iff for all i, $\Delta_i(t) = 0$. As a matter of fact, it can readily be verified that L(t) acts as a valid norm for a vector of the distance estimation errors.

As a preface to proving that L(t) is also non-increasing, we define $\mathcal{K}_+(t)$ as a set comprising all nodes whose error equals $\Delta^+(t)$. More precisely:

$$\mathcal{K}_{+}(t) = \left\{ i \in V | \Delta_{i}(t) = \Delta^{+}(t) \right\}.$$
(3.8)

Similarly,

$$\mathcal{K}_{-}(t) = \left\{ i \in V | \Delta_i(t) = -\Delta^{-}(t) \right\}.$$
(3.9)

If $\Delta^+(t) \neq 0$ then each member of $\mathcal{K}_+(t)$ has the largest estimation error. This is however, not necessarily true if $\Delta^+(t) = 0$, as then $\Delta_i(t) \leq 0$, for all $i \in V$. If $\Delta^-(t) \neq 0$ then its members $\mathcal{K}_-(t)$ have the most negative estimation error. We now prove the non-increasing property of $\Delta^+(t)$.

Lemma 3.2.2. Consider (2.13) under Assumption 1. Then with Δ^+ defined in (3.4), for all t,

$$\Delta^+(t+1) \le \Delta^+(t). \tag{3.10}$$

Further, consider $\mathcal{K}_{+}(t)$ in (3.8), and suppose $\Delta_{+}(t) > 0$. Then equality in (3.10) holds iff there exists $j \in \mathcal{K}_{+}(t)$ that is both a current and a true constraining node (see definitions 3.1 and 3.2) of a member of $\mathcal{K}_{+}(t+1)$.

Proof. As $\Delta^+(\cdot) \ge 0$, (3.10) holds if $\Delta^+(t+1) = 0$. Assume $\Delta^+(t+1) > 0$ throughout the proof. Consider $l \in \mathcal{K}_+(t+1)$ and any neighbor $j \in \mathcal{N}(l)$ that is a true constraining node of l, i.e. from (2.11) and Definition 3.1,

$$d_l = d_j + e_{lj} \tag{3.11}$$

Then from (3.8) we find that (3.10) is proved through:

$$\Delta^{+}(t+1) = \Delta_{l}(t+1)$$

$$= \hat{d}_{l}(t+1) - d_{l}$$

$$\leq \hat{d}_{j}(t) + e_{lj} - d_{l} \qquad (3.12)$$

$$= \hat{d}_{j}(t) + e_{lj} - e_{lj} - d_{j}$$

$$= \Delta_{j}(t)$$

$$\leq \Delta^{+}(t), \qquad (3.13)$$

where (3.12) comes from (2.13), and (3.13) from (3.4).

Suppose there is a $j \in \mathcal{K}_+(t)$ that is both a constraining and a current constraining node of $l \in \mathcal{K}_+(t+1)$. From Definition 3.1, (3.11) holds; (3.12) is an equality as $j \in \mathcal{C}(l)$; and (3.13) is an equality as $j \in \mathcal{K}_+(t)$. Thus equality in (3.10) holds.

Now suppose equality in (3.10) holds. Then in the sequence of inequalities above one can choose $l \in \mathcal{K}_+(t+1)$ and $j \in \mathcal{C}(l)$, i.e. a j that obeys (3.11), for which both (3.12) and (3.13) are equalities. From (3.8), (3.13) implies that $j \in \mathcal{K}_+(t)$. As (3.11) implies that $j \in \mathcal{C}(l)$ and $l \in \mathcal{K}_+(t+1)$ this must mean a node in $\mathcal{K}_+(t)$ is a true constraining node of $l \in \mathcal{K}_+(t+1)$. As equality also holds in (3.12), this j is also a current constraining node of l. The result follows.

The biggest takeaways from this lemma are that $\Delta^+(t)$ cannot increase, and that a strict decrease eventuates from iteration t to t+1 unless a node with the largest overestimate at time t is both a current and a true constraining node of a node that inherits the largest overestimate at time t + 1. Thus the condition for lack of strict decrease for Δ^+ is very stringent. We next address Δ^- .

There are more subtle properties of (2.13) exposed by the proof. Referring back to the italicized statement at the beginning of this section, the correct comparison point of an overestimate $\Delta_l(t+1)$ is not $\Delta_l(t)$ but in fact the overestimate at t of one of its true constraining nodes j. In particular with $j \in C(l)$,

$$\Delta_l(t+1) \le \Delta_j(t), \tag{3.14}$$

i.e, this new overestimate cannot exceed the overestimates of the true constraining nodes of l.

Similarly, the following lemma shows the non-increasing property of $\Delta^{-}(t)$.

Lemma 3.2.3. Consider (2.13) under Assumption 1. Then with Δ^- defined in (3.5), for all t,

$$\Delta^{-}(t+1) \le \Delta^{-}(t). \tag{3.15}$$

With $\mathcal{K}_{-}(t)$ as in (3.9), unless $\Delta^{-}(t) = 0$, equality in (3.15) holds iff there exists $j \in \mathcal{K}_{-}(t)$ that is both a true and current constraining node of a member of $\mathcal{K}_{-}(t+1)$.

Proof. As $\Delta^{-}(t+1)$ is nonnegative (3.15) holds if $\Delta^{-}(t+1) = 0$. Thus assume $\Delta^{-}(t+1) > 0$. Consider any $l \in \mathcal{K}_{-}(t+1)$. Because of (2.13) there is a $j \in \mathcal{N}(l)$, such that

$$\hat{d}_l(t+1) = \hat{d}_j(t) + e_{lj} \tag{3.16}$$

Further, $\Delta^{-}(t)$ cannot increase as

$$\Delta^{-}(t+1) = -\Delta_{l}(t+1)$$

= $d_{l} - \hat{d}_{l}(t+1)$
= $d_{l} - \hat{d}_{j}(t) - e_{lj}$
 $\leq e_{lj} + d_{j} - \hat{d}_{j}(t) - e_{lj}$ (3.17)

$$= -\Delta_j(t) \tag{3.18}$$

$$\leq \Delta^{-}(t)$$
 (3.19)

where (3.17) comes from (2.11) and (3.19) follows from (3.5).

Suppose equality in (3.15) holds. Then for some $l \in \mathcal{K}_{-}(t+1)$ and a j satisfying (3.16), both (3.17) and (3.19) are equalities. From Definition 3.2, j is a current constraining node of l. From Definition 3.1 equality in (3.17) implies that j is also a true constraining node of l. From (3.9), equality in (3.19) implies that $j \in \mathcal{K}_{-}(t)$. Thus, as $l \in \mathcal{K}_{-}(t+1)$, $\Delta^{-}(t+1) = \Delta^{-}(t)$ only if there exists $j \in \mathcal{K}_{-}(t)$ that is a true constraining node of an $l \in \mathcal{K}_{-}(t+1)$.

On the other hand suppose for some $l \in \mathcal{K}_{-}(t+1)$, there is a $j \in \mathcal{K}_{-}(t)$ that is both a current and true constraining node of l. Then from Definition 3.2, (3.16) holds. Further $j \in \mathcal{K}_{-}(t)$ implies equality holds in (3.19). As j is a true constraining node of l equality also holds in (3.17), proving equality in (3.15).

Lemma 3.2.2 and Lemma 3.2.3 together show that for all t,

$$L(t+1) \le L(t),\tag{3.20}$$

validating the fact that L(t) is indeed a Lyapunov function. Moreover, equality in (3.20) holds under stringent conditions. In fact as shown below in Theorem 3.2.4, a strict decline in L(t) must occur every $\mathcal{D}(\mathcal{G})$ iterations, where $\mathcal{D}(\mathcal{G})$ is the effective diameter in Definition 3.1. Theorem 3.2.4 also provides the aesthetically appealing result that $\hat{d}_i = d_i$, for all $i \in V$ is the only stationary point of ABF.

Theorem 3.2.4. Under the conditions of Lemma 3.2.2 and Lemma 3.2.3, with $\mathcal{D}(\mathcal{G})$ as in Definition 3.1, L(t) as in (3.6), the following holds unless L(t) = 0:

$$L(t + \mathcal{D}(\mathcal{G}) - 1) < L(t) \ \forall t \ge t_0.$$

$$(3.21)$$

Further $\hat{d}_i = d_i$, $\forall i \in V$ is the only stationary point of (2.13).

Proof. Suppose L(t) > 0. From Lemma 3.2.2 and Lemma 3.2.3, (3.20) holds. Suppose now for some t and T and all $s \in \{1, \dots, T-1\}$, L(t+s) = L(t). Then from Lemma 3.2.2 and Lemma 3.2.3 there exists a sequence of nodes n_1, \dots, n_T , such that n_i is a true constraining node of n_{i+1} . From Lemma 3.2.1 this means $T \leq \mathcal{D}(\mathcal{G})$. In fact $T \leq \mathcal{D}(\mathcal{G}) - 1$. To establish a contradiction, suppose $T = \mathcal{D}(\mathcal{G})$. Then in the proofs of Lemma 3.2.2 and Lemma 3.2.3, $j = n_1 \in S$. Thus from (3.13) and (3.19), $\Delta^+(t) = \Delta^-(t) = L(t) = 0$. Thus unless L(t) = 0, $L(t + \mathcal{D}(\mathcal{G}) - 1) < L(t)$.

Suppose for all $i \in V$, $\hat{d}_i = d_i$. For $i \in S$, and all t, $\hat{d}_i(t) = 0 = d_i$, $\hat{d}_i(t+1) = 0 = d_i$, also holds. Now consider any $i \in V \setminus S$. Then from (3.1), there holds:

$$\hat{d}_i(t+1) = \min_{j \in \mathcal{N}(i)} \left\{ \hat{d}_j(t) + e_{ij} \right\}$$
$$= \min_{j \in \mathcal{N}(i)} \left\{ d_j + e_{ij} \right\}$$
$$= d_i.$$

Thus indeed $\hat{d}_i = d_i$, $\forall i \in V$ is a stationary point of (2.13). Now consider any other candidate stationary point $\hat{d}_i = d_i^*$, with

$$d^* = \left[d_1^*, \cdots, d_{|V|}^*\right] \neq \left[d_1, \cdots, d_{|V|}\right].$$
(3.22)

Suppose also at some t, and all $i \in V$, $\hat{d}_i = d_i^*$. Then L(t) > 0. From the first part of this theorem, (3.21) holds and

$$\left[\hat{d}_1(t+\mathcal{D}(\mathcal{G})),\cdots,\hat{d}_{|V|}(t+\mathcal{D}(\mathcal{G}))\right]\neq d^*.$$

Thus d^* cannot be a stationary point.

Of course without establishing a uniform bound from below on the extent of decline in (3.21), we cannot establish global uniform asymptotic stability. The next section does just that.

3.2.2 Global uniform asymptotic stability

This section establishes the global uniform asymptotic stability of ABF and tightly bound its convergence time. Recall that (3.6) has two components (the largest overestimate $\Delta^+(t)$ and the largest underestimate $\Delta^-(t)$), that the classical Bellman-Ford algorithm only copes with overestimates as it initializes to ensure (2.12), and that the motivation behind (2.13) is to permit underestimates.

It turns out that there is a fundamental disparity between the behaviors of under and overestimates in (2.13): Overestimates converge rapidly. Underestimates do not. Why this disparity? The key lies in (3.14). When $\Delta_l(t+1) > 0$ the j in (3.14) is a true constraining node of l, while if $\Delta_l(t+1) < 0$, it is a current constraining

node of l for the algorithm at time t. While true constraining nodes are fixed by the graph, *current constraining nodes may change*. Moreover, a pair of nodes may constrain each other at alternate instants, and should they share a short edge, their distance estimates rise slowly in tandem by small amounts. Dubbed in [14] as the *rising value problem*, this can lead to slow convergence.

By contrast, the following theorem shows that the overestimates all vanish to zero in at most $\mathcal{D}(\mathcal{G}) - 1$ steps, where $\mathcal{D}(\mathcal{G}) - 1$ is the effective diameter defined in Definition 3.1.

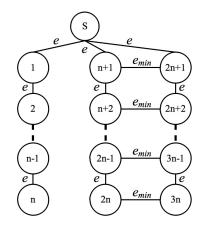


Figure 3.2: Illustration of the tightness of convergence time (figure from [58]). The subgraph comprising the nodes S and $\{1, \dots, n\}$ is used in the proof of Theorem 3.2.5. The entire graph is used in the proof of Theorem 3.2.9.

Theorem 3.2.5. Under Assumption 1, $\Delta^+(t)$ defined in (3.4) obeys

$$\Delta^{+}(t) = 0, \ \forall \ t \ge t_0 + \mathcal{D}(\mathcal{G}) - 1.$$
(3.23)

Further, for every n = |V| > 1 there is a $\mathcal{G} = (V, E)$, obeying Assumption 1 and a set of initial conditions such that $\Delta^+(t) > 0$, for all $t < t_0 + \mathcal{D}(\mathcal{G}) - 1$,

Proof. As G is connected, every node belongs to a sequence of nodes n_1, n_2, \dots, n_T , such that n_i is the true constraining node of n_{i+1} and $n_1 \in S$. From Lemma 3.2.1, $T \leq \mathcal{D}(\mathcal{G})$. We now assert and prove by induction that,

$$\Delta_{n_i}(t) \le 0, \ \forall \ t \ge i - 1 + t_0, \ \text{and} \ i \le T.$$
 (3.24)

Then the result is proved from (3.4). As $n_1 \in S$, (3.24) holds from (2.13). Now suppose it holds for some $i \in \{0, \dots, T-1\}$. As $n_i \in \mathcal{C}(n_{i+1}) \subset \mathcal{N}(n_{i+1})$, from (2.13), (3.1) and the induction hypothesis, for all $t \ge i + 1 + t_0$,

$$\hat{d}_{n_{i+1}}(t) \leq \hat{d}_{n_i}(t-1) + e_{n_{i+1}n_i}$$

 $\leq d_{n_i} + e_{n_{i+1}n_i}$
 $= d_{n_{i+1}}.$

Thus (3.24) and hence (3.23) is true.

For the second part of the theorem, we first observe that if j is both the true

and current constraining node of i. Then

$$\Delta_{i}(t+1) = \hat{d}_{i}(t+1) - d_{i}$$

= $\hat{d}_{i}(t+1) - d_{j} - e_{ij}$
= $\hat{d}_{j}(t) + e_{ij} - d_{j} - e_{ij}$
= $\Delta_{j}(t).$ (3.25)

In the subgraph, $\hat{\mathcal{G}}$ comprising the nodes S and $1, \dots, n$ in Figure 3.2, $\mathcal{D}(\hat{\mathcal{G}}) = n + 1$. Denote S = 0. Suppose for all $i \in \{1, \dots, n\}, 0 < \Delta_i(t_0) < e$. The result is proved if we show that for all $0 \leq j \leq t < i \in \{1, \dots, n\}$

$$0 < \Delta_i(t+t_0) < e, \ \Delta_j(t+t_0) = 0, \tag{3.26}$$

and i-1 is the current constraining node of i. Observe for all $k \in \{1, \dots, n\}$, $d_k = ke$, k-1 is the true constraining node of k, and for all $k \in \{1, \dots, n-1\}$ (3.26) implies that

$$\hat{d}_{k+1}(t+t_0) + e_{k+1,k} = (k+1)e + \Delta_{k+1}(t+t_0) + e \\
\geq (k+2)e \\
= d_{k-1} + 3e \\
> d_{k-1} + \Delta_{k-1}(t+t_0) + 2e \\
> \hat{d}_{k-1}(t+t_0) + e_{k-1,k},$$

i.e. k-1 is the current constraining node of k. As n-1 is the only neighbor of n, this is also true for k = n. Use induction to prove (3.26), which holds for t = 0. Suppose it holds for $1 \le t < m < n$. As $0 < \Delta_i(m + t_0) < e$ for all $i \in \{m + 1, \dots, n\}$, from (3.25), the inequality in (3.26) holds. As $\Delta_{m-1}(t+m-1) = 0$, and m-1 is the current constraining node of m, so does the equality.

The proof of (3.23) implicitly expands on (3.14). Specifically, one can view n_i in the proof as being a node that is effectively $n_i - 1$ hops away from the source set. At the *i*-th instant suppose l in (3.14) is a node that is (i + 1) effective hops away. Then for this node the actual error is forced to be nonpositive, as j its true constraining node, one effective hop closer to the source set, acquires a nonpositive estimation error an iteration earlier. As we show below underestimates lack this property.

Depending on the initial distance estimates and the graph topology, $\Delta^+(t)$ may converge to zero in fewer than $\mathcal{D}(\mathcal{G}) - 1$ rounds. This may happen for example, if n_i acquires a negative error after i - 2 iterations, then convergence of $\Delta^+(t)$ to zero occurs an iteration sooner. Nonetheless the upper bound on the time for Δ^+ to converge is tight in the sense that for all $|V| \ge 2$, there is a graph and an initialization where convergence cannot occur before $\mathcal{D}(\mathcal{G}) - 1$ iterations. The proposition below shows more: If no initial estimate is an underestimate, then convergence occurs in at most $\mathcal{D}(\mathcal{G}) - 1$ iterations. The proof follows from Lemma 3.2.3 and Theorem 3.2.5.

Proposition 1. Under Assumption 1, suppose $\Delta^{-}(0) = 0$. Then the Lyapunov function L(t) defined in (3.6)

$$L(t) = 0, \ \forall t \ge \mathcal{D}(\mathcal{G}) - 1,$$

where $\mathcal{D}(\mathcal{G})$ is defined in Definition 3.1.

This underscores why the classical Bellman-Ford algorithm requires that $\Delta^{-}(0) = 0$.

The next theorem quantifies the rising value problem that causes a slower decline in $\Delta^{-}(t)$. Indeed, the classical Bellman-Ford algorithm, which assumes (2.12) will also yield the same convergence time. But of course it cannot cope with initial underestimates.

Theorem 3.2.6. Under the conditions of Lemma 3.2.3, consider a pair of nodes i and j such that $\Delta_i(t_0) < 0$ and

$$j = \arg\min_{j \in \mathcal{N}(i)} \{e_{ij}\}.$$

Then $\Delta_i(T) \geq 0$ implies

$$T \ge t_0 - \frac{\Delta_i(t_0)}{e_{ij}}.$$

Proof. From (2.13) for any t

$$\hat{d}_j(t+1) \leq \hat{d}_i(t) + e_{ij}.$$
 (3.27)

Likewise, as $j \in \mathcal{N}(i)$, using (3.27) the result follows as,

$$\Delta_i(t+2) = \hat{d}_i(t+2) - d_i$$

$$\leq \hat{d}_j(t+1) + e_{ij} - d_i$$

$$\leq \hat{d}_i(t) + 2e_{ij} - d_i$$

$$= \Delta_i(t) + 2e_{ij}.$$

Thus the rising value problem may occur even if a pair of nodes with underestimated distance estimates do not constrain each other. Rather the rise is limited by the smallest edge length impinging on a node with the largest underestimate.

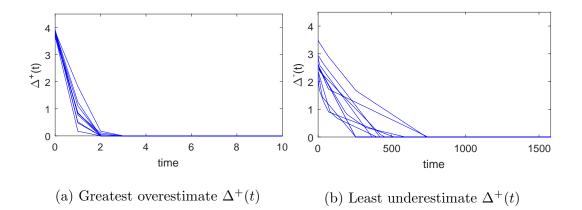


Figure 3.3: Plot from [58] of (a) greatest overestimate $\Delta^+(t)$ and (b) least underestimate $\Delta^-(t)$ for 10 runs of 500 nodes randomly distributed in a 4x1 sq. km area, showing that overestimates correct much faster than underestimates.

This behavior is shown in Figure 3.3, where 500 nodes, including a solitary source, are uniformly distributed in a $4 \times 1 \text{ km}^2$ field. Each has a communication range of 0.25 kilometers, i.e, the average size of $\mathcal{N}(i)$ is 20. The initial distance estimates of non-source nodes are chosen randomly in $\mathcal{U}(0, 4.12)$ kilometers. Figure 3.3 shows the results of 10 simulations, each run synchronously for 2000 seconds with 1 second per round. The results are consistent with our analysis: $\Delta^+(t)$ decreases rapidly to zero within at most 4 rounds, even though $\mathcal{D}(\mathcal{G}) = 14$. On the other hand, $\Delta^-(t)$, limited by close pairs of nodes, has much slower convergence. Observe that the classical Bellman-Ford algorithm would have yielded the same type of trace as in Figure 3.3(a), but cannot deal with the violation of (2.12).

The next lemma helps lower bound the decline in $\Delta^{-}(t)$.

Lemma 3.2.7. Under the conditions of lemmas 3.2.2 and 3.2.3 define,

$$S^{-}(t) = \{ i \in V | \Delta_i(t) < 0 \}, \qquad (3.28)$$

and

$$\hat{d}_{\min}(t) = \min_{i \in \mathcal{S}^{-}(t)} \{ \hat{d}_i(t) \}.$$
(3.29)

Then with e_{\min} defined in (2.8), the following holds unless $S^{-}(t+1)$ is empty:

$$\hat{d}_{\min}(t+1) \ge \hat{d}_{\min}(t) + e_{\min}, \ \forall t \ge t_0.$$
 (3.30)

Proof. Suppose $S^{-}(t+1)$ is not empty. Then from Lemma 3.2.3, and (3.9), $S^{-}(t)$ cannot be empty. Consider any $i \in S^{-}(t+1)$ and suppose j is its current constraining node at t. Then we assert that $j \in S^{-}(t)$. Indeed assume $j \notin S^{-}(t)$. Thus $\hat{d}_{j}(t) \geq d_{j}$. As $j \in \mathcal{N}(i)$, from Definition 3.2 and (2.11),

$$\hat{d}_i(t+1) = \hat{d}_j(t) + e_{ij}$$

$$\geq d_j + e_{ij}$$

$$\geq d_i.$$

Thus $i \notin S^{-}(t+1)$, establishing a contradiction. Hence $j \in S^{-}(t)$. Then from (3.28) and (3.29), (3.30) holds as $\forall i \in S^{-}(t+1)$,

$$\hat{d}_i(t+1) = \hat{d}_j(t) + e_{ij}$$
$$\geq \hat{d}_{\min}(t) + e_{\min}.$$

We now prove that the decline in (3.21) is uniformly lower bounded, proving global uniform asymptotic stability.

Theorem 3.2.8. Under conditions of Theorem 3.2.4, there exists $\alpha > 0$, dependent on the initial conditions but not t_0 , such that

$$0 \le L(t + \mathcal{D}(\mathcal{G}) - 1) \le \max[L(t) - \alpha, 0], \ \forall \ t \ge t_0.$$

$$(3.31)$$

Further the unique stationary point of (2.13), $\Delta_i = 0$ for all $i \in V$, is globally uniformly asymptotically stable.

Proof. As the postulated α in (3.31) is fixed by the initial conditions and is independent of the initial time t_0 , (3.31) proves global uniform asymptotic stability. From Theorem 3.2.5, the set of $\Delta^+(t)$ is a finite countable set that includes zero. Similarly, because of Lemma 3.2.3 and (3.30) of Lemma 3.2.7, so is the set of $\Delta^-(t)$, as \hat{d}_{\min} in (3.29), must equal the true distance it estimates in a finite time. Thus \mathcal{L} the set of L(t), for all $t \geq t_0$ is a finite countable set. Then α , the smallest difference between the elements of this set, is positive unless $L(t_0) = 0$, and while dependent on the initial conditions, is independent of t_0 . From Theorem 3.2.4 a decrease in L(t) occurs every $\mathcal{D}(\mathcal{G})$ iterations. As $L(t) \in \mathcal{L}$, any change must be by at least α .

We now tightly bound the time to convergence.

Theorem 3.2.9. Consider (2.13) under the conditions of Theorem 3.2.8, $\mathcal{D}(\mathcal{G})$ as in Definition 3.1 and e_{\min} in (2.8). Define

$$d_{\max}(\mathcal{G}) = \max_{i \in V} \{d_i\},\tag{3.32}$$

for $\mathcal{G} = (V, E)$. Then $L(t) = 0, \forall t \ge t_0 + T$, where,

$$T = \max\left\{ \mathcal{D}(\mathcal{G}) - 1, \left\lceil \frac{d_{\max} - \hat{d}_{\min}(t_0)}{e_{\min}} \right\rceil \right\}.$$
 (3.33)

Further for every n = |V| > 3, there exists a \mathcal{G} satisfying Assumption 1 for which L(t) > 0 for all t < T.

Proof. From Theorem 3.2.5, $\Delta^+(t_0 + \mathcal{D}(\mathcal{G}) - 1) = 0$, accounting for $\mathcal{D}(\mathcal{G})$ in (3.33). From Lemma 3.2.7, and any $i \in \mathcal{S}^-(t)$ in (3.28), one obtains for any $t \ge t_0$,

$$\begin{aligned} \Delta^{-}(t) &\leq d_{i} - \hat{d}_{i}(t) \\ &\leq d_{\max} - \hat{d}_{\min}(t) \\ &\leq d_{\max} - \hat{d}_{\min}(t_{0}) - (t - t_{0})e_{\min} \end{aligned}$$

Thus $\Delta^{-}(t) = 0$, whenever

$$t - t_0 \ge T_- = \left[\frac{d_{\max}(\mathcal{G}) - \hat{d}_{\min}(t_0)}{e_{\min}}\right].$$
 (3.34)

To prove that convergence time can be as much as T, consider the graph in Figure 3.2 with $e_{\min} < e$. Assume for all $i \in \{1, \dots, n\}$, $\hat{d}_i(t_0) > d_i$, and $\hat{d}_{n+i}(t_0) = \hat{d}_{2n+i}(t_0) = 0$. Then for all $i \in \{1, \dots, n\}$, and all t, $\Delta_i(t) \ge 0$, and $\Delta_{n+i}(t) = \Delta_{2n+i}(t) \le 0$. From the proof of Theorem 3.2.5, it takes exactly $\mathcal{D}(\mathcal{G}) - 1$ iterations for Δ^+ to converge.

Now consider $\Delta^{-}(t)$. For simplicity assume $t_0 = 0$. Suppose

$$\left\lceil \frac{e}{e_{min}} \right\rceil = m. \tag{3.35}$$

Then we assert that for all $i \in \{1, \cdots, n\}$,

$$\hat{d}_{n+i}(t) = \hat{d}_{2n+i}(t) = \begin{cases} te_{\min} & i \in \left\{ \left\lfloor \frac{t}{m} \right\rfloor, \cdots, n \right\} \\ d_{n+i} & \text{otherwise} \end{cases}$$
(3.36)

Call each pair 2n + i and n + i partners. When $t \in \{1, \dots, m-1\}$, as $e > e_{\min}$, and $\hat{d}_{2n+i}(0) = \hat{d}_{n+i}(0) = 0$, these nodes constrain each other and their respective distance estimates rise by increments of m. Thus all $\hat{d}_{2n+i}(m) = \hat{d}_{n+i}(m) = e$. Thus $\hat{d}_{2n+1}(m) = \hat{d}_{n+1}(m) = d_{2n+1} = d_{n+1} = e$, and at t = m the nodes n + 1 and n + 2are constrained by S, while the remaining are constrained by their partners. Since distance estimates cannot fall in value, for all $t \ge m$, n + 1 and n + 2 are constrained by S, and $\hat{d}_{2n+1}(t) = \hat{d}_{n+1}(t) = d_{2n+1}(t) = d_{n+1}(t) = e$, for all $t \ge m$. Continuing this argument (3.36) is readily proved. Thus,

$$\hat{d}_{3n-1}(t) = \hat{d}_{2n-1}(t) < ne = d_{2n-1} = d_{3n-1}, \ \forall t < mn.$$
 (3.37)

Result follows from (3.34) as $\hat{d}_{\min}(0) = 0$, $d_{\max}(\mathcal{G}) = ne$ and

$$mn = n \left[\frac{e}{e_{min}} \right] = \left[\frac{d_{max}(\mathcal{G}) - \hat{d}_{min}(0)}{e_{min}} \right].$$

Thus the bound on convergence time is tight, though the worst case nature of the analysis also makes it conservative.

3.2.3 Robustness under perturbations

We now turn to the robustness of ABF to possibly *persistent perturbations in non-source nodes* with the goal of demonstrating *ultimate boundedness* (per the definition in [45]) of distance estimates around nominal distance values. We consider the behavior of ABF in a framework with two physical interpretations. 1) Nodes experience bounded, potentially perpetual motion around nominal locations. In aggregate

computing this captures an incremental version of the common scenario of mobile computing devices or of imprecise localization. 2) A node receives noisy distance estimates of its neighbors.

Consider first the case where each *non-source* node moves around a nominal position, i.e., edge lengths change from their nominal values e_{ij} as

$$\bar{e}_{ij}(t) = e_{ij} + \epsilon_{ij}(t). \tag{3.38}$$

Mobility is assumed to be both bounded and small, i.e., there exists an ϵ such that with e_{\min} defined in (2.8),

$$|\epsilon_{ij}(t)| < \epsilon < e_{\min}.\tag{3.39}$$

This ensures that no edge length is ever negative. Based on this assumption of bounded mobility, we also assume that the set of neighbors $\mathcal{N}(i)$ of each node *i* does not change.

This also accommodates the setting where noisy estimates of $\hat{d}_i(t)$ available to its neighbors, with $\epsilon_{ij}(t)$ modeling the noise. Unlike the setting of mobility in this case we cannot assume that the noise is symmetric, and permit

$$\bar{e}_{ij}(t) \neq \bar{e}_{ji}(t). \tag{3.40}$$

In particular $\bar{e}_{ij}(t)$ is the noisy edge length seen by node *i* as opposed to node *j*. Thus (2.13) must be interpreted as:

$$\hat{d}_i(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \left\{ \hat{d}_j(t) + \bar{e}_{ij}(t) \right\} & i \notin S \\ 0 & i \in S \end{cases}$$
(3.41)

$$\hat{d}_j(t) + \bar{e}_{ij}(t) = \hat{d}_j(t) + e_{ij} + \epsilon_{ij}(t)$$
$$= (\hat{d}_j(t) + \epsilon_{ij}(t)) + e_{ij},$$

this captures the execution of ABF with noisy measurements of $\hat{d}_j(t)$. The problem formulation is otherwise unchanged. The nominal graph is still $\mathcal{G} = (V, E)$, undirected in that $i \in \mathcal{N}(i)$ implies $j \in \mathcal{N}(j)$ and $e_{ij} = e_{ji}$, and the goal is to study the perturbations of $\hat{d}_i(t)$ from the nominal distances d_i . We now define a shrunken version of \mathcal{G} , which represents the graph with the shortest links permitted by our perturbation model. We will show that the distance estimates provided by ABF when applied to this graph lower bounds all $\hat{d}_i(t)$. This will help provide the lower ultimate bounds and the time to attain them.

