

## MIT Open Access Articles

*Averaging algorithms and distributed  
anonymous discrete function computation*

The MIT Faculty has made this article openly available. **Please share**  
how this access benefits you. Your story matters.

**Citation:** Tsitsiklis, John N. Averaging Algorithms and Distributed Anonymous Discrete Function Computation. In 2011 19th Mediterranean Conference on Control & Automation (MED), 301-302. Institute of Electrical and Electronics Engineers, 2011.

**As Published:** <http://dx.doi.org/10.1109/MED.2011.5983086>

**Publisher:** Institute of Electrical and Electronics Engineers (IEEE)

**Persistent URL:** <http://hdl.handle.net/1721.1/79646>

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

**Terms of use:** Creative Commons Attribution-Noncommercial-Share Alike 3.0



# Averaging algorithms and distributed anonymous discrete function computation

John N. Tsitsiklis

**Abstract**—In distributed consensus and averaging algorithms, processors exchange and update certain values (“estimates” or “opinions”) by forming a local average with the values of their neighbors. Under suitable conditions, such algorithms converge to consensus (every processor ends up holding the same value) or even average-consensus (consensus is achieved on the average of the initial values held by the processors). We first review some convergence rate results. We then abstract the problem by introducing a model of deterministic distributed function computation by a network of identical and anonymous nodes, with limited communication and storage capabilities, and where each node only knows its neighbors, not the entire graph. Our goal is to characterize the class of functions that can be computed within this model. In our main result, we provide a necessary condition for computability which we show to be nearly sufficient, in the sense that every function that satisfies this condition can at least be approximated. The problem of computing (suitably rounded) averages in a distributed manner plays a central role in our development, and we provide an algorithm that solves it in time that grows at most quadratically with the size of the network. (Joint work with J. Hendrickx and A. Olshevsky.)

## I. MOTIVATION, RESULTS, AND OPEN PROBLEMS

Consider a network of  $n$  nodes who can exchange messages along the arcs of a fixed, connected, undirected graph  $G$ . In the averaging problem, each node  $i$  starts with an initial value  $x_i$ , and the nodes execute a distributed algorithm so that they all end up with the value of the average  $\bar{x} = (\sum_{i=1}^n x_i)/n$ . We are interested in algorithms that perform this task with a reasonably small convergence time.

One class of algorithms involves the iteration  $x := Ax$ , where  $A$  is a stochastic matrix with  $a_{ij} \neq 0$  only if  $(i, j)$  is an arc of the graph  $G$ . There is a variety of iterative algorithms of this type and, in the absence of special assumptions on the structure of the graph  $G$ , the best possible convergence time (namely, the time needed to compute the average within some fixed accuracy) is of order  $O(n^2)$  [1]. One possibility here is to form a spanning tree and let  $a_{ij} \neq 0$  only along the arcs  $(i, j)$  of the spanning tree. Even though a spanning tree can be constructed in a distributed manner, such an algorithm is generally not considered to be truly distributed. A more genuinely distributed method, whose convergence time is  $O(n^2)$  [4], is motivated by interpreting  $x_i$  as a “load” and by taking a balancing approach: nodes attempt to locally equalize their values by transferring load to their less loaded neighbors. Still, such an algorithm involves “large” messages: a message is in general a real number (the amount

of load exchanged) or, in the special case where the initial values  $x_i$  are integer, a message is a rational number with arbitrarily large denominator, hence requiring an unbounded number of bits. If such “large” messages are allowed, why consider iterative algorithms in the first place, given that simple “flooding” algorithms can centralize all available information and compute averages in time proportional to the graph diameter?

The above discussion motivates the following question: what is the best possible convergence time if we restrict to algorithms that are genuinely distributed and involve messages with a limited number of bits? To make progress in answering this question, we need to first define a model of computation that, in particular, captures the notion of a “genuinely distributed” algorithm.

We introduce a model, fully described in [2], in which the nodes are anonymous (nodes do not have identity numbers), are not allowed to randomize, do not know the structure of the graph  $G$  (they only know their neighbors), and they all execute the same local algorithm. Furthermore, nodes are limited to transmitting a constant number of bits on each link and their memory is limited to a constant multiple of their degree. The key reason for prohibiting identity numbers or randomization is that otherwise the nodes could elect a leader and essentially revert to a centralized algorithm; see [2] for a more detailed discussion of the reasons for the model choices.

With such a restricted model, one may wonder whether any interesting function of the initial values  $x_i$  can be computed. It turns out [2] that a suitably discretized version of the average (this is the interval-averaging problem, studied in [3]) can be computed under this model, and that the convergence time matches the  $O(n^2)$  bound attained under much more permissive models of computation. It also turns out that the class of functions that can be computed under this model is limited to essentially combinations of various averaging functions; see [2] for precise statements.

The results of [2] establish the central role of average computation under the proposed model. They also raise two important directions for further research:

- (a) Is it possible to improve upon the  $O(n^2)$  bound? Or can one establish a  $\Omega(n^2)$  lower bound?
- (b) Is it possible to extend the positive results of [2] and provide  $O(n^2)$  algorithms (or just polynomial time algorithms) under a variation of the model in which the network structure (the graph  $G$ ) is time-varying (subject to some connectivity assumptions over sufficiently long time intervals)? Indeed, an  $O(n^2)$  algorithm for a time-

Research partially supported by the National Science Foundation under grant ECCS-0701623.

Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; jnt@mit.edu.

varying environment is provided in [4] under a much more permissive model of computation that allows for real-valued messages and unbounded memory. Is the same possible with algorithms with restricted message and memory size (e.g., proportional to a node's degree), and in which the nodes are anonymous and cannot randomize?

#### REFERENCES

- [1] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.
- [2] J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Distributed anonymous discrete function computation and averaging," submitted, 2010; available at: <http://web.mit.edu/jnt/www/Papers/P-10-anonymous-sub.pdf>
- [3] F. Bénézit, P. Thiran, and M. Vetterli, "Interval consensus: from quantized gossip to voting," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3661–3664, April 2009.
- [4] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, November 2009.