Brief Announcement: Communication-Optimal Convex Agreement

Diana Ghinea
ghinead@ethz.ch
ETH Zurich
Zurich, Switzerland

Chen-Da Liu-Zhang
chen-da.liuzhang@hslu.ch
Lucerne University of Applied Sciences and Arts & Web3 Foundation
Zug, Switzerland

Roger Wattenhofer
wattenhofer@ethz.ch
ETH Zurich
Zurich, Switzerland

ABSTRACT

Byzantine Agreement (BA) allows a set of n parties to agree on a value even when up to t of the parties involved are corrupted. While previous works have shown that, for ℓ-bit inputs, BA can be achieved with the optimal communication complexity $O(\ell n)$ for sufficiently large ℓ, BA only ensures that honest parties agree on a meaningful output when they hold the same input, rendering the primitive inadequate for many real-world applications.

This gave rise to the notion of Convex Agreement (CA), introduced by Vaidya and Garg [PODC'13], which requires the honest parties' outputs to be in the convex hull of the honest inputs. Unfortunately, all existing CA protocols incur a communication complexity of at least $\Omega(\ell n^2)$. In this work, we introduce the first CA protocol with the optimal communication of $O(\ell n)$ bits for inputs in $\mathbb{Z}$ of size $\ell = \Omega(\kappa \cdot n^2 \log n)$, where κ is the security parameter.

CCS CONCEPTS

• **Theory of computation → Cryptographic protocols**.

KEYWORDS

convex agreement, optimal communication, long messages

ACM Reference Format:
Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. 2024. Brief Announcement: Communication-Optimal Convex Agreement. In *ACM Symposium on Principles of Distributed Computing (PODC '24), June 17–21, 2024, Nantes, France*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3662158.3662782

Related Version: A full version of this paper is available at [18].

1 INTRODUCTION

Reaching collaborative decisions becomes tricky in decentralized systems, especially when participants might be unreliable or even malicious. This is where agreement protocols come in, acting as crucial tools for finding common ground. One such primitive is Byzantine Agreement (BA), where a group of n parties agree on a value, even if up to t of the parties are byzantine.

The standard BA definition comes with certain limitations when applied to real-world scenarios. Consider, for instance, a network of sensors deployed within a cooling room, responsible for measuring and reporting the room's temperature. One can expect minor discrepancies in the measurements, such as correct sensors obtaining temperatures between $-10.05°C$ and $-10.03°C$. In such a scenario, standard BA allows the honest parties to agree on a value proposed by the byzantine parties, such as $+100°C$, instead of requiring the output to reflect the correct sensors' measurements.

A stronger variant of BA, known as Convex Agreement (CA), addresses this issue, as it requires the honest parties to agree on a value within the convex hull of their inputs (or within the range of their inputs, if the input space is uni-dimensional). The synchronous model, where parties have synchronized clocks and messages get delivered within a publicly known amount of time, facilitates a straightforward approach for achieving CA through Synchronous Broadcast (BC). Essentially, each party sends its input value via BC, which provides the parties with an identical view of the inputs. Afterwards, the parties decide on a common output by applying a deterministic function to the values received. While this approach yields optimal solutions in terms of resilience and round complexity, there is still a gap in terms of communication. Specifically, if the honest parties hold inputs of at most ℓ bits, a lower bound on the communication complexity is $\Omega(\ell n)$ bits [26], and this approach incurs a sub-optimal communication cost of $\Omega(\ell n^2)$ bits. For BA and BC, this gap was long closed in a line of works [4, 14, 15, 22, 26] via so-called *extension protocols*, that achieve a communication complexity of $O(\ell n + \mathsf{poly}(n, \kappa))$ bits, where κ is a security parameter. In this work, we focus on closing this gap in the synchronous model for CA. In this setting, we ask the following question:

Can we achieve CA with the asymptotically optimal communication of $O(\ell n + \mathsf{poly}(n, \kappa))$ bits?

We answer this question in the affirmative. We introduce a deterministic protocol in the plain model (no setup) that achieves the optimal resilience $t < n/3$, optimal asymptotic communication complexity of $O(\ell n + \mathsf{poly}(n, \kappa))$ and round complexity $O(n \log n)$.[1] The protocol makes use of collision-resistant hash functions and takes as inputs ℓ-bit strings interpreted as integers.