Definition 3.3. Given $\mathcal{G} = (V, E)$, the undirected graph $\mathcal{G}^- = (V, E^-)$, has the property that edge $(i, j) \in E^-$ iff $(i, j) \in E$. The edge length e_{ij}^- between nodes i and j obeys

$$e_{ij}^- = e_{ij} - \epsilon, \tag{3.42}$$

with ϵ defined in (3.39). Further \mathcal{G}^- has the same source set S as \mathcal{G} , each i has the same set of neighbors as in \mathcal{G} , and D_i is the shortest distance between i and the nearest source node, i.e. plays the role of d_i in \mathcal{G} .

Our goal is to prove the ultimate boundedness of the $\hat{d}_i(t)$ provided by (3.41) by examining their difference with the nominal distances d_i , through L(t), which

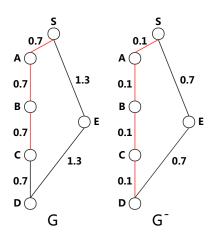


Figure 3.4: Examples of effective diameter (figure from [58]), showing edge length as labels on edges. In the left graph \mathcal{G} , $\mathcal{D}(\mathcal{G}) = 4$ and comes from the sequence S, A, Band C, where each node is the true constraining node of its successor. On the other hand in the right graph \mathcal{G}^- , $\mathcal{D}(\mathcal{G}^-) = 5$ and comes from the sequence S, A, B, C and D.

retains its definition in (3.6). Note $\Delta_i(t) = \hat{d}_i(t) - d_i$ where d_i are the distances in \mathcal{G} . We summarize the underlying assumptions.

Assumption 2. Both the graphs \mathcal{G} and \mathcal{G}^- defined in Definition 3.3 obey Assumption 1. The set of neighbors of node *i*, is time invariant for all $i \in V$ as is the source set. None of the source nodes move. The edge length of each pair of nodes *i* and *j* is given by (3.38) under (3.39) and (2.8). Although (3.40) applies, $e_{ij} = e_{ji}$ still holds. Also assume that $t_0 = 0$.

Note $t_0 = 0$ is without loss of generality because of the fact that its uniform

asymptotic stability, guarantees that the behavior of (2.13) is independent of t_0 . Despite the topological similarity of the graphs \mathcal{G}^- and \mathcal{G} , their respective effective diameters $\mathcal{D}(\mathcal{G}^-)$ and $\mathcal{D}(\mathcal{G})$, defined in Definition 3.1, may differ. This is so as the true constraining nodes in the graphs may be different, as is illustrated by the two graphs in Figure 3.4. In \mathcal{G} the true constraining node of D is E while in \mathcal{G}^- , it is C. Further $\mathcal{D}(\mathcal{G}) = 4 < \mathcal{D}(\mathcal{G}^-) = 5$. Lemma 3.2.12 shows that

$$\mathcal{D}(\mathcal{G}^{-}) \ge \mathcal{D}(\mathcal{G}). \tag{3.43}$$

The next lemma proves the ultimate boundedness of $\Delta^+(t)$.

Lemma 3.2.10. Consider (3.41), under Assumption 2. Then $\Delta^+(t) \leq (\mathcal{D}(G) - 1)\epsilon$ for all $t \geq \mathcal{D}(\mathcal{G}) - 1$, where $\mathcal{D}(\mathcal{G})$ is as in Definition 3.1 and $\Delta^+(t)$ is as in (3.4).

Proof. Consider the sequence $n_1, n_2, ..., n_T$ in the proof of Theorem 3.2.5 where the true constraining nodes are for the graph \mathcal{G} . As $T \leq \mathcal{D}(\mathcal{G})$, the result holds if

$$\hat{d}_{n_i}(t) \le d_{n_i} + (i-1)\epsilon, \ \forall \ i \in \{1, \cdots, T\} \text{ and } t \ge i-1.$$
 (3.44)

We prove (3.44) by induction. It is true for i = 1 from (3.41) as $n_1 \in S$. Thus suppose it holds for some $i \in \{1, \dots, T-1\}$. Then from (3.41), (3.38), (3.39), and the definition of a true constraining node we have for all $t \geq i-1$

$$\begin{aligned} \hat{d}_{n_{i+1}}(t+1) &\leq \hat{d}_{n_i}(t) + \bar{e}_{n_{i+1}n_i}(t) \\ &\leq d_{n_i} + (i-1)\epsilon + e_{n_{i+1}n_i} + \epsilon_{n_{i+1}n_i}(t) \\ &= d_{n_{i+1}} + (i-1)\epsilon + \epsilon_{n_{i+1}n_i}(t) \\ &\leq d_{n_{i+1}} + (i-1)\epsilon + \epsilon \\ &= d_{n_{i+1}} + i\epsilon. \end{aligned}$$

Thus (3.44) and hence the result follows.

To address $\Delta^{-}(t)$ we take an approach like the *comparison principle* [45], the following lemma establishes a connection between distance estimates in \mathcal{G} and those in its shrunken version, \mathcal{G}^{-} , defined in Definition 3.3.

Lemma 3.2.11. Suppose Assumption 2 holds. Consider

$$\hat{D}_i(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \left\{ \hat{D}_j(t) + e_{ij} - \epsilon \right\} & i \notin S \\ 0 & i \in S \end{cases},$$
(3.45)

and (3.41). Suppose for all $i \in V$, $\hat{D}_i(0) \ge \hat{d}_i(0)$. Then $\hat{d}_i(t) \ge \hat{D}_i(t)$, $\forall t \ge 0$ and for all $i \in V$.

Proof. We prove by induction. As the result holds for t = 0, suppose for some $t \ge 0$, $\hat{d}_i(t) \ge \hat{D}_i(t), \ \forall i \in V$. Suppose $j \in \mathcal{N}(i)$ is a current constraining node of i at time tin (3.41) while $k \in \mathcal{N}(i)$ is a current constraining node of i at time t in (3.45). Then

from (3.38) and (3.39), there obtains:

$$\hat{D}_{i}(t+1) = \hat{D}_{k}(t) + e_{ik} - \epsilon$$

$$\leq \hat{D}_{j}(t) + e_{ij} - \epsilon$$

$$\leq \hat{d}_{j}(t) + e_{ij} - \epsilon$$

$$\leq \hat{d}_{j}(t) + e_{ij} + \epsilon_{ij}(t)$$

$$= \hat{d}_{i}(t+1).$$
(3.46)

Thus the estimates offered by (3.41) are uniformly lower bounded by the estimates, $\hat{D}_i(t)$, from the identically initialized Adaptive Bellman-Ford algorithm applied to the graph \mathcal{G}^- . As \mathcal{G}^- satisfies the same assumptions as \mathcal{G} , and is perturbation free, the greatest underestimation error it offers converges to zero. To establish an ultimate bound on Δ^- it thus suffices to relate distances D_i in \mathcal{G}^- , to distances d_i in \mathcal{G} . The next lemma does just that.

Lemma 3.2.12. Suppose Assumption 2 holds. Then for all $i \in V$,

$$d_i \le D_i + (\mathcal{D}(\mathcal{G}^-) - 1)\epsilon, \tag{3.47}$$

where \mathcal{G}^- is in Definition 3.3 and $\mathcal{D}(.)$ is in Definition 3.1. Further, (3.43) holds.

Proof. Consider nodes n_1, \dots, n_{T_1} , such that $n_1 \in S$, and for all $i \in \{1, \dots, T_1 - 1\}$ n_i is a true constraining node of n_{i+1} in \mathcal{G}^- . Every node is in one such sequence. We assert that for all $i \in \{1, \dots, T_1\}$,

$$d_{n_i} \le D_{n_i} + (i-1)\epsilon. \tag{3.48}$$

$$d_{n_{i+1}} \leq d_{n_i} + e_{n_i n_{i+1}}$$

$$\leq D_{n_i} + (i-1)\epsilon + e_{n_i n_{i+1}}$$

$$= D_{n_i} + (i-1)\epsilon + e_{n_i n_{i+1}}^- + \epsilon$$

$$= D_{n_{i+1}} + i\epsilon.$$

Then (3.47) follows as from Lemma 3.2.1, $T_1 \leq \mathcal{D}(\mathcal{G}^-)$.

To prove (3.43) suppose $\mathcal{D}(\mathcal{G}) > \mathcal{D}(\mathcal{G}^-)$. Then there is a sequence of nodes in \mathcal{G} , without loss of generality $\{1, \dots, n\}$, such that $1 \in S$ and for all $i \in \{1, \dots, n\}$, i is a true constraining node of i + 1. Further

$$d_n = \sum_{i=1}^{n-1} e_{i,i+1}.$$
(3.49)

As $\mathcal{D}(\mathcal{G}) > \mathcal{D}(\mathcal{G}^-)$, there is a sequence of nodes $\{p_1, \dots, p_l, n\}$ such that in \mathcal{G}^- , $p_1 \in S$ and p_i is a true constraining node of p_{i+1} , p_l is a true constraining node of nand l+1 < n. Further

$$D_n = \sum_{i=1}^{l-1} \left(e_{p_i, p_{i+1}} - \epsilon \right) + e_{p_l, n} - \epsilon.$$
(3.50)

By definition the distance from S along the path comprising $\{1, \dots, n\}$ in \mathcal{G}^- is less than D_n , i.e.

$$\sum_{i=1}^{l-1} \left(e_{p_i, p_{i+1}} - \epsilon \right) + e_{p_l, n} - \epsilon < \sum_{i=1}^{n-1} \left(e_{i, i+1} - \epsilon \right)$$
$$\Rightarrow \sum_{i=1}^{l-1} e_{p_i, p_{i+1}} + e_{p_l, n} < \sum_{i=1}^{n-1} e_{i, i+1},$$

where the last inequality follows from the fact that l + 1 < n, and violates (3.49). The contradiction proves (3.43).

We now prove the ultimate boundedness of Δ^- .

Lemma 3.2.13. Consider (3.41) under Assumption 2. Then with $\mathcal{D}(\mathcal{G}^-)$, e_{\min} , $d_{\max}(\mathcal{G}^-)$ and \hat{d}_{\min} defined in Definition 3.1, (2.8), Theorem 3.2.9 and Lemma 3.2.7 respectively, for all

$$t \ge T = \frac{d_{\max}(\mathcal{G}^-) - \hat{d}_{\min}(0)}{e_{\min} - \epsilon},\tag{3.51}$$

$$\Delta^{-}(t) \le (\mathcal{D}(\mathcal{G}^{-}) - 1)\epsilon.$$
(3.52)

Proof. Consider the algorithm in (3.45) with the initialization in Lemma 3.2.11. As \mathcal{G}^- satisfies Assumption 1, $\hat{D}_i(t)$ converges to D_i (see Definition 3.3) in a finite time T_1 . Thus from Lemma 3.2.11, for all $t \geq T_1$ and $i \in V$, $\hat{d}_i(t) \geq D_i$. Thus the upper bound in (3.52) follows from Lemma 3.2.12 and

$$\begin{aligned} -\Delta_i(t) &\leq d_i - D_i \\ &\leq D_i + (\mathcal{D}(\mathcal{G}^-) - 1)\epsilon - D_i \\ &= (\mathcal{D}(\mathcal{G}^-) - 1)\epsilon, \end{aligned}$$

Time to attain (3.52) is that for the greatest underestimate in (3.45) to go to zero. This is at most T in (3.51) from the following: (i) Theorem 3.2.9. (ii) The minimum initial estimate in (3.45) is $\hat{d}_{\min}(0)$. (iii) The shortest link in \mathcal{G}^- is at least $e_{\min} - \epsilon$. \Box

Note $d_{max}(\mathcal{G}^-)$ can range from $d_{max}(\mathcal{G}) - \epsilon$, e.g. when the node with the largest distance has a source as its true constraining node, to $d_{max}(\mathcal{G}) - \mathcal{D}(\mathcal{G}^-)\epsilon$.

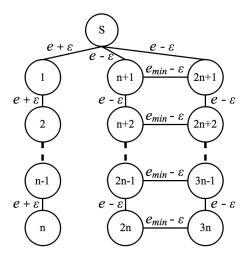


Figure 3.5: Illustration of the tightness of convergence time with perturbations (figure from [58]).

Theorem 3.2.14. Consider (3.41), the conditions of Lemma 3.2.13 and Lemma 3.2.10. With T as in (3.54), L(t) in (3.6) obeys

$$L(t) \le \left(\mathcal{D}(\mathcal{G}) + \mathcal{D}(\mathcal{G}^{-}) - 2\right)\epsilon, \ \forall t \ge T.$$
(3.53)

$$T = \max\left\{ \mathcal{D}(\mathcal{G}) - 1, \frac{d_{\max}(\mathcal{G}^-) - \hat{d}_{\min}(0)}{e_{\min} - \epsilon} \right\}.$$
(3.54)

Further, for every n = |V| > 3, there exists a graph and perturbations conforming to Assumption 2 for which the bound in (3.53) is attained in precisely T iterations.

Proof. Ultimate boundedness follows from Lemma 3.2.13 and Lemma 3.2.10. Consider a perturbed version of the nominal graph \mathcal{G} in Figure 3.2 given in Figure 3.5. The perturbation obeys Assumption 2 and the perturbed graph itself obeys Assumption 1. Since the perturbed graph has fixed edges its distance estimates will converge

to the correct values. In particular \hat{d}_n will converge to $d_n + n\epsilon$ and \hat{d}_{2n} to $d_{2n} - n\epsilon$. It is readily checked that these are in fact the greatest over and underestimates respectively. Thus, as $\mathcal{D}(\mathcal{G}) = \mathcal{D}(\mathcal{G}^-) = n + 1$, (3.53) is precisely met. Suppose now $\hat{d}_i(0) > i(e + \epsilon)$ for all $i \in \{1, \dots, n\}$, and $\hat{d}_i(0) = 0$ for all $i \in \{n + 1, \dots, 3n\}$, then the arguments given in the proof of Theorem 3.2.9 establish that convergence of \hat{d}_n and \hat{d}_{2n} occur precisely by the first and second terms on the right side of (3.54). \Box

Though *tight*, these bounds are conservative as can be seen from the simulations results in Section 3.2.4.

3.2.4 Simulations

In this section, we empirically confirm the results presented in the prior sections through simulations under two classes of persistent perturbations: device movement and *periodic changes of source location to induce large perturbations*. Unless otherwise noted, all use 500 nodes, one of which is a source, distributed randomly in a 4x1 km² field, communicating over a 0.25 km radius, and run synchronously for 2000 simulated 1-second rounds, with $\hat{d}_i(0) \in \mathcal{U}(0, 4.12)$ km. As under perturbations, (2.12) will not be sustained, the classical Bellman-Ford algorithm cannot cope with them.

We first investigate the performance of ABF under device movement. At each t a node is perturbed from its nominal location by $[r \cos \theta, r \sin \theta]^T$ with $r \sim \mathcal{U}(0, 0.5e_{min}), e_{min} = 0.0048$, and $\theta \sim \mathcal{U}(0, 2\pi)$. Thus $\epsilon = e_{\min}$ in (3.38). Note (3.38) and (3.40) are satisfied with probability 1. The results are in Figure 3.6.

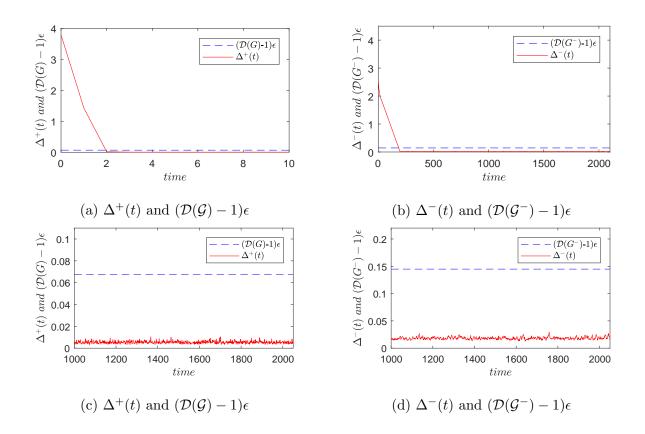


Figure 3.6: Comparison between (a) the greatest overestimate $\Delta^+(t)$ and $(\mathcal{D}(\mathcal{G})-1)\epsilon$, and (b) comparison between the least underestimate $\Delta^-(t)$ and $(\mathcal{D}(\mathcal{G}^-)-1)\epsilon$ (b), as well as their partial enlarged views (c) and (d). In this example, each node is moving within a circle with a radius of $0.5e_{\min}$ (plot from [58]).

Lemma 3.2.10 and Lemma 3.2.13, predict ultimate bounds for Δ^+ and Δ^- of $(\mathcal{D}(\mathcal{G}) - 1)\epsilon$ and $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$, respectively. In Figure 3.6(a) $\Delta^+(t)$ goes lower than $(\mathcal{D}(\mathcal{G}) - 1)\epsilon$ after 3 rounds, $\mathcal{D}(\mathcal{G}) = 17$. On the other hand, Figure 3.6(b) shows that $\Delta^-(t)$ is still constrained by the "rising value problem" and needs a much longer time than $\Delta^+(t)$ to drop below $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$.

Figure 3.6(c) and (d) depict snapshots well beyond the time after the ultimate bounds are attained. Unsurprisingly, due to the worst case nature of the Lyapunov based analysis, which is inherently conservative, there is a significant gap between $\Delta^+(t)$, $\Delta^-(t)$ and their corresponding ultimate upper bounds.

We next simulate large errors by periodic change of the source set in addition to device movements. The graph periodically alternates between two sources nodes at [3.75, 0.5] and [0.25, 0.5]. The remaining 498 nodes are randomly distributed in a 4x1 km field. At each transition the old source inherits a *distance estimate of zero* and other nodes acquire a large estimation error. The non-source nodes still move around their nominal values, in a disk with radius in $\mathcal{U}(0, 0.5e_{\min})$, $e_{min} = 0.0044$ and $\epsilon = e_{\min}$. To mimic frequent changes, we alternate the sources every 50 seconds while the simulation runs for 2000 iterations. The results in Figure 3.7 show that due to the fast convergence rate, $\Delta^+(t)$ will drop below its ultimate upper bound, then bounce back up again when the source changes. The two different "spike" patterns of Δ^+ result from the alternation of the two source nodes. On the other hand, due to its slow convergence rate, $\Delta^-(t)$ in this case never attains its ultimate bound. Yet, even if errors are not small due to the frequent change of source nodes, the algorithm

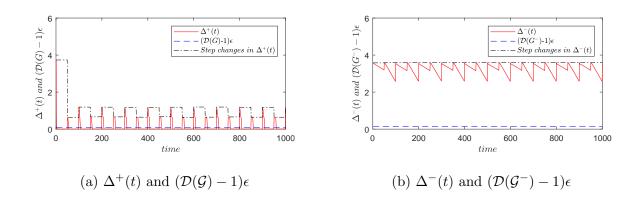


Figure 3.7: Comparison between (a) the greatest overestimate $\Delta^+(t)$, $(\mathcal{D}(\mathcal{G})-1)\epsilon$ and the theoretical bound of $\Delta^+(t)$ for unperturbed system, and (b) comparison between the least underestimate $\Delta^-(t)$, $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$ and the theoretical bound of $\Delta^-(t)$ for unperturbed system. In this example, there are two source nodes in the graph, one is at [3.75, 0.5] and the other is at [0.25, 0.5], switching between source nodes every 50 simulated seconds while all nodes are moving within a circle with a radius of $0.5e_{\min}$ (plot from [58]).

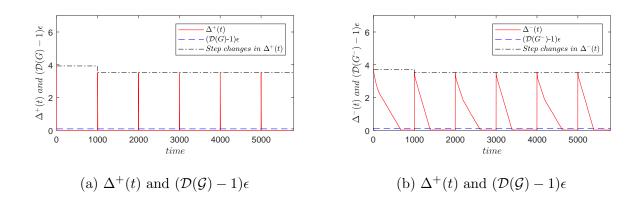


Figure 3.8: Comparison between (a) the greatest overestimate $\Delta^+(t)$, $(\mathcal{D}(\mathcal{G})-1)\epsilon$ and the theoretical bound of $\Delta^+(t)$ for unperturbed system, and (b) comparison between the least underestimate $\Delta^-(t)$, $(\mathcal{D}(\mathcal{G}^-) - 1)\epsilon$ and the theoretical bound of $\Delta^-(t)$ for unperturbed system. In this example, there are two source nodes in the graph, one is at [3.75, 0.5] and the other is at [0.25, 0.5], switching between source nodes every 1000 simulated seconds while all nodes are moving within a circle with a radius of $0.5e_{\min}$ (plot from [58]).

reduces them from the large values they acquire at source transitions. Under less frequent changes $\Delta^{-}(t)$ converges as well, e.g. in Figure 3.8, where the sources switch every 1000 seconds.

3.3 GUES and robustness of MPP

In this section, we prove the robust stability of MPP introduced in Section 2.5, which is a variant of the shortest path problem that is used to find a path between two nodes in a graph such that the multiplication of edge weights is maximized. Similar problems have been studied in [44, 21, 64]. However, these works either involve a computationally expensive operation log or lacks the proof of the time to convergence and robustness under perturbations. In this section, GUES of MPP is proved by using the same Lyapunov function as in ABF. Our analysis shows that MPP is also ultimately bounded under certain persistent perturbations , and the time to convergence, the ultimate bounds, as well as the time to attain those bounds are tight [28]. The following assumption is needed for our proof.

Assumption 3. The graph $\mathcal{G} = (V, E)$ is connected, undirected, $S \neq V$, with e_{ij} obeying (2.19) and the true failure rate of delivery p_i of node *i* from the source set obeying (2.21). Further, the initial time is t_0 , and $\forall i \in V \setminus S, \hat{p}_i(t_0) \in [0, 1)$.

With Assumption 3, the following holds.

Remark. Suppose Assumption 3 holds. Then for all $i \in V$ and all $t, 0 \leq \hat{p}_i(t) < 1$.

3.3.1 A Lyapunov function

Consider (2.20), the constraining node is defined as the following.

Definition 3.4. The minimizing j in the first bullet of (2.20) used to find $\hat{p}_i(t+1)$, is *i*'s current constraining node at t + 1.

Similarly, for p_i the true failure rate of delivery defined in (2.21), the concept of true constraining node is introduced.

Definition 3.5. A minimizing j in the first bullet of (2.21) used to find p_i , is i's true constraining node. As a node may have multiple true constraining nodes, the set of true constraining nodes of a node $i \in V \setminus S$ is defined as C(i).

Define $\Delta_i(t)$ as the probability estimation error of node *i*, and it obeys

$$\Delta_i(t) = \hat{p}_i(t) - p_i. \tag{3.55}$$

Further, define the greatest overestimate of the error $\Delta^+(t)$ and the least underestimate of the error $\Delta^-(t)$ below,

$$\Delta^{+}(t) = \max\left[0, \max_{i} \Delta_{i}(t)\right]$$
(3.56)

$$\Delta^{-}(t) = \max\left[0, -\min_{i} \Delta_{i}(t)\right], \qquad (3.57)$$

and their sum forms our Lyapunov function:

$$L(t) = \Delta^{+}(t) + \Delta^{-}(t).$$
 (3.58)

 $L(t) \geq 0$ as $\Delta^+(t) \geq 0$ and $\Delta^-(t) \geq 0$, and the equality holds iff for all $i \in V$, $\Delta_i(t) = 0$. Indeed, the Lyapunov function we formulate here for MPP is the same as for ABF, and it can readily be verified that L(t) acts as a valid norm for a vector of the probability estimation errors. In order to prove that L(t) is also decreasing, we define $\mathcal{K}_+(t)$ as a set comprising all nodes whose error equals $\Delta^+(t)$.

$$\mathcal{K}_{+}(t) = \left\{ i \in V | \Delta_{i}(t) = \Delta^{+}(t) \right\}.$$
(3.59)

Similarly,

$$\mathcal{K}_{-}(t) = \left\{ i \in V | \Delta_i(t) = -\Delta^{-}(t) \right\}.$$
(3.60)

The following lemma shows that $\Delta^+(t)$ decreases exponentially if $\Delta^+(t) \neq 0$.

Lemma 3.3.1. Consider (2.20) under Assumption 3. With Δ^+ and e_{\max} defined in (3.56) and (2.19), for all t,

$$\Delta^+(t+1) \le e_{\max}\Delta^+(t). \tag{3.61}$$

Proof. Consider $l \in \mathcal{K}_+(t+1)$. Suppose any neighbor $j \in \mathcal{N}(l)$ is a true constraining node of l, i.e. from (2.21) and Definition 3.5,

$$p_l = 1 - (1 - p_j)e_{lj}.$$
(3.62)

Then from (3.8),

$$\begin{aligned} \Delta^{+}(t+1) &= \Delta_{l}(t+1) \\ &= \hat{p}_{l}(t+1) - p_{l} \\ &\leq 1 - (1 - \hat{p}_{j}(t))e_{lj} - p_{l} \\ &= 1 - (1 - \hat{p}_{j}(t))e_{lj} - (1 - (1 - p_{j})e_{lj}) \\ &= (\hat{p}_{j}(t) - p_{j})e_{lj} \\ &= e_{lj}\Delta_{j}(t) \\ &\leq e_{\max}\Delta^{+}(t), \end{aligned}$$
(3.64)

where
$$(3.63)$$
 uses (2.20) , (3.64) uses (2.19) and (3.56) .

Similarly, for $\Delta^{-}(t)$, we have the following lemma.

Lemma 3.3.2. Consider (2.20) under Assumption 3. With Δ^- and e_{max} defined in (3.57) and (2.19), for all t,

$$\Delta^{-}(t+1) \le e_{\max}\Delta^{-}(t). \tag{3.65}$$

Proof. Consider $l \in \mathcal{K}_{-}(t+1)$. Suppose j is the constraining node of l at time t+1, then

$$\Delta^{-}(t+1) = p_{l} - \hat{p}_{l}(t+1)$$

$$\leq 1 - (1 - p_{j})e_{lj} - \hat{p}_{l}(t+1)$$

$$= 1 - (1 - p_{j})e_{lj} - (1 - (1 - \hat{p}_{j}(t))e_{lj}) \qquad (3.66)$$

$$= (p_{j} - \hat{p}_{j}(t))e_{lj}$$

$$= -e_{lj}\Delta_{j}(t)$$

$$= e_{\max}\Delta^{-}(t), \qquad (3.67)$$

where (3.66) uses (2.21) and (3.67) uses (2.19) and (3.57).

With Lemma 3.3.1 and 3.3.2, Theorem 3.3.3 proves that MPP is GUES and $\hat{p}_i = p_i$ for all $i \in V$ is the only stationary point of (2.20).

Theorem 3.3.3. Under conditions of Lemma 3.3.1 and Lemma 3.3.2. With L(t) and e_{\max} defined in (3.58) and (2.19), one obtains,

$$L(t+1) \le e_{\max}L(t), \ \forall t \ge t_0, \tag{3.68}$$

(2.20) is GUES and $\hat{p}_i = p_i$, $\forall i \in V$ is the only stationary point of (2.20).

Proof. This is a direct consequence of Lemma 3.3.1 and Lemma 3.3.2, and the fact that $0 < e_{\text{max}} < 1$.

3.3.2 Tight bound on time to convergence

In this section, we tightly bound the convergence time of MPP. To address this, we derive the convergence time of the greatest overestimate $\Delta^+(t)$ and the least underestimate $\Delta^-(t)$, respectively.

We first introduce the following definition which associates with the convergence time of $\Delta^+(t)$.

Definition 3.6. For a connected graph \mathcal{G} , consider any sequence of nodes such that the predecessor of each node is one of its true constraining nodes. Define $\mathcal{D}(\mathcal{G})$, the effective diameter of \mathcal{G} , as the longest length such a sequence can have in \mathcal{G} (i.e., the diameter of the shortest path forest rooted at S).

The proof showing that $\mathcal{D}(\mathcal{G})$ is finite is similar as in [58] and thus omitted. With Definition 3.6, the following theorem shows that the overestimate will vanish to zero within at most $\mathcal{D}(\mathcal{G}) - 1$ steps.

Theorem 3.3.4. Under Assumption 3, $\Delta^+(t)$ defined in (3.56) obeys

$$\Delta^{+}(t) = 0, \ \forall \ t \ge t_0 + \mathcal{D}(\mathcal{G}) - 1.$$
(3.69)

Proof. As \mathcal{G} is connected, each node belongs to a sequence of nodes n_1, n_2, \dots, n_T , such that n_i is the true constraining node of n_{i+1} and $n_1 \in S$. From Definition 3.6,

 $T \leq \mathcal{D}(\mathcal{G})$. We now assert and prove by induction that,

$$\Delta_{n_i}(t) \le 0, \ \forall \ t \ge i - 1 + t_0, \ \text{and} \ i \le T.$$
 (3.70)

Then the result is proved from (3.56). As $n_1 \in S$, (3.70) holds from (2.20). Now suppose it holds for some $i \in \{0, \dots, T-1\}$. As $n_i \in \mathcal{C}(n_{i+1}) \subset \mathcal{N}(n_{i+1})$, from (2.20), (2.21) and the induction hypothesis, for all $t \ge i + 1 + t_0$,

$$\hat{p}_{n_{i+1}}(t) \leq 1 - (1 - \hat{p}_{n_i}(t-1))e_{n_i n_{i+1}} \\
\leq 1 - (1 - p_{n_i})e_{n_i n_{i+1}} \\
= p_{n_{i+1}}$$
(3.71)

where (3.71) uses (2.21). Thus (3.70) and (3.69) hold.

While for ABF, we lower bound the decline in $\Delta^{-}(t)$ in order to find its convergence time. In MPP, lower bounding the decline ratio makes more sense.

Lemma 3.3.5. Under the conditions of lemmas 3.3.1 and 3.3.2 define,

$$S^{-}(t) = \{ i \in V | \Delta_i(t) < 0 \}, \qquad (3.72)$$

and

$$\hat{p}_{\min}(t) = \min_{i \in \mathcal{S}^{-}(t)} \{ \hat{p}_i(t) \}.$$
(3.73)

Then with e_{\max} defined in (2.19), the following holds unless $\mathcal{S}^{-}(t)$ is empty:

$$1 - \hat{p}_{\min}(t+1) \le e_{\max}(1 - \hat{p}_{\min}(t)) \tag{3.74}$$

Proof. Suppose $S^{-}(t+1)$ is not empty. Then from Lemma 3.3.2, and (3.60), $S^{-}(t)$ cannot be empty. Consider any $i \in S^{-}(t+1)$ and suppose j is its current constraining

node at t. Then we assert that $j \in S^-(t)$. Indeed assume $j \notin S^-(t)$. Thus $\hat{p}_j(t) \ge p_j$. As $j \in \mathcal{N}(i)$, from Definition 3.5 and (2.21),

$$\hat{p}_i(t+1) = 1 - (1 - \hat{p}_j(t))e_{ij}$$

 $\geq 1 - (1 - p_j)e_{ij}$
 $\geq p_i.$

Thus $i \notin S^{-}(t+1)$, leading to a contradiction, and hence $j \in S^{-}(t)$. Then from (2.19), (3.72) and (3.73), there holds:

$$1 - \hat{p}_{i}(t+1) = 1 - \hat{p}_{\min}(t+1)$$

= $(1 - \hat{p}_{j}(t))e_{ij}$
 $\leq (1 - \hat{p}_{\min}(t))e_{ij}$
 $\leq (1 - \hat{p}_{\min}(t))e_{\max}$

From Remark 3.3, $0 \leq \hat{p}_i(t) < 1$ for $i \in V$ and all t. Thus, Lemma 3.3.5 indicates that $\hat{p}_{\min}(t)$ strictly increases unless $S^{-}(t)$ is empty. With Theorem 3.3.4 and Lemma 3.3.5, we now tightly bound the time to convergence in the following theorem.

