Improving Collection Dynamics by Monotonic Filtering

(Article begins on next page)

21 February 2025

Improving Collection Dynamics by Monotonic Filtering

Hunza Zainab*, Giorgio Audrito†, Soura Dasgupta*, Jacob Beal‡

*Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA
email: hunza-zainab@uiowa.edu, soura-dasgupta@uiowa.edu
†Computer Science Department and C3S, University of Torino, Torino, Italy
email: giorgio.audrito@unito.it
‡Raytheon BBN Technologies, Cambridge, MA, USA
email: jakebeal@ieee.org

Abstract—A key coordination problems in distributed open systems is distributed sensing, as achieved by cooperation and interaction among individual devices. An archetypal operation of distributed sensing is data summarization over a region of space, by which many higher level problems can be addressed, including counting items, measuring space, averaging environmental values, etc. A typical coordination strategy to perform data summarization in a peer-to-peer scenario, where devices can communicate only with a neighborhood, is to progressively accumulate information towards one or more collector devices, though this typically exhibits problems of reactivity and fragility. In this paper, we present a *monotonic filtering* strategy for improving the dynamics of single path collection algorithms. The strategy consists of inhibiting communication across devices whose distance towards the collector device is not decreasing. We prove that single path collection in a line graph results in quadratic overestimates after a source change and that these overestimates disappear with the application of monotonic filtering. These preliminary results suggest that monotonic filtering is likely to improve the dynamics of single-path collection algorithms, by preventing excessive overestimates.

Index Terms—edge computing, data aggregation, self-stabilisation

I. Introduction

Physical environments are increasingly filled with heterogeneous connected devices (intelligent and mobile, such as smartphones, drones, robots, and IoT devices). Such settings increasingly call for new mechanisms of collective adaptation, ultimately supporting a view of environments as acting as true *pervasive computing fabric*, where sensing, actuation, and computation are naturally seen as inherently resilient and distributed across physical space [1]. In this paper, we consider the design of a self-adaptive coordination strategy able to realize *distributed sensing* of physical properties of the environment or virtual/digital characteristics of computational resources. By the strict cooperation and interaction of dynamic sets of mobile entities situated in physical proximity, distributed sensing can generally support forms of complex situation recognition [2], better monitoring of the physical environment [1], and observation (and then control) of teams of agents [3].

A paradigmatic coordination operation of distributed sensing is data summarization performed on devices filling a region of space. This is a key component with which one can then realize other operations such as counting, integration, averaging, maximization, and the like. In fact, data summarization corresponds to the *reduce* phase of the MapReduce paradigm [4], extended into a "spatial" context of agents spread in a physical environment and communicating by proximity, and has close analogs for wireless sensor networks [5]. Data summarization can be performed by *distributed collection*, where information propagates towards one or more collector devices, and combines *en-route* until reaching a unique value, i.e., the result of collection. This component of self-organizing behavior (also referred to as the "C" building block [6]), is one of the most basic and widely used components of collective adaptive systems (CASs), as it can be instantiated for values of any data type with an associative and commutative aggregation operator, and can be applied to a wide variety of different contexts.

A number of recent papers have focused on characterizing the convergence dynamics of data summarization algorithms [6], [7] and and improving such dynamics [8]–[10]. A common result across these works is that, despite being self-stabilizing, such algorithms can give rise to excessive transient overestimates. Such overestimates can have deleterious effects in many settings. For example if the goal is to apprise a leader of the net resources in a network, then temporary overestimates may cause a leader to agree to accept more tasks than the network is capable of performing.

In this paper, we address this issue by presenting a *monotonic filtering* strategy for improving the dynamics of single-path collection algorithms by removing overestimations. This strategy operates by inhibiting communication between devices whose distance towards the collector device is not

Table I
TABLE OF SYMBOLS

Symbol	Property
$d_i(t)$	Distance estimate
d_i	True distance
$c_i(t)$	Constraining node
c_i	True constraining node
$C_i(t)$	Set of nodes constrained by i at time t
$a_i(t)$	Partial accumulate
$S(t)$	Set of sources
e_{ij}	Length of the edge between i and j
v_i	Input value in node i
$N(i)$	Set of neighbour nodes of i
V	Set of all vertices

decreasing. We prove that single-path collection in a line graph can incurs quadratic overestimates after a source change; the provide that these overestimates disappear when monotonic filtering is applied. These preliminary results suggest that monotonic filtering could improve the dynamics of single-path collection algorithms, by preventing excessive overestimates.