2 RELATED WORK

Convex-Hull Validity. The requirement of obtaining outputs within the honest inputs' range has been first introduced in [9]

PODC '24, June 17–21, 2024, Nantes, France

ACM ISBN 979-8-4007-0668-4/24/06
https://doi.org/10.1145/3662158.3662782

[1]With randomization, our protocol can be made to achieve $O(\kappa \log n) = \tilde{O}(1)$ rounds.

for Approximate Agreement (AA). AA relaxes the agreement requirement, where parties' outputs may deviate by a predefined error $\varepsilon > 0$. This allows for deterministic asynchronous protocols, circumventing the FLP result [13], and it also has advantages in the synchronous model if n is $\Omega(\ell)$: the runtime of deterministic AA protocols may only depend on ℓ, bypassing the $O(n)$ rounds requirement [10]. AA has been a subject of an extensive line of works, focusing on optimal convergence rates [3, 11, 12], higher resilience [1, 16, 21], and different input spaces, such as multidimensional inputs [17, 24, 30], or abstract convexity spaces [2, 8, 20, 27].

CA was formally defined by Vaidya and Garg in [25, 30], assuming that the input space consists of multidimensional values. Feasibility with optimal resilience has been considered for abstract convexity spaces as well [8, 27]. Another line of works has investigated the feasibility of an even stronger requirement for inputs in $\mathbb{R}$, i.e. that the output is *close* to the median of the honest inputs [7, 28], or, more generally, to the k-th lowest honest input [23].

Extension Protocols. The problem of reducing the communication complexity of BA on multi-valued inputs was first addressed by Turpin and Coan [29], where the authors assume $t < n/3$ and give a reduction from long-messages BA to short-messages BA with a communication cost of $\Omega(\ell n^2)$ bits. Fitzi and Hirt [14] later achieve BA in the honest majority setting with the asymptotically optimal communication complexity $O(\ell n + \text{poly}(n, \kappa))$ bits, assuming a universal hash function. Further works have provided error-free solutions focusing on reducing the additional $\text{poly}(n, \kappa)$ factor in the communication complexity both in the $t < n/3$ [15, 22, 26] setting and in the honest-majority setting [4, 15, 26].

Extension protocols have also been a topic of interest for problems related to BA, such as BC in the $t < n$ setting [6, 19], or asynchronous Reliable Broadcast [5, 26].

Comparison to previous works. In terms of techniques, our solution differs significantly from both prior works on BA extension protocols and prior works on CA or AA. In comparison to BA, the honest-range requirement adds a new level of challenges when it comes to reducing communication. Roughly, in prior works on communication-optimal BA, each party first computes a short κ-bit encoding of its long input value (using e.g. a hash function). Afterwards, the parties agree on an encoding $z^\star$ using a BA protocol for short messages. Finally, parties holding the (unique) input $v^\star$ matching $z^\star$ non-trivially distribute $v^\star$ to all the parties. The main issue when trying to adapt this approach to CA is that the short κ-bit encodings cannot reflect the honest inputs' range. On the other hand, existing CA or AA protocols involve some step where all parties send their ℓ-bit values to all other parties. It might seem intuitive that the parties need a view of the actual inputs to decide on a valid output. However, we show that this intuition is not true.

3 OUR RESULT

We state our main theorem below. In the remainder of this paper, we describe the construction behind the theorem, outlining the main challenges and techniques.

THEOREM 1. *Assume a BA protocol resilient against $t < n/3$ corruptions with round complexity R and communication complexity B_κ for κ-bit inputs. Additionally, assume a collision-resistant hash function $H_\kappa : \{0,1\}^\star \to \{0,1\}^\kappa$. Then, there is a protocol achieving CA on $\mathbb{Z}$ resilient against $t < n/3$ corruptions, with round complexity $O(\log n) \cdot R$ and communication complexity $O(\ell n + \kappa \cdot n^3 \log n) + O(n \log n) \cdot B_\kappa$ for ℓ-bit inputs.*

Note that $O(\ell n + \text{poly}(n, k))$ bits do not allow for a step where the parties distribute their ℓ-bit values. Instead, we aim to only work with the values' prefixes. In the following, we solely concentrate on inputs in $\mathbb{N}$. To extend the protocol to $\mathbb{Z}$, the parties may first agree on their input values' sign using a BA protocol. Parties holding input values with a different sign set their value to 0. Afterwards, the parties run the steps we describe below on inputs in $\mathbb{N}$.

