



**HAL**  
open science

## Brief Announcement: Know Your Audience

Bernadette Charron-Bost, Patrick Lambein-Monette

► **To cite this version:**

Bernadette Charron-Bost, Patrick Lambein-Monette. Brief Announcement: Know Your Audience. PODC '24: 43rd ACM Symposium on Principles of Distributed Computing, Jun 2024, Nantes (France), France. pp.243-246, 10.1145/3662158.3662784 . hal-04796749

**HAL Id: hal-04796749**

**<https://hal.science/hal-04796749v1>**

Submitted on 21 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Brief Announcement: Know your audience

## Communication models and computability in anonymous networks

Bernadette Charron-Bost  
CNRS, DI ENS  
École Normale Supérieure  
Paris, France  
charron@di.ens.fr

Patrick Lambein-Monette  
Unaffiliated  
Paris, France  
patrick@lambein.name

### ABSTRACT

In distributed computing, questions of computability are exquisitely sensitive to minute details of the model assumptions, and there is no universally agreed upon model of network computing. Here, we study which functions are computable by deterministic and anonymous agents in either static or dynamic networks. We consider various communication assumptions common in the literature, and in each case we strive to characterize the set of computable functions, organizing existing results as well as offering new ones, alongside new proofs which bring new understanding of this computability landscape.

### CCS CONCEPTS

• **Theory of computation** → **Computability; Distributed algorithms.**

### KEYWORDS

anonymous networks, distributed computability

#### ACM Reference Format:

Bernadette Charron-Bost and Patrick Lambein-Monette. 2024. Brief Announcement: Know your audience: Communication models and computability in anonymous networks. 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/10.1145/3662158.3662784>

## 1 INTRODUCTION

Function computation [1–4, 6, 8, 10, 11, 13–15, 17–23, 25–32, 34, 35] is a fundamental problem of distributed computing in networks. Often, it is assumed that each agent in the network starts with a unique identifier, e.g., in the well-known LOCAL model, in which case the question of *which* functions can be computed becomes essentially moot. Indeed, by running a straightforward flooding algorithm, it is possible for all  $n$  agents to recover all pairs  $((id_i, val_i))_{1 \leq i \leq n}$ , after which any function  $f(val_1, \dots, val_n)$  can be computed, as long as it is invariant under permutation of its arguments. Similarly, if we allow randomized algorithms, then once again various methods allow for computing the same functions, possibly with some caveats such as “with high probability”. In short, with either identifiers or

randomization, questions of distributed computability in networks are rather trivial.

If, however, we look at anonymous networks of deterministic agents, then computability becomes a non-trivial question of much importance, as function computation is an extremely commonplace building block of many distributed applications. Here, our objective is to understand the fundamental limitations and capabilities for function computation by networked systems of anonymous, deterministic agents, operating under asynchronous starts in either static or dynamic networks. We do this by describing a minimalist model of network computing, and then studying how *varying the model assumptions*, in particular as it relates to the communication primitives, impacts the set of computable functions. We cover the common communication models found in the literature, and, in each case, strive to characterize computable functions. The current paper only gives a high-level summary of our work. While Tables 1 and 2 give a high-level summary of our argument, we refer the reader to our preprint [12] for precise definitions of our models, methods, results, as well as their proofs.

We define three classes of functions of arbitrary arity, i.e., of type  $f : \bigcup_{n \geq 1} \Omega^n \rightarrow X$ . First, *set-based* functions, whose value only depend on the set of its input values. Second, *frequency-based* functions, which can depend on the *relative frequencies* of the input. Third, *multiset-based* functions, which can depend on the multiset of the input. A prototypical example of a set-based function is the maximum, an example of a frequency-based function is the average, and an example of a multiset-based function is the sum. As these examples illustrate, the inclusion between the three classes is strict.

