# Decentralized Estimation in Open Multi-Agent Systems

Sebastien Colla, Charles Monnoyer de Galland and Julien M. Hendrickx

***Abstract*— We present a new decentralized estimation algorithm that operates in open multi-agent systems which are subject to arrivals and departures of agents. The algorithm averages the data of all the agents that are or have been in the system in order to estimate, in a decentralized manner, the mean of the data distribution. The main challenge of this problem is to correctly incorporate the information from new agents without (i) forgetting the information of the agents having left the system and (ii) being impacted by the perturbations inherent to an arrival or a departure. This work establishes empirical performance limitations for this problem in Open Multi-Agent Systems and proposes a new algorithm (SPS) that approaches the performance limitations (within 1%) in most cases.**

## I. Introduction

We consider open multi-agent systems, i.e. systems where the agents may leave or join the system freely during its operation. This open character may be inherent to the situation, e.g. when the agents are moving or when there is a kind of birth process in the system. It is also relevant in large systems where the arrival and departure of agents may become unavoidable. Finally, one may desire openness of the system to ensure robustness against possible failures. Theoretical analyses of open multi-agent systems are relatively recent. In [1], [2], the authors have analyzed the behavior of the gossip averaging algorithm in open system; in [3] the performance limitations for the consensus problem on the average of the agents currently present in the open systems; in [4] the maximum consensus and in [5], [6] the dynamic consensus on the average and the median of time-varying signals, in open systems.

In this work, we focus on a new decentralized estimation problem for open multi-agent systems. We suppose that every agent holds a noisy measurement of a mean $\mu$, and the goal of the system is to estimate $\mu$, in a decentralized manner.
In this context, the open character of the system will at the same time be beneficial, because each new agent brings a new measurement allowing for potentially more accurate estimates, and challenging, because each arrival and departure creates a perturbation, and the information from former agents should not be forgotten by the system.
In the context of sensor networks, this estimation problem corresponds to a multi-agent system where each entering agent gets a noisy measurement of a value, e.g. a state of the environment, that it cannot access once it is inside the system, and the agents would like to estimate accurately this external value. In the context of opinion dynamics, our problem corresponds to estimating the average public opinion in a population where the open system is composed of random people from that population, who can enter and leave.

This new estimation problem can be posed when the system is subject to arrivals and departures of agents. We focus here on the simpler situation where arrivals and departures are synchronized, such that the system size remains constant at any time and where the current agents perform all-to-all pairwise communications.

## II. Problem statement and assumptions

*a) The system:* for simplicity, we consider an open multi-agent system only subject to replacements of agents during its operation. This means that the departure of an agent is immediately followed by the arrival of a new one. The agents present in the system at time $t$ are described by the set of indices $\mathcal{N}(t)$. The size of the system, denoted $n = |\mathcal{N}(t)|$, is constant due to the replacement assumption.

*b) The agents:* agents are assumed identical. They have a bounded memory, and are capable of local computation and pairwise communications. They all execute the same algorithm. Each agent $i \in \mathcal{N}(t)$ holds an initial value $z_i$ (also called initial measurement) independently drawn from an identical distribution of constant mean $\mu$ and variance $\sigma^2$; an estimated solution $y_i(t)$; and potentially some additional internal variables.

*c) The goal:* the goal of each agent is to estimate *as accurately as possible* the distribution mean $\mu$:

$$y_i(t) \to \mu, \qquad \text{for all } i \in \mathcal{N}(t).$$

*d) The quality metric:* the quality metric used to quantify the accuracy and the success of an algorithm in achieving this goal is the mean squared error criterion (MSE) computed between the agents estimates $y_i(t)$ and the value of interest $\mu$. It is defined as:

$$\text{MSE}(t) = \frac{1}{n} \sum_{i \in \mathcal{N}(t)} (y_i(t) - \mu)^2.$$

The measurements $z_i$ of the agents are random variables and the sequence of events is not fully deterministic, as explained below. Therefore, this metric is also a random variable and we will generally evaluate its expectation: $\mathbb{E}[\text{MSE}(t)]$.