Theorem 3.3.6. Under conditions of Lemma 3.3.1 and Lemma 3.3.2. Consider $(2.20), \mathcal{D}(\mathcal{G})$ defined in Definition (3.6) and e_{\max} defined in (2.19). Define

$$p_{\max}(\mathcal{G}) = \max_{i \in V} \{p_i\},\tag{3.75}$$

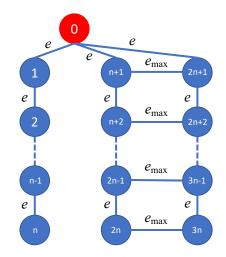


Figure 3.9: Illustration of the tightness of convergence time. The entire graph is used in the proof of Theorem 3.3.6.

for $\mathcal{G} = (V, E)$. Then $L(t) = 0, \forall t \ge t_0 + T$ with T obeying

$$T = \max\left\{ \mathcal{D}(\mathcal{G}) - 1, \left\lceil \log_{e_{\max}} \left(\frac{1 - p_{\max}(\mathcal{G})}{1 - \hat{p}_{\min}(t_0)} \right) \right\rceil \right\}$$
(3.76)

Further, for every n = |V| > 3, there exists a \mathcal{G} satisfying Assumption 3 for which L(t) > 0 for all t < T.

Proof. From Theorem 3.3.4, $\Delta^+(t)$ vanished to 0 within $\mathcal{D}(\mathcal{G}) - 1$ steps, this accounts for the $\mathcal{D}(\mathcal{G}) - 1$ in (3.76). For $\Delta^-(t)$, from Lemma 3.3.5, $\Delta^-(t) = 0$ whenever

$$t - t_0 \ge T_- = \left[\log_{e_{\max}} \left(\frac{1 - p_{\max}(\mathcal{G})}{1 - \hat{p}_{\min}(t_0)} \right) \right].$$
 (3.77)

To prove that (3.76) is a tight bound for convergence time, consider the graph in Figure 3.2 with $e_{\text{max}} > e$ and $\log e$ is a multiple of $\log e_{\text{max}}$. Assume for all $i \in$

$$\{1, \cdots, n\}, \hat{p}_i(t_0) > p_{\max}(\mathcal{G}), \text{ and } \hat{p}_{n+i}(t_0) = \hat{p}_{2n+i}(t_0) = 0.$$
 Then from Theorem 3.3.4,

it takes exactly $\mathcal{D}(\mathcal{G}) - 1$ iterations for $\Delta^+(t)$ to converge,

For $\Delta^{-}(t)$, w.l.o.g., suppose $t_0 = 0$, and

$$\left\lceil \frac{\log e}{\log e_{\max}} \right\rceil = \frac{\log e}{\log e_{\max}} = m.$$
(3.78)

Then we assert that for all $i \in \{1, \cdots, n\}$,

$$1 - \hat{p}_{n+i}(t) = 1 - \hat{p}_{2n+i}(t) = \begin{cases} e_{\max}^t & i \in \{\lfloor \frac{t}{m} \rfloor + 1, \cdots, n\} \\ \\ p_{n+i} & \text{otherwsie} \end{cases}$$
(3.79)

Here we call each pair n+i and 2n+i partners. Clearly, the true probability for node n+i and 2n+i is $1-e^i$. For $t \in \{1, \dots, m-1\}$, as $e_{\max} > e$ and $\hat{p}_{n+i}(0) = \hat{p}_{2n+i} = 0$, partner nodes will constrain each other until t = m when $e_{\max}^m = e$, leading to $\hat{p}_{n+1}(m) = \hat{p}_{2n+1}(m) = p_{n+1} = p_{2n+1} = 1 - e$. Notice that $\hat{p}_{n+i}(t) \ge 1 - e$ and $\hat{p}_{2n+i}(t) \ge 1 - e$ for $t \ge m$, and hence n+1 and n+2 are constrained by the source for $t \ge m$, establishing $\hat{p}_{n+1}(t) = \hat{p}_{2n+1}(t) = p_{n+1} = p_{2n+1}$ for $t \ge m$. Continuing this argument (3.79), there holds:

$$\hat{p}_{2n}(t) = \hat{p}_{3n}(t) < 1 - e^n = p_{2n} = p_{3n}, \forall t < mn.$$
(3.80)

Our result follows as

$$mn = n \frac{\log e}{\log e_{\max}}$$

$$= n \left[\frac{\log e}{\log e_{\max}} \right]$$

$$= \left[\log_{e_{\max}} e^{n} \right]$$

$$= \left[\log_{e_{\max}} \left(\frac{1 - p_{\max}(\mathcal{G})}{1 - \hat{p}_{\min}(t_{0})} \right) \right] \qquad (3.81)$$

where (3.81) uses
$$p_{\max}(\mathcal{G}) = 1 - e^n$$
 and $\hat{p}_{\min}(t_0) = 0$.

3.3.3 Robustness under perturbations

In this section, we analyze the robustness of MPP under *persistent perturba*tions on edges values, and demonstrate ultimate boundedness of probability estimates around the nominal values. Here we assume a node receives noisy probability estimate of its neighbors, and the noise is reflected in the value of edges, i.e., edge values, which represent success rate of delivery, change from their nominal values e_{ij} as

$$\bar{e}_{ij}(t) = e_{ij} + \epsilon_{ij}(t). \tag{3.82}$$

Such noise is assumed to be bounded and small, i.e., there exists an ϵ such that with e_{\min} and e_{\max} defined in (2.19),

$$|\epsilon_{ij}(t)| < \epsilon < \min\{e_{\min}, 1 - e_{\max}\}.$$
(3.83)

(3.83) ensures that edge value is still in (0, 1). Moreover, as the noise seen by node *i* differs from that seen by node *j*, we assume the noise is asymmetric,

$$\bar{e}_{ij}(t) \neq \bar{e}_{ji}(t). \tag{3.84}$$

Thus, (2.20) under perturbations should be interpreted as:

$$\hat{p}_{i}(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \{1 - (1 - \hat{p}_{j}(t)) \cdot \bar{e}_{ij}(t)\} & i \notin S \\ 0 & i \in S \end{cases}$$
(3.85)

We begin with the ultimate boundedness of $\Delta^+(t)$. We first introduce the following definition.

Definition 3.7. Consider $\mathcal{G} = (V, E)$, we call a path from a node *i* to the source set a most probable path, if it starts at *i*, ends with a source node and each node in the path is a true constraining node of its predecessor. We call a most probable path from *i*, the longest most probable path if it has the most nodes among all most probable paths of *i*. The set \mathcal{F}_i is the set of nodes whose longest most probable paths to the source have i + 1 nodes.

Thus if a node *i* has two most probable paths one having two and the other three nodes then $i \notin \mathcal{F}_1$ but $i \in \mathcal{F}_2$. From Definition 3.6, \mathcal{F}_i follows

$$\mathcal{F}_0 = S. \ \mathcal{F}_i \neq \emptyset, \ i \in \{0, 1, \cdots, \mathcal{D}(\mathcal{G}) - 1\},$$
(3.86)

and each node in \mathcal{F}_{i+1} has a true constraining node in \mathcal{F}_i :

$$\mathcal{C}(j) \cap \mathcal{F}_i \neq \emptyset, \ \forall j \in \mathcal{F}_{i+1}.$$
(3.87)

We define the smallest true failure rate of delivery in \mathcal{F}_i as

$$p_{i\min} = \min_{j \in \mathcal{F}_i} \{p_j\}.$$
(3.88)

Observe that $p_{0\min} = 0$ as $\mathcal{F}_0 = S$. Then the following lemma proves the ultimate boundedness of $\Delta^+(t)$.

Lemma 3.3.7. Consider (3.85). Then $\Delta^+(t) \leq \epsilon \sum_{i=0}^{\mathcal{D}(\mathcal{G})-2} (1-p_{i\min})(e_{\max}-\epsilon)^{\mathcal{D}(\mathcal{G})-2-i}$ for all $t \geq \mathcal{D}(\mathcal{G}) - 1$ with $\mathcal{D}(\mathcal{G})$, $\Delta^+(t)$, $p_{i\min}$, e_{\max} and ϵ defined in Definition 3.6, (3.56), (3.88), (2.19) and (3.83) respectively.

Proof. Consider the sequence of nodes n_1, n_2, \dots, n_T in the proof of Theorem 3.3.4. It follows that $T \leq \mathcal{D}(\mathcal{G})$ and $\hat{p}_{n_1}(t) = p_{n_1} = 0$ for all t in (3.85). The result holds if for $t \geq i - 1$,

$$\hat{p}_{n_i}(t) \le p_{n_i} + \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}) (e_{\max} - \epsilon)^{i-2-l}, \forall i \in \{2, \cdots, T\}.$$
(3.89)

We prove (3.89) by induction. For i = 2 and $t \ge 1$, from (3.85), there holds:

$$\hat{p}_{n_2}(t) \leq 1 - (1 - \hat{p}_{n_1}(t-1))e_{n_2n_1}(t-1)$$

$$= 1 - (1 - p_{n_1})e_{n_2n_1}(t-1)$$
(3.90)

$$\leq 1 - (1 - p_{n_1})(e_{n_2 n_1} - \epsilon) \tag{3.91}$$

$$\leq p_{n_2} + (1 - p_{n_1})\epsilon$$
 (3.92)

$$= p_{n_2} + (1 - p_{0\min})\epsilon \tag{3.93}$$

where (3.90) uses $\hat{p}_{n_1}(t) = p_{n_1} = 0$ for all t, (3.91) uses (3.83), (3.92) uses (2.21), and (3.93) uses (3.88).

Now suppose (3.89) holds for some $i \in \{2, \dots, T-1\}$. From (2.21), (3.83)

and our induction hypothesis, for all $t \ge i$, $\hat{p}_{n_{i+1}}(t)$ obeys

$$\begin{split} \hat{p}_{n_{i+1}}(t) \\ \leq & 1 - (1 - \hat{p}_{n_i}(t-1))e_{n_{i+1}n_i}(t-1) \\ \leq & 1 - (1 - p_{n_i} - \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min})(e_{\max} - \epsilon)^{i-2-l})(e_{n_{i+1}n_i} - \epsilon) \\ = & 1 - (1 - p_{n_i})e_{n_{i+1}n_i} + (1 - p_{n_i})\epsilon + (e_{n_{i+1}n_i} - \epsilon)\epsilon \sum_{l=0}^{i-2} (1 - p_{l\min})(e_{\max} - \epsilon)^{i-2-l} \\ = & p_{n_{i+1}} + (1 - p_{n_i})\epsilon + (e_{n_{i+1}n_i} - \epsilon)\epsilon \sum_{l=0}^{i-2} (1 - p_{l\min})(e_{\max} - \epsilon)^{i-2-l} \\ \leq & p_{n_{i+1}} + (1 - p_{i-1\min})\epsilon + (e_{\max} - \epsilon)\epsilon \sum_{l=0}^{i-2} (1 - p_{l\min})(e_{\max} - \epsilon)^{i-2-l} \\ = & p_{n_{i+1}} + \epsilon \sum_{l=0}^{i-1} (1 - p_{l\min})(e_{\max} - \epsilon)^{i-1-l} \end{split}$$

In order to address the ultimate boundedness of $\Delta^{-}(t)$, we utilize an approach like the *comparison principle* [45]. Instead of utilizing a *shrunken verision* of graph as we did for ABF [58]. Here, for a graph $\mathcal{G} = (V, E)$, we define its extended version, $\mathcal{G}^{+} = (V, E^{+})$ in the following definition.

Definition 3.8. Given $\mathcal{G} = (V, E)$, the undirected graph $\mathcal{G}^+ = (V, E^+)$, has the property that edge $(i, j) \in E^+$ iff $(i, j) \in E$. The edge value e_{ij}^+ between nodes i and j obeys

$$e_{ij}^+ = e_{ij} + \epsilon, \tag{3.94}$$

with ϵ defined in (3.83). Further \mathcal{G}^+ has the same source set S as \mathcal{G} , each i has the same set of neighbors as in \mathcal{G} , and P_i is the true failure rate of delivery between i and

the nearest source node, i.e. plays the role of p_i in \mathcal{G} . Further, we define $\mathcal{D}(\mathcal{G}^+)$ as the effective diameter of \mathcal{G}^+ , and \mathcal{F}_i^+ the set of nodes whose longest most probable paths defined in Definition 3.7 to the source have i + 1 nodes in \mathcal{G}^+ .

Similarly, for \mathcal{G}^+ , we define the smallest true failure rate of delivery in \mathcal{F}_i^+ as

$$p_{i\min}^{+} = \min_{j \in \mathcal{F}_{i}^{+}} \{P_{j}\}, \qquad (3.95)$$

It holds that $p_{0\min}^+ = 0$ as $\mathcal{F}_0^+ = S$.

We summarize the following assumptions for \mathcal{G} and \mathcal{G}^+ .

Assumption 4. Both the graphs \mathcal{G} and \mathcal{G}^+ defined in Definition 3.8 obey Assumption 3. The set of neighbors of node *i*, is time invariant for all $i \in V$ as is the source set. None of the source nodes move. The edge value of each pair of nodes *i* and *j* is given by (3.82) under (3.83) and (2.19). Although (3.84) holds, $e_{ij} = e_{ji}$ still holds. We also assume $t_0 = 0$ is the initial time.

Under Assumption 4, define $\hat{P}_i(t)$ as the estimated failure rate of delivery for $i \in V$ in \mathcal{G}^+ , when \mathcal{G}^+ is perturbations free, $\hat{P}_i(t)$ obeys

$$\hat{P}_{i}(t+1) = \begin{cases} \min_{j \in \mathcal{N}(i)} \{1 - (1 - \hat{P}_{j}(t)) \cdot (e_{ij} + \epsilon)\} & i \notin S \\ 0 & i \in S \end{cases}$$
(3.96)

The following lemma establishes a connection between probability estimates in \mathcal{G} and those in its extended version, \mathcal{G}^+ , defined in Definition 3.8.

Lemma 3.3.8. Suppose Assumption 4 holds. Consider $\hat{P}_i(t)$ defined in (3.96) and $\hat{p}_i(t)$ defined in (3.85). Suppose for all $i \in V$, $\hat{P}_i(0) \geq \hat{p}_i(0)$. Then $\hat{p}_i(t) \geq \hat{P}_i(t)$ $\forall t \geq 0$ and for all $i \in V$. Proof. We prove by induction. The result holds for t = 0. Suppose for some $t \ge 0$, $\hat{p}_i(t) \ge \hat{P}_i(t)$ for all $i \in V$. Suppose $j \in \mathcal{N}(i)$ is a current constraining node of i at time t + 1 in (3.85) while $k \in \mathcal{N}(i)$ is a current constraining node of i at time t + 1in (3.96). Then there holds:

$$\hat{P}_{i}(t+1) = 1 - (1 - \hat{P}_{k}(t))(e_{ik} + \epsilon)
\leq 1 - (1 - \hat{P}_{j}(t))(e_{ij} + \epsilon)$$
(3.97)

$$\leq 1 - (1 - \hat{p}_j(t))(e_{ij} + \epsilon)$$
 (3.98)

$$\leq 1 - (1 - \hat{p}_j(t))e_{ij}(t)$$
 (3.99)

$$= \hat{p}_i(t+1) \tag{3.100}$$

where (3.97) uses (3.96), (3.98) uses the induction hypothesis that $\hat{p}_i(t) \ge \hat{P}_i(t)$ for all $i \in V$, and (3.99) uses (3.83).

Thus, for all $i \in V$, the probability estimates $\hat{p}_i(t)$ in (3.85) are uniformly lower bounded by $\hat{P}_i(t)$ in (3.96) if initial condition obeys $\hat{P}_i(0) \geq \hat{p}_i(0)$. Further, as \mathcal{G}^+ satisfies the same assumptions as \mathcal{G} , and is perturbations free, both $\Delta^+(t)$ and $\Delta^-(t)$ of (3.96) converge to zero. To establish an ultimate bound on Δ^- in (3.85), we need to relate the true probability P_i in \mathcal{G}^+ to p_i in \mathcal{G} , and this is stated in the following theorem.

Lemma 3.3.9. Suppose Assumption 4 holds. Then for all $i \in V$,

$$p_i \le P_i + \epsilon \sum_{l=0}^{\mathcal{D}(\mathcal{G}^+)-2} (1 - p_{l\min}^+) e_{\max}^{\mathcal{D}(\mathcal{G}^+)-2-l}.$$
 (3.101)

with e_{\max} , $\mathcal{D}(\mathcal{G}^+)$ and $p_{l\min}^+$ defined in (2.19), Definition 3.8 and (3.95), respectively.

Proof. Consider a sequence of nodes n_1, n_2, \dots, n_{T_1} , such that $n_1 \in S$, and for all $i \in \{1, \dots, T_1 - 1\}$, n_i is a true constraining node of n_{i+1} in \mathcal{G}^+ . Every node in \mathcal{G}^+ is in one such sequence, and $T_1 \leq \mathcal{D}(\mathcal{G}^+)$. As $p_{n_1} = P_{n_1} = 0$, (3.101) holds for $i \in S$. Then the result holds if for all $i \in \{2, \dots, T_1\}$,

$$p_{n_i} \le P_{n_i} + \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}^+) e_{\max}^{i-2-l}.$$
 (3.102)

For $i = 2, p_{n_2}$ obeys

$$p_{n_2} \leq 1 - (1 - p_{n_1})e_{n_2 n_1} \tag{3.103}$$

$$= 1 - (1 - P_{n_1})e_{n_2n_1} \tag{3.104}$$

$$= P_{n_2} + (1 - P_{n_1})\epsilon \tag{3.105}$$

$$= P_{n_2} + (1 - p_{0\min}^+)\epsilon \tag{3.106}$$

where (3.103) uses (2.21), (3.104) uses $p_{n_1} = P_{n_1} = 0$, (3.105) uses that n_1 is a true constraining node of n_2 in \mathcal{G}^+ , and (3.106) uses (3.95). Thus, (3.102) holds for i = 2.

 $= 1 - (1 - P_{n_1})(e_{n_2n_1} + \epsilon) + (1 - P_{n_1})\epsilon$

Now suppose (3.102) holds for some $i \in \{2, \dots, T_1 - 1\}$. As n_i and n_{i+1} are neighbors in both \mathcal{G} and \mathcal{G}^+ , n_i is a true constraining node of n_{i+1} in \mathcal{G}^+ and (3.83) holds, $p_{n_{i+1}}$ obeys

$$p_{n_{i+1}} \leq 1 - (1 - p_{n_i})e_{n_{i+1}n_i}$$

$$\leq 1 - (1 - P_{n_i} - \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}^+)e_{\max}^{i-2-l})e_{n_{i+1}n_i}$$

$$= 1 - (1 - P_{n_i} - \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}^+)e_{\max}^{i-2-l})(e_{n_{i+1}n_i} + \epsilon) + (1 - P_{n_i} - \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}^+)e_{\max}^{i-2-l})\epsilon$$

$$= 1 - (1 - P_{n_i})(e_{n_{i+1}n_i} + \epsilon) + (1 - P_{n_i})\epsilon + \epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}^+)e_{\max}^{i-2-l}e_{n_{i+1}n_i}$$

$$\leq P_{n_{i+1}} + (1 - p_{i-1}^+)\epsilon + e_{\max}\epsilon \sum_{l=0}^{i-2} (1 - p_{l\min}^+)e_{\max}^{i-2-l}$$

$$= P_{n_{i+1}} + \epsilon \sum_{l=0}^{i-1} (1 - p_{l\min}^+)e_{\max}^{i-1-l}$$

With Lemma 3.3.8 and Lemma 3.3.9, we now prove the ultimate boundedness of Δ^- .

Lemma 3.3.10. Considering (3.85) under Assumption 4. With $\mathcal{D}(\mathcal{G}^+)$, e_{\max} , e_{\min} , $p_{\max}(\mathcal{G}^+)$ and \hat{p}_{\min} defined in Definition 3.8, (2.19), Theorem 3.3.6 and Lemma 3.3.5 respectively, for all

$$t \ge T_1 = \left\lceil \log_{e_{\max}+\epsilon} \left(\frac{1 - p_{\max}(\mathcal{G}^+)}{1 - \hat{p}_{\min}(t_0)} \right) \right\rceil,$$
(3.107)

there holds:

$$\Delta^{-}(t) \le \epsilon \sum_{l=0}^{\mathcal{D}(\mathcal{G}^{+})-2} (1-p_{l\min}^{+}) e_{\max}^{\mathcal{D}(\mathcal{G}^{+})-2-l}.$$
(3.108)

Proof. Consider (3.96) in \mathcal{G}^+ with the initialization in Lemma 3.3.8. As \mathcal{G}^+ satisfies Assumption 3, $\hat{P}_i(t)$ for all $i \in V$ converge to P_i within a finite time T_1 . Thus from Lemma 3.3.8, $\hat{p}_i(t) \ge P_i$ for all $t \ge T_1$ and $i \in V$. With Lemma 3.3.9, for $t \ge T_1$, there holds:

$$\begin{aligned} -\Delta_i(t) &\leq p_i - P_i \\ &\leq P_i + \epsilon \sum_{l=0}^{\mathcal{D}(\mathcal{G}^+) - 2} (1 - p_{l\min}^+) e_{\max}^{i-2-l} - P_i \\ &= \epsilon \sum_{l=0}^{\mathcal{D}(\mathcal{G}^+) - 2} (1 - p_{l\min}^+) e_{\max}^{i-2-l}. \end{aligned}$$

From Theorem 3.3.6, T_1 is the time when $\Delta^-(t)$ of (3.96) goes to zero. With Lemma 3.3.5 and Theorem 3.3.6, T_1 obeys (3.107).

The following theorem provides a tight bound on the ultimate bound and the time to attain it.

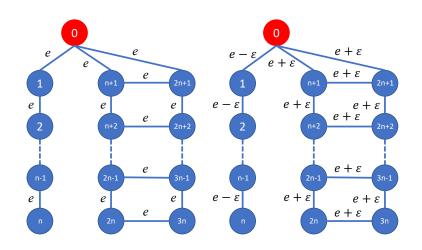


Figure 3.10: Illustration of the tightness of convergence time with perturbations

L(t) in (3.58) obeys

$$L(t) \leq \epsilon \sum_{i=0}^{\mathcal{D}(\mathcal{G})-2} (1-p_{i\min})(e_{\max}-\epsilon)^{\mathcal{D}(\mathcal{G})-2-i} + \epsilon \sum_{i=0}^{\mathcal{D}(\mathcal{G}^+)-2} (1-p_{i\min}^+)e_{\max}^{\mathcal{D}(\mathcal{G}^+)-2-i}, \forall t \geq T$$
(3.109)

with T obeying

$$T = \max\left\{ \mathcal{D}(\mathcal{G}) - 1, \left\lceil \log_{e_{\max} + \epsilon} \left(\frac{1 - p_{\max}(\mathcal{G}^+)}{1 - \hat{p}_{\min}(t_0)} \right) \right\rceil \right\}$$
(3.110)

Further, for every n = |V| > 3, there exists a graph and perturbations conforming to Assumption 4 for which the bound in (3.109) is attained in precisely T steps.

Proof. Ultimate boundedness follows directly from Lemma 3.3.7 and Lemma 3.3.10. For the tightness of the ultimate bound and the time to attain it. Consider the two graphs in Figure 3.10, the right figure is the perturbed version of the left one. The perturbation obeys Assumption 4 and the perturbed graph obeys Assumption 3. The probability estimates in the perturbed graph will converge to the correct values as the edge values are fixed. As $e = e_{\max}$ in this case, $p_n = 1 - (e_{\max} - \epsilon)^n$, and $p_{2n} = p_{3n} =$ $1 - (e_{\max} + \epsilon)^n$. It is readily checked that node *n* has the greatest overestimate shown in Lemma 3.3.7, while node 2n and 3n have the greatest underestimate shown in Lemma 3.3.10, and hence (3.109) is precisely met. Further, as $\mathcal{D}(\mathcal{G}) = \mathcal{D}(\mathcal{G}^+) = n + 1$ and $p_{\max}(\mathcal{G}^+) = p_{2n} = p_{3n}$, by setting $\hat{p}_i(0) > p_n = 1 - (e_{\max} - \epsilon)^n$ for all $i \in \{1, \dots, n\}$, and $\hat{p}_i(0) = 0$ for all $i \in \{n + 1, \dots, 3n\}$, then the time to attain the ultimate bound of $\Delta^+(t)$ is precisely the same as the first term in (3.110), and the ultimate bound of $\Delta^-(t)$ occurs precisely by the second term in (3.110), which is *n*.

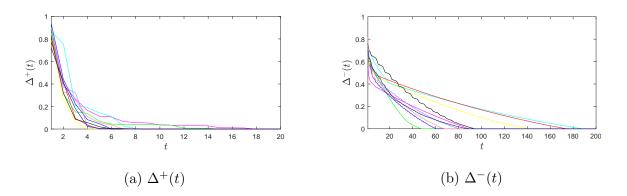


Figure 3.11: Trace of (a) greatest overestimate $\Delta^+(t)$ and (b) least underestimate $\Delta^-(t)$ for 10 runs, showing that $\Delta^+(t)$ will converge in $\mathcal{D}(\mathcal{G}) - 1$ rounds while $\Delta^-(t)$ takes longer to converge.

3.3.4 Simulations

In this section, we empirically confirm the previous results through simulations. Our simulation is a connected graph with 500 nodes, one of which is a source, randomly distributed in a 4×1 km² area, and communicate using a 0.25 km radius, ensuring that each node has approximately 20 neighbors.

We first verify the exponential decay of $\Delta^+(t)$ and $\Delta^-(t)$ defined in (3.56) and (3.57) without perturbations. The initial probability estimates and the edge values in this case are both uniformly distributed between 0 and 1. The simulation is run 10 times.

Figure 3.11 shows the results of 10 trials, which are consistent with our analysis. $\Delta^+(t)$ decreases rapidly to zeros within at most 18 rounds, while $\mathcal{D}(\mathcal{G})$ is ranging from 24 to 45 in those trials. For $\Delta^-(t)$, as can be seen from (3.76) in Theorem 3.3.6,

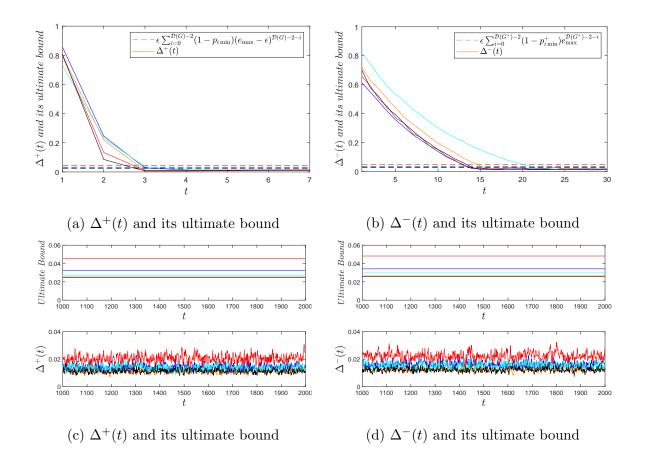


Figure 3.12: Trace of (a) greatest overestimate $\Delta^+(t)$ and its ultimate bound, and (b) least underestimate $\Delta^-(t)$ and its ultimate bound for 5 runs, as well as their partial enlarged views (c) and (d).

its convergence time depends on e_{\max} , $\hat{p}_{\min}(t_0)$ and $p_{\max}(\mathcal{G})$, and normally $\Delta^-(t)$ has slower convergence than $\Delta^+(t)$.

Now we turn to the robustness of the algorithm. Here we consider static nodes with asymmetric noise in the estimated e_{ij} , the edge values under perturbations will change from their nominal values as shown in (3.82), and the noise $\epsilon_{ij}(t)$ is uniformly distributed between $-\min\{e_{\min}, 1 - e_{\max}\}$ and $\min\{e_{\min}, 1 - e_{\max}\}$, and thus obeys (3.83). Further, the noise is asymmetric in that $\epsilon_{ij}(t) \neq \epsilon_{ji}(t)$. The simulation is run for 5 times.

Simulation results are shown in Figure 3.12, $\epsilon = 5.9 \times 10^{-3}, 7.5 \times 10^{-3}, 7.9 \times 10^{-3}, 1.05 \times 10^{-2}, 5.7 \times 10^{-3}$ for those 5 trials. $\Delta^+(t)$ for all trials becomes lower than its predict ultimate bound after 4 rounds, while $\mathcal{D}(\mathcal{G})$ ranges from 24 to 36. Figure 3.12 (b) shows that $\Delta^-(t)$ needs a longer time to drop below its ultimate bound. This is reasonable as from (3.107) in Lemma 3.3.10, where the time for $\Delta^-(t)$ to be ultimately bounded depends on e_{\max} , $\hat{p}_{\min}(t_0)$ and $p_{\max}(\mathcal{G}^+)$. Figure 3.12 (c) and (d) are partially enlarged views of (a) and (b), as Lyapunov based analysis are based on worst cases, there is a gap between $\Delta^+(t)$, $\Delta^-(t)$ and their corresponding ultimate upper bounds.

3.4 GUAS of GABF

Though in Section 3.2 ABF has proved to be GUAS, its convergence can be very slow in graphs will short edges if some initial estimates are smaller than their true values. As can be seen from Lemma 3.2.6, the rising value problem may occur even if a pair of nodes with underestimated distance estimates do not constrain each other. In this section, we revisit GABF introduced in Section 2.4. We prove it to be GUAS, bounding the time to converge, and show via simulations. Moreover, the robustness of GABF is not impaired while permitting fast convergence. Indeed, by setting the addition parameter D defined in (2.18) over some threshold, GABF will have the identical ultimate bounds as ABF under the same structural perturbations. Proof for this part will be included in the robust stability proof for the generalized Gblock as a byproduct in Chapter 4.

The convergence analysis of this algorithm is nontrivial as unlike ABF $\Delta^+(t)$ and $\Delta^-(t)$ defined in (3.4) and (3.5) here may well intermittently increase, and is thus not a Lyapunov function. In fact the proof of GUAS in the next section does not use a Lyapunov function at all.

3.4.1 Preliminaries

Assumption 5 is the standing assumption in this section.

Assumption 5. Graph \mathcal{G} is connected, $S \neq \emptyset$, $S \neq V$, $e_{\max} \ge e_{ik} = e_{ki} \ge e_{\min} > 0$, $D \ge 0$ and $g(\cdot)$ obeys (2.16).