In terms of communication, we consider four distinct models. In the base case of *blind (local) broadcast*, in a round  $t$ , an agent  $i$  prepares and sends a message  $m_i(t)$  which is delivered to each of  $i$ 's round  $t$  outneighbors. This message is entirely determined from the agent's state obtained at the end of the previous round. Our second model, *outdegree awareness*, only differs in one way: the message  $m_i(t)$  may now also depend on agent  $i$ 's outdegree in the round  $t$  communication graph. Transmissions in these two models are *isotropic*, in the sense that, in any given round, any two messages coming from the same sender are the same. In contrast, in our third model of *output port awareness*, each agent  $i$  has access to different ports, each linking to a different outneighbor, and sends one message per port in each round.<sup>1</sup> Finally, our fourth model of *symmetric communications* corresponds to the restriction of the blind broadcast model, to the class of networks with bidirectional links.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. Request permissions from owner/author(s).  
PODC '24, June 17–21, 2024, Nantes, France

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0668-4/24/06  
<https://doi.org/10.1145/3662158.3662784>

<sup>1</sup>In this work, we only consider output port awareness in the context of static networks, as it is difficult to meaningfully define output ports in a dynamic context.

**Table 1: Computable functions in static, strongly connected networks of  $n$  anonymous agents**

	SIMPLE BROADCAST	OUTDEGREE AWARENESS	SYMMETRIC COMMUNICATIONS	OUTPUT PORT AWARENESS
no centralized help	set-based	frequency-based	frequency-based	frequency-based
a bound over $n$ is known	set-based	frequency-based	frequency-based	frequency-based
$n$ is known	set-based <sup>a</sup>	multiset-based	multiset-based	multiset-based
one leader	set-based	multiset-based	multiset-based	multiset-based

<sup>a</sup> J. Chalopin pointed out to us that this result does not hold for  $n = 2$  or  $3$ .

## 2 PRELIMINARIES

*Network and computation model.* We adopt a standard model of distributed computing in networks. In each round  $t = 1, 2, \dots$  of an execution, each agent  $i \in [n] := \{1, \dots, n\}$  sends a message, receives some messages from the network, then undergoes an internal transition to a new state before starting the next round. The patterns of message reception in a given round  $t$  define a directed graph  $\mathbb{G}(t) = ([n], E_t)$ , where  $(i, j) \in E_t$  means that agent  $j$  received agent  $i$ 's message in round  $t$ . The infinite sequence  $\mathbb{G} = \mathbb{G}(t)_{t \geq 1}$  is the *dynamic graph* modeling the behavior of the network over the execution.

The knowledge that each agent may possess about the network is captured by a set of possible dynamic graphs, called a *network class*, over which the execution might take place: the larger the class, the less information the agents have about the network. Since we consider *anonymous* networks, we restrict our attention to network classes that are closed under graph isomorphisms. Then, for example, the class of static networks can be expressed as  $\bigcup_{n \geq 1} \{([n], E_t) \mid \forall t, t' : E_t = E_{t'}\}$ , and similarly for other network classes. As we are looking for algorithms that tolerate asynchronous starts, we remark that an agent who begins its execution in some round  $s + 1$  can be modeled as an isolated vertex in each communication graph  $\mathbb{G}(t)$  for  $1 \leq t \leq s$ , so that our dynamic graph model captures the case of asynchronous starts. In contrast, a self-stabilizing algorithm is expected to produce a correct result under synchronous starts and arbitrary initial states of the agents. Self-stabilizing algorithms obviously tolerate asynchronous starts, while the converse is not necessarily true.