The remainder of this paper is structured as follows. Section II presents the background and related work on the building blocks investigated in this paper. Section III shows that single-path collection can incur quadratic overestimates during transients. Section IV shows that the introduction of monotonic filtering removes overestimates for line graphs, and Section V discusses conclusions.

II. BACKGROUND AND RELATED WORK

Recent works have promoted an approach to engineering complex field-based coordination algorithms by combination of basic building blocks [6], capturing key mechanisms of self-organisation such as spreading (block "G"), collection (block "C"), time evolution (block "T"), and leader election and partitioning (block "S").

The most basic and versatile building block is called *gradient* (G block), which provides distance estimation, creation of spanning trees, and execution of broadcast operations. In particular, the estimated distances from a source are a crucial input of every data aggregation routine (C block), providing means to guide the direction of aggregation. Accurately computing distances in a distributed and volatile scenario is a demanding task, which can be tackled in different ways depending on the context [11], [12]. In this paper, we focus on the most basic *adaptive Bellman-Ford* (ABF) algorithm [6], [13], although the monotonic filtering strategy may as well be applied to single-path collection algorithms using other distance estimation routines as input. In this algorithm, the distance estimate $d_i(t)$ of node i at time t is computed as
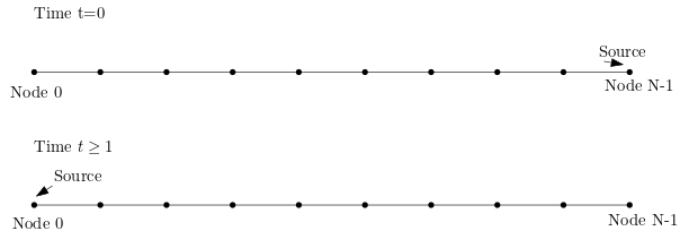
Time t=0

Time $t \geq 1$

Figure 1. Representation of an N-node line graph (N-line) with a source switch from time $t = 0$ to $t = 1$.

follows:[1]

$$d_i(t) = \begin{cases} 0 & \text{if } i \in S(t) \\ d_j(t-1) + e_{ji} \text{ for } j = c_i(t) & \text{otherwise} \end{cases} \quad (1)$$

$$c_i(t) = \begin{cases} i & \text{if } i \in S(t) \\ \underset{j \in N(i)}{\operatorname{argmin}}\{d_j(t-1) + e_{ji}\} & \text{otherwise} \end{cases} \quad (2)$$

where $S(t)$ is the set of *sources*, e_{ij} is the distance between neighbouring nodes i and j, $c_i(t)$ is the *constraining node* of i at time t, and $N(i)$ is the set of neighboring nodes of i (c.f. Table I). Notice that the ABF algorithm produces two different outputs: the *distance estimates* $d_i(t)$ and the *constraining nodes* $c_i(t)$. Both of these evolve with time t until reaching the self-stabilising limit d_i, c_i for sufficiently large t [13].

The constraining nodes implicitly define a *spanning tree* in a network: every node i has exactly one ancestor in the spanning tree (its true constraining node c_i, assuming stable tie-breaking), except for the source (which becomes the root of the tree). This spanning tree can be used to guide a process of data collection, via a strategy that we call here *single-path collection* (SP collection for short). In this algorithm, partial accumulates $a_i(t)$ of node i at time t are computed as in the following:

$$a_i(t) = \sum_{j \in C_i(t)} (a_j(t-1)) + v_i \quad (3)$$

where v_i are input values to be accumulated and

$$C_i(t) = \{j \in N(i) | \ i = c_j(t-1)\} \quad (4)$$

is the set of nodes *constrained by* i. In the next sections, we will study the dynamics of this algorithm, showing that it naturally occurs in quadratic overestimates, which can be avoid by resorting to a "monotonic filtering" guided by the distance estimates $d_i(t)$ produced by ABF.

