# A Lyapunov Analysis for the Robust Stability of an Adaptive Bellman-Ford Algorithm

Soura Dasgupta and Jacob Beal*

## Abstract

*Self-stabilizing (asymptotically stable) distance estimation algorithms are an important building block of many distributed systems featuring in Spatial or Aggregate computing, but the dynamics of their convergence to correct distance estimates has not previously been formally analyzed. As a first step to understanding, how they behave in interconnections involving other building blocks, it is important to develop a Lyapunov framework to demonstrate their robust stability. This paper addresses this shortcoming by providing the first Lyapunov-based analysis of an adaptive Bellman-Ford algorithm, by formulating a simple Lyapunov function. This analysis proves global uniform asymptotic stability of such algorithms, a property which the classical Bellman-Ford algorithm lacks, thus demonstrating a measure of robustness to structural perturbations, empirically observed by us in a previous work.*

## 1. Introduction

Our world is increasingly dependent on complex networked and distributed systems, often composed of many different subsystems (physical or logical) that are themselves distributed. The stability, safety, and dynamical behavior of such systems is of great importance, but in general, such analysis of arbitrary compositions of arbitrary distributed algorithms appears to be an intractable problem.

In the controls literature the study of the robust stability of limited classes of interconnections of large scale distributed systems using a mature set of tools in stability theory, [1], dates back decades, [2]. More recently interest has been rekindled by the modern incarnation of the consensus literature, (see [3] and references therein) and formation control, (see [4] - [6] and references therein).

This paper is motivated by the interconnections of distributed algorithms appearing in the emerging field of *aggregate or spatial computing.* One promising recent approach in studying such interconnections has been to simplify the problem by a generative approach using a basis set of "building block" distributed algorithms [7–9]. Rather than attempting to analyze arbitrary systems, this approach identifies a set of distributed algorithms with properties that can be predicted for arbitrary compositions. Any system that can be mapped onto such a basis set then inherits its properties. While empirically the dynamics of these basis set systems appear to be amenable to effective composition [10–13], to date formal analysis has been limited to eventually-true properties such as self-stabilization. In the parlance of stability theory, these results demonstrate asymptotic stability of each individual block. Yet as a first step in understanding the robust stability of interconnections, it is important to understand how these individual systems behave under perturbations.

More precisely are their stability properties robust to these perturbations? Does stability in the ideal unperturbed setting translate to acceptable behavior in the face of perturbations? Such robust behavior cannot be deduced by the mere demonstration of asymptotic stability. Rather, as is now well understood in the adaptive systems literature, [14], [15], [16] one must instead show *uniform asymptotic stability* of the unperturbed system. This is so as uniform asymptotic stability implies total stability, [17], an ability to sustain modest departures from idealizing assumptions.

This paper begins to address this shortcoming, taking a Lyapunov-based approach toward a framework for analyzing the stability, safety, and dynamical behavior of arbitrary compositions of a basis set of distributed algorithms. For this first analysis presented in this paper, we focus on distributed distance estimates of the nodes of a undirected graph from a set of source nodes. This is a widely used building block, and is in particular a continuously adaptive variant of the Bellman-Ford

*Soura Dasgupta is with the Department of of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242, USA. He is also with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks. dasgupta@engineering.uiowa.edu. Jacob Beal is with the Raytheon BBN Technologies Cambridge, MA, USA 02138. jakebeal@bbn.com.