Finally, let us briefly recall what is meant by “computing” a function in an anonymous context. Unless the algorithm has a constant runtime, it is generally impossible to get the agents to irrevocably decide on a return value. Indeed, this would essentially amount to computing a bound  $N \geq n$ , which is impossible, as is shown here. Instead, each agent  $i$  is equipped with an output variable  $x_i$ . Given a metric space  $(X, \delta)$ , we say that an execution  $\delta$ -computes some value  $x^*$  if  $\lim_{t \rightarrow \infty} \delta(x_i(t), x^*) = 0$  for each agent  $i$ . An algorithm  $\mathcal{A}$  is then said to  $\delta$ -compute some function  $f : \bigcup_{n \geq 1} \Omega \rightarrow X$  on some dynamic graph  $\mathbb{G}$  with  $n$  nodes if any of its executions starting with input values  $v_1, \dots, v_n$   $\delta$ -computes the value  $f(v_1, \dots, v_n)$ . Finally, an algorithm  $\mathcal{A}$   $\delta$ -computes the function  $f$  on the *network class*  $\mathcal{C}$  if it is the case for every graph of the class, in which case the function  $f$  is said to be  $\delta$ -computable. Note that if  $\delta$  is the discrete metric, then in any execution there exists some round  $t^*$  after which

all variables  $x_i(t)$  are constant. We therefore say that the algorithm computes  $f$  in *finite time*, in which case  $f$  is computable for every other metric. Similarly, if  $X = \mathbb{R}^k$ , an algorithm which computes  $f$  for the Euclidean metric is said to compute  $f$  *asymptotically* or *approximately*.

*Graph fibrations.* Over static graphs, the homotopic notion of *graph fibration* [7] captures the idea of indistinguishability of nodes with respect to the executions of a distributed algorithm. Recall that a (directed, multi-)graph  $G$  is composed of a non-empty set of vertices  $V_G = \{1, \dots, n_G\}$ , and a set  $E_G$  of edges defined by two functions  $s_G, t_G : E_G \rightarrow V_G$ , which specify the source and the target vertices of each edge. A *graph morphism* between two graphs  $G$  and  $H$ , which we denote  $\varphi : G \rightarrow H$ , is defined by a pair of functions  $\varphi_V : V_G \rightarrow V_H$  and  $\varphi_E : E_G \rightarrow E_H$  which commute with the source and targets functions. A *fibration* is a surjective morphism  $\varphi : G \rightarrow B$  with the additional following property: for every edge  $e$  of the graph  $B$  (called the *base* of  $\varphi$ ) and for each vertex  $i$  of the graph  $G$  with  $\varphi_V(i) = t_B(e)$ , there exists a *unique* edge  $\tilde{e}^i$  of  $G$  such that  $\varphi_E(\tilde{e}^i) = e$  and  $t_G(\tilde{e}^i) = i$ . The *fibres*  $\varphi^{-1}(i)$  of some vertex  $i \in V_B$  is the set of vertices of  $G$  that are mapped to  $i$ . A fibration  $\varphi : G \rightarrow B$  defines an equivalence relation on  $V_G$  whose classes are the fibres: for any  $i, j, k$  such that  $j, k \in \varphi^{-1}(i)$ , the nodes  $i, j, k$  have similar neighborhoods, and there is a bijective correspondence between the edges of  $G$  coming to  $j$  and those coming to  $k$ . Informally, a morphism is a fibration when it is a local isomorphism with respect to the incoming edges of each nodes. All of the above notions extend to the case where either the edges or vertices of the graphs are colored, with the additional condition that morphisms ought to be compatible with the coloring functions.

Graph fibrations were developed by Boldi and Vigna [5–7, 9] as a refinement of the concept of graph coverings, which Angluin used in her seminal work [1], and in subsequent works by others [16, 24, 33, 35] to study computability in anonymous networks. The power of fibrations lie in the fundamental *Lifting lemma* [9, Lemma 2], which, informally, states that for any fibration  $\varphi : G \rightarrow B$  and any execution of a distributed algorithm  $\mathcal{A}$  over  $B$ , there exists an execution of  $\mathcal{A}$  over  $G$  in which all nodes of the same fiber  $\varphi^{-1}(i)$  are indistinguishable, and their behavior is the same as that of  $i$  in the original execution over  $B$ . The execution of  $\mathcal{A}$  over the graph  $B$  is entirely determined by the input values of the nodes in  $B$ , and the execution over the graph  $G$  is obtained by copying fibre-wise the input values, i.e., each node  $j \in V_G$  gets input  $\omega_j^\varphi := \omega_{\varphi_V(j)}$ ,