A. Line Graph

In the remainder of this paper, we will use the following graph (represented in Fig. II-A) to test the effectiveness of the collection algorithms under consideration.

---

[1]In this paper, we assume that the network topology in which the algorithms are computed is fixed, and that the computation of nodes happen in synchronous rounds. However, all algorithms presented here can easily be extended to mutable network topologies and asynchronous networks (c.f. [6]).

Definition 1 (Line Graph). An N-node line graph $\mathcal{L}_N$ (N-line for short) is a graph with N nodes $i = 0 \dots N-1$ and the following topology:

$$N(i) = \begin{cases} \{i+1\} & \text{if } i = 0 \\ \{i-1\} & \text{if } i = N-1 \\ \{i+1, i-1\} & \text{otherwise} \end{cases} \quad (5)$$

$$e_{ji} = 1 \quad \text{for all } j \in N(i) \quad (6)$$

Furthermore, we assume that there is a single source (switching from time $t = 0$ to time $t = 1$, c.f. Fig II-A) as follows:

$$S(t) = \begin{cases} \{N-1\} & \text{if } t = 0 \\ \{0\} & \text{otherwise} \end{cases} \quad (7)$$

and initial values for the G block are the true distances and constraining nodes towards the initial source $S(0) = N-1$:

$$d_i(0) = N - 1 - i \text{ for all } i \quad (8)$$

$$c_i(0) = \begin{cases} i & \text{if } i = N-1 \\ i+1 & \text{otherwise} \end{cases} \quad (9)$$

Finally, the input value for every node is $v_i = 1$ and the initial values for the partial accumulates of the C block are:

$$a_i(0) = i + 1 \text{ for all } i. \quad (10)$$

Notice that according to Definition 1, for every $t > 0$ the source is node 0, while all of the initial values $d_i(0)$, $c_i(0)$, $a_i(0)$ correspond to a stable state where the source is node $N - 1$. This represents the situation right after a source change, where both the ABF estimates and collection results will need to adjust. In particular, the adjustment process of ABF is described by the following lemma.

Lemma 1. In $\mathcal{L}_N$, the constraining nodes $c_i(t)$ as described in equation (2) for nodes $i = 0 \dots N-1$ and times $t \geq 0$, change exactly once from right to left. More precisely, for $i = 1 \dots N-2$:

$$c_i(t) = \begin{cases} i+1 \text{ if } t \leq i \\ i-1 \text{ otherwise} \end{cases} \quad (11)$$

while $c_0(t) = 0$ and $c_{N-1}(t) = N - 2$ for all $t > 0$.

Proof. For node $i = 0$ and $t > 0$, from equation (7) and (2), we have that $c_0(t) = 0$ since node 0 is the source after time $t = 0$. For node $i = N - 1$, from (5) there is only one neighbour node (i.e. node $N - 2$), thus for $t > 0$

$$c_{N-1}(t) = \operatorname*{argmin}_{j=N-2}(d_{N-2}(t-1)+1) = N - 2 \quad (12)$$

For $i = 1 \dots N - 2$, we proceed by induction on t.
   Base case: Substituting $t = 0$ in equation (11), we obtain:

$$c_i(0) = i + 1 \quad (13)$$

which holds as it is exactly equation (9).
   Inductive Step: Assume that equation (11) holds for some $t > 0$. To complete our proof, we must show that it also holds for time $t' = t + 1$. Substituting for t' in equation (2).

$$c_i(t') = \operatorname*{argmin}_{j=i+1,i-1} (d_j(t) + 1) \quad (14)$$

The distance estimates decrease as you move closer to the source and initially the source was node $N - 1$. The initial condition for time $t = 0$ given in equation (8) corresponds to that. We assume that the distance estimates remain the same even after the source has changed until the time right before the change in constraining nodes. We assert that the distance estimates change at $t = i$.
   Then for time $t' \leq i$, as distance estimates do not change $d_{i-1}(t) < d_{i+1}(t)$

$$c_i(t+1) = i - 1 \quad (15)$$

While for time $t' > i$, according to our assumption the distance estimates have changed at time t and are $d_{i-1}(t) > d_{i+1}(t)$, hence

$$c_i(t+1) = i + 1 \quad (16)$$

Thus, the constraining node at time t was $i + 1$, while at time $t + 1$ it is $i - 1$, concluding the proof. $\square$

III. Quadratic Overestimates in SP Collection

We now show that the single-path collection algorithm in equations (1)-(4) reaches quadratic overestimates on the N-node line graph $\mathcal{L}_N$. In the first phase (where $t \leq i$), every node except the last two ($i \leq N-3$) does not change is partial accumulate, as shown in the following lemma.