For intuition, it will be useful to arrange the honest inputs' range in a so-called *prefix tree* (or *trie*). As shown in Figure 1, a prefix tree is a (rooted) tree where each node stores a string's prefix. The edges from nodes to their children are labelled with characters (0 or 1) indicating the prefixes stored on the children. To achieve CA, it is sufficient for the parties to find a leaf in this prefix tree.

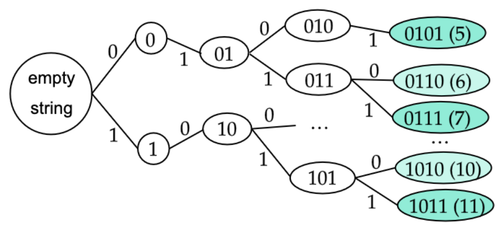

Figure 1: Prefix tree storing the honest inputs' range, assuming that $\ell = 4$ and the honest inputs are 5, 7 and 11.

As the inputs may be of different lengths, to compare prefixes effectively, we enable the parties to agree on a value $\ell_{\text{EST}} \le \ell + n$ such that every party can modify its input to a valid ℓ_{EST}-bits value.

3.1 Warm-up

As a starting point towards our solution, we describe a simple (yet inefficient) approach that finds a leaf in the prefix tree of the honest inputs' range using ℓ_{EST} iterations.

We need to establish a few notations. For a value $v \in \mathbb{N}$, we define its binary representation $\textsc{bits}(v) := \textsc{b}_1\textsc{b}_2 \dots \textsc{b}_k$ such that $2^{k-1} \le v < 2^k$, $\textsc{b}_i \in \{0,1\}$ for every $1 \le i \le k$, and $\sum_{i=1}^{k} \textsc{b}_i \cdot 2^{k-i} = v$. The reverse operation will be $\textsc{val}(\textsc{bits})$. For $\ell \ge k$, we additionally define $\textsc{bits}_\ell(v)$ as the ℓ-bit string obtained by prepending $\ell - k$ zeroes to $\textsc{bits}(v)$. The length of a bitstring $\textsc{bits}$ is denoted by $|\textsc{bits}|$.

In iteration i, the parties hold valid values v such that the bit representations $\textsc{bits}_{\ell_{\text{EST}}}(v)$ have a common prefix $\textsc{prefix}^\star$ of $i - 1$ bits. The parties extend the common prefix with one bit using a BA protocol Π_{BA}: they join Π_{BA} with input $\textsc{b}_i :=$ the i-th bit of $\textsc{bits}_{\ell_{\text{EST}}}(v)$ and agree on bit $\textsc{prefix}_i^\star$. Parties holding $\textsc{b}_i \ne \textsc{prefix}_i^\star$ need to update their value v to some *valid* value matching the prefix agreed upon. We know that $\textsc{prefix}_i^\star$ was proposed by an honest party, hence $\textsc{prefix} \parallel \textsc{prefix}_i^\star$ is the prefix of a valid ℓ_{EST}-bit value $v^\star$. This allows the parties to update their values as follows: if $\textsc{b}_i = 0$ and $\textsc{prefix}_i^\star = 1$, meaning that $v < v^\star$, then the lowest ℓ_{EST}-bit value having prefix $\textsc{prefix}^\star \parallel \textsc{prefix}_i^\star$ is in $[v, v^\star]$ and therefore is valid. Similarly, if $\textsc{b}_i = 1$ and $\textsc{prefix}_i^\star = 0$, meaning that $v > v^\star$, then the highest ℓ_{EST}-bit value having prefix $\textsc{prefix}^\star \parallel \textsc{prefix}_i^\star$ is in $[v^\star, v]$ and therefore is valid.