**Table 2: Computable functions in dynamic networks of  $n$  anonymous agents with finite diameter**

	SIMPLE BROADCAST	OUTDEGREE AWARENESS	SYMMETRIC COMMUNICATIONS
no centralized help	set-based	?	? <sup>a</sup>
a bound over $n$ is known	set-based	frequency-based	frequency-based
$n$ is known	set-based	multiset-based	multiset-based
one leader	set-based	?	? <sup>a</sup>

<sup>a</sup> Di Luna and Viglietta's algorithms [22, 23] exactly compute frequency-based functions only in the model of synchronous starts.

where  $\omega \in \Omega^{V_B}$  is the input vector for the graph  $B$ . From the Lifting lemma, we get the following corollary for computability.

**LEMMA 2.1.** *Let  $\varphi : G \rightarrow B$  be a fibration, and  $f$  be a function of variable arity over  $\Omega$ . If some algorithm  $\delta$ -computes  $f$  on both graphs  $G$  and  $B$ , then for any  $\omega \in \Omega^{V_B}$  we have  $f(\omega) = f(\omega^\varphi)$ .*

As we consider network classes that are closed under graph isomorphisms, Lemma 2.1 implies that computable functions have to be insensitive to the permutation of their arguments.

### 3 COMPUTABILITY RESULTS

*Static networks.* The notion of graph fibrations allows us to establish our main theorem for static networks: an exact characterization of computable functions when assuming one of a) output port awareness, b) symmetric communication links, or c) outdegree awareness.

**THEOREM 3.1.** *Let  $(X, \delta)$  be a metric space. In any of the three communication models with either output port awareness, symmetric communications, or outdegree awareness, and for any function  $f : \bigcup_{n \in \mathbb{N}_{>0}} \Omega^n \rightarrow X$ , the following assertions are equivalent:*

- (i)  $f$  is frequency-based;
- (ii)  $f$  is  $\delta$ -computable in the class of static strongly connected networks.

The proof can be found in our pre-print [12]. For the impossibility result, we extend an existing argument for symmetric communications [18]. For the positive results, we start with Boldi and Vigna's universal self-stabilizing algorithm [9], which is finite-state and computes in finite time a *minimum base* of the network graph  $G$ , i.e., a graph  $B^*$  such that i) there exists a fibration  $\varphi^* : G \rightarrow B^*$ , ii) any fibration  $B^* \rightarrow B'$  is an isomorphism (the graph  $B^*$  is *fibration prime*). Under any of our three communication assumptions, it is possible to show that, from the minimum base  $B^*$ , one can reconstruct the vector of the cardinalities of the fibres of  $\varphi^*$  up to a constant multiplicative vector, from which we deduce the relative frequencies of the input values of the execution. This allows us to compute any frequency-based function for the discrete metric, and thus for any metric. While it was known that output port awareness or symmetric communications make it possible to reconstruct the frequency vector, the novelty of our approach is to show the same for outdegree awareness, as well as to show that the three approaches work because they imply the same constraint on the

vector of fibre cardinalities for any fibration with base  $B^*$  – namely, that they lie in a subspace of rank 1 which can be computed from the graph  $B^*$ .

In addition to Theorem 3.1, we deduce additional results by considering additional assumptions: namely, if some agent is initially distinguished as leader, if all agents know the number  $n$  of agents, or if all agents know some bound  $N \geq n$ . Our computability results are collected in Table 1.

