Fast, Robust, Quantizable Approximate Consensus*

Bernadette Charron-Bost[1], Matthias Függer[2], and Thomas Nowak[3]

1 **CNRS, École polytechnique, Paris, France**
charron@lix.polytechnique.fr
2 **Max-Planck-Institut für Informatik, Saarbrücken, Germany**
mfuegger@mpi-inf.mpg.de
3 **Université Paris-Sud, Paris, France**
thomas.nowak@lri.fr

Abstract

We introduce a new class of distributed algorithms for the approximate consensus problem in dynamic rooted networks, which we call *amortized averaging algorithms*. They are deduced from ordinary averaging algorithms by adding a value-gathering phase before each value update. This results in a drastic drop in decision times, from being exponential in the number n of processes to being polynomial under the assumption that each process knows n. In particular, the *amortized midpoint algorithm* is the first algorithm that achieves a linear decision time in dynamic rooted networks with an optimal contraction rate of $1/2$ at each update step.

We then show robustness of the amortized midpoint algorithm under violation of network assumptions: it gracefully degrades if communication graphs from time to time are non rooted, or under a wrong estimate of the number of processes. Finally, we prove that the amortized midpoint algorithm behaves well if processes can store and send only quantized values, rendering it well-suited for the design of dynamic networked systems. As a corollary we obtain that the 2-set consensus problem is solvable in linear time in any dynamic rooted network model.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases approximate consensus, dynamic networks, averaging algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.137

1 Introduction

This paper studies the problem of *approximate consensus*: given a set of processes, each of them with an initial real value, for any positive real number ε, every process has to achieve ε*-consensus*, i.e., decides on one value that lies in the range of the initial values and that differs from the other decision values by at most ε. This problem, which is a weakening of *exact* ($\varepsilon = 0$) *consensus*, has a large variety of applications in distributed computing or multi-agent control (e.g., clock synchronization or geometric coordination tasks) and often has to be solved in the context of dynamic networks, using local information, and under adverse constraints.

In recent work [8], we proved that approximate consensus is solvable in a dynamic network model if and only if the directed time-varying communication graph is always rooted, i.e.,

* This work was partially supported by an X-DGA contract and the Austrian Science Foundation (FWF) project SIC (P26436).

contains a rooted spanning tree. The spanning tree can change completely from time to time. Hence no stability condition is required to solve approximate consensus.

In fact, we showed that when addressing the solvability issue of approximate consensus, it suffices to consider the class of *averaging algorithms* where each process maintains a single scalar state variable and repeatedly updates to a weighted average of the values it has just received. Indeed, the problem is solvable if and only if it can be solved by an averaging algorithm.

We proved that, if there is a positive lower bound ϱ on the weights used in averaging steps, then the averaging algorithm achieves ε-agreement in $O\big(n\varrho^{-n}\log\frac{1}{\varepsilon}\big)$ rounds where n is the number of processes. Moreover, we showed that the classical averaging algorithm, called the *equal-neighbor* algorithm, exhibits an exponentially large decision time in the case of the *Butterfly network model*.

Contribution. In this work we present a new approximate consensus algorithm that we show to be *fast*, being the first algorithm in highly dynamic networks with linear time complexity in the number of processes, *robust* to violation of network assumptions, and allowing for *quantization* of its variables – leading to a space efficient algorithm.

We list the main contributions in detail in the following:

(i) We consider the property of *ϱ-safety* for averaging algorithms originally introduced by Chazelle [12], which is a generalization of the lower bound condition on positive weights. This property focuses on the set of transmitted *values* and not on the linear *functions* (stochastic matrices) applied in the averaging steps, as done classically. It thus captures the essential properties needed for contracting the range of current values in the system. Using a graph-theoretic reduction that already played a key role in [8], we first give simple proofs of correctness and of upper bounds on the contraction rate of ϱ-safe averaging algorithms.
More importantly, this approach allows us to propose a new averaging algorithm for approximate consensus in dynamic rooted networks, that we call the *midpoint algorithm*, with an optimal contraction rate of $1/2$.

(ii) We introduce the notion of *amortization* of averaging algorithms, which consists in inserting a value-gathering phase before each averaging step. This additional phase transforms ϱ-safe averaging algorithms into "turbo versions" of themselves in that decision times pass from being exponential [8] to being polynomial in the number n of processes.
Amortization assumes that each process knows the size n of the network. Furthermore, processes ought to be able to accumulate and to relay values from the past, as opposed to averaging algorithms that are memoryless in the sense that processes deal only with fresh values. More precisely, we show that the amortized version of a ϱ-safe averaging algorithm solves approximate consensus with polynomial decision times in any dynamic rooted network, and *a priori* under the conditions that each process knows the size of the network, and can store and transmit in every message up to $n(n-1)$ scalar values.
Then we combine the two above ideas in the design of the *amortized midpoint algorithm*, which achieves ε-agreement in $O\big(n\log\frac{1}{\varepsilon}\big)$ rounds. Apart its linear decision times, the positive features of this amortized averaging algorithm are that it works in anonymous networks (without process identifier) and that processes are required to store and to transmit in each message only *two* scalar values, whatever the size of the network is.