As in any iteration the estimated distance of a node is obtained by one of the bullets in (2.18), at each t, we partition V into two sets defined below.

Definition 3.9. The ABF set $\mathcal{A}(t)$ comprises all nodes that use the first case in (2.18) to obtain $\hat{d}_i(t+1)$. Define a set the set of extraordinary nodes $\mathcal{E}(t) = V \setminus \mathcal{A}(t)$ to be those that use the second case in (2.18) to obtain $\hat{d}_i(t+1)$.

We also introduce the notion of a constraining node of i that determines $\hat{d}_i(t+1).$

Definition 3.10. For $i \in \mathcal{A}(t)$, the minimizing j in (2.14) used to find $\tilde{d}_i(j+1)$, is i's current constraining node at t. For $i \in \mathcal{E}(t)$, i is its own current constraining node. Sources are their own constraining nodes.

It is important to note that a source may well be in the extraordinary set \mathcal{E} . d_i the true distance in GABF is the same as in ABF, and obeys the following recursion.

$$d_{i} = \begin{cases} 0, & i \in S\\ \min_{k \in \mathcal{N}(i)} \{e_{ik} + d_{k}\} & i \notin S \end{cases}$$

$$(3.111)$$

with S the set of sources, and similarly a *true constraining node* is defined as

Definition 3.11. A j that minimizes the right hand side of (3.111) is a true constraining node of $i \in V \setminus S$. As a node may have multiple true constraining nodes, the set of true constraining nodes of a node $i \in V \setminus S$ is C(i).

Based on Definition 3.11, we introduce another.

Definition 3.12. Call a path from a node i to the source set a shortest path, if it starts at i, ends with a source node and each node in the path is a true constraining node of its predecessor. Call a shortest path from i, the longest shortest path if it has the most nodes among all shortest paths of i. The set \mathcal{F}_i is the set of nodes whose longest shortest paths to the source have i+1 nodes. Call $\mathcal{D}(\mathcal{G})$ the effective diameter of \mathcal{G} if the longest shortest path among all $i \in V$ has $\mathcal{D}(\mathcal{G})$ nodes. Thus if a node *i* has two shortest paths one having two and the other three nodes then $i \notin \mathcal{F}_1$ but $i \in \mathcal{F}_2$. From Definition 3.12, \mathcal{F}_i follows

$$\mathcal{F}_0 = S. \ \mathcal{F}_i \neq \emptyset, \ i \in \{0, 1, \dots, \mathcal{D}(\mathcal{G}) - 1\}.$$

$$(3.112)$$

Every node in \mathcal{F}_{i+1} has a true constraining node in \mathcal{F}_i :

$$\mathcal{C}(j) \bigcap \mathcal{F}_i \neq \emptyset, \ \forall \ j \in \mathcal{F}_{i+1}.$$
(3.113)

Further it has been shown in [58] that $\mathcal{D}(\mathcal{G})$ is finite.

3.4.2 Global uniform asymptotic stability

We define two sets requiring recursive definitions below.

Definition 3.13. The set of nodes $\mathcal{R}(t)$ rooted to S with nodes "rooted" to the source set obeys $\mathcal{R}(t_0) = S$. Further $\mathcal{R}(t+1)$ comprises all nodes whose current constraining node at t, is in $\mathcal{R}(t)$. We also define a set $\mathcal{U}(t)$ as unrooted to the source set if $\mathcal{U}(t_0) = V \setminus S$, and $\mathcal{U}(t+1)$ comprises all nodes whose current constraining node at t, is in $\mathcal{U}(t)$.

Lemma 3.4.1 shows that $\mathcal{U}(t)$ and $\mathcal{R}(t)$, partition V.

Lemma 3.4.1. Consider (2.14), (2.16) and (2.18) under Assumption 5, with $\mathcal{R}(t)$ and $\mathcal{U}(t)$ defined in Definition 3.13. Then for all initialization as in (2.17) and for all t,

$$\mathcal{U}(t) \bigcup \mathcal{R}(t) = V, \tag{3.114}$$

and

$$\mathcal{U}(t) \bigcap \mathcal{R}(t) = \emptyset. \tag{3.115}$$

Proof. First observe at each t, a node has only one current constraining node. Thus (3.115) holds. We prove (3.114) by induction. From Definition 3.13, (3.114) holds for $t = t_0$. Suppose (3.114) holds for some $t = t_1$. Consider $t = t_1 + 1$. Every node ihas a constraining node at t_1 . By the inductive hypothesis this is either in $\mathcal{R}(t)$ or in $\mathcal{U}(t)$. In the former case $i \in \mathcal{R}(t+1)$. In the latter case $i \in \mathcal{U}(t+1)$.

Observe that by definition of the unrooted set

$$\mathcal{U}(t) = \emptyset \Rightarrow \mathcal{U}(t+1) = \emptyset. \tag{3.116}$$

This is so because at t + 1 every node i has a current constraining node j. As $\mathcal{U}(t)$ is empty, from Lemma 3.4.1, $j \in \mathcal{R}(t)$ and thus $i \in \mathcal{R}(t+1)$. We further define the following function:

$$\hat{d}_{\min}(t) = \min_{j \in \mathcal{U}(t)} \{ \hat{d}_j(t) \} \text{ if } \mathcal{U}(t) \neq \emptyset.$$
(3.117)

The next lemma proves that the lower bound of $\hat{d}_i(t)$ with $i \in \mathcal{U}(t)$ strictly increases.

Lemma 3.4.2. Consider (2.14), (2.16) and (2.18) under Assumption 5, $\mathcal{U}(t)$ and $\hat{d}_{\min}(t)$ defined in Definition 3.13 and (3.117), respectively, for all initialization as in (2.17). The following holds while the set $\mathcal{U}(t) \neq \emptyset$:

$$\hat{d}_i(t) \ge \hat{d}_{\min}(t_0) + \min\{e_{\min}, \delta\}(t - t_0), \ \forall i \in \mathcal{U}(t).$$
 (3.118)

Proof. We prove by induction. Clearly, (3.118) holds for $t = t_0$. Suppose (3.118) holds for $t = t_1$. If $\mathcal{U}(t_1 + 1)$ is nonempty then from (3.116) so is $\mathcal{U}(t_1)$. Suppose *i* is the node with the minimum distance estimate in $\mathcal{U}(t_1 + 1)$, i.e

$$\hat{d}_i(t_1+1) = \hat{d}_{\min}(t_1+1).$$
 (3.119)

By definition, j the current constraining node of $i \in \mathcal{U}(t_1 + 1)$ is in $\mathcal{U}(t_1)$ Thus from the induction hypothesis

$$\hat{d}_j(t_1) \ge \hat{d}_{\min}(t_1) \ge \hat{d}_{\min}(t_0) + \min\{e_{\min}, \delta\}(t_1 - t_0)$$
(3.120)

If $i \in \mathcal{E}(t_1 + 1)$, then j = i and from (3.119) and (3.120) we have

$$\hat{d}_i(t_1+1) = \hat{d}_{\min}(t_1+1) \ge \hat{d}_j(t_1) + \delta \ge \hat{d}_{\min}(t_1)$$
$$\ge \hat{d}_{\min}(t_0) + \min\{e_{\min}, \delta\}(t_1+1-t_0).$$

If $i \in \mathcal{A}(t_1+1)$

$$\hat{d}_{i}(t_{1}+1) = \hat{d}_{\min}(t_{1}+1) = \hat{d}_{j}(t_{1}) + e_{ij} \\
\geq \hat{d}_{\min}(t_{1}) + e_{\min} \ge \hat{d}_{\min}(t_{1}) \\
\geq \hat{d}_{\min}(t_{0}) + \min\{e_{\min}, \delta\}(t_{1}+1-t_{0}).$$

The result follows.

We now show that the estimates of all nodes in $\mathcal{R}(t)$ are overestimates.

Lemma 3.4.3. Consider (2.14), (2.16) and (2.18) under Assumption 5, with $\mathcal{A}(t)$, $\mathcal{E}(t)$, $\mathcal{R}(t)$ and d_i defined in Definition 3.9, 3.13 and (2.21), respectively. Then for all initialization as in (2.17) and $i \in \mathcal{R}(t)$, $\hat{d}_i(t)$ obeys

$$\hat{d}_i(t) \ge d_i \tag{3.121}$$

Proof. We prove by induction. As $\mathcal{R}(t_0) = S$ and $d_i = 0$, $\forall i \in S$, under (2.17), (3.121) holds for $t = t_0$. Suppose (3.121) holds for $t = t_1$. Consider $i \in \mathcal{R}(t_1 + 1)$. If

L			I
L			I
-	-	-	J

 $i \in \mathcal{E}(t_1 + 1)$, from (2.18), $\hat{d}_i(t_1 + 1)$ obeys

$$\hat{d}_i(t_1+1) > \hat{d}_i(t_1) + \delta$$

$$\geq d_i + \delta$$
(3.122)

If $i \in V \setminus \mathcal{E}(t_1 + 1) = \mathcal{A}(t_1 + 1)$, from (2.14), $\hat{d}_i(t_1 + 1)$ obeys

$$\hat{d}_{i}(t_{1}+1) = \hat{d}_{k}(t_{1}) + e_{ik}$$

$$\geq d_{k} + e_{ik}$$

$$\geq d_{i}.$$
(3.123)

Define:

$$d_{\max} = \max_{k \in V} \{d_k\} \tag{3.124}$$

and

$$T^* = \left\lceil \frac{d_{\max}}{\min\{\delta, e_{\min}\}} \right\rceil.$$
(3.125)

Then from Lemma 3.4.2, we have that

$$\hat{d}_i(t) \ge d_i, \forall i \in \mathcal{U}(t), \forall t \ge t_0 + T^*.$$

Thus, as $\mathcal{U}(\cdot)$ and $\mathcal{R}(\cdot)$ partition V, from Lemma 3.4.3,

$$\hat{d}_i(t) \ge d_i, \forall i \in V, \forall t \ge t_0 + T^*.$$
(3.126)

Thus underestimates are eliminated at T^* . We now define the smallest distance in \mathcal{F}_i

as

$$d_{i\min} = \min_{j \in \mathcal{F}_i} \{d_j\}.$$
(3.127)

Observe $d_{0\min} = 0$ as $\mathcal{F}_0 = S$. Define a sequence

$$T_i = \max\left\{0, \left\lceil \frac{M - d_{i\min} - D}{\delta} \right\rceil\right\} + 2.$$
(3.128)

Then we have the following lemma.

Lemma 3.4.4. Consider (2.14), (2.16) and (2.18) under Assumption 5, and \mathcal{F}_i defined in Definition 3.12. Suppose at a time $t_L > T^*$ defined in (3.125) and $L \in \{0, 1, \dots, \mathcal{D}(\mathcal{G}) - 2\}$

$$\hat{d}_i(t) = d_i, \ \forall i \in \bigcup_{i=0}^L \mathcal{F}_i, \ \forall t \ge t_L,$$
(3.129)

i.e., all distance estimates of nodes in $\mathcal{F}_0, \dots, \mathcal{F}_L$ have converged by t_L . Then with T_i defined in (3.128), there holds:

$$\hat{d}_i(t) = d_i, \ \forall i \in \bigcup_{i=0}^{L+1} \mathcal{F}_i, \ \forall t \ge t_L + T_{L+1}.$$
(3.130)

Proof. As $t_L > T^*$, from (3.126), no underestimates remain at or beyond t_L . Consider $i \in \mathcal{F}_{i+1}$. Suppose $i \in \mathcal{E}(t_L+1)$, i.e. in the extraordinary set whose distance estimates obey the second bullet of (2.18). As there are no underestimates, this mandates that with $d_{L+1,\min}$ given in (3.127), from the second bullet in (2.18) the following holds:

$$\hat{d}_i(t_L+1) \ge D + d_i \ge D + d_{L+1,\min}, \ \forall i \in \mathcal{F}_{L+1}.$$
 (3.131)

Then from (2.16) and (3.128), $\hat{d}_i(t_i) > M$ for some $t_L \leq t_i \leq t_L + T_{L+1} - 1$ and $i \in \mathcal{A}(t_i + 1)$. Thus there is always a $t_L \leq t_i \leq t_L + T_{L+1} - 1$ for which $i \in \mathcal{A}(t_i + 1)$. From (3.113) there is a $j \in \mathcal{F}_L$ that is a true constraining node of i. From (2.14) and (2.18) one has for k the current constraining node of i

$$\hat{d}_i(t_i + 1) = \tilde{d}_i(t_i + 1)$$

= $\hat{d}_k(t_i) + e_{ik}$ (3.132)

$$\geq d_k + e_{ik} \tag{3.133}$$

$$\geq d_j + e_{ij} \tag{3.134}$$

$$= \hat{d}_j(t_i) + e_{ij} \tag{3.135}$$

where (3.132) uses the definition of the current constraining node, (3.133) uses the fact that there are no underestimates, (3.134) uses (3.111) and the fact that j is a true constraining node of i and (3.135) uses the fact that nodes in \mathcal{F}_L have converged. The inequalities from (3.132-3.135) show that k can be the current constraining node of i only if equality holds in (3.133) and (3.134). Thus we have

$$\hat{d}_i(t_i+1) = \tilde{d}_i(t_i+1) = d_j + e_{ij} = d_i,$$

the last equality stemming from the fact that j is a true constraining of i. Then as

$$\tilde{d}_i(t_i+2) = d_j + e_{ij} = d_i = \hat{d}_i(t_i+1),$$

 $i \in \mathcal{A}(t_i + 2)$, the ABF set where estimates obey the first bullet of (2.18). A simple induction then proves that $\hat{d}_i(t) = d_i, \forall t \ge t_i + 1$.

Then the main theorem below proves GUAS and furnishes an upper bound on the convergence time. **Theorem 3.4.5.** Consider (2.14), (2.16) and (2.18) under Assumption 5, with T_i defined in (3.128) and T^* in (3.125), define

$$T^* = \max\{T^*, T_0\}. \tag{3.136}$$

Then for all initialization as in (2.17), for all $i \in V$,

$$\hat{d}_i(t) = d_i, \ \forall t > t_0 + T^* + \sum_{i=1}^{\mathcal{D}(\mathcal{G})-1} T_i$$
(3.137)

with $\mathcal{D}(\mathcal{G})$ defined in Definition 3.12.

Proof. Consider a source $i \in \mathcal{E}(t_0)$. Because of (2.16) and the definition of T_0 in (3.128), $\hat{d}_i(t) > M$ for some $t \le t_0 + T_0 - 1$. Thus from (2.18),

$$\hat{d}_i(t+1) = \tilde{d}_i(t+1) = 0 = \tilde{d}_i(t+2).$$

Then as in the proof of Lemma 3.4.4, distance estimates of all source nodes converge at $t = T_0$. Recall that $t \ge T^*$ implies that all underestimates have been removed. As $S = \mathcal{F}_0$, the repeated application of Lemma 6 proves the result.

3.4.3 Simulations

In this section, we compare the relative performance of ABF and GABF. In the simulations, 500 nodes, one of which is a source, are randomly distributed in a $4 \times 1 \text{ km}^2$ field, communicating over a 0.25 km radius, running synchronously. The initial distance estimates are picked as $\hat{d}_i(0) \in U(0, \sqrt{17})$ km. Each simulation is run 10 times.

In the sequel Δ^- and Δ^+ are as in (3.5) and (3.4). Thus, $\Delta^+(t) = \Delta^-(t) = 0$ indicates convergence at time t. The simulations show that unlike ABF, Δ^- and Δ^+

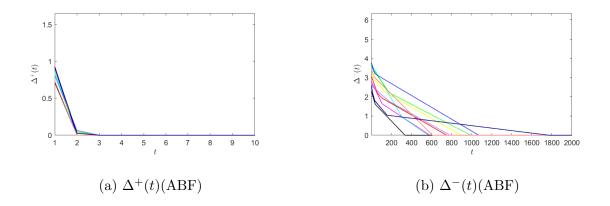


Figure 3.13: Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations.

need not be non-increasing in GABF. Figure 3.14 concerns GABF with $M = \delta = 2\sqrt{17}$ km, D = 0, i.e. with $M > d_{\text{max}}$: $\Delta^{-}(t)$ is rid of the "rising value problem" and converges in 2 rounds; $\Delta^{+}(t)$, takes slightly longer than ABF. Yet the overall convergence is much faster than ABF as shown in Figure 3.13.

In Figure 3.15 with D = 0.06, GABF converges slower than with D = 0. However, compared with ABF in Figure 3.13, the convergence of GABF is still much faster.

Effect of M is demonstrated in Figure 3.16 and 3.17, In Figure 3.16, we choose $M = \delta = 0.1$ km and D = 0, thus M is smaller than true distances of most nodes. It shows that in this case the behavior of $\Delta^+(t)$ and $\Delta^-(t)$ are very close to that in ABF. Evidently with small M, most nodes use the first bullet of (2.18) and the rising value problem persists. Also the use of the second bullet in (2.18), may cause $\Delta^+(t)$ not to converge in $\mathcal{D}(\mathcal{G})$ rounds. Figure 3.17 shows the simulation results with

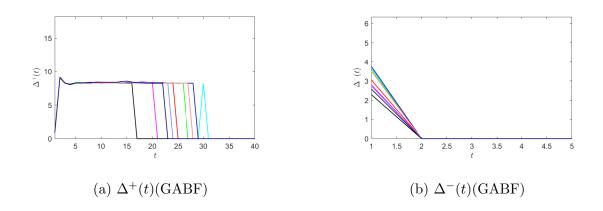


Figure 3.14: Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = \delta = 2\sqrt{17}$ and D = 0.

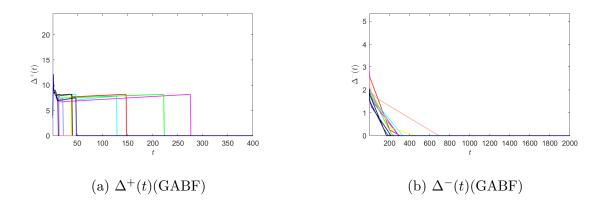


Figure 3.15: Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = \delta = 2\sqrt{17}$ and D = 0.06.

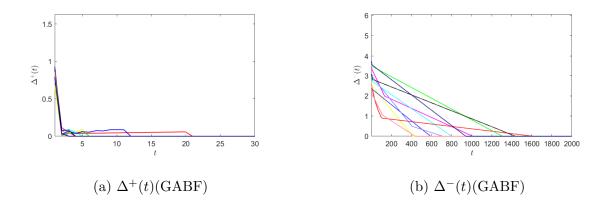


Figure 3.16: Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = \delta = 0.1$ and D = 0.

 $M = 2\sqrt{17}$ km, $\delta = 0.1$ km, D = 0. In this case, M is larger than d_{max} , and GABF converges slightly faster.

For the effect of δ , we compare Figure 3.17 and Figure 3.14. In Figure 3.17, by setting $\delta = 0.1$, $\Delta^+(t)$ needs a much longer time to converge than with a larger δ (e.g., Figure 3.14) as estimates take longer to rise to M, which shows that a smaller δ leads to a slower convergence.

3.5 Conclusion

In this chapter, we have presented the Lyapunov analyses of two specific Gblocks ABF and MPP, proving their GUAS and GUES respectively. Both ABF and MPP are ultimately bounded under bounded persistent perturbations. Moreover, the bounds are tight so as the bound of time to attain them. We also provide a non-

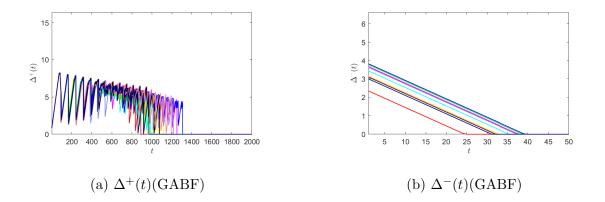


Figure 3.17: Plot from [57] of (a) the greatest overestimate $\Delta^+(t)$ and (b) the least underestimate $\Delta^-(t)$ for 10 runs without perturbations under $M = 2\sqrt{17}, \delta = 0.1$ and D = 0.

Lyapunov analysis for GABF to prove its GUAS and ultimate boundedness, and this proof gives us rich insights in proving the GUAS of the generalized G block in next chapter.

CHAPTER 4 GLOBAL STABILITY OF THE MOST GENERAL G BLOCK

4.1 Introduction

In this chapter, we focus on the generalized G block, which is the primitive type of G block introduced in [71]. One distinct difference between the proof of the generalized G block and GABF lies in that, in the generalized G block we cannot even *a priori* assume the existence of a stationary point. Rather, we will first prove the existence of the stationary point in Section 4.3. As GABF, the GUAS of the generalized G block is proved without using a Lyapunov function in Section 4.4. Despite this fact, in Section 4.5, with an additional Lipschitz condition on the update kernel, we are able to establish ultimate bounds on the state error under persistent perturbations. Section 4.6 illustrates and confirms the results with simulations, and Section 4.7 concludes.

4.2 Preliminaries

As stated in Section 2.2, the generalized G block obeys

$$\hat{x}_i(t+1) = F\left(\tilde{x}_i(t+1), \hat{x}_i(t), v_i\right)$$
(4.1)

with $\tilde{x}_i(t)$ and $F(\cdot, \cdot, v)$ satisfying

$$\tilde{x}_i(t+1) = \min\left\{\min_{k\in\mathcal{N}(i)}\left\{f\left(\hat{x}_k(t), e_{ik}\right)\right\}, s_i\right\}, \forall t \ge t_0,$$
(4.2)

and

$$F(\ell_1, \ell_2, v) = \begin{cases} \ell_1 & \ell_2 > M \text{ or } |\ell_2 - \ell_1| \le D \\ g(\ell_2) & \text{otherwise} \end{cases},$$
(4.3)

respectively, and g(x) obeys $g(x) \ge x + \delta$ for some $\delta > 0$.

Define \mathcal{S}^* as the set of nodes with finite maximum values.

$$\mathcal{S}^* = \{ i \in V | s_i < \infty \}. \tag{4.4}$$

With (4.4), we define S(t) comprising nodes in S^* that acquire their maximum values at time t,

$$\mathcal{S}(t) = \{ i \in \mathcal{S}^* \mid \hat{x}_i(t) = s_i \}, \tag{4.5}$$

and we say i is a source at time t if $i \in S(t)$. One difference between the generalized G block and its variants (e.g., ABF, GABF and MPP) is that, in the generalized G block a non-source node may have a finite maximum value, while in ABF, GABF or MPP, only sources have a finite maximum value of 0. This has been demonstrated through the tactical wireless network example in Section 2.6.

The following assumption holds for the generalized G block.

Assumption 6. Graph \mathcal{G} is connected, $e_{ik} = e_{ki} \ge e_{\min} > 0$, D defined in (4.3) obeys $D \ge 0$. \mathcal{S}^* defined in (4.4) in nonempty, Moreover,

$$s_i \ge s_{\min} = \min_{j \in \mathcal{S}^*} \{s_j\} \ge 0, \ \forall i \in V.$$

$$(4.6)$$

As in any given iteration the estimated state of a node is obtained by one of the bullet in (4.3), at each t, we partition V into two sets defined below.

Definition 4.1. The set $\mathcal{A}(t)$ comprises all nodes that use the first case in (4.3) to obtain $\hat{x}_i(t)$. Define a set the set of extraordinary nodes $\mathcal{E}(t) = V \setminus \mathcal{A}(t)$ to be those that use the second case in (4.3) to obtain $\hat{x}_i(t)$.

The next definition introduces the notion of the current constraining node of *i* that determines $\hat{x}_i(t)$.

Definition 4.2. For $i \in \mathcal{A}(t)$, the minimizing k in (4.2) used to find $\hat{x}_i(t)$, is i's current constraining node at t. For $i \in \mathcal{E}(t)$, i is its own current constraining node. Sources are their own constraining nodes.

Based on Definition 4.2, we give a definition of the smallest estimate from j to i.

Definition 4.3. Consider $i, j \in V$, as the graph is connected, there exist at least one path from j to i, e.g., $j = l_0 \rightarrow l_1 \rightarrow \cdots, \rightarrow l_L = i$, with which i will obtain a value based on the following recursion,

$$x_{l_{k}}^{*} = \begin{cases} s_{l_{k}} & k = 0 \text{ and } s_{l_{0}} \neq \infty \\ f(x_{l_{k-1}}^{*}, e_{l_{k-1}l_{k}}) & k \neq 0 \text{ and } s_{l_{0}} \neq \infty \\ \infty & \text{otherwise} \end{cases}$$
(4.7)

with $k \in \{0, 1, \dots, L\}$. We define x_{ij} as the smallest value x_i^* can have among all the paths from j to i.

Observe from Definition 4.3 that x_{ij} may be greater than s_i . The key steps in our proof are summarized as the following:

- We first define a set S_{∞} , based on which the stationary point of (4.1) is predefined. The concept of true constraining node is also introduced.
- We identify two time varying sets U(t) and R(t) that parition V, and show that lower bound of states of nodes in U(t) is increasing and states of nodes in R(t) are always overestimates, indicating that beyond a finite time all states are overestimates.
- We then show that the state of a node will converge to its stationary state after a finite time once its true constraining node has converged.
- Convergence is proved by showing that all nodes in \mathcal{S}_{∞} will converge to their stationary states.

4.3 The stationary point

We begin by proving that all states are lower bounded. To this end we define

$$\bar{x}(t) = \min_{i \in V} \hat{x}_i(t). \tag{4.8}$$

Lemma 4.3.1. Consider (4.1), (4.2) and (4.3) under Assumption 6, s_{\min} defined in (4.6). There exists a T such that, $\forall i \in V$, $\hat{x}_i(t) \geq s_{\min}$ for $t \geq T$.

Proof. Consider $\bar{x}(t+1) = \hat{x}_i(t+1)$. If $i \in \mathcal{A}(t+1)$, then either $\hat{x}_i(t+1) = s_i$ or

$$\hat{x}_i(t+1) = f(\hat{x}_j(t), e_{ij})$$
(4.9)

$$> \hat{x}_j(t) + \sigma \tag{4.10}$$

$$\geq \bar{x}(t) + \sigma$$

where in (4.9) we assume j is the current constraining node of i, and (4.10) uses the

progressive property of $f(\cdot, \cdot)$.

If $i \in \mathcal{E}(t+1)$, then from (2.7), it follows

$$\hat{x}_i(t+1) = g(\hat{x}_i(t))$$

$$\geq \hat{x}_i(t) + \delta$$
(4.11)

$$\geq \bar{x}(t) + \delta \tag{4.12}$$

Thus, either $\bar{x}(t+1) > \bar{x}(t)$ or $\bar{x}(t+1) = \hat{x}_i(t+1) = s_i$ holds. From (4.6), there exists a T such that, $\bar{x}(t) \ge s_{\min}$ for $t \ge T$.

Define a set S_{\min} comprising the nodes whose maximum values are s_{\min} ,

$$\mathcal{S}_{\min} = \{ i \in V | s_i = s_{\min} \}.$$

$$(4.13)$$

From Assumption 6, $S_{\min} \neq \emptyset$. The next lemma shows that states of nodes in S_{\min} will converge after a certain time.

Lemma 4.3.2. Consider (4.1), (4.2) and (4.3) under Assumption 6, M, s_{\min} , T and S_{\min} defined in (4.3), (4.6), Lemma 4.3.1 and (4.13), respectively. There exists a $T_s \leq \max\{0, \lceil \frac{M-s_{\min}}{\delta} \rceil\} + 1$ such that, $\forall i \in S_{\min}, \hat{x}_i(t) = s_{\min}$ for $t \geq T + T_s + 1$.

Proof. With Lemma 4.3.1, $\forall i \in V$, $\hat{x}_i(t) \geq s_{\min}$ for $t \geq T$. Consider $i \in S_{\min}$, suppose $i \in \mathcal{E}(T)$, i.e. the first bullet of (4.3) applies. Then from (4.3) and (2.7), $\hat{x}_i(t) > M$ for some $T \leq t \leq T + T_s$. By definition, $i \in \mathcal{A}(t+1)$. From (4.2) and (4.1), $\hat{x}_i(t+1)$

obeys

$$\hat{x}_{i}(t+1) = \tilde{x}_{i}(t+1) = \min\left\{\min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{x}_{k}(t), e_{ik}\right) \right\}, s_{i} \right\} = s_{i},$$
(4.14)

where (4.14) uses the fact the $\forall k \in V$, $\hat{x}_k(t) \geq s_{\min} = s_i$ for $t \geq T$ and $f(\cdot, \cdot)$ is progressive. Then as

$$\tilde{x}_i(t+2) = \min\left\{\min_{k\in\mathcal{N}(i)}\left\{f\left(\hat{x}_k(t+1), e_{ik}\right)\right\}, s_i\right\}$$
$$= s_i$$

 $|\hat{x}_i(t+1) - \tilde{x}_i(t+2)| \le D$, from (4.3), $\hat{x}_i(t+2) = s_i$. A simple induction then proves that $\hat{x}_i(t) = s_i$ for $t \ge T + T_s + 1$.

We now explore some properties of x_{ij} defined in Definition 4.3.

Lemma 4.3.3. Consider x_{ij} defined in Definition 4.3, suppose $s_j \neq \infty$ and there exists a k such that, $x_{jk} < s_j$, then $\forall i \in V$, $x_{ik} < x_{ij}$.

Proof. From Definition 4.3, $s_k \neq \infty$. Suppose x_{jk} is obtained from the path $k = l_0 \rightarrow l_1 \rightarrow \cdots, \rightarrow l_L = j$, while x_{ij} is obtained from the path $j = l_L \rightarrow l_{L+1} \rightarrow \cdots, \rightarrow l_{L'} = i$, then there exists a path $k = l_0 \rightarrow l_1 \rightarrow \cdots, \rightarrow l_L \rightarrow \cdots, \rightarrow l'_L = i$ from k to i such that, i will obtain an value x_{ik}^* using (4.7). From the definition of $x_{ik}, x_{ik}^* \ge x_{ik}$. Further, as $x_{jk} < s_j$, from (4.7), $x_{ik}^* < x_{ij}$, leading to $x_{ik} < x_{ij}$.

Lemma 4.3.4. Consider x_{ij} defined in Definition 4.3, suppose $s_j \neq \infty$, then $x_{ij} > s_j$.

Proof. This is a direct consequence of the progressive property of $f(\cdot, \cdot)$.

Define \mathcal{S}_{∞} as the following:

$$\mathcal{S}_{\infty} = \{ i \in V | x_{ij} \ge s_i, \ \forall j \in V \setminus \{i\} \}$$

$$(4.15)$$

From Lemma 4.3.4, with s_{\min} and S_{\min} defined in (4.6) and (4.13), respectively, there holds:

$$i \in \mathcal{S}_{\min} \implies i \in \mathcal{S}_{\infty},$$
 (4.16)

which also indicates that $S_{\infty} \neq \emptyset$ from Lemma 4.3.2. We introduce the following lemma.

Lemma 4.3.5. Consider x_{ij} defined in Definition 4.3, then $\forall i \notin S_{\infty}$, there exists a $j \in V$ such that, x_{ij} is finite.