Lemma 2. In $\mathcal{L}_N$, the partial accumulate $a_i(t)$ at any node for time $t \leq i$, where i goes from $i = 0$ to $i = N - 3$ is

$$a_i(t) = i + 1 \quad (17)$$

Proof. To prove this, we will use equation (3), where constraining nodes are given from Lemma 1 and proceed by induction on t.
   Base case For $t = 0$ and all $i \leq N - 3$, $a_i(0) = i + 1$ by Definition 1 of the input line graph G.
   Inductive Step: Assume that equation (17) holds for some $t \geq 0$. We must show that it also holds true for $t + 1$ provided that $t + 1 \leq i$, that is:

$$a_i(t+1) = i + 1 \quad (18)$$

Substituting for time $t + 1$ in equation (3):

$$a_i(t+1) = \sum_{j \in C_i(t+1)} a_j(t) + 1 \quad (19)$$

Using equation (4) and Lemma 1:

$$C_i(t+1) = \{j \in N(i) | c_j(t) = i\} = \{i-1\} \quad (20)$$

Then equation (19) simplifies to:

$$a_i(t+1) = a_{i-1}(t) + 1 \quad (21)$$

Using the inductive hypothesis:

$$a_{i-1}(t) = (i-1) + 1 = i \quad (22)$$

Then equation (21) simplifies to:

$$a_i(t+1) = i+1 \tag{23}$$

concluding the proof. □

Meanwhile, the partial accumulates at the last two nodes $N-2$, $N-1$ keep increasing before suddenly dropping to the stable state, as characterised by the following lemma.

Lemma 3. *In $\mathcal{L}_N$, the partial accumulate $a_i(t)$ at the old source node $N-1$ for time $t \leq N-1$ is given by*

$$a_{N-1}(t) = \lceil \frac{t}{2} \rceil N \tag{24}$$

And for $t \geq N$

$$a_{N-1}(t) = 1 \tag{25}$$

Proof. We will prove this by induction.
Base case: For $t = 1$ and $t = 2$ from Lemma 2,

$$a_{N-1}(1) = (N-1) + 1 = N \tag{26}$$

$$a_{N-1}(2) = (N-1) + 1 = N \tag{27}$$

Inductive Step: Assume that equation (24) holds true for all $t' < t$. To complete the induction, we must show that it also holds for t.

Substituting for time t and node $i = N-1$ in equation (3) where $C_i(t)$ contains only one node $j = N-2$ as given by equation (4):

$$a_{N-1}(t) = a_{N-2}(t-1) + 1 \tag{28}$$

Then again through equation (3):

$$a_{N-2}(t-1) = a_{N-1}(t-2) + a_{N-3}(t-2) + 1 \tag{29}$$

Then using Lemma 2, $a_{N-3}(t-2) = N-2$:

$$a_{N-1}(t-2) = a_{N-1}(t-2) + (N-2) + 1 + 1 \tag{30}$$

From equation (24), we have:

$$a_{N-1}(t-2) = \lceil \frac{t-2}{2} \rceil N = \lceil \frac{t}{2} \rceil N - N \tag{31}$$

Thus:

$$a_{N-1}(t) = \lceil \frac{t}{2} \rceil N - N + N \tag{32}$$

And finally:

$$a_{N-1}(t) = \lceil \frac{t}{2} \rceil N \tag{33}$$

This completes the proof of equation (24), and we now move on to the proof of equation (25).