(iii) The correctness proof of amortized averaging algorithms relies on two fundamental hypotheses: common knowledge of the number n of processes and rootedness of all occurring communication graphs. However we show that, even with an erroneous estimate on n or when communication graphs sometimes fail to be rooted, the amortized version of a ϱ-safe averaging algorithm still solves approximate consensus, albeit with larger decision time. This graceful degradation property demonstrates that the robustness of a ϱ-safe averaging algorithm carries over to some extent to its amortized versions.

(iv) Finally we show that if processes can only use variables with finite precision, then the amortized midpoint algorithm achieves ε-consensus in linear time whenever possible, i.e., when ε is larger than the variables' precision. The decision time with quantization does not degrade and remains in the same order as without quantization. The robustness to rounding errors seems to be specific to the amortized midpoint algorithm as our proof does not work for general amortized ϱ-safe averaging algorithms.

As a corollary, we get that the 2-*set consensus problem* – another weakening of exact consensus where disagreement is limited to at most two different decision values – is solvable in linear time in all dynamic rooted network models.

While the practical implications of a versatile approximate consensus algorithm which is both fast and robust are clearly appealing for the design of man-made dynamic networked systems, our results go even beyond that and are interesting also from a different, more analytic, perspective.

Many natural phenomena, like bird flocking [11], synchronization of coupled oscillators [27, 20], opinion dynamics [16], or firefly synchronization [19], can be modeled with agents that execute averaging algorithms. However, there is a large mismatch between the fast agreement times observed in nature and the theoretical worst-case times, which may be exponential [11]. In this sense, our results provide a step towards closing this gap, suggesting that the agents in these natural systems in fact operate on a different, slower, time scale than their environments, and that this inertia actually contributes positively to faster consensus.

Related work. The time complexity of approximate consensus has been extensively studied in the past. Several papers (e.g., [20, 6, 1, 5, 2]) handle the case of time-varying communication graphs and consider averaging algorithms with decision times which are exponential in the number of processes.

The first algorithm with polynomial decision times has been proposed by Olshevsky and Tsitsiklis [24], based on ideas from load balancing. They proved that their algorithm achieves ε-consensus in $O\left(n^3 \log \frac{1}{\varepsilon}\right)$ rounds with a time-varying *bidirectional* topology under the quite strong stability condition that graphs do not change during each 3 round slot $3k$, $3k+1$, $3k+2$. A refined analysis of this algorithm [22] results in an improved quadratic bound. In the meantime, Chazelle [12] proposed a remarkable averaging algorithm with cubic decision times, which works in any bidirectional connected network model.

Another approach to speed up decision times – also developed in the context of load balancing – consists in considering *second-order* algorithms where the update rule for process p at round $k+1$ involves the old value held by p at round $k-1$, in addition to the values at round k of p's neighbors; see [21]. Following this approach, Olshevsky [23] has recently proposed a linear decision time algorithm that requires each process to know n, and to be able to store and to transmit in every message two scalar values, like the amortized midpoint algorithm. Unfortunately Olshevsky's algorithm works only under the assumption of a *fixed bidirectional* communication graph. Another linear approximate consensus algorithm has been proposed in [28] in the case of a fixed topology, possibly non-bidirectional, but it assumes processes with strong computational power and large memory (processes ought to store all values of the past). The sum of these efforts makes the time-linearity of the relatively simple amortized midpoint algorithm in arbitrarily dynamic directed anonymous networks all the more so striking.

As for the effects of quantization, they have been studied in a number of papers, but most of them assume a fixed topology to bound the time complexity; see [18, 15]. In their seminal paper [22], Nedić et al. analyzed the quantized version of their quadratic averaging algorithm for a time-varying bidirectional topology and proved that decision times with quantized values are of the same order as with continuous values.

2 Approximate consensus and averaging algorithms

We assume a distributed, round-based computational model in the spirit of the Heard-Of model [9]. A system consists in a set of processes $[n] = \{1, \ldots, n\}$. Computation proceeds in *rounds*: In a round, each process sends its state to its outgoing neighbors, receives values from its incoming neighbors, and finally updates its state based. The value of the updated state is determined by a deterministic algorithm, i.e., a transition function that maps the values in the incoming messages to a new state value. Rounds are communication closed in the sense that no process receives values in round k that are sent in a round different from k.

Communications that occur in a round are modeled by a directed graph $G = ([n], E(G))$ with a self-loop at each node. The latter requirement is quite natural as a process can obviously communicate with itself instantaneously. Such a directed graph is called a *communication graph*. We denote by $\mathrm{In}_p(G)$ and $\mathrm{Out}_p(G)$ the sets of incoming and outgoing neighbors of p, respectively.

The *product* of two communication graphs G and H, denoted $G \circ H$, is the directed graph with an edge from (p, q) if there exists $r \in [n]$ such that $(p, r) \in E(G)$ and $(r, q) \in E(H)$. We easily see that $G \circ H$ is a communication graph and, because of the self-loops, $E(G) \cup E(H) \subseteq E(G \circ H)$.