Proof. From Assumption 6, $S_{\min} \neq \emptyset$. Consider $j \in S_{\min} \subseteq V$, as the graph is connected, with (4.7) and the progressive property of $f(\cdot, \cdot)$, $\forall i \in V \setminus S_{\infty}$, x_{ij} is finite.

Lemma 4.3.5 implies that $s_i \neq \infty, \forall i \in S_{\infty}$. The next lemma shows a relation between nodes in S_{∞} and those are not.

Lemma 4.3.6. Consider x_{ij} and S_{∞} defined in Definition 4.3 and (4.15), respectively. Suppose $i \notin S_{\infty}$, then there exists a $j \in S_{\infty}$ such that, $x_{ij} < s_i$.

Proof. We prove by contradiction. Without loss of generality, consider $0 \notin S_{\infty}$, suppose $\forall j \in S_{\infty}, x_{0j} > s_0$. As $0 \notin S_{\infty}$, from (4.15), there exists a $1 \in V \setminus S_{\infty}$ such that, $x_{01} < s_0$. Further, as $1 \notin S_{\infty}$, there exists a $2 \notin S_{\infty}$ such that, $x_{12} < s_1$, leading to $x_{02} < x_{01} < s_0$ as a result of Lemma 4.3.3. However, as $|V \setminus S_{\infty}|$ is finite, there must exists a $k + 1 \in \{0, 1, \dots, k\}$ such that $x_{kk+1} < s_k$ and $x_{ii+1} < s_i$ for $i \in \{0, 1, \dots, k\}$.

Obviously, $k + 1 \neq k$, suppose $k + 1 = k' \in \{0, 1, \dots, k - 1\}$, then

$$s_{k'} = s_{k+1} < x_{kk+1} < s_k. ag{4.17}$$

Now we claim and prove by induction that

$$x_{lk} < s_l, \ \forall l \in \{k', k'+1, \cdots, k-1\}.$$
 (4.18)

(4.18) holds for l = k - 1. Suppose (4.18) holds for some $l \in \{k' + 1, \dots, k - 1\}$, as $x_{l-1,l} < s_{l-1}$ and $x_{lk} < s_l$ by our induction hypothesis, from Lemma 4.3.3, $x_{l-1,k} < x_{l-1,l} < s_{l-1}$, and hence (4.18) holds and $x_{k'k} < s_{k'}$. Further, by definition of $x_{k'k}$, it follows $s_k < x_{k'k}$, leading to $s_k < s_{k'}$, which contradicts (4.17).

Now we aforehand define the stationary point of (4.1) $x = [x_1, \ldots, x_N]^T$ as the following:

$$x_{i} = \begin{cases} s_{i} & i \in \mathcal{S}_{\infty} \\ \\ \min_{k \in \mathcal{N}(i)} \{f(x_{k}, e_{ik})\} & i \notin \mathcal{S}_{\infty} \end{cases}$$
(4.19)

with \mathcal{S}_{∞} defined in (4.15).

From Definition 4.3 and Lemma 4.3.6, for $i \notin S_{\infty}$, there exists a $j \in S_{\infty}$ such that,

$$x_{ij} = x_i. (4.20)$$

Further from (4.6), x_i obeys

$$s_{\min} \le x_i \le s_i, \ \forall i \in V. \tag{4.21}$$

With (4.19), we make the following definitions.

Definition 4.4. In (4.19), if $x_i = s_i$, then we say that *i* is its own true constraining node. Otherwise, any minimizing *k* in the second bullet of (4.19) is a true constraining node of *i*. As *i* may have more than one true constraining node, its set of true constraining nodes is designated as C(i).

Definition 4.5. We call a path from a node i to $j \in S_{\infty}$ a shortest path, if it starts at i, ends with $j \in S_{\infty}$ and each node in the path is a true constraining node of its predecessor. We call a shortest path from i, the longest shortest path if it has the most nodes among all shortest paths of i. The set \mathcal{F}_i is the set of nodes whose longest shortest paths to the source have i + 1 nodes. We call $\mathcal{D}(\mathcal{G})$ the effective diameter of \mathcal{G} if the longest shortest path among all $i \in V$ has $\mathcal{D}(\mathcal{G})$ nodes.

Thus if a node *i* has two shortest paths one has two and the other has three nodes, then $i \notin \mathcal{F}_1$ but $i \in \mathcal{F}_2$. From Definition 4.5, \mathcal{F}_i follows

$$\mathcal{F}_0 = \mathcal{S}_{\infty}. \ \mathcal{F}_i \neq \emptyset, \ i \in \{0, 1, \cdots, \mathcal{D}(\mathcal{G}) - 1\}.$$

$$(4.22)$$

Also, each node in \mathcal{F}_{i+1} has a true constraining node in \mathcal{F}_i :

$$\mathcal{C}(j) \bigcap \mathcal{F}_i \neq \emptyset, \ \forall j \in \mathcal{F}_{i+1}.$$
(4.23)

The effective diameter is always finite, per the following:

Lemma 4.3.7. Under Assumption 6, $\mathcal{D}(\mathcal{G})$ defined in Definition 4.5 is finite.

Proof. As defined in Definition 4.5, consider a sequence of nodes k_i in \mathcal{G} such that, k_{i-1} is a true constraining node of k_i . Since there are only a finite number of nodes in the graph, the only way that $\mathcal{D}(\mathcal{G})$ can be infinite is if for some i > j, $k_i = k_j$. From the progressive property of $f(\cdot, \cdot)$, this leads to the contradiction:

$$x_{k_i} > x_{k_j} = x_{k_i}.$$
 (4.24)

The next section proves the global uniform and asymptotic convergence of (4.1).

4.4 Global uniform and asymptotic convergence

Define two time varying sets $\mathcal{U}(t)$ and $\mathcal{R}(t)$ requiring recursive definitions below.

Definition 4.6. The set $\mathcal{R}(t+1) = \mathcal{S}(t+1) \bigcup P(t+1)$ with $\mathcal{S}(t+1)$ defined in (4.5) and P(t+1) comprising nodes constrained at time t+1 by a member of $\mathcal{R}(t)$ is a set of nodes rooted to the source. Further $\mathcal{R}(t_0) = \mathcal{S}(t_0)$. Correspondingly, we define a set $\mathcal{U}(t)$ as unrooted to the source if $\mathcal{U}(t_0) = V \setminus \mathcal{S}(t_0)$, and $\mathcal{U}(t+1)$ comprises nodes whose current constraining nodes at t+1, are not themselves and in $\mathcal{U}(t)$.

Observe from the definition of the unrooted set that

$$\mathcal{U}(t) = \emptyset \implies \mathcal{U}(t+1) = \emptyset. \tag{4.25}$$

The sets presented in Definition 4.1 and Definition 4.6 are exemplified through GABF in Figure 4.1. In this case, $M = 4, \delta = 1$ and D = 0. $\mathcal{S}(0) = \{1, 5\}$ as $\hat{x}_i(0) = 0 = s_i$ for i = 1 or 5. At t = 1, $\tilde{x}_2(1) = \hat{x}_1(0) + e_{12} = \hat{x}_2(0) = 1$, as D = 0, node 2 will take 1 as the current constraining node and use the first bullet of (2.6) to update its estimate, leading to $2 \in \mathcal{A}(1) \cap \mathcal{R}(1)$. Meanwhile, as $\tilde{x}_4(1) = \hat{x}_5(0) + e_{45} = 1 \neq \hat{x}_4(0)$, D = 0 and $\hat{x}_4(0) < M$, node 4 will update its estimate using the second bullet of (4.3) and take itself as the current constraining node, then $4 \in \mathcal{E}(1) \cap \mathcal{U}(1)$.

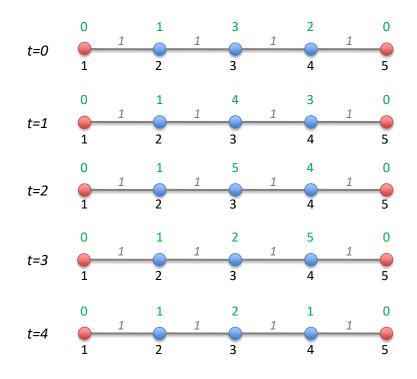


Figure 4.1: Illustration of sets $\mathcal{A}(t), \mathcal{E}(t), \mathcal{R}(t)$ and $\mathcal{U}(t)$. Each edge length in the graph is 1, $M = 4, \delta = 1, D = 0, s_i = 0$ for i = 1, 5 and $s_i = \infty$ for i = 2, 3, 4. In this case, $\mathcal{R}(1) = \mathcal{A}(1) = \{1, 2, 5\}, \mathcal{U}(1) = \mathcal{E}(1) = \{3, 4\}.$

Lemma 4.4.1 shows that $\mathcal{U}(t)$ and $\mathcal{R}(t)$, partition V.

Lemma 4.4.1. Consider (4.1), (4.2) and (4.3) under Assumption 6, with $\mathcal{U}(t)$ and $\mathcal{R}(t)$ defined in Definition 4.6, then for all t,

$$\mathcal{U}(t) \bigcup \mathcal{R}(t) = V \text{ and } \mathcal{U}(t) \bigcap \mathcal{R}(t) = \emptyset.$$
 (4.26)

Proof. Since at each t a node has only one current constraining node, $\mathcal{U}(t)$ and $\mathcal{R}(t)$ are disjoint. We prove (4.26) by induction. From Definition 4.4.1, (4.26) holds for $t = t_0$. Suppose (4.26) holds for some $t = t_1$. Consider $t = t_1 + 1$. Every node i has a current constraining node at $t_1 + 1$. By the induction hypothesis this is either in $\mathcal{R}(t)$ or in $\mathcal{U}(t)$. In the former case $i \in \mathcal{R}(t+1)$. In the latter case $i \in \mathcal{U}(t+1)$. \Box

Define the following function:

$$\hat{x}_{\min}(t) = \min_{j \in \mathcal{U}(t)} \{ \hat{x}_j(t) \} \text{ if } \mathcal{U}(t) \neq \emptyset.$$
(4.27)

The next lemma shows that the lower bound of $\hat{x}_i(t)$ with $i \in \mathcal{U}(t)$ strictly increases.

Lemma 4.4.2. Consider (4.1), (4.2) and (4.3) under Assumption 6, with $\mathcal{A}(t)$, $\mathcal{E}(t)$, $\mathcal{U}(t)$ and $\hat{x}_{\min}(t)$ defined in Definition 4.1, Definition 4.6 and (4.27), respectively. The following holds while the set $\mathcal{U}(t) \neq \emptyset$:

$$\hat{x}_i(t) \ge \hat{x}_{\min}(t_0) + \min\{\sigma, \delta\}(t - t_0), \ \forall i \in \mathcal{U}(t).$$

$$(4.28)$$

Proof. We prove by induction. (4.28) holds for $t = t_0$. Suppose (4.28) holds for $t = t_1$. If $\mathcal{U}(t_1+1)$ is nonempty, then from (4.25) so is $\mathcal{U}(t_1)$. Suppose $\hat{x}_i(t_1+1) = \hat{x}_{\min}(t_1+1)$, by definition, j the current constraining node of $i \in \mathcal{U}(t_1 + 1)$ is in $\mathcal{U}(t_1)$. Thus from the induction hypothesis

$$\hat{x}_j(t_1) \ge \hat{x}_{\min}(t_1) \ge \hat{x}_{\min}(t_0) + \min\{\sigma, \delta\}(t_1 - t_0).$$
 (4.29)

If $i \in \mathcal{E}(t_1 + 1)$, then j = i and from (4.29) and (2.7) we have

$$\hat{x}_{i}(t_{1}+1) = \hat{x}_{\min}(t_{1}+1)$$

 $\geq \hat{x}_{j}(t_{1}) + \delta$
 $\geq \hat{x}_{\min}(t_{0}) + \min\{\sigma, \delta\}(t_{1}+1-t_{0}).$

If $i \in \mathcal{A}(t_1+1)$, then

$$\hat{x}_{i}(t_{1}+1) = \hat{x}_{\min}(t_{1}+1)
= \min\{f(\hat{x}_{j}(t_{1}), e_{ij}), s_{i}\}
= f(\hat{x}_{j}(t_{1}), e_{ij})
\geq \hat{x}_{j}(t_{1}) + \sigma
\geq \min\{\sigma, \delta\}(t_{1}+1-t_{0}).$$
(4.30)

where (4.30) uses the fact that $i \notin \mathcal{R}(t_1 + 1)$ and thus $\hat{x}_i(t_1 + 1) \neq s_i$.

We now show that all estimates of nodes in $\mathcal{R}(t)$ are overestimates.

Lemma 4.4.3. Consider (4.1), (4.2) and (4.3) under Assumption 6, with $\mathcal{A}(t)$, $\mathcal{E}(t)$, $\mathcal{R}(t)$ and x_i defined in Definition 4.1, Definition 4.6 and (4.19), respectively. For $i \in \mathcal{R}(t)$, $\hat{x}_i(t)$ obeys

$$\hat{x}_i(t) \ge x_i. \tag{4.31}$$

Proof. We prove by induction. From Lemma 4.3.2, $\mathcal{R}(t) \neq \emptyset$ for $t \geq T + T_s + 1$ with T and T_s defined in Lemma 4.3.1 and 4.3.2, respectively. Further, with the definition of $\mathcal{R}(t)$, there exists a $s \geq t_0$ such that, $\mathcal{R}(s) = \mathcal{S}(s) \neq \emptyset$, and $\mathcal{R}(t) = \emptyset$ for t < s.

Consider $i \in \mathcal{R}(s) = \mathcal{S}(s)$, as $\hat{x}_i(s) = s_i$, from (4.19) and (4.21), $\hat{x}_i(s) \ge x_i$. Suppose (4.31) holds for $t = t_1 \ge s$. Consider $i \in \mathcal{R}(t_1 + 1)$. If $i \in \mathcal{E}(t_1 + 1)$, from (2.7), $\hat{x}_i(t_1 + 1)$ obeys

$$\hat{x}_i(t_1+1) > \hat{x}_i(t_1) + \delta$$
(4.32)

$$\geq x_i + \delta \tag{4.33}$$

where (4.33) uses $i \in \mathcal{R}(t_1)$ and $\hat{x}_i(t_1) \ge x_1$. If $i \in V \setminus \mathcal{E}(t_1+1) = \mathcal{A}(t_1+1)$, suppose j is the current constraining node of i. As $i \in \mathcal{R}(t_1+1)$, then $j \in \mathcal{R}(t_1)$, by our induction hypothesis, $\hat{x}_j(t_1) \ge x_j$. From (4.2), either j = i and $\hat{x}_i(t_1+1) = s_i \ge x_i$ or

$$\hat{x}_{i}(t_{1}+1) = f(\hat{x}_{j}(t_{1}), e_{ij})$$

$$\geq f(x_{j}, e_{ij})$$

$$\geq x_{i}$$
(4.34)

where equality in (4.34) holds if j is also a true constraining node of i.

Define:

$$x_{\max} = \max_{k \in V} \{x_k\},$$
 (4.35)

and

$$T^* = \left\lceil \frac{x_{\max} - \hat{x}_{\min}(t_0)}{\min\{\sigma, \delta\}} \right\rceil.$$
(4.36)

$$\hat{x}_i(t) \ge x_{\max} \ge x_i, \ \forall i \in \mathcal{U}(t), \ \forall t \ge t_0 + T^*.$$

$$(4.37)$$

Further, as $\mathcal{U}(t)$ and $\mathcal{R}(t)$ partition V, from Lemma 4.4.1,

$$\hat{x}_i(t) \ge x_i, \ \forall i \in V, \ \forall t \ge t_0 + T^*.$$

$$(4.38)$$

We now define the smallest estimate in \mathcal{F}_i as

$$x_{i\min} = \min_{j \in \mathcal{F}_i} \{x_j\}.$$
 (4.39)

From (4.19) and (4.22), we have $x_{0 \min} = s_{\min}$. Define a sequence

$$T_i = \max\left\{0, \left\lceil \frac{M - x_{i\min}}{\delta} \right\rceil\right\} + 1.$$
(4.40)

Now we have the following lemma.

Lemma 4.4.4. Consider (4.1), (4.2) and (4.3) under Assumption 6, \mathcal{F}_i defined in Definition 4.5. Suppose at a time $t_L \geq t_0 + T^*$ with T^* defined in (4.36) and $L \in \{0, 1, \dots, \mathcal{D}(\mathcal{G}) - 2\}$

$$\hat{x}_i(t) = x_i, \ \forall i \in \bigcup_{i=0}^L \mathcal{F}_i, \ \forall t \ge t_L.$$
(4.41)

Then with T_i defined in (4.40), there holds:

$$\hat{x}_i(t) = x_i, \ \forall i \in \bigcup_{i=0}^{L+1} \mathcal{F}_i, \ \forall t \ge t_L + T_{L+1}.$$

$$(4.42)$$

Proof. As $t_L \ge t_0 + T^*$, from (4.38), no underestimates remain at or beyond t_L . Consider $i \in \mathcal{F}_{L+1}$. Suppose $i \in \mathcal{E}(t_L + 1)$, we begin with deriving the lower bound of $\hat{x}_i(t_L)$. We consider two cases. First, if $i \in \mathcal{R}(t_L)$, from Lemma 4.4.3, $\hat{x}_i(t_L) \ge x_i$. We prove by contradiction that in this case $\hat{x}_i(t_L) \neq x_i$. Suppose $\hat{x}_i(t_L) = x_i$, from (4.23), there is a $j \in \mathcal{F}_L$ that is a true constraining node of *i*, then it follows:

$$\tilde{x}_{i}(t_{L}+1) = \min\left\{\min_{k\in\mathcal{N}(i)}\left\{f\left(\hat{x}_{k}(t), e_{ik}\right)\right\}, s_{i}\right\} \\
\geq \min\left\{\min_{k\in\mathcal{N}(i)}\left\{f\left(x_{k}, e_{ik}\right)\right\}, s_{i}\right\}$$
(4.43)

$$= \min \{f(x_j, e_{ij}), s_i\}$$
(4.44)

$$= \min\{x_i, s_i\} \tag{4.45}$$

$$= x_i. (4.46)$$

where (4.43) uses that there are no underestimates at time t_L , (4.44) and (4.45) use the definition of true constraining node, and (4.46) uses (4.21). From (4.41), $\hat{x}_j(t) = x_j$ for $t \ge t_L$, and hence $\tilde{x}_i(t_L + 1) = \hat{x}_i(t_L) = x_i$. With (4.3), $i \in \mathcal{A}(t_L + 1)$, contradicting our assumption. Thus, $\hat{x}_i(t_L) > x_i$ if $i \in \mathcal{R}(t_L)$.

Next we consider $i \in V \setminus \mathcal{R}(t_L) = \mathcal{U}(t_L)$. From (4.37), $\hat{x}_i(t_L) \ge x_{\max} \ge x_i$. If $\hat{x}_i(t_L) = x_i = x_{\max}$, from (4.43) to (4.46), $i \notin \mathcal{E}(t_L + 1)$, leading to a contradiction, hence $\hat{x}_i(t_L) > x_i$ if $i \in \mathcal{U}(t_L)$. As $\mathcal{U}(t)$ and $\mathcal{R}(t)$ partition V, the following holds:

$$\hat{x}_i(t_L) > x_i \ge x_{L+1\min}.\tag{4.47}$$

From (2.7), (4.3) and (4.40), $\hat{x}_i(t) > M$ for some $t_L \leq t_i \leq t_L + T_{L+1} - 1$ and $i \in \mathcal{A}(t_i + 1)$. Thus there is always a $t_L \leq t_i \leq t_L + T_{L+1} - 1$ for which $i \in \mathcal{A}(t_i + 1)$. From (4.23), there is a $j \in \mathcal{F}_L$ that is a true constraining node of i. Suppose k is the

$$\hat{x}_{i}(t_{i}+1) = \min\{f(\hat{x}_{k}(t_{i}), e_{ik}), s_{i}\} \\
\geq \min\{f(x_{k}, e_{ik}), s_{i}\}$$
(4.48)

$$\geq \min\{f(x_j, e_{ij}), s_i\} \tag{4.49}$$

$$\geq \min\{x_i, s_i\} \tag{4.50}$$

$$= x_i \tag{4.51}$$

where (4.48) uses that there are no underestimates, (4.49) uses (4.19), (4.50) uses the fact that j is a true constraining node of i, and (4.51) uses (4.21).

From (4.41), $\hat{x}_j(t) = x_j$ for $t \ge t_L$, it follows

$$\hat{x}_i(t_i+1) = x_i.$$

Then as

$$\begin{aligned} \tilde{x}(t_i + 2) &= \min \left\{ \min_{k \in \mathcal{N}(i)} \{ f(\hat{x}_k(t_i + 1), e_{ik}) \}, s_i \right\} \\ &\geq \min \left\{ \min_{k \in \mathcal{N}(i)} \{ f(x_k, e_{ik}) \}, s_i \right\} \\ &\geq \min \{ f(x_j, e_{ij}), s_i \} \\ &= x_i. \end{aligned}$$

From (4.41), $\tilde{x}_i(t_i+2) = \hat{x}_i(t_i+1)$. With (4.3), we have $\hat{x}_i(t_i+2) = x_i$, and a simple induction proves that $\hat{x}_i(t) = x_i$, $\forall t \ge t_i + 1$.

Define:

$$T_0^* = \max\left\{0, \left\lceil \frac{M - \min\{\delta + s_{\min}, x_{\max}\}}{\delta} \right\rceil\right\} + 2.$$
(4.52)

The theorem below proves the global, uniform and asymptotic stability of (4.1) and provides a tight bound on time to convergence.

Theorem 4.4.5. Consider (4.1), (4.2) and (4.3) under Assumption 6, with T_i , T^* and T_0^* defined in (4.40), (4.36) and (4.52) respectively, for all $i \in V$,

$$\hat{x}_i(t) = x_i, \ \forall t > t_0 + T^* + T_0^* + \sum_{i=1}^{\mathcal{D}(\mathcal{G})-1} T_i.$$
 (4.53)

Further $\hat{x}_i = x_i, \forall i \in V$ is the only stationary point of (4.1).

Proof. Consider $i \in \mathcal{F}_0 = \mathcal{S}_\infty$. Suppose $i \in \mathcal{E}(t_0 + T^* + 1)$. We first derive the lower bound of $\hat{x}_i(t_0 + T^*)$.

Suppose $i \in \mathcal{R}(t_0 + T^*)$. We prove by contradiction that in this case $i \notin \mathcal{A}(t_0 + T^*)$. If $i \in \mathcal{A}(t_0 + T^*)$, from (4.2), $\hat{x}_i(t_0 + T^*) \leq s_i$. Further, as there are no underestimates at or after $t_0 + T^*$ and $i \in \mathcal{S}_\infty$, it follows that $\hat{x}_i(t_0 + T^*) = x_i = s_i$ and $\tilde{x}_i(t_0 + T^* + 1) = \hat{x}_i(t_0 + T^*)$. From (4.3), $i \notin \mathcal{E}(t_0 + T^* + 1)$, leading to a contradiction. Thus, $i \in V \setminus \mathcal{A}(t_0 + T^*) = \mathcal{E}(t_0 + T^*)$. Then as $i \in \mathcal{R}(t_0 + T^*)$, there exists a sequence of nodes i_0, i_1, \cdots, i_L with $L \leq T^* - 1$ such that $i_0 = i$, i_k is a constraining node of i_{k-1} at time $t_0 + T^* - k + 1$, $i_k \in \mathcal{R}(t_0 + T^* - k)$ for $k \in \{1, 2, \cdots, L\}$, and $\hat{x}_{i_L}(t_0 + T^* - L) = s_{i_L}$. Then $\hat{x}_i(t_0 + T^*)$ obeys

$$\hat{x}_i(t_0 + T^*) = \hat{x}_{i_1}(t_0 + T^* - 1) + \delta$$
(4.54)

$$\geq x_{i_1} + \delta \tag{4.55}$$

$$\geq s_{\min} + \delta$$
 (4.56)

where (4.54) uses $i \in \mathcal{E}(t_0 + T^*)$, (4.55) uses $i_1 \in \mathcal{R}(t_0 + T^* - 1)$ and (4.56) uses (4.21). In the case where $i \in \mathcal{U}(t_0 + T^*)$, from (4.37), $\hat{x}_i(t_0 + T^*) \ge x_{\text{max}}$. As $\mathcal{U}(t)$ and $\mathcal{R}(t)$ partition V, $\hat{x}_i(t_0 + T^*)$ obeys

$$\hat{x}_i(t_0 + T^*) \ge \min\{s_{\min} + \delta, x_{\max}\}$$

$$(4.57)$$

Then $\hat{x}_i(t) > M$ for some $t_0 + T^* \le t_i \le t_0 + T^* + T_0^* - 1$ and $i \in \mathcal{A}(t_i + 1)$, from (4.2) and (4.1), $\hat{x}_i(t_i + 1)$ obeys

$$\hat{x}_{i}(t_{i}+1) = \min\{\min_{j \in \mathcal{N}(i)} \{f(\hat{x}_{j}(t_{i}), e_{ij})\}, s_{i}\} \\
\geq \min\{\min_{j \in \mathcal{N}(i)} \{f(x_{j}, e_{ij})\}, s_{i}\}$$
(4.58)

$$= s_i. (4.59)$$

where (4.58) uses the fact that there are no underestimates for $t \ge t_0 + T^*$, (4.59) uses that $i \in S_{\infty}$. Further we have $\tilde{x}_i(t_i + 2) = \hat{x}_i(t_i + 1) = s_i$ and hence $\hat{x}_i(t_i + 2) = s_i$. Repeating the above arguments, we have $\hat{x}_i(t) = x_i$ for $t \ge t_i + 1$. As $\mathcal{F}_0 = \mathcal{S}_{\infty}$, the repeated application of Lemma 4.4.4 proves the result. Thus $\hat{x}_i = x_i, \forall i \in V$ is a stationary point of (4.1). Further, from the definition of x_i in (4.19), the stationary point is unique.

4.5 Robustness under perturbations

In this section, under some additional assumptions, we prove that the behavior of (4.1) is ultimately bounded. The first of these additional assumptions extends the monotonicity property to the second argument of $f(\cdot, \cdot)$ as well. Given that this argument represents edge lengths in most applications, this is an entirely reasonable assumption. As is also standard in most stability analysis, we also impose a Lipschitz condition. **Assumption 7.** The function $f(\cdot, \cdot)$ is monotonic with respect to its second argument, i.e. f(a, b) obeys

$$f(a, b_1) \ge f(a, b_2), \text{ if } b_1 \ge b_2.$$
 (4.60)

Further, there exist $L_i > 0$, such that

$$|f(a,b_1) - f(a,b_2)| \le L_1 |b_1 - b_2| \tag{4.61}$$

and

$$|f(a_1, b) - f(a_2, b)| \le L_2 |a_1 - a_2|$$
(4.62)

The perturbation we consider here is specifically on e_{ij} . Such perturbation could reflect noise or localization error. Specifically,

$$e_{ij}(t) = e_{ij} + \epsilon_{ij}(t) \tag{4.63}$$

with

$$|\epsilon_{ij}(t)| \le \epsilon < e_{\min},\tag{4.64}$$

where e_{\min} is defined in Assumption 6. Notice that the perturbations need not be symmetric, i.e. we permit

$$\epsilon_{ij}(t) \neq \epsilon_{ji}(t). \tag{4.65}$$

In this case, (4.2) becomes

$$\tilde{x}_{i}(t+1) = \min\left\{\min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{x}_{k}(t), e_{ik}(t)\right) \right\}, s_{i} \right\}.$$
(4.66)

Define a function in the following form:

$$W(L_2, D) = \sum_{i=0}^{D-1} L_2^i = \begin{cases} \frac{L_2^D - 1}{L_2 - 1} & L_2 \neq 1\\ D & L_2 = 1. \end{cases}$$
(4.67)

To address the ultimate bound of $\hat{x}_i(t)$, we first define a *shrunken graph* that corresponds the smallest possible values of e_{ij} .

Definition 4.7. Given a graph \mathcal{G} , we define a \mathcal{G}^- , and \mathcal{G}^- is a shrunken version of \mathcal{G} such that, $\forall i \in V$ and $j \in \mathcal{N}(i)$ in \mathcal{G} , e_{ij} becomes e_{ij}^- in \mathcal{G}^- , and e_{ij}^- obeys

$$e_{ij}^{-} = e_{ij} - \epsilon \tag{4.68}$$

with ϵ defined in (4.64). Also consider (4.1) implemented on this shrunken graph, i.e.

$$\hat{X}_i(t+1) = F(\tilde{X}_i(t+1), \hat{X}_i(t), v_i).$$
(4.69)

with $\tilde{X}_i(t+1)$ obeying

$$\tilde{X}_{i}(t+1) = \min\left\{\min_{k\in\mathcal{N}(i)}\left\{f\left(\hat{X}_{k}(t), e_{ik}^{-}\right)\right\}, s_{i}\right\}$$
(4.70)

As \mathcal{G}^- satisfies the same assumptions as \mathcal{G} and is perturbation free, define $X = [X_1, \dots, X_N]$ as a stationary point in \mathcal{G}^- to which (4.69) converges.

Correspondingly, we define X_{ij} , \mathcal{S}_{∞}^{-} and $\mathcal{D}(\mathcal{G}^{-})$ for \mathcal{G}^{-} as we have defined x_{ij} , \mathcal{S}_{∞} and $\mathcal{D}(\mathcal{G})$ for \mathcal{G} in Definition 4.3, (4.15) and Definition 4.5.

From (4.15), (4.19) and (4.21), the stationary point in \mathcal{G}^- obeys

$$X_{i} = \begin{cases} s_{i} & i \in \mathcal{S}_{\infty}^{-} \\ \\ \min_{k \in \mathcal{N}(i)} \{f(X_{k}, e_{ik}^{-})\} & i \notin \mathcal{S}_{\infty}^{-}. \end{cases}$$
(4.71)

with

$$\mathcal{S}_{\infty}^{-} = \{ i \in V | X_{ij} \ge s_i, \ \forall j \in V \setminus \{i\} \},$$

$$(4.72)$$

Similarly, the following holds in \mathcal{G}^- ,

$$s_{\min} \le X_i \le s_i, \ \forall i \in V, \tag{4.73}$$

and

$$\mathcal{S}_{\min} \subseteq \mathcal{S}_{\infty}^{-}.$$
 (4.74)

As $e_{ij}(t) > 0$ for all t under perturbations, the lower bound of $\hat{x}_i(t)$ for $i \in \mathcal{U}(t)$ still increases under perturbations as shown in the following lemma.

Lemma 4.5.1. Consider (4.66), (4.1) and (4.3), with $\mathcal{U}(t)$ and $\hat{x}_{\min}(t)$ defined in Definition 4.6 and (4.27), respectively. The following holds while the set $\mathcal{U}(t) \neq \emptyset$:

$$\hat{x}_i(t) \ge \hat{x}_{\min}(t_0) + \min\{\sigma, \delta\}(t - t_0), \ \forall i \in \mathcal{U}(t).$$

$$(4.75)$$

Proof. The proof is similar to that of Lemma 4.4.2, and thus omitted. \Box

Based on Definition 4.7, we now turn to $\mathcal{R}(t)$, and prove that under perturbations all estimates of nodes in $\mathcal{R}(t)$ are lower bounded by their true estimates in \mathcal{G}^- .