*Dynamic networks.* Positive results are, unsurprisingly, harder to obtain in the dynamic case, and, in contrast to the static case, the computability landscape isn't complete. Recently, Di Luna and Viglietta have proposed a remarkable method for dynamic networks with symmetric communications [22, 23], which allows for computing frequency-based functions in finite time, with a runtime that is linear in the dynamic diameter of the network. In particular, if every communication graph is strongly connected, then the runtime of the algorithm is linear in  $n$ . Unfortunately, this algorithm is not self-stabilizing, nor does it tolerate asynchronous starts. Moreover, as the computation depends on the construction of an infinitely growing *history tree*, the algorithm requires both an infinite number of states and an infinite bandwidth.

Here, we take another approach to compute frequency-based functions, based on *average consensus* algorithms derived from statistical physics – namely, the *Metropolis* and *Push-Sum* algorithms, both of which use iterated averaging to asymptotically compute the average of input values. Each algorithm works for a different model: Push-Sum assumes outdegree awareness, while Metropolis requires both outdegree awareness and symmetric communications; we previously proposed a variant of the latter [11] which doesn't require outdegree awareness to compute the average, at the cost of a worse runtime. Compared to Di Luna and Viglietta's linear-time algorithm, these methods only compute the average in an approximate manner, and their runtime is non-linear; however, these algorithms are conceptually very simple, they tolerate asynchronous starts, and they use no persistent memory, in the sense that state-update rules correspond to discrete differential equations of order one. Assuming either a bound  $N \geq n$  or knowledge of  $n$ , we can turn these asymptotic algorithms into exact, finite-state algorithms for computing frequency-based functions (resp. multiset-based functions). Our computability results are summarized in Table 2.