A *communication pattern* is a sequence $(G(k))_{k \geqslant 1}$ of communication graphs. For a given communication pattern, $\mathrm{In}_p(k)$ and $\mathrm{Out}_p(k)$ stand for $\mathrm{In}_p(G(k))$ and $\mathrm{Out}_p(G(k))$, respectively.

Each process p has a *local state* s_p the value of which at the end of round $k \geqslant 1$ is denoted by $s_p(k)$. Process p's initial state, i.e., its state at the beginning of round 1, is denoted by $s_p(0)$. Let the *global state* at the end of round k be the collection $s(k) = (s_p(k))_{p \in [n]}$. The *execution* of an algorithm from global initial state $s(0)$, with communication pattern $(G(k))_{k \geqslant 1}$ is the unique sequence $(s(k))_{k \geqslant 0}$ of global states defined as follows: for each round $k \geqslant 1$, process p sends $s_p(k-1)$ to all the processes in $\mathrm{Out}_p(k)$, receives $s_q(k-1)$ from each process q in $\mathrm{In}_p(k)$, and computes $s_p(k)$ from the incoming messages, according to the algorithm's transition function.

When the structure of states allows each process to record and to relay information it has received during any period of K rounds for some positive integer K, we may be led to modify time-scale and to consider blocks of K consecutive rounds, called *macro-rounds*: macro-round ℓ is the sequence of rounds $(\ell - 1)K + 1, \ldots, \ell K$ and the corresponding information flow graph, called *communication graph at macro-round ℓ*, is the product of the communication graphs $G((\ell - 1)K + 1) \circ \ldots \circ G(\ell K)$.

2.1 Approximate consensus

A crucial problem in distributed systems is to achieve agreement among local process states from arbitrary initial local states. It is a well-known fact that this goal is not easily achievable in the context of dynamic network changes [26, 8], and restrictions on communication patterns are required for that. We define a *network model* as a non-empty set $\mathcal{N}$ of communication graphs, those that may occur in communication patterns.

We now consider the above round-based algorithms in which the local state of process p contains two variables x_p and dec_p. Initially the range of x_p is $[0,1]$ and $\mathrm{dec}_p = \perp$ (which informally means that p has not decided yet).[1] Process p is allowed to set dec_p to the current value of x_p, and so to a value v different from $\perp$, only once; in that case we say that p *decides* v.

For any $\varepsilon \geqslant 0$, an algorithm *achieves ε-agreement with communication pattern* $(G(k))_{k\geqslant 1}$ if each execution from a global initial state as specified above and with the communication pattern $(G(k))_{k\geqslant 1}$ fulfills the following three conditions:

ε-Agreement. The decision values of any two processes are within ε.
Validity. All decided values are in the range of the initial values of processes.
Termination. All processes eventually decide.

An algorithm *solves approximate consensus* in a network model $\mathcal{N}$ if for any $\varepsilon > 0$, it achieves ε-agreement with each communication pattern formed with graphs all in $\mathcal{N}$. It *solves exact consensus* in $\mathcal{N}$ if f it achieves 0-agreement with each communication pattern with graphs in $\mathcal{N}$.

In a previous paper [8], we proved the following characterization of network models in which approximate consensus is solvable:

▶ **Theorem 1** ([8]). *The approximate consensus problem is solvable in a network model $\mathcal{N}$ if and only if each graph in $\mathcal{N}$ has a rooted spanning tree.*

2.2 Averaging algorithms

We focus on *averaging algorithms* in which at each round k, every process p updates x_p to some weighted average of the values it has just received: formally, if $m_p(k-1)$ is the minimum of the values $\{x_q(k-1) \mid q \in \mathrm{In}_p(k)\}$ received by p in round k and $M_p(k-1)$ is its maximum, then

$$m_p(k-1) \leqslant x_p(k) \leqslant M_p(k-1)\,.$$

In other words, at each round k, every process adopts a new value within the interval formed by the values of its incoming neighbors in the communication graph $G(k)$.

Since we strive for distributed implementations of averaging algorithms, weights in the average update rule of process p are required to be locally computable by p. They may depend only on the set of values received by p at round k, as is the case, for instance, with the update rule of the *mean-value algorithm*:

$$x_p(k) = \frac{1}{|V_p(k)|} \sum_{v \in V_p(k)} v \tag{1}$$

where $V_p(k) = \{x_q(k-1) \mid q \in \mathrm{In}_p(k)\}$. In contrast, even in anonymous networks, weights in the update rule for p may depend on the *multiset* of values received by p at round k counted with their multiplicities: as an example, p's update rule in the *equal-neighbor algorithm* is given by:

$$x_p(k) = \frac{1}{|\mathrm{In}_p(k)|} \sum_{q \in \mathrm{In}_p(k)} x_q(k-1)\,. \tag{2}$$

[1] In the case of *binary consensus*, x_p is restricted to be initially from $\{0,1\}$.

Observe that the decision rule is not specified in the above definition of averaging algorithms: the decision time immediately follows from the number of rounds that is proven to be sufficient to reach ε-agreement.