Lemma 4.5.2. Consider (4.66), (4.1) and (4.3) under Assumption 7, with $\mathcal{A}(t)$, $\mathcal{E}(t)$, $\mathcal{R}(t)$ and X_i defined in Definition 4.1, 4.6 and 4.7, respectively. For $i \in \mathcal{R}(t)$, $\hat{x}_i(t)$ obeys

$$\hat{x}_i(t) \ge X_i. \tag{4.76}$$

Proof. We prove by induction. Notice that Lemma 4.3.1 and 4.3.2 still hold under perturbations, leading to $\mathcal{R}(t) \neq \emptyset$ for $t \geq T + T_s + 1$ with T and T_s defined in Lemma

4.3.1 and 4.3.2, respectively. From the definition of $\mathcal{R}(t)$, there exists a $s \ge t_0$ such that, $\mathcal{R}(s) = \mathcal{S}(s)$ and $\mathcal{R}(t) = \emptyset$ for t < s. $\forall i \in \mathcal{R}(s) = \mathcal{S}(s)$, as $\hat{x}_i(s) = s_i$, from (4.73), $\hat{x}_i(s) \ge X_i$. Suppose (4.76) holds for $t = t_1 \ge s$. For $i \in \mathcal{R}(t_1 + 1)$, if $i \in \mathcal{E}(t_1 + 1)$, from (2.7), $\hat{x}_i(t_1 + 1)$ obeys

$$\hat{x}_i(t_1+1) > \hat{x}_i(t_1) + \delta$$
(4.77)

$$\geq X_i + \delta \tag{4.78}$$

where (4.78) uses $i \in \mathcal{R}(t_1)$ and $\hat{x}_i(t_1) \geq X_1$. If $i \in V \setminus \mathcal{E}(t_1 + 1) = \mathcal{A}(t_1 + 1)$, suppose j is the current constraining node of i. As $i \in \mathcal{R}(t_1 + 1)$, then $j \in \mathcal{R}(t_1)$, from our induction hypothesis, $\hat{x}_j(t_1) \geq X_j$. From (4.66) and (4.73), either j = i and $\hat{x}_i(t_1 + 1) = s_i \geq X_i$ or

$$\hat{x}_i(t_1+1) = f(\hat{x}_j(t_1), e_{ij}(t))$$

 $\geq f(X_j, e_{ij})$
(4.79)

$$\geq X_i$$
 (4.80)

where (4.79) uses the fact that $f(\cdot, \cdot)$ is monotonic increasing on both arguments and $e_{ij}(t) \ge e_{ij}^{-}$, and equality in (4.80) holds if j is also a true constraining node of i in \mathcal{G}^{-} .

The next lemma relates the stationary points of \mathcal{G} and \mathcal{G}^- .

Lemma 4.5.3. With X_i and \mathcal{G}^- defined in Definition 4.7, x_i and \mathcal{G} in (4.19) and Assumption 7, ϵ in (4.64), $W(\cdot)$ in (4.67) and $\mathcal{D}(\cdot)$ in Definition 4.5. $\forall i \in V$, $x_i \leq X_i + W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1\epsilon$. Proof. Consider nodes n_1, n_2, \dots, n_T such that $n_1 \in \mathcal{S}_{\infty}^-$, $X_{n_1} = s_{n_1}$, and for all $i \in \{1, \dots, T-1\}$, n_i is a true constraining node of n_{i+1} in \mathcal{G}^- . Each node in \mathcal{G}^- is in one such sequence. As from Definition 4.5, $T \leq \mathcal{D}(\mathcal{G}^-)$, the result will follow if $\forall i \in \{1, \dots, T\}$

$$x_{n_i} - X_{n_i} \le W(L_2, i - 1)L_1\epsilon, \tag{4.81}$$

where $W(L_2, i-1) = 0$ if i = 1. From 4.21, $x_{n_i} \leq s_{n_i}$, (4.81) holds for i = 1. Suppose it holds for some $i \in \{1, \dots, T-1\}$. As n_i and n_{i+1} are neighbors in both \mathcal{G} and \mathcal{G}^- , n_i is a true constraining node of n_{i+1} in \mathcal{G}^- , $x_{n_{i+1}}$ obeys

$$x_{n_{i+1}} \le f(x_{n_i}, e_{n_i n_{i+1}})$$

$$\le f(X_{n_i} + W(L_2, i-1)L_1\epsilon, e_{n_i n_{i+1}})$$
(4.82)

$$\leq f(X_{n_i}, e_{n_i n_{i+1}}) + L_2 W(L_2, i-1) L_1 \epsilon$$
(4.83)

$$= f(X_{n_i}, e_{n_i n_{i+1}}^- + \epsilon) + L_2 W(L_2, i-1) L_1 \epsilon$$
(4.84)

$$\leq f(X_{n_i}, e_{n_i n_{i+1}}) + L_1 \epsilon + L_2 W(L_2, i-1) L_1 \epsilon$$
(4.85)

$$= X_{n_{i+1}} + W(L_1, i)L_1\epsilon$$
(4.86)

where (4.82) uses the induction hypothesis that $x_{n_i} \leq X_{n_i} + W(L_2, i-1)L_1\epsilon$, (4.83) uses (4.62), (4.84) uses (4.68), (4.85) uses (4.61), and (4.86) uses the fact that n_i is a true constraining node of n_{i+1} .

We define the largest true estimate in \mathcal{G}^- as

$$X_{\max} = \max_{k \in V} \{X_k\},$$
 (4.87)

and T^- as

$$T^{-} = \left\lceil \frac{X_{\max} - \hat{x}_{\min}(t_0)}{\min\{\delta, \sigma\}} \right\rceil$$
(4.88)

Thus, from Lemma 4.5.1, we have

$$\hat{x}_i(t) \ge X_{\max}, \ \forall i \in \mathcal{U}(t), \ \forall t \ge t_0 + T^-.$$

$$(4.89)$$

As $\mathcal{U}(t)$ and $\mathcal{R}(t)$ partition V, from Lemma 4.5.2,

$$\hat{x}_i(t) \ge X_i, \ \forall i \in V, \ \forall t \ge t_0 + T^-.$$

$$(4.90)$$

With \mathcal{F}_i defined in Definition 4.5, define $X_{i\min}$ as

$$X_{i\min} = \min_{j \in \mathcal{F}_i} \{X_j\}$$
(4.91)

Define a sequence

$$T_i^- = \max\left\{0, \left\lceil \frac{M - X_{i\min}}{\delta} \right\rceil\right\} + 1.$$
(4.92)

Then we have the following lemma.

Lemma 4.5.4. Consider (4.66), (4.1) and (4.3) under Assumption 7, ϵ defined in (4.64), $W(\cdot)$ defined in (4.67). Suppose D in (4.3) obeys

$$D \ge (W(L_2, \mathcal{D}(\mathcal{G}^-) - 1) + W(L_2, \mathcal{D}(\mathcal{G}) - 1))L_1\epsilon,$$
(4.93)

and at a time $t_L \ge t_0 + T^-$ defined in (4.88), for $L \in \{0, 1, \cdots, \mathcal{D}(\mathcal{G}) - 2\}$

$$\hat{x}_i(t) \le x_i + W(L_2, L)L_1\epsilon, \ \forall i \in \mathcal{F}_L, \ \forall t \ge t_L.$$

$$(4.94)$$

with $W(L_2, L) = 0$ if L = 0. Then with T_i^- define in (4.92), for all $i \in \mathcal{F}_{L+1}$ and $t \ge t_L + T_{L+1}^-$, there holds:

$$x_i - W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1 \epsilon \le \hat{x}_i(t) \le x_i + W(L_2, L+1)L_1 \epsilon.$$
 (4.95)

Proof. As $t_L \ge t_0 + T^-$, from (4.90), $\forall i \in V$, $\hat{x}_i(t) \ge X_i$ for $t \ge t_L$. Further, with Lemma 4.5.3, the left side of (4.95) has been proved.

For the right hand side of (4.95), consider $i \in \mathcal{F}_{L+1}$. Suppose $i \in \mathcal{E}(t_L + 1)$. We first derive the lower bound of $\hat{x}_i(t_L)$. If $i \in \mathcal{R}(t_L)$, from (4.90), $\hat{x}_i(t_L) \ge X_i$. We now prove by contradiction that in this case $\hat{x}_i(t_L) \ne X_i$.

From (4.23), there exists a $j \in \mathcal{F}_L$ that is a true constraining node of i in \mathcal{G} such that,

$$\tilde{x}_{i}(t_{L}+1) = \min\left\{\min_{k \in \mathcal{N}(i)} \left\{f\left(\hat{x}_{k}(t_{L}), e_{ik}(t_{L})\right)\right\}, s_{i}\right\}$$

$$\leq \min\{f(\hat{x}_{j}(t_{L}), e_{ij}(t_{L})), s_{i}\}$$

$$\leq f\left(\hat{x}_{j}(t_{L}), e_{ij} + \epsilon\right)$$
(4.96)

$$\leq f\left(x_j + W(L_2, L)L_1\epsilon, e_{ij}\right) + L_1\epsilon \tag{4.97}$$

$$\leq x_i + L_2 W(L_2, L) L_1 \epsilon + L_1 \epsilon \tag{4.98}$$

$$= x_i + W(L_2, L+1)L_1\epsilon$$
(4.99)

where (4.96) uses (4.63) and (4.64), (4.97) uses (4.61) and (4.94), and (4.98) uses (4.62) and the definition of the true constraining node. Further,

$$\min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{x}_k(t_L), e_{ik}(t_L)\right) \right\} \ge \min_{k \in \mathcal{N}(i)} \left\{ f\left(X_k, e_{ik}^-\right) \right\}$$
(4.100)

$$=X_i \tag{4.101}$$

where (4.100) uses that $x_i(t) \ge X_i, \forall i \in V$ for $t \ge t_L$, $e_{ik}(t_L) \ge e_{ik}^-$ and $f(\cdot, \cdot)$ is monotonic increasing with respect to its two arguments, and (4.101) uses (4.71). From (4.66) and (4.73), (4.101) leads to

$$\tilde{x}_i(t_L+1) \ge X_i \tag{4.102}$$

From (4.102) and (4.99), if $\hat{x}_i(t_L) = X_i$, then it follows

$$\begin{aligned} |\tilde{x}_{i}(t_{L}+1) - \hat{x}_{i}(t_{L})| &\leq |x_{i} + W(L_{2}, L+1)L_{1}\epsilon - X_{i}| \\ &\leq W(L_{2}, \mathcal{D}(\mathcal{G}^{-}) - 1)L_{1}\epsilon + W(L_{2}, L+1)L_{1}\epsilon \quad (4.103) \\ &\leq D \qquad (4.104) \end{aligned}$$

where (4.103) uses Lemma 4.5.3. (4.104) indicates $i \in \mathcal{A}(t_L + 1)$, contradicting our assumption that $i \in \mathcal{E}(t_L + 1)$. Thus $\hat{x}_i(t_L) > X_i$ if $i \in \mathcal{R}(t_L)$.

In the case where $i \in \mathcal{U}(t_L)$. From (4.89), $\hat{x}_i(t_L) \geq X_{\max} \geq X_i$. For $\hat{x}_i(t_L) = X_i$, from (4.97) to (4.104), $i \notin \mathcal{E}(t_L+1)$, leading to a contradiction. Thus, $\hat{x}_i(t_L) > X_i$ if $i \in \mathcal{U}(t_L)$, and the following holds

$$\hat{x}_i(t_L) > X_i \ge X_{L+1\min}, \ \forall i \in \mathcal{F}_{L+1}.$$

$$(4.105)$$

Then from (4.92), $\hat{x}_i(t) > M$ for some $t_L \leq t_i \leq t_L + T_{L+1}^- - 1$ and $i \in \mathcal{A}(t_i + 1)$. From (4.19) and Definition 4.5, there is a $j \in \mathcal{F}_L$ that is a true constraining node of i in \mathcal{G} . From (4.66) and (4.1), $\hat{x}_i(t_i + 1)$ obeys

$$\hat{x}_{i}(t_{i}+1) \leq \min\{f(\hat{x}_{j}(t_{i}), e_{ij}(t_{i})), s_{i}\}$$

$$\leq f(\hat{x}_{j}(t_{i}), e_{ij}(t_{i}))$$
(4.106)

$$\leq f(x_j + W(L_2, L)L_1\epsilon, e_{ij} + \epsilon) \tag{4.107}$$

$$\leq f(x_j, e_{ij}) + L_2 W(L_2, L) L_1 \epsilon + L_1 \epsilon$$
(4.108)

$$\leq x_i + W(L_2, L+1)L_1\epsilon \tag{4.109}$$

where (4.107) uses (4.63), (4.64) and (4.94), (4.108) uses (4.61) and (4.62).

Now we claim and prove by induction that

$$\hat{x}_i(t) \le x_i + W(L_2, L+1)L_1\epsilon, \ \forall t \ge t_i + 1.$$
(4.110)

(4.110) holds for $t = t_i + 1$. Suppose (4.110) holds for some $t_1 \ge t_i + 1$. For $t = t_1 + 1$, suppose j is the true constraining node of i, from (4.2) and (4.1),

$$\tilde{x}_{i}(t_{1}+1) \leq \min\{f(\hat{x}_{j}(t_{1}), e_{ij}(t_{1})), s_{i}\}$$

 $\leq x_{i} + W(L_{2}, L+1)L_{1}\epsilon$
(4.111)

where (4.111) uses (4.106-4.109). As

$$\min_{k \in \mathcal{N}(i)} \left\{ f\left(\hat{x}_{k}(t_{1}), e_{ik}(t_{1})\right) \right\} \geq \min_{k \in \mathcal{N}(i)} \left\{ f\left(X_{k}, e_{ik}^{-}\right) \right\} \\
\geq X_{i}$$
(4.112)

and $X_i \le s_i$ by (4.73). With (4.66)

$$\tilde{x}_i(t_1+1) \ge X_i \tag{4.113}$$

Further, as the left side of (4.95) holds and $\hat{x}_i(t_1) \leq x_i + W(L_2, L+1)L_1\epsilon$ by our induction hypothesis, with D obeying (4.93),

$$\begin{aligned} |\tilde{x}_i(t_1+1) - \hat{x}_i(t_1)| &\leq |x_i + W(L_2, L+1)L_1\epsilon - X_i| \\ &= W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1\epsilon + \\ &\qquad W(L_2, L+1)L_1\epsilon \\ &\leq D. \end{aligned}$$

From (4.3), $\hat{x}_i(t_1+1) = \tilde{x}_i(t_1+1) \le x_i + W(L_2, L+1)L_1\epsilon$, and our assertion follows. \Box

The next theorem proves that the generalized G block is ultimately bounded under bounded persistent perturbations and provides an upper bound on time to attain it.

Theorem 4.5.5. Consider (4.66), (4.1) and (4.3) under Assumption 7, $\mathcal{D}(\mathcal{G})$ and $\mathcal{D}(\mathcal{G}^-)$ defined in Definition 4.5 and 4.7, ϵ in (4.64), T^- in (4.88) and T_i^- in (4.92). Suppose D in (4.3) obeys

$$D \ge (W(L_2, \mathcal{D}(\mathcal{G}^-) - 1) + W(L_2, \mathcal{D}(\mathcal{G}) - 1))L_1\epsilon, \qquad (4.114)$$

with $W(\cdot, \cdot)$ defined in (4.67), then for all $i \in V$ and $t \ge t_0 + T^- + \overline{T}_0 + \sum_{i=1}^{\mathcal{D}(\mathcal{G})-1} T_i^-$,

$$x_i - W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1\epsilon \le \hat{x}_i(t) \le x_i + W(L_2, \mathcal{D}(\mathcal{G}) - 1)L_1\epsilon$$
(4.115)

Proof. From (4.90), $\forall i \in V, \hat{x}_i(t) \geq X_i$ for $t \geq T^-$. With Lemma 4.5.3, the left side of (4.115) has been proved.

Consider $i \in \mathcal{F}_0 = \mathcal{S}_\infty$. Suppose $i \in \mathcal{E}(t_0 + T^- + 1)$. We first address the lower bound of $\hat{x}_i(t_0 + T^-)$. If $i \in \mathcal{R}(t_0 + T^-)$, we prove by contradiction that in this case $i \notin \mathcal{A}(t_0 + T^-)$. If $i \in \mathcal{A}(t_0 + T^-)$, from (4.66) and Lemma 4.5.2, $\hat{x}_i(t_0 + T^-)$ obeys

$$X_i \le \hat{x}_i (t_0 + T^-) \le s_i \tag{4.116}$$

Further,

$$\min_{k \in \mathcal{N}(i)} f\left(\hat{x}_{k}(t_{0} + T^{-}), e_{ik}(t_{0} + T^{-})\right) \\
\geq \min_{k \in \mathcal{N}(i)} f\left(X_{k}, e_{ik}^{-}\right)$$
(4.117)

$$=X_i \tag{4.118}$$

where (4.117) uses that $\hat{x}_i(t) \ge X_i$ for $t \ge t_0 + T^-$ and all $i \in V$, $e_{ik}(t) \ge e_{ik}^-$ for all t and $f(\cdot, \cdot)$ is monotonic increasing with respect to its two arguments, and (4.118) uses (4.71). As $X_i \le s_i$ for all $i \in V$, from (4.66), $\tilde{x}_i(t_0 + T^- + 1)$ obeys

$$X_i \le \tilde{x}_i (t_0 + T^- + 1) \le s_i. \tag{4.119}$$

As $i \in \mathcal{S}_{\infty}$, $x_i = s_i$, (4.116) and (4.119) imply that

$$\begin{aligned} |\tilde{x}_i(t_0 + T^- + 1) - \hat{x}_i(t_0 + T^-)| &\leq x_i - X_i \\ &\leq W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1 \epsilon \\ &\leq D \end{aligned}$$

leading to $i \notin \mathcal{E}(t_0 + T^- + 1)$, contradicting our assumption. Thus $i \in V \setminus \mathcal{A}(t_0 + T^-) = \mathcal{E}(t_0 + T^-)$. As $i \in \mathcal{R}(t_0 + T^-)$ as we have assumed, there exists a sequence of nodes i_0, i_1, \cdots, i_L with $L \leq T^- - 1$ such that $i_0 = i, i_k$ is a constraining node of i_{k-1} at time $t_0 + T^- - k + 1, i_k \in \mathcal{R}(t_0 + T^- - k)$ for $k \in \{1, 2, \cdots, L\}$. Then $\hat{x}_i(t_0 + T^-)$ obeys

$$\hat{x}_{i}(t_{0} + T^{-}) = \hat{x}_{i_{1}}(t_{0} + T^{-} - 1) + \delta$$

 $\geq X_{i_{1}} + \delta$
(4.120)

$$\geq s_{\min} + \delta$$
 (4.121)

where (4.120) uses $i_1 \in \mathcal{R}(t_0 + T^- - 1)$ and (4.121) uses (4.73).

In the case where $i \in \mathcal{U}(t_0 + T^-)$, from (4.89), $\hat{x}_i(t_0 + T^-) \geq X_{\text{max}}$. As $\mathcal{U}(t) \cup \mathcal{R}(t) = V$, the following holds:

$$\hat{x}_i(t_0 + T^-) \ge \min\{X_{\max}, s_{\min} + \delta\}.$$
 (4.122)

From (4.3) and (2.7), $\hat{x}_i(t) > M$ for some $t_0 + T^- \le t_i \le t_0 + T^- + \overline{T}_0 - 1$ and $i \in \mathcal{A}(t_i + 1)$. $\hat{x}_i(t_i + 1)$ obeys

$$\hat{x}_i(t_i + 1) = \min\{f(\hat{x}_j(t_i), e_{ij}(t)), s_i\}$$
$$\leq s_i$$
$$= x_i$$

Now we prove by induction that

$$\hat{x}_i(t) \le s_i = x_i, \ \forall t \ge t_i + 1.$$
 (4.123)

(4.123) holds for $t = t_i + 1$. Suppose (4.123) holds for some $t_1 \ge t_i + 1$. For $t = t_1 + 1$, from (4.117) to (4.119), the following holds:

$$X_i \le \tilde{x}_i(t_1 + 1) \le x_i \tag{4.124}$$

Further, as $\hat{x}_i(t_1) \leq x_i$ by our induction hypothesis and the left side of (4.115) holds,

$$\begin{aligned} |\tilde{x}_i(t_1+1) - \hat{x}_i(t_1)| &\leq |x_i - X_i| \\ &= W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1\epsilon \\ &\leq D \end{aligned}$$

From (4.3), $\hat{x}_i(t_1+1) = \tilde{x}_i(t_1+1) \leq s_i$. Thus, $\forall i \in \mathcal{F}_0$, the following holds:

$$x_i - W(L_2, \mathcal{D}(\mathcal{G}^-) - 1)L_1 \epsilon \le \hat{x}_i(t) \le x_i, \ \forall t \ge t_0 + T^- + T_0^-.$$
(4.125)

Then the repeated application of Lemma 4.5.4 proves our result.

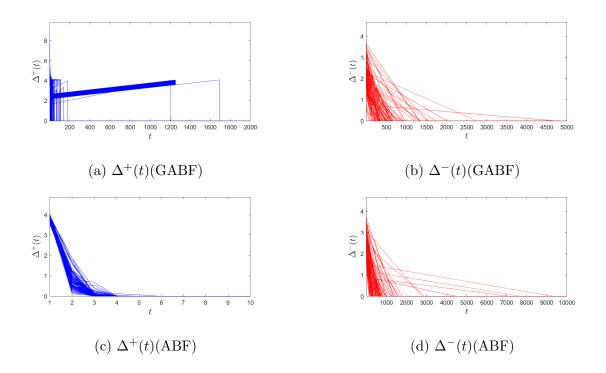


Figure 4.2: Comparison of time to convergence between GABF and ABF for 100 runs of 500 nodes randomly distributed in a $4 \times 1 \text{ km}^2$ area, showing that GABF converges faster than ABF.

4.6 Simulations

In this section, we empirically confirm the results presented in the prior sections through three scenarios in simulation. Specifically, we compare the performance of ABF and GABF. GABF follows (2.18) defined in Section 2.4, while ABF follows (2.13) defined in Section 2.3. ABF and GABF are both instances of (4.1). While in ABF, M in (4.3) is set to be $-\infty$, GABF has no restrictions on M.

In this case, 500 nodes, one of which is a source, are randomly distributed in a $4 \times 1 \text{ km}^2$ area, each node communicate over a 0.25 km radius and runs simultaneously.

The simulation is run 100 times. For perturbations, we consider static nodes with asymmetric noise in the estimated e_{ij} , then edge lengths change from their nominal value e_{ij} as $\bar{e}_{ij}(t) = e_{ij} + \epsilon_{ij}(t)$, and $\epsilon_{ij}(t)$ obeys

$$|\epsilon_{ij}(t)| < \epsilon < e_{\min},\tag{4.126}$$

ensuring no edge length is negative. Further, the noise is asymmetric, i.e., $e_{ij}(t) \neq e_{ji}(t)$. In all simulations, $\epsilon_{ij} \in \mathcal{U}(-\epsilon, \epsilon)$ with $\epsilon = e_{\min}$. D, M and δ in GABF are set as $D = (\mathcal{D}(\mathcal{G}) + \mathcal{D}(\mathcal{G}^-) - 2)\epsilon$ and $M = \delta = \sqrt{17}$ km, in which case $\Delta^+(t)$ and $\Delta^-(t)$ in GABF will have the same ultimate bounds as in ABF according to our Theorem 4.5.5.

We first compare their performance without perturbations. From Figure 4.2 (c) and (d), while $\Delta^+(t)$ in ABF converges within $\mathcal{D}(\mathcal{G})$ rounds, $\Delta^-(t)$, constrained by the "rising value problem", takes much longer to converge, and the average convergence time for all 100 trials is 946 rounds. In GABF, as shown in Figure 4.2 (a) and (b), the "rising value problem" is alleviated, and the average convergence time is 616 rounds.

For the robustness of GABF and ABF under perturbations. We randomly pick 5 trials and apply the perturbations described above. As shown in Figure 4.3, $\Delta^+(t)$ and $\Delta^-(t)$ in both ABF and GABF will have the same ultimate bounds, which confirms our Theorem 4.5.5.

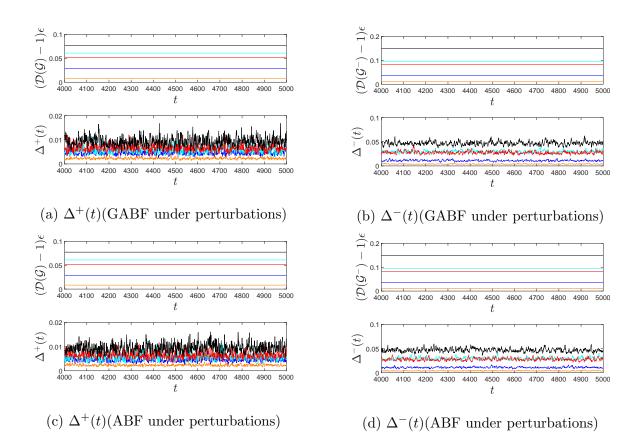


Figure 4.3: Comparison of robustness between GABF and ABF for 100 runs of 500 nodes randomly distributed in a $4 \times 1 \text{ km}^2$ area, showing that both GABF and ABF have the same ultimate bounds.

4.7 Conclusion

In this chapter, we provide a global uniform asymptotic stability of the generalized G block. With an additional Lipschitz condition on the update kernel, we have also established ultimate bounds in face of persistent perturbations. The analysis for the generalized G block, as well as the analyses for its specific variants, will serve to provide design insights in interconnections involving other basis blocks,

CHAPTER 5 ANALYSIS OF COMPOSITIONS OF BASIS BLOCKS

5.1 Introduction

Having completed the stability analysis of G blocks in both specialized and general settings, we now turn to compositions, by considering two examples. While in the context of this thesis this serves the purpose of illustrating the power of stability analysis framework to the analysis of such compositions, these examples are of independent interest. The first, a G - C combination, serves to estimate such collective states in a network of devices, like the total resources available to the network. The other is a *feedback* interconnection of G and C blocks for leader election. Though leader election is a well studied topic, intriguingly this feedback combination of Aggregate Computing basis blocks provides an algorithm that has certain superior attributes to others in the literature. For the G - C combination we analyze the error bounds and dynamics. For leader election we characterize the conditions for its GUAS and demonstrate its resilience. The analysis of the leader election algorithm in particular serves as the first example of analysis of feedback compositions of Aggregate Computing basis blocks.

Sections 5.2 and 5.3 respectively consider the G - C combination and leader election.

5.2 A G-C combination

We first analyze the G-C combination, which is a state estimation algorithm based on spanning trees. Efficient resilient algorithms have been found for special cases of collective state estimation, such as the use of gossip for estimating monotonic functions (e.g. [19], [65] and [62]). Here we focus our study on spanning-tree aggregation, one of the most frequently used approaches to collective state estimation(e.g., [55, 69, 54]). Amongst the many variants of spanning tree distributed state estimation, we focus on one introduced in [16]: a G block is used to construct a spanning tree, where each node chooses its parent as its current constraining node, and the Cblock accumulates resource estimates on this spanning tree as the following:

$$O_i(t+1) = O_i(t) + \sum_{j \in \mathcal{C}_i(t+1)} O_j(t)$$
(5.1)

where $C_i(t)$ denotes the nodes *i* constrains at time *t* and $O_i(t)$ is the resource estimate of node *i* at time *t*. In other words, under *C* block, each node accumulates resource estimates from all the nodes it constrains, and combines with its own estimate to be further collected by its constraining node. In this section, we consider a *C* block using summation. For a general *C* block, it may use other commutative and associative operations including idempotent ones.

In an *aggregate computing* setting, the G and C blocks run simultaneously, the latter does not wait for the spanning tree construction to be completed before it starts collecting.

5.2.1 Error bounds and dynamics

In this section, we first analyze the worst-case errors in the collective state estimate that can be produced by the C block, then extend to consider the interaction of C block with the distance estimation algorithms that may be used to produce its potential input.

We first introduce two definitions:

Definition 5.1. Given a spanning tree T, a node is said to be at level i if it is i hops away from the source node.

Definition 5.2. Given two nodes a and b in a spanning tree, if a and b are connected and b is one level higher than a, then a is defined as the parent of b and b as the child of a.

By definition of a spanning tree at a given iteration a node can have at most one parent. We consider the perturbation of estimates computed by the C block caused by a spanning tree that varies with time. Specifically, in each cycle, the parent of a node may change. In such a case the child may inform both the new and the old parents when it transmits its estimate. Cycles are not synchronized, so updates can occur in any order. As such:

A sub-tree estimate is duplicated if node a switches parent from b to c, and the nodes update in order b < a < c, since b sends an estimate including a as a child, then a notifies b and c of the switch, and finally c sends an estimate also including a as a child.

A sub-tree estimate is lost if node a switches parent from b to c and the nodes update in order c < a < b, since c sends an estimate without a as a child, then a notifies b and c of the switch, and finally b sends an estimate also without a as a child.

The following assumption holds in our analysis.

Assumption 8. Under the duplicating and loss perturbations described above, we assume the level of a node does not change.

We will evaluate duplication and loss potential by considering the case where the C block is applied to count devices, i.e., summing a local constant of 1 from each node. First, we have the following lemma:

Lemma 5.2.1. Suppose Assumption 8 holds. Under duplicating perturbation, if there is more than one node at level *i*, then the sum of transmitted values O_j of all nodes $j \in S_i$, where S_i denotes the set of nodes at level *i*, satisfies

$$\sum_{j \in S_i} O_j \le 2 \sum_{j \in S_{i+1}} O_j + |S_i|$$

Proof. Suppose in this case, there are n nodes at level i, and the number of nodes with decreased values in the current cycle is m. Then we have

$$\sum_{j \in S_i} O_j = \sum_{j=1}^m O_j + \sum_{j=m+1}^n O_j$$

$$\leq \sum_{j \in S_{i+1}} O_j + m + \sum_{j=m+1}^n O_j$$
(5.2)

$$\leq \sum_{j \in S_{i+1}} O_j + m + \sum_{j \in S_{i+1}} O_j + n - m$$

$$= 2 \sum_{j \in S_{i+1}} O_j + |S_j|$$
(5.3)

$$= 2 \sum_{j \in S_{i+1}} O_j + |S_i|$$

Equality in (5.2) holds when in the previous cycle, all the nodes at level i + 1 are children of those m nodes at level i while the remaining n - m nodes at level i have no children. Equality in (5.3) holds when those m nodes lose all their children and all their children choose the remaining n - m nodes as their parents in the current cycle.

Based on Lemma 5.2.1, we have the following theorem.