*e) The communications:* they occur in the system according to a Poisson clock of global rate $n\lambda_c$. Whenever a communication occurs, two randomly uniformly and independently selected agents $i, j \in \mathcal{N}(t)$ (possibly twice the same agent) exchange information with each other. In a nutshell, communications are said to be asynchronous, pairwise, symmetric and random.

S. Colla, C. Monnoyer de Galland and J. M. Hendrickx are with the ICTEAM institute, UCLouvain, Louvain-la-Neuve, Belgium. S. Colla and C. Monnoyer de Galland are supported by the French Community of Belgium through a FRIA fellowship (F.R.S.-FNRS). J. M. Hendrickx is supported by the "RevealFlight" Concerted Research Action (ARC) of the Federation Wallonie-Bruxelles and by the Incentive Grant for Scientific Research (MIS) "Learning from Pairwise Comparisons" of the F.R.S.-FNRS. Email addresses: { sebastien.colla, charles.monnoyer, julien.hendrickx } @uclouvain.be

*f) The replacements:* they occur in the system according to a Poisson clock of global rate $n\lambda_r$. A replacement consists of the departure of one randomly and uniformly chosen agent from the system, instantaneously followed by the arrival of a new agent into the system. The value of $\lambda_r$ is therefore the individual replacement rate. Also, a former agent cannot come back into the system.

## III. Optimal algorithm - empirical lower bound

Since the measurements $z$ are i.i.d. drawn from a distribution with mean $\mu$, the best way to estimate $\mu$ at time $t$ through a linear unbiased estimator is to compute the average of the initial measurements $z_k$ of all agents $k$ that have ever been in the system until time $t$. Let us call this average the external average of the system.

$$\bar{z}_{\text{ext}}(t) = \frac{1}{|\cup_{s\leq t}\mathcal{N}(s)|}\sum_{k\in\cup_{s\leq t}\mathcal{N}(s)} z_k \tag{1}$$

The expected performance achieved by this external average provides therefore a lower bound on the expected performance of any algorithm that generates linear unbiased estimators of $\mu$. This lower bound is not tight since every agent cannot always be able to compute this external average. Indeed, an agent cannot always know all the other current and previous agents. The acquaintance of an agent depends on the sequence of events (communications and replacements) that occur in the system. Intuitively, the best thing an agent $i$ can do for estimating $\mu$ is to average the initial measurements $z_k$ of all agents $k$ it knows, i.e. the measurements of all the agents it has been in contact (directly or indirectly) since its arrival into the system.

Implementing such an algorithm within our framework is challenging, and could be impossible. A direct implementation based on the definition would indeed require the agents to use unique identifiers and a growing memory, which contradicts our assumptions. Nevertheless, the results of this algorithm will serve as a lower performance bound. Indeed, it can be proven (see [7], Chap. 2) that it provides the smallest expected mean squared error among all the algorithms within our framework that generate linear and unbiased estimates of $\mu$. Therefore, we will refer to this algorithm as the optimal algorithm. Analytical performance lower bound expressions were obtained in [7], Chap. 2, by bounding the expected performance of this optimal algorithm. These resulting theoretical lower bounds are very conservative. Hence, we will use here an empirical lower bound, obtained by averaging the performance of various executions of the optimal algorithm on the specific system configuration we are considering.

A first attempt to solve our problem is to directly apply the well-known gossip averaging, which would recover the best estimate of $\mu$ in a closed system. Indeed, in a closed system, the set of agents is fixed and the gossip converges exponentially to the average of their initial measurements [8]. But in open systems, Fig. 2 shows that the expected performance of the gossip algorithm is far from the empirical lower bound in many situations, especially when the system is small. The analysis pursued in [7], Chap. 3 shows that these poor performances are explained by the algorithm giving too much weight to new arriving agents and tending to forget information from former agents. In the asymptotic situation where the replacements are so infrequent that the system has time to converge between each of them, i.e. when $\lambda_r \to 0$, we can show that the expected weight of each former agent in the system estimates decreases with a factor $\frac{n-1}{n}$ at each replacement. This forgetting rate is thus more important for small systems, which explains the observations in Fig. 2a.