## REFERENCES

- [1] Dana Angluin. 1980. Local and Global Properties in Networks of Processors (Extended Abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28–30, 1980, Los Angeles, California, USA*, Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton (Eds.). ACM, 82–93. <https://doi.org/10.1145/800141.804655>
- [2] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. 2006. Computation in networks of passively mobile finite-state sensors. *Distributed Comput.* 18, 4 (2006), 235–253. <https://doi.org/10.1007/s00446-005-0138-3>
- [3] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. 2007. The computational power of population protocols. *Distributed Comput.* 20, 4 (2007), 279–304. <https://doi.org/10.1007/s00446-007-0040-2>
- [4] Florence Bénézit, Vincent D. Blondel, Patrick Thiran, John N. Tsitsiklis, and Martin Vetterli. 2010. Weighted Gossip: Distributed Averaging using non-doubly stochastic matrices. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13–18, 2010, Austin, Texas, USA, Proceedings*. IEEE, 1753–1757. <https://doi.org/10.1109/ISIT.2010.5513273>
- [5] Paolo Boldi, Shella Shammah, Sebastiano Vigna, Bruno Codenotti, Peter Gemmel, and Janos Simon. 1996. Symmetry Breaking in Anonymous Networks: Characterizations. In *Fourth Israel Symposium on Theory of Computing and Systems, ISTCS 1996, Jerusalem, Israel, June 10–12, 1996, Proceedings*. IEEE Computer Society, 16–26.
- [6] Paolo Boldi and Sebastiano Vigna. 1997. Computing Vector Functions on Anonymous Networks. In *SIROCCO'97, 4th International Colloquium on Structural Information & Communication Complexity, Monte Verita, Ascona, Switzerland, July 24–26, 1997*, Danny Krizanc and Peter Widmayer (Eds.). Carleton Scientific, 201–214.
- [7] Paolo Boldi and Sebastiano Vigna. 2002. Fibrations of graphs. *Discret. Math.* 243, 1–3 (2002), 21–66. [https://doi.org/10.1016/S0012-365X\(00\)00455-6](https://doi.org/10.1016/S0012-365X(00)00455-6)
- [8] Paolo Boldi and Sebastiano Vigna. 2002. Universal Dynamic Synchronous Self-Stabilization. *Distributed Computing* 15, 3 (2002), 137–153. <https://doi.org/10.1007/s004460100062>
- [9] Paolo Boldi and Sebastiano Vigna. 2002. Universal dynamic synchronous self-stabilization. *Distributed Comput.* 15, 3 (2002), 137–153. <https://doi.org/10.1007/s004460100062>
- [10] Bernadette Charron-Bost and Patrick Lambein-Monette. 2018. Randomization and Quantization for Average Consensus. In *2018 IEEE Conference on Decision and Control (CDC)* (Miami Beach, FL). IEEE, 3716–3721. <https://doi.org/10.1109/CDC.2018.8619817>
- [11] Bernadette Charron-Bost and Patrick Lambein-Monette. 2022. Computing Outside the Box: Average Consensus over Dynamic Networks. In *1st Symposium on Algorithmic Foundations of Dynamic Networks, SAND 2022, March 28–30, 2022, Virtual Conference (LIPIcs, Vol. 221)*, James Aspnes and Othon Michail (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 10:1–10:16. <https://doi.org/10.4230/LIPIcs.SAND.2022.10>
- [12] Bernadette Charron-Bost and Patrick Lambein-Monette. 2023. Know Your Audience. <https://doi.org/10.48550/arXiv.2311.17728> arXiv:2311.17728 [cs]
- [13] George Cybenko. 1989. Dynamic Load Balancing for Distributed Memory Multiprocessors. *J. Parallel and Distrib. Comput.* 7, 2 (1989), 279–301. [https://doi.org/10.1016/0743-7315\(89\)90021-X](https://doi.org/10.1016/0743-7315(89)90021-X)
- [14] Michael Dinitz, Jeremy Fineman, Seth Gilbert, and Calvin Newport. 2017. Load Balancing with Bounded Convergence in Dynamic Networks. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications* (Atlanta, GA, USA). IEEE, 1–9. <https://doi.org/10.1109/INFOCOM.2017.8057000>
- [15] Leonhard Euler. 1755. *Institutiones calculi differentialis*.
- [16] Faith E. Fich and Eric Ruppert. 2003. Hundreds of impossibility results for distributed computing. *Distributed Comput.* 16, 2–3 (2003), 121–163. <https://doi.org/10.1007/s00446-003-0091-y>
- [17] Balazs Gerencsér and Julien M. Hendrickx. 2019. Push-Sum With Transmission Failures. *IEEE Trans. Automat. Control* 64, 3 (2019), 1019–1033. <https://doi.org/10.1109/TAC.2018.2836861>
- [18] Julien M. Hendrickx, Alexander Olshevsky, and John N. Tsitsiklis. 2011. Distributed Anonymous Discrete Function Computation. *IEEE Trans. Autom. Control* 56, 10 (2011), 2276–2289. <https://doi.org/10.1109/TAC.2011.2163874>