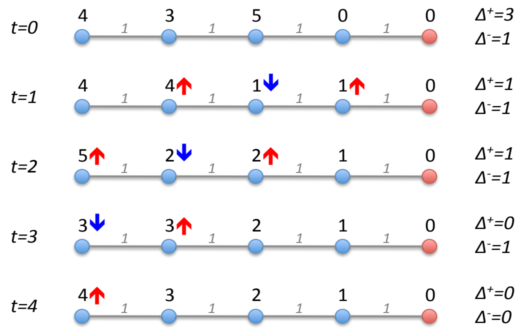



Figure 1: Illustration of the Adaptive Bellman-Ford algorithm. From [10]. Individual distance estimates may go up and down, but the greatest overestimate ($\Delta^+$) and least underestimate ($\Delta^-$) are monotonic. This example shows a line network of five devices (circles, source red, others blue) 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.

algorithm [18], [19]. As explained later this variant is needed as the classical Bellman-Ford algorithm is not globally uniformly asymptotically stable. Here, we propose a novel Lyapunov function, hypothesized on the basis of experiments, but never formally analyzed in our earlier work [10] and use it to demonstrate the global uniform asymptotic convergence of the distance estimates to their true values. In doing so we thus formally explain empirical observations made in [10], and validate the robustness observed there to structural perturbations of the underlying graph.

In Section 2 we introduce the graph framework and state the assumptions underlying this work. Section 3 introduces what we call an Adaptive Bellman-Ford algorithm and explains why the traditional Bellman-Ford algorithm is not a suitable building block of the framework postulated in [7–9]. Section 4 provides the Lyapunov analysis. Section 5 consolidates the analysis with implications to robustness. Section 6 concludes and outlines an agenda for future work. Most proofs are omitted due to space constraints.

## 2. Framework and Assumptions

We consider undirected graphs of the type depicted in Figure 1. We say that node $i$ is a neighbor of node $j$ if an edge exists between $i$ and $j$. The set of neighbors of node $i$ are $\mathscr{N}(i)$. If $j \in \mathscr{N}(i)$ then $i \in \mathscr{N}(j)$. Further

$$i \notin \mathscr{N}(i),$$

i.e. no node is deemed to be its own neighbor.

In the sequel $e_{ij}$ denotes the *edge length between the neighbors i and j*. We will assume that

$$e_{ij} > 0, \; \forall j \in \mathscr{N}(i), \tag{1}$$

i.e. distinct neighbors have nonzero edge lengths between them. Thus in the graph in Figure 1 each edge length is 1. Further, The distance $d_{ij}$ between two nodes is the shortest walk from $i$ to $j$. Thus, the distance between the third and the last nodes in Figure 1 is 2. From the principle optimality one has the recursion

$$d_{ij} = \min_{k \in \mathscr{N}(i)} \{e_{ik} + d_{jk}\}. \tag{2}$$

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*. Our goal is to find the distance between each node and the source set $S$. In particular we must find for each node $i$, $d_i$ the distance between the source set and node $i$. More precisely we must find

$$d_i = \min_{k \in S} \{d_{ik}\}. \tag{3}$$

In view of (2) $d_i$ obeys the recursion

$$d_i = \min_{k \in \mathscr{N}(i)} \{e_{ik} + d_k\} \tag{4}$$

Observe

$$d_i = 0, \; \forall i \in S. \tag{5}$$

The distance estimation we desire must be recursive and distributed in the sense that in executing the algorithm at time $t$, the $i$-th node knows only

(A) Its edge length from each of its neighbors.

(B) The current estimated distance of its neighbors from the source set, i.e. $\hat{d}_j(t)$ for all $j \in \mathscr{N}(i)$.

The classical Bellman-Ford algorithm [18, 19], is a well known solution to this problem. We will explain in Section 3 why this algorithm fails to meet our requirements and will present instead an adaptive version of this algorithm which we have empirically studied in [10] both in isolation and in interconnections in some cases involving feedback loops.

## 3. Adaptive Bellman-Ford Algorithm

In the classical Bellman-Ford algorithm [18, 19] 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. for all $i$

$$\hat{d}_i(0) \geq d_i. \tag{6}$$

*Thus by definition the classical Bellman-Ford algorithm is not globally uniformly asymptotically stable.* In the dynamic interconnected environment the inputs may be graph topology, or even the source set, and indeed these may change over time. Consequently at a given instant the current estimate may well fall below the true current distance. The classical Bellman-Ford simply cannot cope with such perturbations, and hence the adaptive version studied here.