For $t \geq N$, using equation (4), $C_{N-1}(t) = \{j \in N(i)|i = c_j(t-1)\} = \emptyset$. Then the algorithm in equation (3) gives:

$$a_{N-1}(t) = \sum_{j \in C_i(t) = \emptyset} (a_j(t-1)) + 1 = 1 \tag{34}$$

□

Finally, in the last phase $t \geq N-1$ the large overestimates in the last two nodes flow towards the new source, as shown in the following lemma.

Lemma 4. *In $\mathcal{L}_N$, once all the constraining nodes are in a steady state, that is, for every time $t \geq N-1$ and index $x \geq 0$:*

$$a_{N-1-x}(t+x) = a_{N-1}(t) + x \tag{35}$$

Proof. **Base case**: For $x = 0$, the equation holds trivially:

$$a_{N-1-0}(t+0) = a_{N-1}(t) = a_{N-1}(t) + 0 \tag{36}$$

Inductive Step: Assume that equation (35) holds for some $x \geq 0$ and $t \geq N-1$. To complete the induction, we must show that it also holds for $x+1$ and the same t. Let $i = N-1-(x+1)$. From the algorithm definition in equation (3):

$$a_i(t+x+1) = \sum_{j \in C_i(t+x+1)} (a_j(t+x)) + 1 \tag{37}$$

Since $t+x \geq t \geq N-1$, we can apply Lemma 1 to obtain that $C_i(t+x+1) = \{j \in N(i)|i = c_j(t+x)\} = \{i+1\}$. Then, substituting i and applying the inductive hypothesis:

$$a_i(t+x+1) = a_{i+1}(t+x) + 1$$
$$= a_{N-1-x}(t+x) + 1$$
$$= a_{N-1}(t) + x + 1$$

concluding the proof. □

Altogether, the lemmas proved so far imply that a quadratic overestimate is reached in the new source $i = 0$ just before reaching the final stable state, as detailed in the following theorem.

Theorem 1. *In $\mathcal{L}_N$, the maximum partial accumulate $a_i(t)$ reached by the source is obtained at time $t = 2N-2$ and is:*

$$a_0(2N-2) = \left\lceil \frac{N-1}{2} \right\rceil N + N - 1 \geq \frac{N(N+1)}{2} - 1 \in O(N^2) \tag{38}$$

before reaching the correct value at time $t = 2N-1$

$$a_0(2N-1) = N \tag{39}$$

Proof. Consider Lemma 4 where $t = x = N-1$, so that:

$$a_0(2N-2) = a_{N-1-x}(t+x) = a_{N-1}(t) + x = a_{N-1}(N-1) + N - 1. \tag{40}$$

Then by Lemma 3:

$$a_0(2N-2) = a_{N-1}(N-1) + N - 1 = \left\lceil \frac{N-1}{2} \right\rceil N + N - 1 \tag{41}$$

For $a_0(2N-1)$, consider Lemma 4 where $t = N$, $x = N-1$, so that:

$$a_0(2N-1) = a_{N-1-x}(t+x) = a_{N-1}(t) + x = a_{N-1}(N) + N - 1. \tag{42}$$

Then by equation (25) in Lemma 3:

$$a_{N-1}(N) + N - 1 = 1 + N - 1 = N. \tag{□}$$

IV. Improved Dynamics with Monotonic Filtering

We now propose an improved version of the single path C-block, by imposing a *monotonic filter* that reduces the set of considered constraining nodes as follows:

$$C_i(t) = \{j \in N(i)| \; i = c_j(t-1) \wedge d_i(t) < d_j(t-1) \wedge i \neq j\} \tag{43}$$

Using this filtered set $C_i(t)$, we compute the partial accumulates exactly as before with equation (3). This constraint, which intuitively ensures that data is collected by always descending distances, is satisfied by every node in a stable state; however, it may not be satisfied during transients. In the following, we will show that monotonic filtering is enough to completely eliminate the overestimates in $\mathcal{L}_N$. First, we characterise the partial accumulates for a set of intermediate times $i+1 \leq t \leq 2N-3-i$ and every node except for the last two $i \leq N-3$, as in the following lemma.