Let $\varrho \in]0, 1/2]$; an averaging algorithm is *ϱ-safe in $\mathcal{N}$* if at any round k of each of its executions with communication patterns in $\mathcal{N}$, each process adopts a new value within the interval formed by its neighbors in $G(k)$ not too close to the boundary:

$$\varrho M_p(k-1) + (1-\varrho) m_p(k-1) \leqslant x_p(k) \leqslant (1-\varrho) M_p(k-1) + \varrho\, m_p(k-1)\,.$$

Clearly if an averaging algorithm is ϱ-safe, then it is ϱ'-safe for any $\varrho' \leqslant \varrho$. We also easily check the following property of the equal-neighbor and mean-value algorithms.

▶ **Proposition 2.** *In any network model with n processes, the equal-neighbor algorithm and the mean-value algorithm are both $(1/n)$-safe.*

Finally, an averaging algorithm is *α-contracting in $\mathcal{N}$* if at each round k of any of its executions with communication patterns in $\mathcal{N}$, we have

$$\delta\big(x(k)\big) \leqslant \alpha\, \delta\big(x(k-1)\big)$$

where δ is the seminorm on $\mathbb{R}^n$ defined by $\delta(x) = \max_p(x_p) - \min_p(x_p)$.

▶ **Proposition 3.** *In any network model, every α-contracting averaging algorithm with $\alpha \in [0, 1[$ solves approximate consensus and achieves ε-agreement in $\left\lceil \log_{\frac{1}{\alpha}} \left(\frac{1}{\varepsilon}\right) \right\rceil$ rounds.*

3 Averaging algorithms, nonsplitness, and contraction rates

In a previous paper [8], we proved solvability of approximate consensus in rooted network models by a reduction to *nonsplit* network models: a directed graph is *nonsplit* if any two nodes have a common incoming neighbor. The crucial point of nonsplitness lies in the following result:

▶ **Theorem 4.** *In a nonsplit network model, a ϱ-safe averaging algorithm is $(1-\varrho)$-contracting. Thus it solves approximate consensus and achieves ε-agreement in $\left\lceil \log_{\frac{1}{1-\varrho}} \left(\frac{1}{\varepsilon}\right) \right\rceil$ rounds.*

Combined with Proposition 2, we obtain an elementary proof of the classical result [4, 7] that approximate consensus is solvable in rooted network models, which does not make use anymore of Dobrushin's coefficients of scrambling matrices [14], which can be defined as $\delta(A) = \sup_{\delta(x)\neq 0} \delta(Ax)/\delta(x)$.

As an immediate consequence of Theorem 4, we deduce that any ϱ-safe algorithm is at best $(1/2)$-contracting since by definition the safety coefficient ϱ cannot be larger than $1/2$. Indeed, in Section 4 we shall present an averaging algorithm that is $(1/2)$-safe in any nonsplit network model. But do there exist other averaging algorithms with a contraction rate less than $1/2$? As stated in the following theorem, the answer is no in the network model of nonsplit communication graphs, except in the case of two processes for which the best contraction rate is $1/3$.

▶ **Theorem 5.** *In the network model of nonsplit communication graphs with n processes, there is no averaging algorithm that is α-contracting for any $\alpha > 1/3$ if $n = 2$ and for any $\alpha > 1/2$ if $n \geqslant 3$.*

Both lower bounds in Theorem 5 are tight. Indeed the *midpoint algorithm* that we will introduce in Section 4.3 is $(1/2)$-safe, and so $(1/2)$-contracting in the network model of nonsplit communication graphs. For a two process network, we check that Algorithm 1 has a contraction rate of $1/3$.

Algorithm 1 Averaging algorithm for two processes with contraction rate 1/3

```
Initialization:
1: $x_p \in [0, 1]$
In round $k \geqslant 1$ do:
2: send $x_p$ to other process and receive $x_q$ if $(q, p) \in E(k)$
3: if $x_q$ was received then
4:     $x_p \leftarrow x_p/3 + 2x_q/3$
5: end if
```

4 Speedup by amortization

In [8] we proved the following reduction from rooted to nonsplit network models to show that averaging algorithms solve approximate consensus in exponential time.

▶ **Proposition 6** ([8]). *Every product of $n-1$ rooted graphs with n nodes and self-loops at all nodes is nonsplit.*

We next show that one can in fact push this reduction to the extreme, obtaining significantly faster algorithms: Proposition 6 and Theorem 4 suggest to consider a new type of distributed algorithms, which we call *amortized averaging algorithms*, in which each process repeatedly first collects values during $n-1$ rounds, and then computes a weighted average of values it has just collected. In other words, an amortized averaging algorithm is an averaging algorithm with the granularity of macro-rounds consisting in blocks of $n-1$ consecutive rounds.