The algorithm we empirically studied in [10] is based closely on the classic Bellman-Ford algorithm, but unlike the classical algorithm computes distances to the nearest member of a set of source nodes rather than just a single node. Moreover, we wish to be able to support the case where either the set of sources or the graph may change. It is thus an *adaptive algorithm* that addresses these differences by a) setting the distance estimate of every source node to zero, and b) for all other devices, rather than starting at infinity and always decreasing, recomputes distance estimates periodically, ignoring the current estimate at a device and using only the minimum of the triangle inequality constraints of its neighbors. For simplicity in analysis, we formulate this algorithm in terms of synchronized rounds of computation (though in practice it is typically executed without synchronization).

In particular, suppose $\hat{d}_i(t)$ is the current estimated distance of $i$ from the source set. Then the algorithm is

$$\hat{d}_i(t+1) = \begin{cases} \min_{j\in\mathcal{N}(i)} \left\{\hat{d}_j(t) + e_{ij}\right\} & i \notin S \\ 0 & i \in S \end{cases} \tag{7}$$

Observe (7) respects the information structure imposed in (A) and (B) of Section 2, and seeks to emulate (4) treating the available distance estimates as their true values.

Note that the behavior of this algorithm reduces to something very close to classical Bellman-Ford in the case where there is precisely one source device and neither the graph nor the source ever change. We have previously presented an empirical analysis and experimental results on the dynamics of this algorithm in [10]; in this paper we extend to formal analysis of the relevant properties.

## 4. A Lyapunov Based Analysis

Under the initialization (6), the classical Bellman-Ford is known to be self stabilizing. In other words absent any topological changes the estimated distances are known to asymptotically converge to their true value.

As noted earlier this algorithm is ill-equipped to deliver the sort of robustness to perturbations we seek especially in the face of interconnections of program blocks, interconnections of the type empirically studied in [10]. Even for the adaptive algorithm in (7) mere asymptotic convergence, for a fixed graph, does not suffice for robustness. As argued in [14] uniform asymptotic, rather than just asymptotic convergence in the ideal case of a fixed graph is needed for robustness of adaptive algorithms.

Of course one device for establishing uniform asymptotic stability is to formulate a suitable Lyapunov function [1]. Indeed this section formulates such a Lyapunov function. An obvious measure of algorithm performance is of course the distance estimation error,

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

However, an instance of (7) depicted in Figure 1 shows that $\Delta_i(t)$ may well increase in magnitude for individual nodes. On the other hand at least for the particular instance of the algorithm considered in Figure 1 each of the greatest overestimate of the error $\Delta^+$ and the least underestimate $\Delta^-$, defined below, appear to be non-increasing

$$\Delta^+(t) = \max\left[0, \max_i \Delta_i(t)\right] \tag{8}$$

$$\Delta^-(t) = \max\left[0, -\min_i \Delta_i(t)\right]. \tag{9}$$

This together with their eventual decrescence has been empirically verified in [10]. Consequently, in this section we examine the behavior of the candidate Lyapunov function

$$L(t) = \Delta^+(t) + \Delta^-(t). \tag{10}$$

That this is a valid Lyapunov function follows from the fact that clearly

$$L(t) \geq 0 \tag{11}$$

with equality holding iff for all $l$,

$$\Delta_l(t) = 0.$$

Indeed we formally verify the empirical observations of [10], in the belief that the Lyapunov framework thus established can be exploited in future work to quantify the effect of perturbations on (7).

Our analysis requires some further definitions. In the sequel $k$ in (4) will be called a *true constraining* *node of $i$* while $j$ in (7) will be called a *constraining node of $i$*. Evidently, in view of (4)

$$d_l < d_i \tag{12}$$

whenever $l$ is a true constraining node of $i$.