Theorem 5.2.2. Suppose Assumption 8 holds. Under the perturbation described above, for a spanning tree with n nodes, value of the source node satisfies the tight bound

$$O_s \leq \begin{cases} 2^{\frac{n+1}{2}} - 1 & n \text{ is odd} \\ \\ 2^{\frac{n}{2}} + 2^{\frac{n}{2}-1} - 1 & n \text{ is even} \end{cases}$$

where O_s denotes the value of source node.

Proof. First, we consider the case that all levels in the spanning tree have more than one node, then we consider the case that some or all levels have only one node, and show that value of the source node resulted from the former case will be larger than that in the latter case.

For the former case where each level has more than one node, when n is odd, suppose value of the source node is maximized when there are l levels in the spanning tree(here we assume the bottom level is level l), according to Lemma 5.2.1, O_s satisfies

$$O_s \le \sum_{i=1}^l 2^{i-1} |S_i| + 1 \tag{5.4}$$

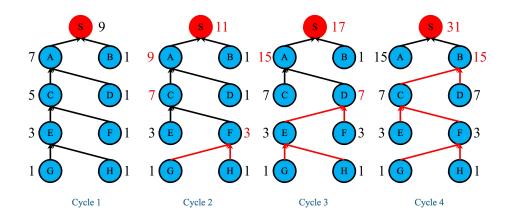


Figure 5.1: Example of achieving the maximum source value (adapted from [56]).

where S_i denotes the set comprised of the nodes at level i.

As we can see from (5.4), the sum grows exponentially with the number of levels and grows linearly with the number of nodes at each level. Thus, value of the source node will be maximized by maximizing the number of levels, which means that each level should have two nodes and l = (n - 1)/2, then value of the source node satisfies

$$O_s \leq \sum_{i=1}^{(n-1)/2} 2^{i-1} |S_i| + 1$$

= $2^{\frac{n+1}{2}} - 1$ (5.5)

where $|S_i| = 2$ for i = 1, 2, ..., l.

In (5.5), we assume that all values of nodes transmitted from higher levels to lower levels all get doubled. This is a tight bound, and an example is shown in Figure 5.1. As shown in Figure 5.1, all nodes in the same level share the same parent in every cycle, and nodes at the highest level change their common parent back and forth from the second cycle, nodes at the second highest cycle follow this from the third cycle, and so on. The source value grows exponentially and finally achieves its maximum value defined in (5.5).

If the number of nodes in the graph is even and value of the source node is maximized when there are l levels in the spanning tree. Since the number of nodes is even, there will be one level with a single node. The question is that where to put this single node in order to achieve the maximum value of the source node. Here we claim that value of the source node will be maximized when the single node is at the highest level. The proof (omitted due to space constraints) follows similar reasoning as before, in essence finding that it is always better to replace two layers of single nodes with one layer of two nodes, and that such replacements are better at lower layers.

Next, we show the lower bound of value of the source node. First, we provide the following lemma.

Lemma 5.2.3. Suppose Assumption 8 holds. Under loss perturbation, if there is more than one node at level i, then the sum of values of all nodes transmitted from level i satisfies

$$\sum_{j \in S_i} O_j \ge |S_i|$$

Proof. Suppose in this case, there are n nodes at level i, and the number of nodes

with increased values in the current cycle is m. Then we have

$$\sum_{j \in S_i} O_j = \sum_{j=1}^m O_j + \sum_{j=m+1}^n O_j$$

$$\geq m + \sum_{j=m+1}^n O_j$$
(5.6)

$$\geq m + n - m \tag{5.7}$$

$$= |S_i|$$

Equality in (5.6) holds when those m nodes have no children in the previous cycle, and equality in (5.7) holds when the remaining n - m nodes lose all their children in the current cycle.

Based on this, we have the following theorem.

Theorem 5.2.4. Suppose Assumption 8 holds. Under the perturbation described above, for a spanning tree with n nodes, value of the source node satisfies

$$O_s \ge \begin{cases} n & n \le 2\\ 3 & n \ge 3 \end{cases}$$

Proof. Cases n = 1 and n = 2 are trivial, since there is no parent switching. When $n \ge 3$, we have

$$O_s \ge \sum_{j \in S_1} O_j + 1 \tag{5.8}$$

If there is only one node at level 1, then $O_s \ge 3$ since there must be at least one node at level 2. According to Lemma 5.2.3, if there are more than one node at level 1, then

$$O_s \geq |S_1| + 1 \tag{5.9}$$

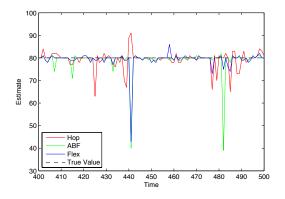
$$\geq 3$$
 (5.10)

Equality in (5.9) holds when nodes in level 1 satisfy the conditions mentioned in Lemma 5.2.3, and equality in (5.10) holds when there are two nodes at level 1. \Box

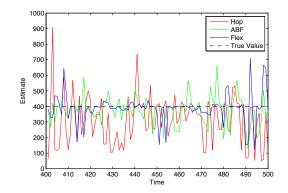
5.2.2 Empirical comparison

To test these predictions, we ran experiments using using MIT Proto [13] to simulate a network of unsynchronized devices. In particular, we ran tests on devices randomly arranged in a rectangular region with neighbors determined by a unit disc graph. The G block here used to calculate the distance from a single fixed source devices is in a different manner for each instance: one using hop-count (i.e., Adaptive Bellman-Ford with unit distance), one by straight ABF, and one by Flex-Gradient [11]. Specifically, Flex-Gradient is designed to limit the radius of disruption from network oscillations. It accomplishes this by tolerating a set fraction of error in estimate per hope, such that the distance that a correction propagates is proportional to the size of the correction.

Error in state estimation is then the difference between the number of devices in the network and the state estimate output from the C block at the source device. To test the effects of scale, we ranged the width of the arena from 2 to 20 units in steps of 2 units, while keeping its other dimension at 2 units, and placed the source device initially at the one extreme of the long axis of the arena. The number of devices



(a) Example estimate variation over time in a 4 unit width space



(b) Example estimate variation over timein a 20 unit width space

Figure 5.2: Examples of typical estimate variation over time (figure from [56]), showing excerpts from a individual simulation run on a network dispersed through a space (a) 4 units in width and (b) 20 units in width. Notice that most transients are relatively short, but they are much more frequent for hop-count than for Adaptive Bellman-Ford or Flex-Gradient, and they tend to be underestimates rather than overestimates, particularly for hop-count.

was scaled proportional to the area of the arena, at 10 devices per square unit, i.e., from 40 devices in a 2×2 arena to 400 devices in a 20×2 arena. To inject continual perturbation into this network, each device moved randomly following a reactive Levy walk (a scale-free form of constrained random walk [12]) at a rate of 0.0025 units/second. Each trial was run for 1000 simulated seconds, 10 trials per condition, recording values at each second. Our analysis, however, drops the first 100 seconds of each trial as being potentially still affected by convergence from initialization. When diameter is low, it is expected that there should not be many opportunities for disruption on any given chain of parents to the source, and indeed we see that the estimates are often correct, but with frequent transients, sometimes causing large transients in estimate value. At high diameter, most devices are far from the source, so there are many opportunities for information to be duplicated or lost. Moreover, many of the devices share at least part of their path to the source, creating critical links whose disruption is likely to cause large transients. Figure 5.2 shows excerpts from two of trials, illustrating that the typical patterns in how estimates were observed to vary from the true value follow these predictions. Most individual transients are relatively short, but they are much more frequent for hop-count than for Adaptive Bellman-Ford or Flex-Gradient, and they tend to be underestimates rather than overestimates, particularly for hop-count.

Analysis of the overall statistics of errors versus width bears out these observations. Figure 5.3a shows that as the width of the space increases (and thus the diameter of the network rises), the mean relative error in estimates rises approximately linearly—though the high degree of variation in the behavior of Adaptive Bellman-Ford makes difficult to verify for that case. Complementarily, Figure 5.3b shows that the amount of time that the estimate spends equal to the true value decreases. approximately exponentially with increasing width. This is as would be expected if we consider reconfiguration (and thus transient duplication or loss) to be equally likely to occur at any location in the network. In general, having a smoother input potential produces better results: Adaptive Bellman-Ford slightly outperforms hop-count

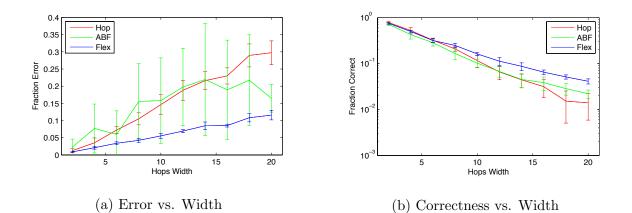


Figure 5.3: The more hops spanned by the network, the higher the relative error in estimates (a) and the smaller amount of time that the estimate is correct. Furthermore, the smoother the input potential, the better estimation performs (figure from [56]).

distance values, and Flex-Gradient produces much better performance than both.

A deeper inspection of the errors finds that the distribution of individual error values is also consistent with the prediction of the importance of smoothness of potential from our analysis. Figure 5.4 shows a typical histogram of error ratios, in this case from the collection of trials with a width of 16 hops. Running the C block with all three potential algorithms results in a clear "spike" with a plurality of values being equal (or almost equal) to the true value, and all three have nearly identical rates of overestimates from transient duplication of values. The three potential algorithms differ starkly, however, in the distribution of underestimates. Hop-count distances are the least able to distinguish between alternative paths, and appear to pay for this in generally increased volatility. Adaptive Bellman-Ford usually performs bet-

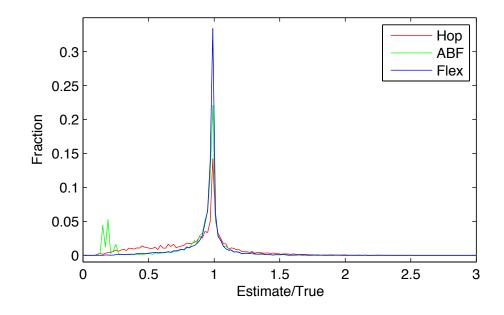


Figure 5.4: Plot from [56] of typical histogram of error values, showing that smooth adaptation of potential prevents transient value loss.

ter, but in certain circumstances can experience long underestimate transients due to the "rising value problem" [14, 27]: when this problem occurs, it can cause a severe underestimate to last for a long time, as seen in the distribution spikes low values. Flex-Gradient, on the other hand, because it preserves smoothness at the cost of accuracy in distance estimates, suffers from much less loss of value than both of the others, explaining its superior performance.

5.3 A resilient leader election algorithm

Leader election is widely recognized as a fundamental problem in distributed systems. It permits a network of nodes to collectively choose a node as a leader in a distributed, and in our case, resilient fashion. A resilient algorithm should must ensure the eventual election of a *single* leader and should have the ability to recover from any transient perturbations like the disappearance of leaders, temporary, though nonpersistent emergence of false leaders and link failures that leave the graph connected.

In the sequel we use the following standard notation. An algorithm has a time complexity of $\Theta(D)$ if there exist positive constants k_1, k_2 and D_0 such that, $k_1D \leq$ running time $\leq k_2D$ for $D \geq D_0$. An algorithm has a time complexity of $\Omega(D)$ if there exist positive constants k_1 and D_0 such that, $k_1D \leq$ running time for $D \geq D_0$. An algorithm has a time complexity of O(D) if there exist positive constants k_1 and D_0 such that, $k_1D \geq$ running time for $D \geq D_0$.

Starting from the earliest formulation of leader election algorithm in 1977 e.g. [38] and [53], the study of leader election algorithms analyzes various attributes, including complexity (time, space and message complexities) of *deterministic* leader election on general networks with identifiers, the complexity of *probabilistic* leader election on anonymous networks, the complexity of leader election on some specific networks such as ring graphs and complete graphs, and complexity of leader election on asynchronous graphs. In this thesis we assume that each node carries a unique identifier of size $O(\log N)$ bits with N the number of nodes, and each node exchanges messages with its neighbors in synchronous rounds. Moreover, time, space and message complexities are measured by communication rounds, bits and $\log N$ bits respectively.

In [61], a simple leader election algorithm is presented for general networks via the construction of a *breadth first search* tree (BFS) rooted in the leader. Time

and message complexities of the algorithm are $O(\mathcal{D})$ and $O(\mathcal{D}\mathcal{E})$ respectively, where \mathcal{D} is the diameter of the network and \mathcal{E} is the number of edges. This algorithm is time optimal as the global lower bound on the time required by the leader election algorithm in general synchronous networks is $\Omega(\mathcal{D})$ rounds [51]. On the other hand, [24] proposes a solution that elects a leader in $O(\mathcal{D} + \log N)$ rounds by only sending messages of size O(1) instead of $O(\log N)$ bits. However, both these algorithms assume the graph is perturbation free.

Some time optimal solutions consider perturbations but also assume nodes have such prior knowledge of the graph as the number of nodes N or the diameter \mathcal{D} . In face of loss of nodes or network growth such assumptions cannot be sustained. In [8], the authors propose a time optimal leader election algorithm by implementing the Bellman-Ford algorithm with IDs. This algorithm requires nodes know an upper bound on the diameter of the network. Similarly, [22] provides a time-optimal leader election algorithm by assuming nodes know some upper bound on \mathcal{D} . The space complexity of both solutions is $O(\log N \log \mathcal{D})$ as they require $\Theta(\log N \log \mathcal{D})$ bits per node.

There are algorithms that accommodate perturbations without advance knowledge of the network. In [1] and [48], authors present time optimal leader election algorithms without any prior knowledge of the network size or diameter. In the former, an *Extend-ID* mechanism is used to eliminate a corrupted leader, in which case the bits of a message may increase to unbounded length. While in the latter, the author implicitly uses the network size by implementing a synchronizer requiring the knowledge of upper bound on N. Recently, many papers have considered leader election on asynchronous networks. Solutions proposed by Datta et al. in [29, 30] and Altisen et al. in [2] assume a distributed unfair daemon and have a stabilization time in O(N)rounds as well as a space complexity of $O(\log N)$ bits per node. As stabilization time is not $O(\mathcal{D})$, these are not time optimal.

In this section, we introduce a GUAS leader election algorithm via a feedback interconnection of the building blocks in *aggregate computing*, that is time optimal, in the sense that its stabilization time is $O(\mathcal{D})$, with a space complexity of $O(\log N)$ bits per node. This space complexity is asymptotically optimal since a node needs to store its own ID using $O(\log N)$ bits, and the message complexity of our algorithm is $O(\mathcal{DE})$. Inspired by [8], while running a slightly different Bellman-Ford algorithm with IDs, our algorithm estimates the diameter of the network simultaneously, and thus no prior knowledge of the network is needed. Like most feedback systems, our algorithm has one free design parameter that defines certain important performance attributes. Too large a value will accelerate convergence but impair resilience by delaying recovery from loss of the current leader. Too small a value will improve resilience but make convergence slower or even lead to multiple leaders. Moreover, any synchronous algorithm can be made to run on an asynchronous network as long as it requires every node to send messages to all its neighbors at each time step. As our algorithm obeys this property, it can be applied to an asynchronous network by using a simple synchronizer.

Section 5.3.1, makes some important definitions and assumptions which will be

used for the stability proof later. In Section 5.3.2, we give a formal description of our leader election algorithm. In Section 5.3.3, we first demonstrate the resilience of the leader election algorithm by upper bounding the time on recovering from transient perturbations, and then prove its GUAS. Section 5.3.4 validates our results through simulations.

5.3.1 Preliminaries

We consider an undirected graph $\mathcal{G} = (V, E)$, with $V = \{1, 2, ..., N\}$ the set of nodes, E the set of undirected edges. We assume the index of a node represents its ID and also reflects its priority. A node with a lower ID has a higher priority, in that node i has a higher priority than node i+1. The goal of our leader election algorithm is to elect the node with the highest priority to be the single leader in the graph. An edge indicates the existence of communication link between nodes and, we call node i a neighbor of node j if there is an edge between i and j. We define $\mathcal{N}(i)$ as the set of all neighbors of node i. We assume each edge length is 1.

Suppose node 1 is the only source in the graph, define d_i as the shortest distance from node *i* to the source. Then d_i for all $i \in V$ obeys the recursion [58]:

$$d_{i} = \begin{cases} 0, & i = 1\\ \\ \min_{k \in \mathcal{N}(i)} \{d_{k} + 1\}, & i \neq 1 \end{cases}$$
(5.11)

Based on (5.11), we introduce the following definition.

Definition 5.3. A k that minimizes the right hand side of (5.11) is a *true constraining* node of i. As there may be two neighbors k and l of i such that $d_l = d_k$, a node may have multiple true constraining nodes. The set of true constraining nodes of i is defined as C(i). Moreover, the true constraining node of node 1 is itself.

Further, we define the following related definition.

Definition 5.4. For a connected graph \mathcal{G} , consider any sequence of nodes such that the predecessor of each node is one of its true constraining nodes. Define $\mathcal{D}(\mathcal{G})$, the effective diameter of \mathcal{G} , as the longest length such a sequence can have in \mathcal{G} . It follows $\mathcal{D}(\mathcal{G}) \leq \mathcal{D}$ with \mathcal{D} the diameter of \mathcal{G} .

5.3.2 Algorithm

We give a formal description of our leader election algorithm in this section. The block diagram of the leader election algorithm is shown in Figure 5.5. It uses a feedback interconnection comprising three building blocks in aggregate computing, which are themselves distributed algorithms.

Before providing details of this algorithm we summarize the basic approach. A node *i*, is *attached* to the leader estimated as being nearest to it. A leader is attached to itself. The distance estimates are obtained using an ABF like algorithm. Node *i* carries the diameter estimate D_i of the sub-network of the nodes attached to the same leader as itself. With a design parameter K, a leader relinquishes its status if it finds a neighbor *j* that is attached to a higher priority node within $KD_j + 1$ hops from *j*. If no such neighbor exists then it assumes or retains the role of a leader.

It is this diameter estimate based election strategy that endows the algorithm with resilience. Thus suppose a leader is lost. Each of its neighbors is thereafter constrained by others. Their distance estimates keep increasing until at least one node, i, is unable to find a neighbor j which is within $KD_j + 1$ hops from the leader it is attached to. This node i then becomes a leader.

In Figure 5.5 MABF block is a modified adaptive Bellman-Ford algorithm. It has the dual role of determining current leaders and obtaining distance estimates of nodes attached to it. The C block collects diameter estimates. The B-block is a G-block that broadcasts this diameter estimate.

5.3.2.1 The MABF block

The *MABF* block in the forward path is a variant of the \mathcal{G} block in *aggregate* computing, and is used to estimate the shortest distance of each node from its nearest source (leader). As introduced in Section 5.3.1, the node ID, represented by node index, is unique and cannot be falsified. Further, we define $\sigma_i(t)$ as the estimated leader ID of node i, referring to the node ID that i believes to be the leader at time tand $\sigma_i(t)$ can be corrupted under perturbations. The purpose of leader election is to elect the highest priority node as the unique leader, i.e., $\sigma_i(t) = 1$ for all $i \in V$ and $t \geq T$ with some finite T.

More specifically node *i* is a current leader if $\sigma_i(t) = i$. Furthermore *i* is *attached* to a leader *j* if $\sigma_i(t) = j$.

In MABF, there is first an implicit selection of whether a node *i* should be a leader. This done by seeing if: (i) *i* has a neighbor *j* that carries a distance estimate (hop count) \hat{d}_j that is within $KD_j(t) + 1$ hops from the leader it is attached to; and

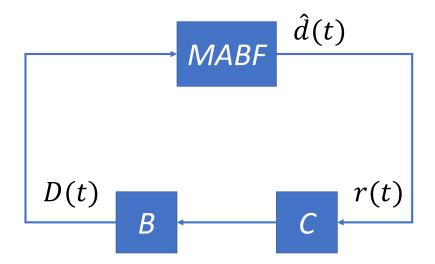


Figure 5.5: Block diagram of the leader election algorithm

(ii) if this leader has a higher priority than *i*. Specifically this is equivalent to seeing whether a valid set of neighbors of *i*, $V_i(t)$ defined below is empty. To define $V_i(t)$, we first define $W_i(t)$ comprising the neighbors of *i* whose distance estimates are within a certain range:

$$W_i(t) = \{ j \in \mathcal{N}(i) \mid \hat{d}_j(t) \le g(D_j(t)) \}$$
(5.12)

where $g(D_i(t)) = KD_j(t) + 1$ with $K \ge 1$, and $D_j(t)$ is j's effective diameter estimate which is defined in Section 5.3.2.3. Then the valid set $V_i(t)$ comprises members j of $W_i(t)$ that carry the smallest $\sigma_j(t)$ lower than i:

$$V_i(t) = \{ j \in W_i(t) \mid \sigma_j(t) = \min_{l \in \mathcal{N}(i)} \{ \sigma_l(t) \} \text{ and } \sigma_j(t) < i \}$$

$$(5.13)$$

If $V_i(t)$ is empty, then i becomes a leader. Otherwise it cannot be one, and will attach itself to the leader one of the members of $V_i(t)$ is attached to. This neighbor also becomes the constraining node of i used to update its distance estimate $\hat{d}_i(t)$, and is a member of $V_i(t)$ that carry the smallest distance and the largest diameter estimate. Thus the constraining node is a member of $\tilde{V}_i(t)$ defined below. Define $\bar{V}_i(t)$ as the set comprising nodes from $V_i(t)$ that have the smallest distance estimate at time t:

$$\bar{V}_i(t) = \{ l \in V_i(t) | l = \arg\min_{j \in V_i(t)} \{ \hat{d}_j(t) \} \},$$
(5.14)

and

$$\tilde{V}_i(t) = \{ l \in \bar{V}_i(t) | l = \arg \max_{j \in \bar{V}_i(t)} \{ D_j(t) \} \}.$$
(5.15)

Then each node updates its distance estimate according to:

$$\hat{d}_i(t+1) = \begin{cases} \hat{d}_j(t) + 1 & V_i(t) \neq \emptyset, j \in \tilde{V}_i(t) \\ 0 & V_i(t) = \emptyset. \end{cases}$$
(5.16)

We provide a definition a part of which mirrors the definition of constraining nodes in ABF.

Definition 5.5. Define j in the first case of (5.16) as the current constraining node $c_i(t+1)$ of i at time t+1. Further at t+1, if the second case is used then i is its own current constraining node. We define $C_i(t)$ as the set of nodes taking i as the current constraining node at time t except i itself.

Node *i* is elected as a leader if distance estimates of all its neighbors are greater than g(D(t)) with D(t) their effective diameter estimates, or the neighbors having distance estimates lower than g(D(t)) all carry estimated leader IDs with lower priority, and *i* is prevented from being a leader if there exists a neighbor of *i* with a distance estimate lower than g(D(t)) and an estimated leader ID with higher priority.

$$\sigma_i(t+1) = \begin{cases} i & c_i(t+1) = i \\ \sigma_j(t) & c_i(t+1) = j \end{cases}$$
(5.17)

Finally as explained in the foregoing, the set of leaders (sources) set at time t is defined as

$$S(t) = \{i \mid \sigma_i(t) = i \text{ and } \hat{d}_i(t) = 0\}.$$
(5.18)

5.3.2.2 The C block

The C block collects and sends to each source its current estimated effective diameter as the following:

$$r_i(t+1) = \max\{d_i(t+1), \{r_j(t)|j \in \mathcal{C}_i(t+1)\}\}$$
(5.19)

with $C_i(t)$ defined in Definition 5.5.

5.3.2.3 The *B* block

The B block is a G block broadcasting the estimated effective diameter from each source to nodes attached to it. Each node updates its effective diameter estimate to the value held at its current constraining node:

$$D_{i}(t+1) = \begin{cases} r_{i}(t+1) & c_{i}(t+1) = i \\ D_{j}(t) & c_{i}(t+1) = j \end{cases}$$
(5.20)

where $c_i(t)$ is defined in Definition 5.5.

5.3.3 Global uniform asymptotic stability

We now prove the global uniform and asymptotic stability of the leader election algorithm described in Section 5.3.2, while bounding the time required for its convergence. Throughout the section, we shall use the following assumptions.

Assumption 9. Graph $\mathcal{G} = (V, E)$ is connected and undirected, $t_0 = 0$ is the initial time, d_i , $\mathcal{N}(i)$, $\mathcal{C}(i)$, $\hat{d}_i(t)$, $c_i(t)$, $r_i(t)$, $D_i(t)$, $\sigma_i(t)$, g(D) = KD + 1 with $K \ge 1$ are defined as in Section 5.3.2, and for all $i \in V$, the quantities $\hat{d}_i(0)$, $r_i(0)$, $D_i(0)$ are non-negative integers.

Notice that this last assumption is not restrictive, since the algorithm is only able to produce non-negative values for those quantities, hence any negative value can be easily recognised as spurious.¹ In general, however, we cannot assume that $\sigma_i(0)$ is positive (nor integer), since under certain transient perturbations (e.g., node crashes or is corrupted by an adversary), some node may carry a fake ID, i.e., for $i \in V$, $\sigma_i(0) = k \notin V$. However, we can prove that the influence of incorrect starting values tends to decay in favour of the values produced from existing nodes: More precisely, that estimates on *unrooted nodes* (i.e., whose estimates depend on starting values) tend to worsen as time passes, until they are finally discarded. In our proof, we only consider the transient perturbations to be a node corruption, but it is readily verified that our algorithm works for node crashes. Suppose in this case node j is a crashed node, then $j \notin V$.

¹By the same reasoning, we could also assume that $\hat{d}_i(0) \leq g(D_i(0))$ (but it is not needed in the proofs).

Definition 5.6 (Unrooted Nodes). Define the set of *unrooted nodes* $\mathcal{U}(t)$ as $\mathcal{U}(0) = V$, and

$$\mathcal{U}(t+1) = \{i \in V | i \neq c_i(t+1) \in \mathcal{U}(t)\}.$$

In other words members $\mathcal{U}(t+1)$ comprises non-source nodes constrained by members of $\mathcal{U}(t)$. Let $\mathcal{L}(t) = \{\sigma_i(t) | i \in \mathcal{U}(t)\}$ be the set of unrooted leaders, and $\mathcal{U}^k(t)$ be the set of unrooted nodes with leader k:

$$\mathcal{U}^k(t) = \{ i \in \mathcal{U}(t) | \sigma_i(t) = k \}.$$

Furthermore, define

$$\hat{d}_{\min}^{k}(t) = \min\{\hat{d}_{i}(t)|i \in \mathcal{U}^{k}(t)\},\$$
$$D_{\max}^{k}(t) = \max\{D_{i}(t)|i \in \mathcal{U}^{k}(t)\}.$$

We first show that the set of unrooted leaders cannot grow.

Lemma 5.3.1 (Unrooted Leader Set's Nonincrease). Consider (5.12-5.20) under Assumption 9 and Definition 5.6. The set of unrooted leaders cannot expand over time, i.e. obeys $\mathcal{L}(t+1) \subseteq \mathcal{L}(t)$. Furthermore, leaders $\sigma_i(t)$ of rooted nodes $i \in V \setminus \mathcal{U}(t)$ are in V.

Proof. We proceed by induction on t. Suppose $j \in \mathcal{L}(t+1)$. From Definition 5.6, there exists $i \in \mathcal{U}(t+1)$ such that $j = \sigma_i(t+1)$. Thus from definitions 5.6 and 5.5, there exists a $k, i \neq k = c_i(t+1) \in \mathcal{U}(t)$ such that $\sigma_k(t) = j$. Then by Definition 5.6, $j \in \mathcal{L}(t)$, proving the first part of the theorem.

For the second part, observe from Definition 5.5 and Definition 5.6 that the constraining node of $i \in V \setminus \mathcal{U}(t+1)$ can only be such that either $c_i(t+1) = i$

or $c_i(t+1) = j \notin \mathcal{U}(t)$. In the former case, $\sigma_i(t+1) = i \in V$. In the latter, $\sigma_i(t+1) = \sigma_j(t)$ which is in V by the inductive hypothesis.

We next provide a lemma that helps show that the unrooted set decays.

Lemma 5.3.2 (Unrooted Decay). Let $t \ge 0$ and $k \in \mathcal{L}(t)$. Then, $\hat{d}_{\min}^k(t) \ge \hat{d}_{\min}^k(0) + t$ and $D_{\max}^k(t) \le D_{\max}^k(0)$.

Proof. We proceed by induction on t. For t = 0 the hypothesis is trivially true. Thus assume that t > 0 and $i \in \mathcal{U}^k(t)$. By Lemma 5.3.1, $k \in \mathcal{L}(0)$ hence $\hat{d}_{\min}^k(0)$, $D_{\max}^k(0)$ are well-defined. By definition of unrooted node $j = c_i(t) \in \mathcal{U}(t-1)$, by the algorithm definition $\sigma_i(t) = \sigma_j(t-1) = k$, thus $j \in \mathcal{U}^k(t-1)$. Thus $D_i(t) = D_j(t-1) \leq D_{\max}^k(0)$, and $\hat{d}_i(t) = \hat{d}_j(t-1) + 1 \geq \hat{d}_{\min}^k(0) + (t-1) + 1$, concluding the proof.

Combining the monotonic progression of $\hat{d}_{\min}^{k}(t)$ and $D_{\max}^{k}(t)$ in Lemma 5.3.2 with the algorithm restriction that $\hat{d}_{i}(t) \leq g(D_{i}(t))$, we obtain both that the impact of fake ID, $k \in \mathcal{L}(0)$ but $k \notin V$ disappears after a time \hat{T} (Lemma 5.3.3), and that underestimates for the correct leader (w.o.l.g., assume the existing highest priority node in the network is node 1) disappear after a time depending on the node distance (Lemma 5.3.4).

Lemma 5.3.3 (Fake IDs Decay). Let:

$$\hat{T} = 1 + \max\{g(D_{\max}^k(0)) - \hat{d}_{\min}^k(0) | k \in \mathcal{L}(0) \text{ with } k < 1\}.$$

Then $\sigma_i(t) \in V$ for every node $i \in V$ and $t \geq \hat{T}$.

Proof. Assume by contradiction that $i \in V$, $t \geq \hat{T}$ are such that $\sigma_i(t) = k \in V$. By Lemma 5.3.1, $k \in \mathcal{L}(0)$ and $i \in \mathcal{U}^k(t)$ since $k \notin V$. By Lemma 5.3.2 and definition of $\hat{d}^k_{\min}(t)$:

$$\hat{d}_i(t) \ge \hat{d}^k_{\min}(t) \ge \hat{d}^k_{\min}(0) + t \ge \hat{d}^k_{\min}(0) + \hat{T}$$
 (5.21)

>
$$\hat{d}_{\min}^k(0) + g(D_{\max}^k(0)) - \hat{d}_{\min}^k(0)$$
 (5.22)

$$= g(D_{\max}^k(0))$$
 (5.23)

By Lemma 5.3.2 and definition of $\hat{D}_{\max}^k(t)$:

$$D_i(t) \le \hat{D}_{\max}^k(t) \le \hat{D}_{\max}^k(0)$$

Thus $\hat{d}_i(t) > g(D_i(t))$, which is a contradiction by the third bullet of (2.2).