We now make this notion more precise. First let us fix some notation. Macro-round ℓ is the sequence of rounds $(\ell-1)(n-1)+1, \dots, \ell(n-1)$. We consider algorithms for which every variable x_p is updated only at the end of macro-rounds; $x_p(\ell)$ will denote the value of x_p at the end of round $\ell(n-1)$, as no confusion can arise. Given some communication pattern $(G(k))_{k\geqslant 1}$, the communication graph at macro-round ℓ is equal to:

$$\hat{G}(\ell) = G((\ell-1)(n-1)+1) \circ \ldots \circ G(\ell(n-1)) \,.$$

Each process p can record the set of values it has received during macro-round ℓ, namely the set $V_p(\ell) = \{x_q(\ell-1) \mid q \in \mathrm{In}_p\big(\hat{G}(\ell)\big)\}$, but in anonymous networks, p cannot determine the set of its incoming neighbors in $\hat{G}(\ell)$. This is in contrast to networks with unique process identifiers where each process p can determine the membership of $\mathrm{In}_p\big(\hat{G}(\ell)\big)$ by piggybacking the name of the sender onto every message, and so can compute the set $W_p(\ell) = \{\big(q, x_q(\ell-1)\big) \mid q \in \mathrm{In}_p\big(\hat{G}(\ell)\big)\}$. Each process can then determine the multiset of values that it has received during a macro-round, counted with their multiplicities.

In consequence in any anonymous network, we can define the *amortized version of* an averaging algorithm $\mathcal{A}$ with weights in update rules that depend only on the sets of received values: at the end of every macro-round ℓ, each process p adopts a new value by applying the same update rule as in $\mathcal{A}$ with the macro-set $V_p(\ell)$. Based on the above discussion, this definition can be extended to averaging algorithms with update rules involving the sets of incoming neighbors when processes have unique identifiers. For instance, the amortized version of the mean-value algorithm is defined in any anonymous network while the amortized equal-neighbor algorithm requires to have unique process identifiers.

In both cases, the new value adopted by process p at the end of macro-round ℓ lies within the interval formed by the values of its incoming neighbors in the communication graph $\hat{G}(\ell)$: if $\hat{m}_p(\ell-1)$ is the minimum of the values in $V_p(\ell)$ and $\hat{M}_p(\ell-1)$ is the maximum, then

$$\hat{m}_p(\ell-1) \leqslant x_p(\ell) \leqslant \hat{M}_p(\ell-1) \,.$$

Algorithm 2 Amortized equal-neighbor algorithm

```
Initialization:
 1: $x_p \in [0,1]$ and $W_p \leftarrow \{(p, x_p)\}$
In round $k \geqslant 1$ do:
 2: send $W_p$ to all processes in $\mathrm{Out}_p(k)$ and receive $W_q$ from all processes $q$ in $\mathrm{In}_p(k)$
 3: $W_p \leftarrow \bigcup_{q \in \mathrm{In}_p(k)} W_q$
 4: if $k \equiv 0 \mod n-1$ then
 5:     $x_p \leftarrow \frac{1}{|W_p|} \sum_{(q,v) \in W_p} v$
 6:     $W_p \leftarrow \{(p, x_p)\}$
 7: end if
```

If the original averaging algorithm is ϱ-safe, then we have the additional guarantee that

$$\varrho \hat{M}_p(\ell-1) + (1-\varrho)\hat{m}_p(\ell-1) \leqslant x_p(\ell) \leqslant (1-\varrho)\hat{M}_p(\ell-1) + \varrho\, \hat{m}_p(\ell-1)\,.$$

We can then combine Proposition 6 and Theorem 4 to derive the following central result for amortized averaging algorithms.

▶ **Theorem 7.** *In any rooted network model, the amortized version of a ϱ-safe averaging algorithm solves approximate consensus and achieves ε-agreement in $(n-1)\left\lceil \log_{\frac{1}{1-\varrho}} \left(\frac{1}{\varepsilon}\right)\right\rceil$ rounds.*

In order to fix decision times, we have assumed from the beginning that each process knows the number n of processes or at least a common upper bound on n. However, regarding the *asymptotic consensus* problem, obtained by substituting limit values for decision values in the specification of approximate consensus, this common knowledge on n is actually useless for averaging algorithms.

In contrast, update rules in amortized averaging algorithms require that the number of processes in the network is known to all processes. In other words, amortized averaging algorithms are defined only under the assumption of this common knowledge in the network, even for solving asymptotic consensus. In fact, we can adapt the definition of amortized averaging algorithms to the case where n is a fixed parameter and then, because of the self-loops, obtain that Theorem 7 still holds when n is only an upper bound on the number of processes.

4.1 A quadratic-time algorithm in rooted networks with process identifiers

In [8], we proposed an approximate consensus algorithm with a quadratic decision time. The algorithm (cf. Algorithm 2) does not work in anonymous networks as it uses process identifiers so that each process can determine whether some value that it has received several times during a macro-round is originated from the same process or not.

The update rule (line 5) rewrites as:

$$x_p(\ell) = \frac{1}{\left|\,\mathrm{In}_p\left(\hat{G}(\ell)\right)\right|} \sum_{q \in \mathrm{In}_p\left(\hat{G}(\ell)\right)} x_q(\ell-1)\,.$$

Hence Algorithm 2 is actually the amortized version of the equal-neighbor algorithm.

From Proposition 2 and Theorem 7, it follows that Algorithm 2 solves approximate consensus. Moreover, because of the inequality $\log(1-a) \leqslant -a$ when $0 \leqslant a$ and because $\delta\left(x(0)\right) \leqslant 1$, it follows that if $\ell \geqslant n \log \frac{1}{\varepsilon}$, then $\delta\left(x(\ell)\right) \leqslant \varepsilon$. Hence Algorithm 2 achieves ε-agreement in $O\left(n^2 \log \frac{1}{\varepsilon}\right)$ rounds.