Further, we will define as $k_+(t)$ a node that has the maximum error. More precisely, $k_+(t)$ is an index for which $\Delta^+(t) = \Delta_{k_+(t)}(t)$, i.e.

$$k_+(t) = \arg\max_i \{\Delta_i(t)\}. \tag{13}$$

Similarly, $k_-(t)$ is an index for which $\Delta^-(t) = \Delta_{k_-(t)}(t)$, i.e.

$$k_-(t) = \arg\min_i \{\Delta_i(t)\}. \tag{14}$$

We will first show the non-increasing nature of $\Delta^+(t)$.

**Lemma 1.** *Consider (7) with the assumptions stated in Section 2. Then with $\Delta^+$ defined in (8), for all $t$, $\Delta^+(t+1) \leq \Delta^+(t)$. Further, equality holds only if $k_+(t)$ defined in (13) is the true constraining node of $k_+(t+1)$.*

**Remark 1.** *In fact the omitted proof shows that $\Delta^+(t+1) = \Delta^+(t)$ iff $k_+(t)$ defined in (13) is the true constraining node as well as the constraining node of $k_+(t+1)$.*

Next we prove the non-increasing nature of $\Delta^-(t)$.

**Lemma 2.** *Consider (7) with the assumptions stated in Section 2. Then with $\Delta^-$ defined in (9), for all $t$, $\Delta^-(t+1) \leq \Delta^-(t)$. Further, equality holds only if $k_-(t)$ defined in (14) is the true constraining node of $k_-(t+1)$.*

**Remark 2.** *Again the omitted proof shows that $\Delta^-(t+1) = \Delta^-(t)$ iff $k_-(t)$ defined in (14) is the true constraining node as well as the constraining node of $k_-(t+1)$.*

Lemma 1 and Lemma 2 together show that for all $t$, the Lyapunov function $L(t)$ in (11) obeys for all $t$

$$L(t+1) \leq L(t). \tag{15}$$

The proofs also show that conditions for equality in (15) are quite stringent. The next lemma helps to show that in fact in a connected graph these conditions cannot be sustained beyond a number of steps equaling at most the diameter of the graph.

**Lemma 3.** *With every node connected to the source set $S$, suppose there is a sequence of nodes $k(t)$ such that, for all $t \in \{t_0+1, t_0+2, \cdots, t_0+T\}$, $k(t-1)$ is a true constraining node of $k(t)$. Then $T$ is no larger than the diameter of the graph.*

Lemma 1 to Lemma 3 can be used to prove the uniform decrescence of $L(t)$ in (10), when the graph is fixed.

**Theorem 1.** *Suppose in the underlying graph, each node is connected to the source set $S$. Consider the Lyapunov function in (10) under (8) and (9). Then under (7) there exists a positive integer $T$ no larger than the diameter of the graph, and an $\alpha > 0$, possibly dependent on the initial conditions, but not on the initial time, such that $L(t)$ in (10) obeys*

$$L(t+T) - L(t) \leq \begin{cases} -\alpha & L(t) > 0 \\ 0 & L(t) = 0 \end{cases}. \tag{16}$$

This confirms the empirical observation of [10] that $\Delta^+(t)$ and $\Delta^-(t)$ are non-increasing and must decline over time intervals bounded from above by the graph diameter. As importantly, from systems theory perspective, Theorem 1 establishes $L(t)$ in (10) as a valid Lyapunov function. Though, as with most Lyapunov analyses Theorem 1 is conservative, particularly in the bound on $T$, and possibly $\alpha$, though it does not quantify $\alpha$, it is used in the next theorem to establish the promised uniform asymptotic stability.