Lemma 5. *In $\mathcal{L}_N$, the partial accumulate $a_i(t)$ for time $i+1 \leq t \leq 2N-3-i$ and nodes $i = 0$ to $i = N-3$ is given by*

$$a_i(t) = \left\lceil \frac{t-i}{2} \right\rceil \tag{44}$$

Proof. **Base Case:** For a node i at time $t = i+1$ and time $t = i+2$, using equation (3):

$$a_i(i+1) = \sum_{j \in C_i(i+1)} (a_j(i)) + 1 \tag{45}$$

and:

$$a_i(i+2) = \sum_{j \in C_i(i+2)} (a_j(i+1)) + 1 \tag{46}$$

To determine the constraining node set $C_i(i+1)$ and $C_i(i+2)$, using equation (43):

$$C_i(i+1) = \{j \in N(i) \; | \tag{47}$$
$$c_j(i) = i \wedge d_i(i+1) < d_j(i) \wedge j \neq i\}$$
$$= \emptyset$$

$$C_i(i+2) = \{j \in N(i) \; | \tag{48}$$
$$c_j(i+1) = 0 \wedge d_0(i+2) < d_j(i+1) \wedge j \neq i\}$$
$$= \emptyset$$

These sets are empty as from Lemma 1 for time $t = i$ and $t = i+1$, no node has node i as their constraining node. Then equations (45)-(46) do not have the first term, thus $a_i(t) = 1$. For $t = 3 \ldots 2N-1$ we proceed by induction on t.

Inductive Step: Assume that equation (44) holds for some $t \geq i+2$. We must show that the same equation holds for $t+1$ as well. Using equations (43) and (3), $C_i(t)$ is no longer empty. By Lemma 1, for time $t > i+1$ the only node constrained by i is $j = i+1$. Thus:

$$C_i(t+1) = \{i+1\} \tag{49}$$

Then using equation (3):

$$a_i(t+1) = a_{i+1}(t) + 1 \tag{50}$$

And from our inductive hypothesis:

$$a_{i+1}(t) = \left\lceil \frac{t-i-1}{2} \right\rceil \tag{51}$$

Then:

$$a_i(t+1) = \left\lceil \frac{t-i-1}{2} \right\rceil + 1 = \left\lceil \frac{t-i+1}{2} \right\rceil \tag{52}$$

concluding the proof. □

Notice that this last lemma holds also for the traditional SP collection without monotonic filtering. In the following lemma, we prove that for times $t \geq 2N-2-i$ and nodes i the partial accumulates stabilise.

Lemma 6. *In $\mathcal{L}_N$ with monotonic filtering, for time $t \geq 2N-2-i$*

$$a_i(t) = N - i \tag{53}$$

Proof. We prove the hypothesis by backwards induction on i, starting from $i = N-1$ down to $i = 0$.

Base case: The distance condition in equation (43) along with Lemma 1 for $t > 0$ implies that:

$$C_{N-1}(t) = \emptyset \tag{54}$$

For $t > 0$:

$$C_{N-1}(t) = \{j \in N(N-1)|c_j(t-1) = N-1 \text{ and} \tag{55}$$
$$d_{N-1}(t) < d_j(t-1), j \neq N-1\} = \emptyset$$

where $j = N-2$, but the distance condition fails to meet as $d_{N-1}(t) > d_j(t-1)$ and hence the set is empty.

Thus:

$$a_{N-1}(t) = \sum_{j \in C_{N-1}(t) = \emptyset} (a_j(t-1)) + 1 = 1 \tag{56}$$

concluding the proof for $i = N-1$.

Inductive Step: Assume now that the hypothesis holds for $i+1 \leq N-1$. We need to prove that it holds for $i \leq N-2$. Notice that $i \leq N-2$ implies that $t \geq 2N-2-i \geq 2N-2-N+2 = N$. Hence by Lemma 1 the constraining nodes and distance estimates have already stabilised at t, thus:

$$C_i(t) = \{i+1\} \tag{57}$$

Then, by inductive hypothesis:

$$a_i(t) = a_{i+1}(t-1) + 1 = N - (i+1) + 1 = N - i \tag{58}$$

concluding the proof. □

Altogether, the previous lemmas imply that the new source $i = 0$ does not have overestimates during convergence of the SP collection, and in fact, it has underestimates that are no worse than without monotonic filtering (since Lemma 5 holds for collection without monotonic filtering as well).