For a bitstring PREFIX of at most ℓ bits, $\textsc{max}_\ell(\textsc{prefix})$ denotes the highest ℓ-bit value having PREFIX as prefix (obtained by concatenating PREFIX with $\ell - |\textsc{prefix}|$ ones). Similarly, $\textsc{min}_\ell(\textsc{prefix})$ denotes the lowest ℓ-bit value having PREFIX as prefix (obtained by concatenating PREFIX with $\ell - |\textsc{prefix}|$ zeroes). The remark below then ensures that the update step indeed leads to valid values, therefore achieving CA at the end of iteration ℓ_{EST}.

Remark 1. *Consider two values $v, v' \in \mathbb{N}$ satisfying $v \le v' < 2^\ell$, and let COMMON_PREFIX be the **longest** common prefix of $\textsc{bits}_\ell(v)$ and $\textsc{bits}_\ell(v')$. If $|\textsc{common\_prefix}| < \ell$, then $\textsc{max}_\ell(\textsc{common\_prefix} \,\|\, 0), \textsc{min}_\ell(\textsc{common\_prefix} \,\|\, 1) \in [v, v']$.*

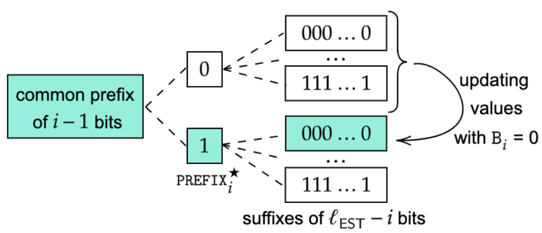

Figure 2: In iteration i, the parties hold values with a common prefix of $i - 1$ bits, and agree on the i-th bit $\textsc{prefix}^\star_i$. If $\textsc{prefix}^\star_i = 1$, parties holding $\textsc{b}_i = 0$ update their values.

3.2 From bits to blocks

The above warm-up solution has communication $O(\ell n^2)$. To achieve an asymptotically optimal solution, instead of building some valid values' prefix *bit by bit*, we may do so *block by block*. Assume without loss of generality that ℓ_{EST} is a multiple of n. Then, for $v \in \mathbb{N}$ with $|\textsc{bits}(v)| \le \ell_{\text{EST}}$, let $\textsc{blocks}(v) := (\textsc{block}_1, \textsc{block}_2, \dots, \textsc{block}_n)$ such that $\textsc{bits}_{\ell_{\text{EST}}}(v) = \textsc{block}_1 \,\|\, \textsc{block}_2 \,\|\, \dots \,\|\, \textsc{block}_n$, and, for any $1 \le i \le n$, $|\textsc{block}_i| = \ell_{\text{EST}}/n$. For $1 \le i \le n$, use $\textsc{block}_i(v)$ to refer to $\textsc{block}_i$. We use the term *block* to refer to such sequences of ℓ_{EST}/n bits. Following the outline of the warm-up approach, in iteration i, the parties hold valid ℓ_{EST}-bit values v with a common prefix $\textsc{prefix}^\star$ of $i - 1$ blocks. In an attempt to extend $\textsc{prefix}^\star$ by one block $\textsc{prefix}^\star_i$, the parties join a BA protocol $\Pi_{\ell\text{BA}}$ (for long messages) with $\textsc{block}_i(v)$ as input.

When the parties agree on a block. If the parties agree on a block $\textsc{prefix}^\star_i$, the honest parties holding $\textsc{block}_i \neq \textsc{prefix}^\star_i$ should update their values v to match the prefix agreed upon. However, unless all honest parties hold $\textsc{block}_i = \textsc{prefix}^\star_i$, $\textsc{prefix}^\star_i$ may be a block proposed by a corrupted party, forcing the updated values outside the honest range. To prevent this, we make use of the special symbol $\perp$, and we require $\Pi_{\ell\text{BA}}$ the following property.