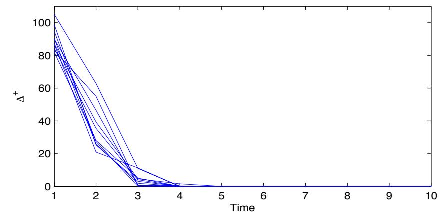
**Theorem 2.** *Suppose in the underlying graph, each node is connected to the source set $S$. Then the algorithm in (7) induces all $\hat{d}_i(t)$ to converge globally, uniformly, asymptotically to zero.*

*Proof.* As $L(t)$ is non-negative (16) ensures that $L(t)$ and hence all $\Delta_l(t)$ globally converge to zero. As neither $\alpha$ nor $T$, depend on the initial time, uniformity follows. □

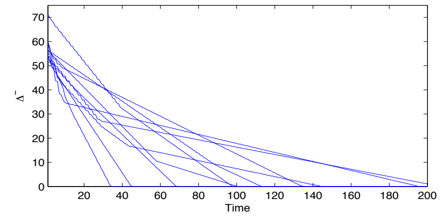
## 5. Discussion and Implication to Robustness

We now discuss various aspects of Theorem 1 and Theorem 2. First as already noted by its very nature Lyapunov analysis caters to worst case scenarios, and is prone to be conservative. For example, $T$ the time it takes $\Delta^+$ and $\Delta^-$ to strictly decrease, is upper bounded by the diameter of the graph. Yet, simulations from [10] involving 200-node disk graphs, reproduced in Figure 2, show that the overestimate $\Delta^+(t)$ converges at a much faster rate. In these graphs the nodes are uniformly distributed over a 100x100 meter surface, with a communication range of 15 meters.

By contrast, the convergence rate of the underestimate $\Delta^-(t)$ is slower. This has been attributed in [12] and [10] to the so called "rising value problem". Whereas it is impossible to have a periodic cycle where each node is a true constraining node of its successor, there can be cycles where each node is a constraining node of its successor. Indeed transmission lags can create such loops which restrict the rise of $\hat{d}_i(t)$ values to




(a) Greatest overestimate ($\Delta^+$)




(b) Least underestimate ($\Delta^-$)

Figure 2: Plot from [10] of greatest overestimate ($\Delta^+$) (a) and least underestimate $\Delta^-$, (b) for 10 runs of 200 devices uniformly distributed in a 100x100 meter surface, with a communication range of 15 meters.

as little as

$$\frac{1}{2} \min_{i,j \in \mathcal{N}(i)} e_{ij}$$

per round.

Despite this fact the results in the previous section are conservative. For example in the proofs, we have only utilized the fact that strict decresence of the over and underestimates requires that the node sequence with largest over and underestimates be such that each be a true constraining node of its successor. As noted in Remark 1 and Remark 2, the prevention of decresence has the additional requirement that in these sequences each must *also be a constraining node of its successor.*

Yet Theorem 1 does formally prove robustness to perturbations in the graph topology. For example, should perturbations be upper bounded by $\alpha$ in Theorem 1, and be less frequent than $T$, then the Adaptive Bellman-Ford algorithm should sustain them. Again this is conservative, as for example decline may often exceed the minimal value of $\alpha$ and may occur more frequently than a diameter interval. Similarly, in a large graph, by the time the effect of a perturbation at a remote point propagates through the graph, the inherent stabilizing property of (7) attenuates it. Indeed [10] documents several instances of this type, and even shows the ability to sustain perturbations induced by feedback interconnections of multiple blocks, involving the separate execution of (7) in these blocks.

# 6. Conclusion

Motivated by the recent recognition of the need to study the robustness properties of certain ubiquitous building blocks in spatial or aggregate computing, we formally analyze one such block, namely the Adaptive Bellman-Ford algorithm. We formulate a simple Lyapunov function for this algorithm and use it to demonstrate uniform asymptotic convergence and discuss its implications to robustness experimentally verified in [10].

We regard this as a first step toward a similar Lyapunov based study of other building blocks, and eventually to their interconnections, possibly involving feedback. In the long run we anticipate even the notion of identifying suitable analogs of the celebrated passivity, [20], or small gain, [21], theorems.