4.2 A quadratic-time algorithm in anonymous rooted networks

Interestingly, Algorithm 2 still works when processes just collect values without taking into account processes from which they originate.

Algorithm 3 Amortized mean-value algorithm

```
Initialization:
 1: $x_p \in [0,1]$
 2: $V_p \subseteq V$, initially $\emptyset$
In round $k \geqslant 1$ do:
 3: send $V_p$ to all processes in $\mathrm{Out}_p(k)$ and receive $V_q$ from all processes $q$ in $\mathrm{In}_p(k)$
 4: $V_p \leftarrow \bigcup_{q \in \mathrm{In}_p(k)} V_q$
 5: if $k \equiv 0 \mod n-1$ then
 6:     $x_p \leftarrow \frac{1}{|V_p|} \sum_{v \in V_p} v$
 7:     $V_p \leftarrow \emptyset$
 8: end if
```

At each macro-round ℓ, the update rule in the resulting algorithm for anonymous networks (cf. Algorithm 3, line 6) coincides with the update rule in the mean-value algorithm. In other words, Algorithm 3 is the amortized version of the mean-value algorithm. Since the latter algorithm is a $(1/n)$-safe averaging algorithm (Proposition 2), we may apply Theorem 7 and obtain the following result.

▶ **Theorem 8.** *In a rooted network model of n processes, the amortized mean-value algorithm solves approximate consensus and achieves ε-agreement in $O(n^2 \log \frac{1}{\varepsilon})$ rounds.*

4.3 A linear-time algorithm in anonymous rooted networks

We improve the above quadratic upper bound on decision times with a linear amortized averaging algorithm which differs from our previous algorithm in the update rule: each process adopts the midpoint of the range of values it has received during a macro-round (cf. Algorithm 4).

Algorithm 4 Amortized midpoint algorithm

```
Initialization:
 1: $x_p \in [0,1]$
 2: $m_p \in [0,1]$, initially $x_p$
 3: $M_p \in [0,1]$, initially $x_p$
In round $k \geqslant 1$ do:
 4: send $(m_p, M_p)$ to all processes in $\mathrm{Out}_p(k)$ and receive $(m_q, M_q)$ from all processes $q$ in $\mathrm{In}_p(k)$
 5: $m_p \leftarrow \min \{ m_q \mid q \in \mathrm{In}_p(k) \}$
 6: $M_p \leftarrow \max \{ M_q \mid q \in \mathrm{In}_p(k) \}$
 7: if $k \equiv 0 \mod n-1$ then
 8:     $x_p \leftarrow (m_p + M_p)/2$
 9:     $m_p \leftarrow x_p$
10:     $M_p \leftarrow x_p$
11: end if
```

Let us now consider the *midpoint algorithm* that is the simple averaging algorithm with the midpoint update rule. In other words, Algorithm 4 is the amortized version of the midpoint algorithm.

By definition, the midpoint algorithm is $(1/2)$-safe. By Theorem 4, it follows that this algorithm is a $(1/2)$-contracting approximate consensus algorithm in any nonsplit network model, with a constant decision time. This improves the linear time complexity of the equal neighbor averaging algorithm [8] for this type of network models and demonstrates that the 1/2 lower bound in Theorem 5 for networks with $n \geqslant 3$ processes is tight.

Furthermore Theorem 7 implies that Algorithm 4 solves approximate consensus in any rooted network model with a decision time in $O(n \log \frac{1}{\varepsilon})$ rounds.

▶ **Theorem 9.** *In a rooted network model of n processes, the amortized midpoint algorithm solves approximate consensus and achieves ε-agreement in $(n-1)\lceil \log_2 \frac{1}{\varepsilon} \rceil$ rounds.*

Hence the amortized mean-value algorithm and the amortized midpoint algorithm both work in any anonymous network model, under the assumption that the number of processes is known to all processes. However while the first algorithm has quadratic decision times and requires each process to store n values per round and to transmit n values per message, the amortized midpoint algorithm solves approximate consensus in linear-time and with only a constant number of (namely, two) values per process and per message.

5 Robustness of amortized averaging algorithms

In this section, we discuss the resiliency of our amortized averaging algorithms against a wrong estimate of the number of processes or against a partial failure of the assumption that communication graphs are permanently rooted.

Firstly consider some communication pattern in which only part of communication graphs are rooted: suppose that $N-1$ communication graphs in any macro-round of $n-1$ consecutive rounds are guaranteed to be rooted. Then there are at least $n-1$ rooted communication graphs in any sequence of $L = \left\lceil \frac{n-1}{N-1} \right\rceil$ macro-rounds of length $n-1$, and every product of L communication graphs of macro-rounds is nonsplit.

Secondly suppose that the network model is indeed rooted but processes do not know the exact number n of processes and only knows an estimate N on n. Macro-rounds in the amortized averaging algorithms then consist of $N-1$ rounds (instead of $n-1$), and so the cumulative communication graphs in such macro-rounds may be not nonsplit when $N < n$. However the communication graph over any block of $L = \left\lceil \frac{n-1}{N-1} \right\rceil$ macro-rounds of length $N-1$ is nonsplit.