Definition 1. *Honest Output: If the honest parties output $v \neq \perp$, then v is some honest party's input.*

If $\Pi_{\ell\text{BA}}$ satisfies Honest Output and the parties agree on a block $\textsc{prefix}^\star_i$, then $\textsc{prefix}^\star \,\|\, \textsc{prefix}^\star_i$ is the prefix of an honest party's (valid) value. If a party P holds value v with $\textsc{block}_i(v) \neq \textsc{prefix}^\star_i$, it updates v to match the prefix agreed upon. If $\textsc{block}_i < \textsc{prefix}^\star_i$, then P updates its value as $v := \textsc{min}_{\ell_{\text{EST}}}(\textsc{prefix}^\star \,\|\, \textsc{prefix}^\star_i)$, and, if $\textsc{block}_i > \textsc{prefix}^\star_i$, P updates its value as $v := \textsc{max}_{\ell_{\text{EST}}}(\textsc{prefix}^\star \,\|\, \textsc{prefix}^\star_i)$. In both cases, the updated value remains valid.

When the parties agree on $\perp$. If $\Pi_{\ell\text{BA}}$ returns $\perp$ in some iteration $i^\star \le n$, honest parties hold different blocks $\textsc{block}_{i^\star}$. In fact, this means that we are very close to finding a valid output.

Looking at Figure 1, a crucial observation is that nodes that have two children, and hence that store valid values' longest common prefixes, reveal subsets of the honest inputs' range. For example, the node storing 01 indicates that the highest 4-bit value with prefix 010 (in this case, this is 5) and the lowest value with prefix 011 (namely, 6) are valid. This means that, once the parties identify some valid values' longest common prefix, they may immediately derive an output with the help of Remark 1. However, this property applies to valid values' longest common prefix *of bits*, while the parties are only aware of a longest common prefix *of blocks*: some of the bits in block $i^\star$ may be common. We then enable the honest parties to find the common bits in block $i^\star$ by adding one more property to $\Pi_{\ell\text{BA}}$, defined below. Then, parties may distribute their blocks $i^\star$ via BC, and this additional property allows us to identify two different blocks $i^\star$ that lead to prefixes of valid values.

Definition 2. *Bounded Pre-agreement: If the honest parties output $\perp$, then at most t honest parties hold the same input value.*

3.3 A round-efficient approach

Although the approach described achieves our goal regarding communication complexity, the round complexity is $O(n)$ times the round complexity of $\Pi_{\ell\text{BA}}$. We reduce the number of iterations from $O(n)$ to $O(\log n)$ (while maintaining the communication complexity) by employing *binary search*: the parties are looking for an index $i^\star$ such that, roughly, $\Pi_{\ell\text{BA}}$ returns $\perp$ on valid values' prefixes of $i^\star$, but not on valid values' prefixes of $i^\star - 1$ blocks. Then, we proceed as follows: in the first iteration, the parties check whether $\Pi_{\ell\text{BA}}$ returns $\perp$ on the first half of their blocks $\textsc{block}_1 \,\|\, \dots \,\|\, \textsc{block}_{\textsc{mid}}$. If $\Pi_{\ell\text{BA}}$ returns $\perp$, MID is an upper bound for $i^\star$, and we continue the search for $i^\star$ within the first half of the blocks $\textsc{block}_1, \dots, \textsc{block}_{\textsc{mid}-1}$ in the next iteration, using an identical approach. Otherwise, if $\Pi_{\ell\text{BA}}$ returns a bitstring of MID blocks $\textsc{prefix}^\star_1 \,\|\, \dots \,\|\, \textsc{prefix}^\star_{\textsc{mid}}$, the parties update their values to match this prefix and use the same approach to find $i^\star$ within the second half of their updated values' blocks in the next iteration. After $O(\log n)$ iterations, either $\Pi_{\ell\text{BA}}$ never returned $\perp$ and the parties now hold identical values, or $i^\star$ is found.

This approach introduces some challenges for deciding on the final output once $i^\star$ is found. At the end of $O(\log n)$ iterations, honest parties' values have a common prefix of $i^\star - 1$ blocks. As opposed to the previous solution, these values might have been updated, and now the $i^\star$-th block might also be common. Instead, we make use of the values $v_\perp$ held in the last iteration where $\Pi_{\ell\text{BA}}$ has returned $\perp$: Bounded Pre-Agreement holds for these values' prefixes of $i^\star$ blocks. The parties first obtain a valid values' prefix of $i^\star$ from their values v. Then, each honest party that holds a value $v_\perp$ not matching this prefix *complains* by announcing the first bit where $v_\perp$ differs from the prefix agreed upon. Each honest complaint will lead to a valid value. Due to Bounded Pre-Agreement, there are sufficiently many honest complaints so that parties identify a valid value and agree on it.