Theorem 2. *In $\mathcal{L}_N$ with monotonic filtering, the partial accumulates at the new source $a_0(t)$ do not have any overestimates and for $0 < t \leq 2N-3$ are given by*

$$a_0(t) = \left\lceil \frac{t}{2} \right\rceil \tag{59}$$

while for $t \geq 2N - 2$ are stable $a_0(t) = N$.

Proof. Using Lemma 5 on $i = 0$, for $t = 1 \ldots 2N - 3$:

$$a_0(t) = \lceil \frac{t - 0}{2} \rceil = \lceil \frac{t}{2} \rceil \tag{60}$$

By lemma 6, for $t \geq 2N - 2$:

$$a_0(t) = N - 0 = N \tag{61}$$

concluding the proof. □

V. Conclusions

In this paper, we have introduced the monotonic filtering strategy for single-path collection algorithms. In the sample case of an N-node line graph, we showed that the introduction of monotonic filtering allows to avoid quadratic overestimates, without introducing any additional underestimate.

In future work, we plan to extend the present results by both *(i)* identifying a larger family of graphs for which single-path collection produces quadratic overestimates; and by *(ii)* identifying the assumptions needed to prove that monotonic filtering avoids quadratic overestimates. Based on preliminary investigations, quadratic overestimates seem to possible for all graphs, but are also avoided by monotonic filtering for all graphs, provided that rounds are synchronous. Finally, we plan to empirically validate the effectiveness of monotonic filtering by simulation on asynchronous, mutable networks.

References

[1] N. Bicocchi, M. Mamei, and F. Zambonelli, "Self-organizing virtual macro sensors," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, no. 1, pp. 2:1–2:28, 2012.

[2] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan, "Context is key," *Communications of the ACM*, vol. 48, no. 3, pp. 49–53, 2005.

[3] M. Viroli, D. Pianini, A. Ricci, and A. Croatti, "Aggregate plans for multiagent systems," *International Journal of Agent-Oriented Software Engineering*, vol. 4, no. 5, pp. 336–365, 2017.

[4] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[5] A. K. Talele, S. G. Patil, and N. B. Chopade, "A survey on data routing and aggregation techniques for wireless sensor networks," in *International Conference on Pervasive Computing (ICPC)*. IEEE, 2015, pp. 1–5.

[6] M. Viroli, G. Audrito, J. Beal, F. Damiani, and D. Pianini, "Engineering resilient collective adaptive systems by self-stabilisation," *ACM Transactions on Modeling and Computer Simulation*, vol. 28, no. 2, pp. 16:1–16:28, 2018.

[7] Y. Mo, J. Beal, and S. Dasgupta, "Error in self-stabilizing spanning-tree estimation of collective state," in *2nd IEEE International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. IEEE Computer Society, 2017, pp. 1–6.

[8] G. Audrito and S. Bergamini, "Resilient blocks for summarising distributed data," in *1st Workshop on Architectures, Languages and Paradigms for IoT (ALP4IoT)*, ser. EPTCS, vol. 264, 2017, pp. 23–26.

[9] G. Audrito, S. Bergamini, F. Damiani, and M. Viroli, "Effective collective summarisation of distributed data in mobile multi-agent systems," in *18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1618–1626. [Online]. Available: http://dl.acm.org/citation.cfm?id=3331882

[10] ——, "Resilient distributed collection through information speed thresholds," in *22th International Conference on Coordination Models and Languages (COORDINATION)*, ser. Lecture Notes in Computer Science. Springer, 2020, to appear.

[11] Q. Liu, A. Pruteanu, and S. Dulman, "Gradient-based distance estimation for spatial computers," *Comput. J.*, vol. 56, no. 12, pp. 1469–1499, 2013.

[12] G. Audrito, F. Damiani, and M. Viroli, "Optimal single-path information propagation in gradient-based algorithms," *Science of Computer Programming*, vol. 166, pp. 146–166, 2018.

[13] Y. Mo, S. Dasgupta, and J. Beal, "Robustness of the adaptive bellman - ford algorithm: Global stability and ultimate bounds," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4121–4136, 2019.