We next show that a node attached to the desired leader must eventually have an underestimated distance estimate.

Lemma 5.3.4 (Underestimates Decay). For every $i \in V$ and $t \ge 0$ such that $\sigma_i(t) = 1$, we have $\hat{d}_i(t) \ge \min(d_i, t)$.

Proof. Assume that $i \in V, t \geq 0$ are such that $\sigma_i(t) = 1$. If $i \in \mathcal{U}(t)$, then:

$$\hat{d}_i(t) \ge \hat{d}_{\min}^1(t) \ge \hat{d}_{\min}^1(0) + t \ge t.$$

If $i \in V \setminus \mathcal{U}(t)$, then there is a sequence $i = i_0, \ldots, i_k = 1$ such that $c_{i_\ell}(t - \ell) = i_{\ell+1}$ and $c_1(t - k) = 1$, hence $\hat{d}_i(t) = k$. This sequence is a path on the graph, thus its length has to be $\geq d_i$, concluding the proof. As soon as the impact of starting values has sufficiently decayed (as detailed in the previous lemmas), correct stable estimates flow outwards from the source bounded by source diameter estimates (Lemma 5.3.6), while those diameter estimates flow inwards bounded by the stabilization of distances (Lemma 5.3.5).

Henceforth we are concerned with nodes attached to the desired leader 1. The next lemma speaks to how the diameter estimate at 1, evolves.

Lemma 5.3.5 (Diameter Collection). With C(i) and $c_i(t)$ defined in Definition 5.3 and 5.5, assume that T_x integer $0 \le x \le D(\mathcal{G}) - 1$ is such that every device $i \in V$ with $d_i \le x$ stabilizes to $\sigma_i(t) = 1$, $\hat{d}_i(t) = d_i$ and $c_i(t) \in C(i)$ for $t \ge T_x$. Then $D_1(t) \ge x$ for $t \ge T_x + x$.

Proof. Let $i_0 \in V$ be any node such that $d_{i_0} = x$, and let $i_1, \ldots, i_x = 1$ be a shortest path from it to the source, so that $i_{\ell+1} \in \mathcal{C}(i_\ell)$ for $l \in \{0, \cdots, x-1\}$. We prove by induction on $k \leq x$ that the diameter collection $r_{i_k}(t) \geq x$ for $t \geq T_x + k$. For k = 0, it holds since:

$$r_{i_0}(t) = \max\{\hat{d}_{i_0}(t), \{r_j(t-1)|j \in \mathcal{C}_{i_0}(t)\}\}$$
(5.24)

$$\geq \hat{d}_{i_0}(t) = d_{i_0} = x \tag{5.25}$$

For k > 0, it holds since:

$$r_{i_k}(t) = \max\{\hat{d}_{i_k}(t), \{r_j(t-1)|j \in \mathcal{C}_{i_k}(t)\}\}$$
(5.26)

$$\geq r_{i_{k-1}}(t-1) \geq x$$
 (5.27)

Thus, $r_1(t) = r_{i_x}(t) \ge x$ for $t \ge T_x + x$. Since 1 is its own constraining node, $D_1(t) = r_1(t) \ge x$ as well, concluding the proof. The now on how the $D_1(t)$ is broadcast.

Lemma 5.3.6 (Diameter Broadcast). With C(i) and $c_i(t)$ defined in Definition 5.3 and 5.5, assume that $T_x \ge \hat{T}$ defined in Lemma 5.3.3 is such that $\sigma_1(t) = 1$, $\hat{d}_1(t) = 0$ and $D_1(t) \ge x$ for $t \ge T_x$. Then every $i \in V$ with $d_i \le g(x)$ stabilizes to $\sigma_i(t) = 1$, $\hat{d}_i(t) = d_i, c_i(t) \in C(i)$ and $D_i(t) \ge x$ for $t \ge T_x + d_i$.

Proof. We proceed by induction on d_i . If $d_i = 0$, then i = 1 is the source hence $\sigma_i(t) = 1$, $\hat{d}_i(t) = d_i = 0$ for $t \ge T_x$ holds by hypothesis.

Suppose now that $d_i > 0$, and let $t \ge T_x + d_i$, $j \in \mathcal{C}(i)$ so that $d_j = d_i - 1$. By inductive hypothesis $\sigma_j(t-1) = 1$, $\hat{d}_j(t-1) = d_j$, $D_j(t-1) \ge x$. Since $d_j + 1 = d_i \le g(x) \le g(D_j(t-1))$, $j \in W_i(t-1)$ defined in (5.12) is not discarded from the set of possible constraining nodes at the first step. By Lemma 5.3.3 and $T_x \ge \hat{T}$, there are no $k \in \mathcal{N}(i)$ with $\sigma_k(t-1) \notin V$. It follows that j has maximal priority $\sigma_j(t-1) = 1$, hence it is not discarded at the second step, i.e., $j \in V_i(t-1)$, and $\forall k \in V_i(t-1), \sigma_k(t-1) = 1$. From (5.17), $\sigma_i(t) = 1$.

By Lemma 5.3.4, every $k \in \mathcal{N}(i)$ satisfies

$$\hat{d}_k(t-1) \ge \min(d_k, t-1) \ge \hat{d}_i(t-1) = d_i$$

and thus j is not discarded at the third step, i.e., $j \in \overline{V}_i(t-1)$. Assume $c_i(t)$ the current constraining node of i is k, then it follows $\hat{d}_i(t) = \hat{d}_k(t-1) + 1 = d_k + 1 = d_j + 1 = d_i$, and $c_i(t) = k \in \mathcal{C}(i)$. Letting $X \ni j$ as the set of nodes that are not discarded at the final step, i.e., the set $\tilde{V}_i(t-1)$. From (5.20), it follows that

$$D_i(t) = \max\{D_k(t-1)|k \in X\} \ge D_j(t-1) \ge x.$$

Combining the last two lemmas, we can finally recursively characterise the overall convergence time T_x for nodes at distance x (Theorem 5.3.7).

Definition 5.7 (Discrete Inverse). We define $g^{-1}(D) = \left\lceil \frac{D-1}{K} \right\rceil$, which is the smallest number x such that $g(x) = Kx + 1 \ge D$.

Finally we characterize convergence and the time to converge.

Theorem 5.3.7 (Convergence). Let T_x for $x \leq \mathcal{D}(\mathcal{G}) - 1$ be recursively defined as $T_0 = \hat{T}$ with \hat{T} defined in Lemma 5.3.3,

$$T_x = T_{g^{-1}(x)} + g^{-1}(x) + x.$$

Then for $t \ge T_x$ and $i \in V$ such that $d_i \le x$, we have that $\sigma_i(t) = 1$ and $\hat{d}_i(t) = d_i$.

Proof. We prove the thesis by induction on x. If x = 0 and $t \ge \hat{T}$, $\sigma_1(t) \ge 1$ by Lemma 5.3.3, and hence $\sigma_1(t) = 1$ and $\hat{d}_1(t) = 0$ concluding the inductive base case.

Suppose now that the thesis holds for every y < x, and in particular for $g^{-1}(x) < x$. By inductive hypothesis, devices up to $g^{-1}(x)$ distance are stable for $t \ge T_{g^{-1}(x)}$, and then by Lemma 5.3.5 $D_1(t) \ge g^{-1}(x)$ for $t \ge T_{g^{-1}(x)} + g^{-1}(x)$. We can then apply Lemma 5.3.6, obtaining that devices up to x distance are stable for $t \ge T_{g^{-1}(x)} + g^{-1}(x) + x = T_x$, concluding the proof.

Notice that the bound above is strict e.g. whenever $\sigma_i(0) > 1$ for all $i \in V \setminus \{1\}$, and $r_1(0) = D_1(0) = 0$. Furthermore, we can approximate the bound through a closed form formula.

$$T_x = T_{g^{-1}(x)} + g^{-1}(x) + x (5.28)$$

$$= \hat{T} + x + 2\sum_{k=1}^{\infty} g^{-k}(x)$$
 (5.29)

$$= \hat{T} + x + 2\sum_{k=1}^{\infty} \left[\frac{x - \sum_{i=0}^{i < k} K^{i}}{K^{k}} \right]$$
(5.30)

In case K = 1, then $\left\lceil \frac{x - \sum_{i=0}^{i < k} K^i}{K^k} \right\rceil = x - k$ for $x \ge k$ and thus:

$$T_x = \hat{T} + x + 2\sum_{k=1}^{x} (x-k) = \hat{T} + x + x(x-1) = \hat{T} + x^2$$

which is quadratic on x. However, if $K \ge 2$, letting $L = \lceil \log_K(x(K-1)+1) \rceil$ (the first term for which the terms in the summation are zero):

$$\begin{split} T_x &= \hat{T} + x + 2\sum_{k=1}^{\infty} \left\lceil \frac{x - \sum_{i=0}^{i \le k} K^i}{K^k} \right\rceil \\ &= \hat{T} + x + 2\sum_{k=1}^{\infty} \left\lceil \frac{x - (K^k - 1)/(K - 1)}{K^k} \right\rceil \\ &= \hat{T} + x + 2\sum_{k=1}^{L-1} \left\lceil \frac{x}{K^k} - \frac{1 - 1/K^k}{K - 1} \right\rceil \\ &\leq \hat{T} + x + 2L + 2\sum_{k=1}^{L-1} \left(\frac{x}{K^k} - \frac{1 - 1/K^k}{K - 1} \right) \\ &= \hat{T} + x + 2L + 2x \frac{\frac{1}{K^L} - \frac{1}{K}}{\frac{1}{K} - 1} - \frac{2(L - 1)}{K - 1} + \frac{2(\frac{1}{K^L} - \frac{1}{K})}{(K - 1)(\frac{1}{K} - 1)} \\ &= \hat{T} + x \frac{2 - K^L - K^{L-1}}{K^L - K^{L-1}} + 2L \frac{K - 2}{K - 1} + \frac{2}{K - 1} + \frac{2(1 - K^{L-1})}{(K - 1)(K^{L-1} - K^L)} \\ &\leq \hat{T} + 3x + 2L + 4 \end{split}$$

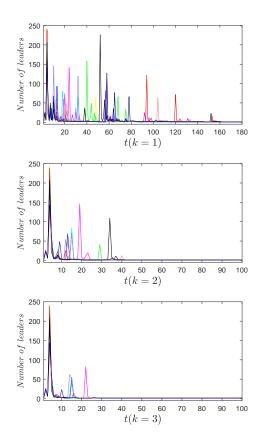
In fact, if K = 2, $T_x \leq \hat{T} + 3x + 4$, which is linear in x. We can also notice that if D is the maximum overall $D_i(0)$ value, $\hat{T} \leq 1 + g(D) = KD + 2$, so that for K = 2,

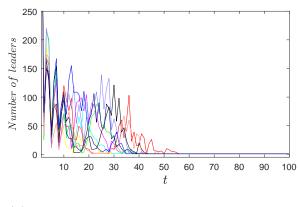
 $T_x \leq 2D + 3x + 6$. Moreover, the time to convergence derived above is independent of the initial time $t_0 = 0$, and thus it proves the global uniform and asymptotic convergence of the leader election algorithm.

5.3.4 Simulations

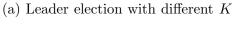
In this section, we confirm our previous results through simulations. We first investigate the global uniform and asymptotic stability of the algorithm, and compare its performance with the algorithm proposed by [30]. In the simulations, 500 nodes are randomly distributed in a 4×1 field, communicating with a 0.25 unit disc. Each node has about 20 neighbors and communicates synchronously, and each node ID is the same as its index ranging from 1 to 500. For our algorithm, the initial conditions are set as the follows, $\forall i \in V$, $\hat{d}_i(0) = r_i(0) = D_i(0) = 0$, and $\sigma_i(0)$ is uniformly distributed between 1 to 500. The simulation is run 10 times.

We run our algorithm with K = 1, K = 2 and K = 3 under the same initial conditions. The results are shown in Figure 5.6a. $\mathcal{D}(\mathcal{G})$ ranges from 11 to 19 for those 10 trials. Compared with K = 2 and K = 3, our leader election algorithm will take a longer time (115 rounds) to converge with K = 1, which verifies our result that in this case the time complexity is $O(\mathcal{D}(\mathcal{G})^2)$. With K = 2 or K = 3, our algorithm has a much better performance. The average time to convergence is 36 rounds and 28 rounds for K = 2 and K = 3 respectively. Figure 5.6b depicts the performance of Datta's algorithm. The average time to convergence in this case is 41 rounds, which is worse than our algorithm with K = 2 or 3. Moreover, Datta's algorithm has more





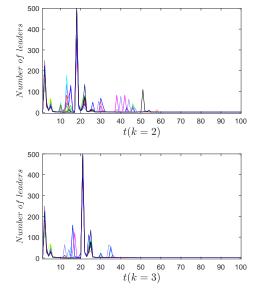
(b) Leader election by Datta in a randomized graph.

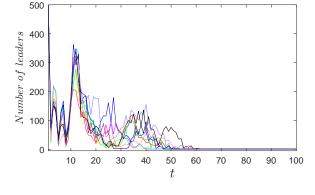


in a randomized graph.

Figure 5.6: Comparison of convergence time between our algorithm and Datta's algorithm without perturbations







(b) Leader election by Datta in a randomized graph.

(a) Leader election with different K

in a randomized graph.

Figure 5.7: Comparison of convergence time between our algorithm and Datta's algorithm with perturbations

frequent oscillations.

We next compare their resilience under the same setup as shown above. In this case, we run our algorithm with K = 2 and 3. The transient perturbation is as the following: With initial conditions unchanged from the perturbation free case, for i = 50 to 100, $\sigma_i(10) = 0.5$. That is, during the process of convergence, 51 nodes are corrupted by a fake ID with the highest priority.

Figure 5.7a demonstrates the results by our algorithm. With K = 2, states of nodes will recover and converge with 52 rounds on the average for the 10 trials. It is almost the same as the Datta's algorithm shown in Figure 5.7b, which will take 53 rounds. With K = 3, our algorithm outperforms Datta's as it will take only 40 rounds to converge. An optimal K that achieves the best convergence time under those transient perturbations remains an interesting problem to be further studied.

5.4 Conclusion

In this Chapter, we have presented an analysis of the error dynamics of state estimation collection via an open loop G-C combination, as well as a feedback interconnection of basis blocks serving for leader election.

According to our analysis, even though individual blocks are proved to be stable, their compositions may misbehave under perturbations. For example, in a asynchronous network, the G-C combination have the theoretical potential for exponential overestimates based on duplicated of data and underestimates based on data loss. However, the resilience manifested by leader election algorithm holds the prospect that stable and robust composability can be realized through these basis blocks.

CHAPTER 6 CONCLUSIONS AND FUTURE WORK

6.1 Summary

In this thesis, we have presented the robust stability of G block in aggregate computing and its variants. We prove their global convergence and ultimate boundedness under certain structural perturbations. A new technique *bounding graph* involving *shrunken graphs* and *extended graphs* is used in deriving the ultimate bounds. Specifically, in Chapter 3, we have shown that ABF, GABF and the generalized Gblock are GUAS, MPP is GUES, and all these algorithms are ultimately bounded under persistent structural perturbations. Such perturbations mainly refer to changes in edge values resulting from measurement errors or node mobility.

We have also analyzed two types of compositions of basis blocks in Chapter 5. The first is an open loop G-C combination, which is used to collect the information through the network via a spanning-tree aggregation approach. We find that in the worst case transients can duplicate values leading to exponential overestimates or can drop values leading to near total loss of information. Finally, we design a leader election algorithm via a feedback interconnection of basis blocks. This algorithm does not assume any prior knowledge of the network diameter and is proved to be GUAS as well as resilient under transient perturbations.

6.2 Open problems

We now present some future research directions that are a logical extension of this work.

1. Lyapunov functions for the generalized *G*-block

In Chapter 4, we have proved the GUAS of the generalized G block without using a Lyapunov function. Finding a Lyapunov function is an area of future research. Inverse Lyapunov theorems [45] prove that GUAS implies the existence of a Lyapunov function. Notice that there is a key distinction between standard inverse Lyapunov theory and that for finite time convergence. For example, exponential convergence implies that there exists a positive definite Lyapunov function that is non-increasing and obeys for some T, $L(t+T) - L(t) \leq -\alpha L(t)$ where $\alpha > 0$. On the other hand, discrete time and finite time convergence GUAS seems to imply that L(t) should instead obey: $L(t+T) \leq \min[L(t)-\alpha, 0]$. There is some inverse Lyapunov theory for the finite time convergence of continuous time systems [63], however, inverse theory of the sort is to our knowledge new, and the Lyapunov function we built for ABF in Section 3.2 satisfies this new type of inequality.

2. Ultimate bounds and the time to attain these bounds under perturbations including link failures, change of source set and neighborhood changes.

In Chapter 3 and Chapter 4, ultimate boundedness has been demonstrated for the generalized G block and its variants but only the perturbations in edge lengths. Perturbations like breaks in links, disappearing links and nodes, neighborhood changes and change in the source sets should also be considered. Apart from clear implications to resilience, both the bounds and the time to attain them will play an important role in stability analysis of feedback interconnections.

3. A small gain theorem for closed loop interconnections

In Section 5.3, we have proved the GUAS of a resilient leader election algorithm. Another approach could be to use variants of the small gain theorem in the classical stability literature [45], that indirectly take ultimate bounds into account [41, 42]. Lyapunov functions and finite time convergence of discrete-time discontinuous systems will assist in formulating such small gain theory [52, 20]. In leader election, perpetual source changes may lead to large perturbations in the distance estimates with correspondingly large ultimate bounds. However, such large changes also *take longer to occur* as the sources suppressed or created must be further away from the previous source set. Thus a small gain type analysis that relies just on ultimate bounds must be augmented by a small gain approach utilizing both the ultimate bounds and the time to attain them are used.

REFERENCES

- Sudhanshu Aggarwal and Shay Kutten. Time optimal self-stabilizing spanning tree algorithms. In International Conference on Foundations of Software Technology and Theoretical Computer Science, pages 400–410. Springer, 1993.
- [2] Karine Altisen, Alain Cournier, Stéphane Devismes, Anaïs Durand, and Franck Petit. Self-stabilizing leader election in polynomial steps. *Information and Computation*, 254:330–366, 2017.
- [3] Brian Anderson, Robert R Bitmead, C Richard Johnson Jr, Petar V Kokotovic, Robert L Kosut, Iven MY Mareels, Laurent Praly, and Bradley D Riedle. *Stability* of adaptive systems: Passivity and averaging analysis. MIT press, 1986.
- [4] Brian DO Anderson. The small-gain theorem, the passivity theorem and their equivalence. *Journal of the Franklin Institute*, 293(2):105–115, 1972.
- [5] Murat Arcak. Passivity as a design tool for group coordination. IEEE Transactions on Automatic Control, 52(8):1380–1390, 2007.
- [6] Oktay Arslan and Panagiotis Tsiotras. Use of relaxation methods in samplingbased algorithms for optimal motion planning. In 2013 IEEE International Conference on Robotics and Automation, pages 2421–2428. IEEE, 2013.

- [7] Giorgio Audrito, Ferruccio Damiani, and Mirko Viroli. Optimally-self-healing distributed gradient structures through bounded information speed. In *International Conference on Coordination Languages and Models*, pages 59–77. Springer, 2017.
- [8] Baruch Awerbuch, Shay Kutten, Yishay Mansour, Boaz Patt-Shamir, and George Varghese. Time optimal self-stabilizing synchronization. In STOC, volume 93, pages 652–661. Citeseer, 1993.
- [9] J Baillieul and A Suri. Information patterns and hedging brockett's theorem in controlling vehicle formations. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 556–563. IEEE, 2003.
- [10] J. Beal, D. Pianini, and M. Viroli. Aggregate programming for the internet of things. *IEEE Computer*, 48(9):22–30, September 2015.
- [11] Jacob Beal. Flexible self-healing gradients. In Proceedings of the 2009 ACM symposium on Applied Computing, pages 1197–1201. ACM, 2009.
- [12] Jacob Beal. Superdiffusive dispersion and mixing of swarms with reactive levy walks. In 2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, pages 141–148. IEEE, 2013.
- [13] Jacob Beal and Jonathan Bachrach. Infrastructure for engineered emergence on sensor/actuator networks. *IEEE Intelligent Systems*, 21(2):10–19, 2006.

- [14] Jacob Beal, Jonathan Bachrach, Dan Vickery, and Mark Tobenkin. Fast selfhealing gradients. In Proceedings of the 2008 ACM symposium on Applied computing, pages 1969–1975. ACM, 2008.
- [15] Jacob Beal, Stefan Dulman, Kyle Usbeck, Mirko Viroli, and Nikolaus Correll. Organizing the aggregate: Languages for spatial computing. In Formal and Practical Aspects of Domain-Specific Languages: Recent Developments, pages 436–501. IGI Global, 2013.
- [16] Jacob Beal and Mirko Viroli. Building blocks for aggregate programming of self-organising applications. In Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on, pages 8–13. IEEE, 2014.
- [17] Jacob Beal and Mirko Viroli. Aggregate programming: From foundations to applications. In International School on Formal Methods for the Design of Computer, Communication and Software Systems, pages 233–260. Springer, 2016.
- [18] Richard Bellman. On a routing problem. Quarterly of applied mathematics, 16(1):87–90, 1958.
- [19] Ken Birman. The promise, and limitations, of gossip protocols. ACM SIGOPS Operating Systems Review, 41(5):8–13, 2007.

- [20] Ruxandra Valentina Bobiti and Mircea Lazar. On input-to-state stability analysis of discrete-time systems via finite-time lyapunov functions. *IFAC Proceedings Volumes*, 47(3):8623–8628, 2014.
- [21] Dirk Brockmann and Dirk Helbing. The hidden geometry of complex, networkdriven contagion phenomena. *science*, 342(6164):1337–1342, 2013.
- [22] Janna Burman and Shay Kutten. Time optimal asynchronous self-stabilizing spanning tree. In International Symposium on Distributed Computing, pages 92–107. Springer, 2007.
- [23] Ming Cao, Changbin Yu, A Stephen Morse, BDO Anderson, and Soura Dasgupta. Generalized controller for directed triangle formations. *IFAC Proceedings Volumes*, 41(2):6590–6595, 2008.
- [24] Arnaud Casteigts, Yves Métivier, John Michael Robson, and Akka Zemmari.
 Deterministic leader election in O(D + log n) time with messages of size O(1).
 In International Symposium on Distributed Computing, pages 16–28. Springer, 2016.
- [25] Ferruccio Damiani, Mirko Viroli, and Jacob Beal. A type-sound calculus of computational fields. Science of Computer Programming, 117:17–44, 2016.

- [26] Ferruccio Damiani, Mirko Viroli, Danilo Pianini, and Jacob Beal. Code mobility meets self-organisation: A higher-order calculus of computational fields. In International Conference on Formal Techniques for Distributed Objects, Components, and Systems, pages 113–128. Springer, 2015.
- [27] Soura Dasgupta and Jacob Beal. A lyapunov analysis for the robust stability of an adaptive bellman-ford algorithm. In *Decision and Control (CDC)*, 2016 *IEEE 55th Conference on*, pages 7282–7287. IEEE, 2016.
- [28] Yuanqiu Mo, Soura Dasgupta, and Jacob Beal. A lyapunov analysis for the robust stability of the most probable path algorithm, Under preparation.
- [29] Ajoy K Datta, Lawrence L Larmore, and Priyanka Vemula. An o (n)-time selfstabilizing leader election algorithm. Journal of Parallel and Distributed Computing, 71(11):1532–1544, 2011.
- [30] Ajoy K Datta, Lawrence L Larmore, and Priyanka Vemula. Self-stabilizing leader election in optimal space under an arbitrary scheduler. *Theoretical Computer Science*, 412(40):5541–5561, 2011.
- [31] Edsger W Dijkstra. A note on two problems in connexion with graphs. Numerische mathematik, 1(1):269–271, 1959.
- [32] Shlomi Dolev. *Self-stabilization*. MIT press, 2000.

- [33] Barış Fidan, Soura Dasgupta, and Brian DO Anderson. Adaptive rangemeasurement-based target pursuit. International Journal of Adaptive Control and Signal Processing, 27(1-2):66–81, 2013.
- [34] Lester R Ford Jr. Network flow theory. Technical report, RAND CORP SANTA MONICA CA, 1956.
- [35] Matteo Francia, Danilo Pianini, Jacob Beal, and Mirko Viroli. Towards a foundational api for resilient distributed systems design. In 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W), pages 27–32. IEEE, 2017.
- [36] Minyue Fu and Soura Dasgupta. Parametric lyapunov functions for uncertain systems: The multiplier approach. In Advances in linear matrix inequality methods in control, pages 95–108. SIAM, 2000.
- [37] Minyue Fu, Soura Dasgupta, and Yeng Chai Soh. Integral quadratic constraint approach vs. multiplier approach. *Automatica*, 41(2):281–287, 2005.
- [38] R. G. Gallager. Finding a leader in network with O(E) + O(NlogN) messages. Unpublished note, MIT, 1977.
- [39] Wolfgang Hahn. Stability of motion, volume 138. Springer, 1967.
- [40] Takeshi Hatanaka, Nikhil Chopra, Masayuki Fujita, and Mark W Spong. Passivity-based control and estimation in networked robotics. Springer, 2016.

- [41] Zhong-Ping Jiang, Iven MY Mareels, and Yuan Wang. A lyapunov formulation of the nonlinear small-gain theorem for interconnected iss systems. *Automatica*, 32(8):1211–1215, 1996.
- [42] Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. Automatica, 37(6):857–869, 2001.
- [43] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. The international journal of robotics research, 30(7):846–894, 2011.
- [44] Hicham Khalife, Satyajeet Ahuja, Naceur Malouch, and Marwan Krunz. Probabilistic path selection in opportunistic cognitive radio networks. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pages 1–5. IEEE, 2008.
- [45] Hassan K Khalil. Noninear systems. Prentice-Hall, New Jersey, 2(5):5–1, 1996.
- [46] Sven Koenig and Maxim Likhachev. D* lite. Aaai/iaai, 15, 2002.
- [47] Sven Koenig, Maxim Likhachev, and David Furcy. Lifelong planning A*. Artificial Intelligence, 155(1-2):93–146, 2004.
- [48] Alex Kravchik and Shay Kutten. Time optimal synchronous self stabilizing spanning tree. In International Symposium on Distributed Computing, pages 91–105. Springer, 2013.

- [49] Laura Krick, Mireille E Broucke, and Bruce A Francis. Stabilisation of infinitesimally rigid formations of multi-robot networks. *International Journal of control*, 82(3):423–439, 2009.
- [50] Amy Kumar, Jacob Beal, Soura Dasgupta, and Raghu Mudumbai. Toward predicting distributed systems dynamics. In Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2015 IEEE International Conference on, pages 68– 73. IEEE, 2015.
- [51] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. On the complexity of universal leader election. *Journal of the ACM* (*JACM*), 62(1):7, 2015.
- [52] Mircea Lazar, WP Maurice H Heemels, and Andy R Teel. Lyapunov functions, stability and input-to-state stability subtleties for discrete-time discontinuous systems. *IEEE Transactions on Automatic Control*, 54(10):2421–2425, 2009.
- [53] Gérard Le Lann. Distributed systems-towards a formal approach. In IFIP congress, volume 7, pages 155–160. Toronto, 1977.
- [54] Marc Lee and Vincent WS Wong. An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks. In PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005., pages 300–303. IEEE, 2005.

- [55] Whay Chiou Lee. Spanning tree method for link state aggregation in large communication networks. In *Proceedings of INFOCOM'95*, volume 1, pages 297– 302. IEEE, 1995.
- [56] Yuanqiu Mo, Jacob Beal, and Soura Dasgupta. Error in self-stabilizing spanningtree estimation of collective state. In *Foundations and Applications of Self* Systems (FAS* W), 2017 IEEE 2nd International Workshops on*, pages 1–6. IEEE, 2017.
- [57] Yuanqiu Mo, Soura Dasgupta, and Jacob Beal. Robust stability of spreading blocks in aggregate computing. In 2018 IEEE Conference on Decision and Control (CDC), pages 6007–6012. IEEE, 2018.
- [58] Yuanqiu Mo, Soura Dasgupta, and Jacob Beal. Robustness of the adaptive bellman-ford algorithm: Global stability and ultimate bounds. *IEEE Transactions on Automatic Control*, 2019.
- [59] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [60] Michael Otte and Emilio Frazzoli. Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning. *The International Journal of Robotics Research*, 35(7):797–822, 2016.

- [61] David Peleg. Time-optimal leader election in general networks. Journal of parallel and distributed computing, 8(1):96–99, 1990.
- [62] Danilo Pianini, Jacob Beal, and Mirko Viroli. Improving gossip dynamics through overlapping replicates. In International Conference on Coordination Languages and Models, pages 192–207. Springer, 2016.
- [63] Andrei Polyakov and Alex Poznyak. Lyapunov function design for finite-time convergence analysis: "twisting" controller for second-order sliding mode realization. Automatica, 45(2):444–448, 2009.
- [64] Enrico Ser-Giacomi, Ruggero Vasile, Emilio Hernández-García, and Cristóbal López. Most probable paths in temporal weighted networks: An application to ocean transport. *Physical review E*, 92(1):012818, 2015.
- [65] Devavrat Shah et al. Gossip algorithms. Foundations and Trends® in Networking, 3(1):1–125, 2009.
- [66] Anthony Stentz et al. The focussed D* algorithm for real-time replanning. In IJCAI, volume 95, pages 1652–1659, 1995.
- [67] Tyler H Summers, Changbin Yu, Brian DO Anderson, and Soura Dasgupta.
 Formation shape control: Global asymptotic stability of a four-agent formation.
 In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, pages 3002–3007. IEEE, 2009.

- [68] Tyler H Summers, Changbin Yu, Soura Dasgupta, and Brian DO Anderson. Control of minimally persistent leader-remote-follower and coleader formations in the plane. *IEEE Transactions on Automatic Control*, 56(12):2778–2792, 2011.
- [69] Hüseyin Ozgür Tan and Ibrahim Körpeoğlu. Power efficient data gathering and aggregation in wireless sensor networks. ACM Sigmod Record, 32(4):66–71, 2003.
- [70] Mathukumalli Vidyasagar. Decomposition techniques for large-scale systems with nonadditive interactions: Stability and stabilizability. *IEEE transactions* on automatic control, 25(4):773–779, 1980.
- [71] Mirko Viroli, Giorgio Audrito, Jacob Beal, Ferruccio Damiani, and Danilo Pianini. Engineering resilient collective adaptive systems by self-stabilisation. ACM Transactions on Modeling and Computer Simulation (TOMACS), 28(2):16, 2018.
- [72] Yinyan Zhang and Shuai Li. Distributed biased min-consensus with applications to shortest path planning. *IEEE Transactions on Automatic Control*, 62(10):5429–5436, 2017.
- [73] Hubert Zimmermann. Osi reference model-the iso model of architecture for open systems interconnection. *IEEE Transactions on communications*, 28(4):425–432, 1980.