REFERENCES

[1] Ittai Abraham, Yonatan Amit, and Danny Dolev. 2005. Optimal Resilience Asynchronous Approximate Agreement. In *Principles of Distributed Systems*, Teruo Higashino (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 229–239.

[2] Dan Alistarh, Faith Ellen, and Joel Rybicki. 2021. Wait-Free Approximate Agreement on Graphs. In *Structural Information and Communication Complexity*, Tomasz Jurdziński and Stefan Schmid (Eds.). Springer International Publishing, Cham, 87–105. https://doi.org/10.1007/978-3-030-79527-6_6

[3] Michael Ben-Or, Danny Dolev, and Ezra N. Hoch. 2010. Brief Announcement: Simple Gradecast Based Algorithms. In *Distributed Computing*, Nancy A. Lynch and Alexander A. Shvartsman (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 194–197.

[4] Amey Bhangale, Chen-Da Liu-Zhang, Julian Loss, and Kartik Nayak. 2023. Efficient adaptively-secure byzantine agreement for long messages. In *Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part I*. Springer, Springer-Verlag, Berlin, Heidelberg, 504–525.

[5] Christian Cachin and Stefano Tessaro. 2005. Asynchronous verifiable information dispersal. In *24th IEEE Symposium on Reliable Distributed Systems (SRDS'05)*. IEEE, IEEE Computer Society, Orlando, FL, USA, 191–201. https://doi.org/10.1109/RELDIS.2005.9

[6] Wutichai Chongchitmate and Rafail Ostrovsky. 2018. Information-Theoretic Broadcast with Dishonest Majority for Long Messages. In *TCC 2018, Part I (LNCS, Vol. 11239)*, Amos Beimel and Stefan Dziembowski (Eds.). Springer, Heidelberg, Cham, 370–388. https://doi.org/10.1007/978-3-030-03807-6_14

[7] Andrei Constantinescu, Diana Ghinea, Lioba Heimbach, Zilin Wang, and Roger Wattenhofer. 2024. A Fair and Resilient Decentralized Clock Network for Transaction Ordering. In *27th International Conference on Principles of Distributed Systems (OPODIS 2023) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 286)*, Alysson Bessani, Xavier Défago, Junya Nakamura, Koichi Wada, and Yukiko Yamauchi (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 8:1–8:20. https://doi.org/10.4230/LIPIcs.OPODIS.2023.8

[8] Andrei Constantinescu, Diana Ghinea, Roger Wattenhofer, and Floris Westermann. 2023. Meeting in a Convex World: Convex Consensus with Asynchronous Fallback. Cryptology ePrint Archive, Paper 2023/1364. https://eprint.iacr.org/2023/1364

[9] Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William E. Weihl. 1986. Reaching Approximate Agreement in the Presence of Faults. *J. ACM* 33, 3 (May 1986), 499–516. https://doi.org/10.1145/5925.5931

[10] Danny Dolev and H. Raymond Strong. 1983. Authenticated algorithms for Byzantine agreement. *SIAM J. Comput.* 12, 4 (1983), 656–666.

[11] A. D. Fekete. 1987. Asynchronous approximate agreement. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing* (Vancouver, British Columbia, Canada) *(PODC '87)*. Association for Computing Machinery, New York, NY, USA, 64–76. https://doi.org/10.1145/41840.41846

[12] Alan David Fekete. 1990. Asymptotically optimal algorithms for approximate agreement. *Distributed Computing* 4, 1 (1990), 9–29.

[13] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. 1985. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)* 32, 2 (1985), 374–382.