In both cases, the above discussion leads to introduce the notion of K-*nonsplit* network model defined as any network model $\mathcal{N}$ such that every product of K communication graphs from $\mathcal{N}$ is nonsplit. Theorem 4 can then be extended as follows:

▶ **Theorem 10.** *In a K-nonsplit network model, a ϱ-safe averaging algorithm is $(1-\varrho^K)$-contracting over each block of K consecutive rounds, i.e., for every non negative integer k, we have*

$$\delta\big(x(k+K)\big) \leqslant (1-\varrho^K)\,\delta\big(x(k)\big)\,.$$

As an immediate corollary of Theorem 10, we obtain that the amortized version of a ϱ-safe averaging algorithm is resilient against a wrong estimate of the number of processes or against a partial failure of the assumption of a rooted network model, and its decision times are multiplied by a factor in $O\big(L\varrho^{-L}\big)$. Note that the theorem measures the number of macro-rounds, and not the number of rounds.

▶ **Theorem 11.** *The amortized version of a ϱ-safe averaging algorithm solves approximate consensus even with an erroneous number of processes or with a communication pattern* *where only part of communication graphs are rooted. Moreover it achieves ε-agreement in $O\left(L\varrho^{-L}\cdot\log\left(\frac{1}{\varepsilon}\right)\right)$ macro-rounds if N denotes the estimate on process number or if only $N-1$ rounds in each block of $n-1$ consecutive rounds are guaranteed to have a rooted communication graph and where $L=\left\lceil\frac{n-1}{N-1}\right\rceil$.*

6 Quantization

In this section, we take into account the additional constraint that processes can only store and transmit quantized values. This model provides a good approximation for networks with storage constraints or with finite bandwidth channels.

We include this constraint by quantizing each averaging update rule. For that, we fix some positive integer Q and choose a rounding function, denoted $[\,.\,]$, which rounds down (or rounds up) to the nearest multiple of $1/Q$. Then the quantized update rule for process p at round k writes

$$x_p(k)=\left[\sum_{q\in\mathrm{In}_p(k)} w_{qp}(k)\,x_q(k-1)\right] \tag{3}$$

where the $w_{qp}(k)$ denote the weights in the average of the original algorithm. Besides we assume that all initial values are multiples of $1/Q$.

6.1 Quantization and midpoint

Nedić et al. [22] proved that in any strongly connected network model, every quantized averaging algorithm with the update rule (3) solves exact consensus (and so approximate consensus). Because of the impossibility result for exact consensus [8], their result does not hold if the strong connectivity assumption is weakened into the one of rooted network models.

For the same reason, if $\varepsilon<1/Q$, then ε-consensus cannot be generally achieved in a rooted network model by a quantized averaging algorithm or its amortized version. In this section, we prove that the quantization of the amortized midpoint algorithm indeed achieves $1/Q$-agreement in any rooted network model.

▶ **Theorem 12.** *In a rooted network model, quantization of the amortized midpoint algorithm achieves $1/Q$-agreement by round $(n-1)\left(\lfloor\log_2(Q-2)\rfloor+2\right)$. Moreover in every execution, the sequence of values of every process p converges to a limit x_p^* that is a multiple of $1/Q$ in finite time, and for every pair of processes p, q, we have either $x_p^*=x_q^*$ or $\left|x_p^*-x_q^*\right|=1/Q$.*

We see that the decision times of the quantized and the non-quantized versions of the amortized midpoint algorithm are in the same order for $\varepsilon=1/Q$. Further, one can show that for $\varepsilon>2/Q$, ε-agreement is achieved earlier, namely in round $(n-1)\left(\left\lfloor\log_2\frac{Q-2}{Q\varepsilon-2}\right\rfloor+1\right)$. When increasing precision, i.e., for $Q\to\infty$, this reduces to $O(n\log\frac{1}{\varepsilon})$.

6.2 Approximate consensus versus 2-set consensus

We now consider the *2-set consensus problem* which is another natural generalization of the consensus problem. Instead of requiring that processes agree to within any positive real-valued tolerance ε, processes have to decide on at most 2 different values:

Agreement. There are at most two different decision values.

Formally, each process starts with an input value from the set V of multiples of $1/Q$ and has to output a decision value from V in such a way that the termination and validity conditions in the consensus specification as well as the above agreement condition are satisfied.

The 2-set consensus problem naturally reduces to approximate consensus: processes round off their $1/Q$-agreement output values. Unfortunately, the use of averaging procedures to solve approximate consensus leads processes to exchange values out of the set V in the resulting 2-set consensus algorithms. The quantized amortized midpoint algorithm allows us to overcome this problem, and Theorem 12 shows that in any rooted network model, this algorithm achieves 2-set consensus in $(n-1)\left(\lfloor \log_2(Q-2) \rfloor + 2\right)$ rounds.