Hence, our idea is to decrease the weights of the arriving agents in a way that compensates for this forgetting rate. It makes sense to assign less and less weights to the new agents for which the initial estimate always contains the same amount of information, coming from their initial measurement of $\mu$, while the estimates of the agents in the system gather progressively more and more information about $\mu$.

## IV. Symmetric Push-Sum (SPS)

In order to assign a different weight to each agent and to control the importance of the new agents entering the system, we have adapted the push-sum algorithm [9], [10] to our problem framework. In this push-sum algorithm, in addition to its measurement $z_i$, each agent $i \in \mathcal{N}(t)$ also holds two internal time-varying variables $x_i(t)$ and $w_i(t)$. The value $x_i(t)$ represents the aggregation of information held by $i$ and the time-varying weight $w_i(t)$ represents the volume of information that $i$ holds within its variable $x_i$ so that, at any time, the estimate of $\mu$ of $i$ is given by the ratio between these two variables $y_i(t) = \frac{x_i(t)}{w_i(t)}$.

In our case, the purpose of the weights is different from their initial design motivation in the classical push-sum. Initially, the push-sum algorithm introduced weights for handling asymmetric communications for the averaging consensus problem in closed multi-agent systems, see e.g. [9] or [10]. The weights were introduced for balancing the volume of information held by each agent in the system, which is not preserved when the communications are asymmetric. Our variant is said to be symmetric since we consider symmetric communications and hence it uses the weights for another purpose: controlling the importance of new agents in the system.

At each symmetric communication between two agents $i, j \in \mathcal{N}(t)$, both of their variables $x$ and $w$ are updated with their average:

$$x_i^+ = x_j^+ = \frac{x_i + x_j}{2}, \qquad w_i^+ = w_j^+ = \frac{w_i + w_j}{2}.$$

This means that the estimates of the agents are updated with the weighted average of their estimates:

$$y_i^+ = y_j^+ = \frac{x_i + x_j}{w_i + w_j} = \frac{w_i y_i + w_j y_j}{w_i + w_j}.$$

The initial estimate of any agent $i$ is its own measurement $z_i$ of $\mu$. Therefore, its initial weight represents the importance that its own measurement will have into the system estimates, until a replacement occurs. The key aspect of this variation of the push-sum algorithm is therefore the choice of the initial weights of the new agents. They should decrease at a rate of $\frac{n-1}{n}$ to ensure that all the agents, even the past ones, have the same importance in the system estimates.

Let us summarize our algorithm properly. For simplicity, let us assume an indexing of agents such that the initial set of agents in the system (at time $t = 0$) is $\mathcal{N}(0) = \{1, \ldots, n\}$ and such that the r$^{th}$ agent to arrive into the system during its operation has index $n + r$.

**Symmetric push-sum algorithm (SPS):**

- Initialization of each agent (before it enters the system):

$$x_i(0) = w_i(0) z_i \qquad \text{for all } i$$

$$w_i(0) = \begin{cases} 1 & \text{for } i = 1, \ldots, n \\ \left(\frac{n-1}{n}\right)^{i-n} & \text{for } i = n+1, \ldots \end{cases} \tag{2}$$

- Interaction between agents $i, j \in \mathcal{N}(t)$:

$$x_i^+ = x_j^+ = \frac{x_i + x_j}{2}, \qquad w_i^+ = w_j^+ = \frac{w_i + w_j}{2}.$$

- Estimate of agent $i \in \mathcal{N}(t)$: $y_i(t) = \frac{x_i(t)}{w_i(t)}$.

This algorithm requires the agents to know the system size $n$ and their own arrival order $i$ in order to set up their initial weight according to SPS (2). The former can be estimated by counting process as in [11] [12], while schemes to approximate the latter through max consensus [4] are presented in [7], Chap. 4.

One can show (see [7], Chap. 4) that in the asymptotic situation where the replacements are so infrequent that the system has time to converge between each of them, i.e. when $\lambda_r \to 0$, SPS algorithm allows converging to the exact external average $\bar{z}_{\text{ext}}$ (1) between each replacement. Determining the expected performance of the algorithm in non-asymptotic situations remains an open question, but these performances are explored numerically in the following section.