[14] Matthias Fitzi and Martin Hirt. 2006. Optimally efficient multi-valued Byzantine agreement. In *25th ACM PODC*, Eric Ruppert and Dahlia Malkhi (Eds.). ACM, New York, NY, USA, 163–168. https://doi.org/10.1145/1146381.1146407

[15] Chaya Ganesh and Arpita Patra. 2016. Broadcast Extensions with Optimal Communication and Round Complexity. In *35th ACM PODC*, George Giakkoupis (Ed.). ACM, New York, NY, USA, 371–380. https://doi.org/10.1145/2933057.2933082

[16] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. 2022. Optimal Synchronous Approximate Agreement with Asynchronous Fallback. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing* (Salerno, Italy) *(PODC'22)*. Association for Computing Machinery, New York, NY, USA, 70–80. https://doi.org/10.1145/3519270.3538442

[17] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. 2023. Multidimensional Approximate Agreement with Asynchronous Fallback. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures* (Orlando, FL, USA) *(SPAA '23)*. Association for Computing Machinery, New York, NY, USA, 141–151. https://doi.org/10.1145/3558481.3591105

[18] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. 2024. Communication-Optimal Convex Agreement. Cryptology ePrint Archive, Paper 2024/251. https://eprint.iacr.org/2024/251

[19] Martin Hirt and Pavel Raykov. 2014. Multi-valued Byzantine Broadcast: The $t < n$ Case. In *ASIACRYPT 2014, Part II (LNCS, Vol. 8874)*, Palash Sarkar and Tetsu Iwata (Eds.). Springer, Heidelberg, Berlin, Heidelberg, 448–465. https://doi.org/10.1007/978-3-662-45608-8_24

[20] Jérémy Ledent. 2021. Brief Announcement: Variants of Approximate Agreement on Graphs and Simplicial Complexes. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing* (Virtual Event, Italy) *(PODC'21)*. Association for Computing Machinery, New York, NY, USA, 427–430. https://doi.org/10.1145/3465084.3467946

[21] Christoph Lenzen and Julian Loss. 2022. Optimal Clock Synchronization with Signatures. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing* (Salerno, Italy) *(PODC'22)*. Association for Computing Machinery, New York, NY, USA, 440–449. https://doi.org/10.1145/3519270.3538444

[22] Guanfeng Liang and Nitin Vaidya. 2011. Error-free multi-valued consensus with Byzantine failures. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*. Association for Computing Machinery, New York, NY, USA, 11–20.

[23] Darya Melnyk and Roger Wattenhofer. 2018. Byzantine Agreement with Interval Validity. In *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*. IEEE Computer Society, Salvador, Brazil, 251–260. https://doi.org/10.1109/SRDS.2018.00036

[24] Hammurabi Mendes and Maurice Herlihy. 2013. Multidimensional approximate agreement in Byzantine asynchronous systems. In *45th ACM STOC*, Dan Boneh, Tim Roughgarden, and Joan Feigenbaum (Eds.). ACM Press, Palo Alto, CA, USA, 391–400. https://doi.org/10.1145/2488608.2488657

[25] Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. 2015. Multidimensional agreement in Byzantine systems. *Distributed Computing* 28, 6 (2015), 423–441.

[26] Kartik Nayak, Ling Ren, Elaine Shi, Nitin H. Vaidya, and Zhuolun Xiang. 2020. Improved Extension Protocols for Byzantine Broadcast and Agreement. In *34th International Symposium on Distributed Computing (DISC 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 179)*, Hagit Attiya (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 28:1–28:17. https://doi.org/10.4230/LIPIcs.DISC.2020.28

[27] Thomas Nowak and Joel Rybicki. 2019. Byzantine Approximate Agreement on Graphs. In *33rd International Symposium on Distributed Computing (DISC 2019) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 146)*, Jukka Suomela (Ed.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 29:1–29:17. https://doi.org/10.4230/LIPIcs.DISC.2019.29

[28] David Stolz and Roger Wattenhofer. 2016. Byzantine Agreement with Median Validity. In *19th International Conference on Principles of Distributed Systems (OPODIS 2015) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 46)*, Emmanuelle Anceaume, Christian Cachin, and Maria Potop-Butucaru (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 1–14. https://doi.org/10.4230/LIPIcs.OPODIS.2015.22

[29] Russell Turpin and Brian A Coan. 1984. Extending binary Byzantine agreement to multivalued Byzantine agreement. *Inform. Process. Lett.* 18, 2 (1984), 73–76.

[30] Nitin H. Vaidya and Vijay K. Garg. 2013. Byzantine vector consensus in complete graphs. In *32nd ACM PODC*, Panagiota Fatourou and Gadi Taubenfeld (Eds.). ACM, Montreal, QC, 65–73. https://doi.org/10.1145/2484239.2484256