The above discussion shows that the 2-set consensus problem is solvable in a dynamic network model if all the communication graphs are rooted. In particular, 2-set consensus is solvable in a asynchronous complete network with a minority of faulty senders since nonsplit rounds can be implemented in this model. Combined with the impossibility result in [13] in the case of a strict majority of faulty processes, we obtain an exact characterization of the sender faulty models for which 2-set consensus is solvable in asynchronous systems if the number of processes n is odd and a small gap (namely $n/2$ faulty processes) when n is even.

Our positive result can be interestingly compared with the 2 faulty processes boundary [3, 17, 25] between possibility and impossibility of the original (and stronger) 2-set consensus problem [10] where decision values ought to be initial values, instead of being in the range of the initial values. That points out the crucial role of the validity condition on the solvability of 2-set consensus.

7 Conclusion

This paper presented a linear-time approximate consensus algorithm in rooted dynamic network models. For that, we introduced the amortization technique to speed up classical averaging algorithms from exponential to polynomial time, while preserving their inherent robustness. Careful study of the properties that make averaging algorithms contracting lead us to identify the midpoint algorithm as having the optimal contraction rate of 1/2. Central to our analysis of the amortized midpoint algorithm was the switch from the commonly employed matrix-based to a value-based view. An important property of the amortized midpoint algorithm is that it can be used almost unaltered with quantized values, which allows to decide on at most two neighboring values, thereby a fortiori solving 2-set consensus. Interestingly, it only needs to store and send two values in each round, which makes it space-efficient and viable for implementation.

References

1 David Angeli and Pierre-Alexandre Bliman. Stability of leaderless discrete-time multi-agent systems. *Mathematics of Control, Signals, and Systems*, 18(4):293–322, 2006.

2 Pierre-Alexandre Bliman, Angelia Nedic, and Asuman E. Ozdaglar. Rate of convergence for consensus with delays. In *Proceedings of the 47th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC)*, pages 2226–2231. IEEE, New York City, 2008.

3 Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 91–100. ACM, New York City, 1993.

4 Ming Cao, A. Stephen Morse, and Brian D. O. Anderson. Reaching a consensus in a dynamically changing environment: a graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.

5 Ming Cao, A. Stephen Morse, and Brian D. O. Anderson. Reaching a consensus in a dynamically changing environment: convergence rates, measurement delays, and asynchronous events. *SIAM Journal on Control and Optimization*, 47(2):601–623, 2008.

6 Ming Cao, Daniel A. Spielman, and A. Stephen Morse. A lower bound on convergence of a distributed network consensus algorithm. In Hannes Frey, Xu Li, and Stefan Rührup, editors, *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC)*, pages 2356–2361. IEEE, New York City, 2005.

7 Bernadette Charron-Bost. Orientation and connectivity based criteria for asymptotic consensus. arXiv:1303.2043v1 [cs.DC], 2013.

8 Bernadette Charron-Bost, Matthias Függer, and Thomas Nowak. Approximate consensus in highly dynamic networks: The role of averaging algorithms. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 528–539. Springer, Heidelberg, 2015.

9 Bernadette Charron-Bost and André Schiper. The Heard-Of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, 2009.

10 Soma Chaudhuri. More choices allow more faults: Set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, 1993.

11 Bernard Chazelle. The convergence of bird flocking. `arXiv:0905.4241 [cs.CG]`, 2009.

12 Bernard Chazelle. The total s-energy of a multiagent system. *SIAM Journal on Control and Optimization*, 49(4):1680–1706, 2011.

13 Roberto De Prisco, Dahlia Malkhi, and Michael K. Reiter. On k-set consensus problems in asynchronous systems. In *Proceedings of the 18th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 257–265. ACM, New York City, 1999.

14 Roland L. Dobrushin. Central limit theorem for non-stationary Markov chains I. *Theory of Probability & Its Applications*, 1(1):65–80, 1956.

15 Paolo Frasca, Ruggero Carli, Fabio Fagnani, and Sandro Zampieri. Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control*, 19(16):1787–1816, 2009.

16 R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3):1–33, 2002.

17 Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 111–120. ACM, New York City, 1993.

18 Akshay Kashyap, Tamer Basar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.

19 Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, December 1990.

20 Luc Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

21 S. Muthukrishnan, Bhaskar Ghosh, and Martin H. Schultz. First- and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory of Computing Systems*, 31(4):331–354, 1998.

22 Angelia Nedic, Alexander Olshevsky, Asuman E. Ozdaglar, and John N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

23 Alex Olshevsky. Linear time average consensus on fixed graphs. *IFAC-PapersOnLine*, 48(22):94–99, 2015.

24 Alex Olshevsky and John N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Review*, 53(4):747–772, 2011.

25 Michael E. Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: the topology of public knowledge. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, pages 101–110. ACM, New York City, 1993.

26 Nicola Santoro and Peter Widmayer. Time is not a healer. In B. Monien and R. Cori, editors, *Proceedings of the 6th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 349 of *LNCS*, pages 304–313. Springer, Heidelberg, 1989.

27 Steven H. Strogatz. From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1–4):1–20, 2000.

28 Y. Yuan, G.-B. Stan, L. Shi, M. Barahona, and J. Goncalves. Decentralized minimum-time consensus. *Automatica*, 49(5):1227–1235, 2013.