# 7. Acknowledgments


This work has been supported by the United States Air Force and the Defense Advanced Research Projects Agency under Contract No. FA8750-10-C-0242, US NSF grants EPS-1101284, CNS-1329657, CCF-1302456, ONR grant N00014-13-1-0202 and under the Thousand Talents Program of the State and Shandong Province, administered by the Shandong Academy of Sciences, China. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for public release; distribution is unlimited.


# References


[1] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, 2002.

[2] M. Vidyasagar, "Decomposition techniques for large-scale systems with nonadditive interactions: Stability and stabilizability," *IEEE Transactions on Automatic Control*, vol. 25, pp. 773 – 779, 1980.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215 – 233, 2007.

[4] J. Baillieul and A. Suri, "Information patterns and hedging brockett's theorem in controlling vehicle formations," in *Proceedings of 42nd IEEE Conference on Decision and Control*, 2003.

[5] S. Dasgupta, and B. D. O. Anderson, "Controlling rectangular formations", in *Proceedings of AUCC*, 2011, Melbourne, Australia.

[6] T. H. Summers, C. Yu, S. Dasgupta, and B. D. O. Anderson, "Control of minimally persistent leader-remote-follower and coleader formations in the plane," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2778 – 2792, 2011.

[7] J. Beal and M. Viroli, "Building blocks for aggregate programming of self-organising applications," in *Workshop on Foundations of Complex Adaptive Systems (FOCAS)*, 2014.

[8] M. Viroli and D. Ferruccio, "A calculus of self-stabilising computational fields," in *Coordination 2014*, 2014, pp. 163–178.

[9] J. Beal, D. Pianini, and M. Viroli, "Aggregate programming for the internet of things," *IEEE Computer*, vol. 48, no. 9, pp. 22–30, September 2015.

[10] A. Kumar, J. Beal, S. Dasgupta, and R. Mudumbai, "Toward predicting distributed systems dynamics," in *Spatial and COllective PErvasive Computing Systems (SCOPES)*. IEEE, September 2015, pp. 68–73.

[11] M. Viroli, J. Beal, F. Damiani, and D. Pianini, "Efficient engineering of complex self-organizing systems by self-stabilising fields," in *IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE, September 2015, pp. 81–90.

[12] J. Beal, J. Bachrach, D. Vickery, and M. Tobenkin, "Fast self-healing gradients," in *ACM Symp. on Applied Computing*, 2008.

[13] J. Beal, "Flexible self-healing gradients," in *Proceedings of the 2009 ACM Symposium on Applied Computing*. 2009, pp. 1197–1201, ACM.

[14] B. D. O. Anderson, R. R. Bitmead, C. R. Johnson, P. V. Kokotovic, R. L. Kosut, I. M. Y. Mareels, L. Praly, and B. D. Riedle, *Stability of Adaptive Systems: Passivity and Averaging Analysis*, MIT Press, 1986.

[15] Dasgupta, S., Anderson, B.D.O and Kaye, R.J., "Identification of physical parameters in structured systems," *Automatica*, March 1988, vol. 24, pp 217-225.

[16] S. Dasgupta, C. R. Johnson Jr and A.M. Baksho, "Sign-sign LMS convergence with independent stochastic inputs", *IEEE Transactions on Information Theory*, pp. 197-201, 1990.

[17] W. Hahn, *Stability of Motion*, Prentice Hall, 1967.

[18] R. E. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[19] L. R. Ford Jr., "Network flow theory," Tech. Rep. Paper P-923, RAND Corporation, 1956.

[20] M. Fu and S. Dasgupta, "Parametric lyapunov functions for uncertain systems: The multiplier approach," *Advances in linear matrix inequality methods in control*, pp. 95–108, 2000.

[21] Z. Jiang, I. M. Y. Mareels, and Y. Wang, "A lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems," *Automatica*, vol. 32, pp. 1211 – 1215, 1996.