- [19] David Kempe, Alin Dobra, and Johannes Gehrke. 2003. Gossip-Based Computation of Aggregate Information. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11–14 October 2003, Cambridge, MA, USA, Proceedings*. IEEE Computer Society, 482–491. <https://doi.org/10.1109/SFCS.2003.1238221>
- [20] Fabian Kuhn, Nancy Lynch, and Rotem Oshman. 2010. Distributed Computation in Dynamic Networks. In *Proceedings of the 42nd ACM Symposium on Theory of Computing - STOC '10* (Cambridge, Massachusetts, USA). ACM Press, 513. <https://doi.org/10.1145/1806689.1806760>
- [21] Patrick Lambein-Monette. 2020. *Average consensus in anonymous dynamic networks: An algorithmic approach. (Consensus de moyenne dans les réseaux dynamiques anonymes: Une approche algorithmique)*. Ph.D. Dissertation. Polytechnic Institute of Paris, France. <https://tel.archives-ouvertes.fr/tel-03168053>
- [22] Giuseppe Antonio Di Luna and Giovanni Viglietta. 2022. Computing in Anonymous Dynamic Networks Is Linear. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 1122–1133. <https://doi.org/10.1109/FOCS54457.2022.00108>
- [23] Giuseppe Antonio Di Luna and Giovanni Viglietta. 2023. Optimal Computation in Leaderless and Multi-Leader Disconnected Anonymous Dynamic Networks. In *37th International Symposium on Distributed Computing, DISC 2023, October 10–12, 2023, L'Aquila, Italy (LIPIcs, Vol. 281)*, Rotem Oshman (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 18:1–18:20. <https://doi.org/10.4230/LIPIcs.DISC.2023.18>
- [24] Shlomo Moran and Manfred K. Warmuth. 1993. Gap Theorems for Distributed Computation. *SIAM J. Comput.* 22, 2 (1993), 379–394. <https://doi.org/10.1137/0222028>
- [25] Angelia Nedic and Ji Liu. 2017–02. On Convergence Rate of Weighted-Averaging Dynamics for Consensus Problems. 62, 2 (2017–02), 766–781. <https://doi.org/10.1109/TAC.2016.2572004>
- [26] Angelia Nedic, Alex Olshevsky, and Michael G. Rabbat. 2018. Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization. *Proc. IEEE* 106, 5 (2018), 953–976. <https://doi.org/10.1109/JPROC.2018.2817461>
- [27] Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. 2009. On Distributed Averaging Algorithms and Quantization Effects. *IEEE Trans. Automat. Control* 54, 11 (2009), 2506–2517. <https://doi.org/10.1109/TAC.2009.2031203>
- [28] Alex Olshevsky. 2017. Linear Time Average Consensus and Distributed Optimization on Fixed Graphs. *SIAM J. Control. Optim.* 55, 6 (2017), 3990–4014. <https://doi.org/10.1137/16M1076629>
- [29] John N. Tsitsiklis. 1984. *Problems in Decentralized Decision Making and Computation*. Ph.D. Dissertation. Massachusetts Institute of Technology. <https://www.mit.edu/~jnt/Papers/PhD-84-jnt.pdf>
- [30] John N. Tsitsiklis, Dimitri P. Bertsekas, and Michael Athans. 1986. Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms. *IEEE Trans. Automat. Control* 31, 9 (1986), 803–812. <https://doi.org/10.1109/TAC.1986.1104412>
- [31] Lin Xiao and Stephen Boyd. 2004. Fast Linear Iterations for Distributed Averaging. *Systems & Control Letters* 53, 1 (2004), 65–78. <https://doi.org/10.1016/j.sysconle.2004.02.022>
- [32] Lin Xiao, Stephen Boyd, and Sanjay Lall. 2005. A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. (Los Angeles, CA, USA). IEEE, 63–70. <https://doi.org/10.1109/IPSN.2005.1440896>
- [33] Masafumi Yamashita and Tiko Kameda. 1989. Electing a Leader when Processor Identity Numbers are not Distinct (Extended Abstract). In *Distributed Algorithms, 3rd International Workshop, Nice, France, September 26–28, 1989, Proceedings (Lecture Notes in Computer Science, Vol. 392)*, Jean-Claude Bermond and Michel Raynal (Eds.). Springer, 303–314. [https://doi.org/10.1007/3-540-51687-5\\_52](https://doi.org/10.1007/3-540-51687-5_52)
- [34] Masafumi Yamashita and Tiko Kameda. 1996. Computing Functions on Asynchronous Anonymous Networks. *Math. Syst. Theory* 29, 4 (1996), 331–356. <https://doi.org/10.1007/bf01192691>
- [35] Masafumi Yamashita and Tsunehiko Kameda. 1996. Computing on Anonymous Networks: Part I- Characterizing the Solvable Cases. *IEEE Trans. Parallel Distributed Syst.* 7, 1 (1996), 69–89. <https://doi.org/10.1109/71.481599>