## V. Numerical Experiments and Extension

Fig. 1 shows the time evolution of the expected mean square error $\mathbb{E}[\text{MSE}]$ for a representative system configuration for the SPS, the gossip, the optimal algorithm and the difference tracking extension of SPS, explained at the end of the section. For each algorithm, the curve appears to have two distinct regimes: a transient regime, where $\mathbb{E}[\text{MSE}]$ decreases, and a steady-state one where it is constant. The duration of the transient regime is identical for both the optimal and the gossip algorithm but the gossip decreases less during that time and it thus stabilizes above the optimal algorithm. The steady-state expected performance of the optimal algorithm is still different from zero since the new agents cannot be directly aware of all the information in the system. It can also happen that all the agents are replaced without having the opportunity to transmit information to new incoming agents.
The transient state of the SPS algorithm is longer but it leads to a steady-state performance that is very close to the empirical lower bound.

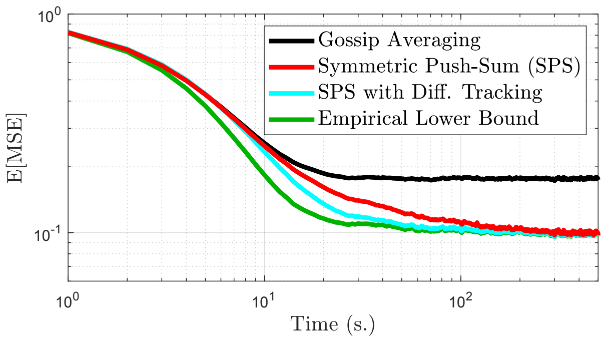



Fig. 1: Time evolution of the expected MSE of the symmetric push-sum algorithm in comparison with the gossip averaging algorithm, the SPS with difference tracking and the empirical lower bound provided by the optimal algorithm (see section III); for a specific open multi-agent system with $n = 50$, $\lambda_c = 0.2$ and $\lambda_r = 0.04$. Expectations have been computed empirically with 5000 realization simulations, by considering a normal distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$.

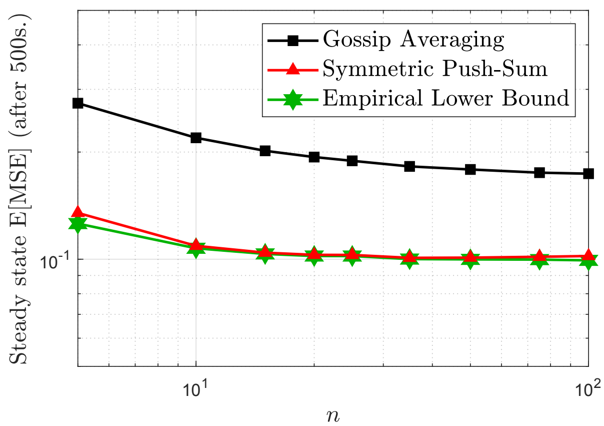
We now analyze the evolution of the steady-state performance value of each algorithm for a variety of different system configurations. To explore the different possible system configurations, we fix the global communication rate in the system: $n\lambda_c$ and we only vary the system size $n$ and the individual replacement rate $\lambda_r$. We choose to set $n\lambda_c = 10$, as it is the case in Fig. 1. With this parameter scaling, the steady-state is reached before 500 seconds in most system configurations. The transient state may still be longer than 500 seconds when the system size is very large and the replacements are infrequent.

Fig. 2 shows the expected mean squared error achieved by the symmetric push-sum algorithm, the gossip, and the optimal algorithm after 500 seconds for different system configurations. The figure only shows system configurations that have (almost) reached their steady-state before 500 seconds. Fig. 2a therefore shows the evolution of the steady-state performance of the algorithm with the system size $n$ and Fig. 2b shows the evolution with the rate ratio $\frac{\lambda_c}{\lambda_r}$. This rate ratio represents the expected number of communications that occur in the system between two replacements.

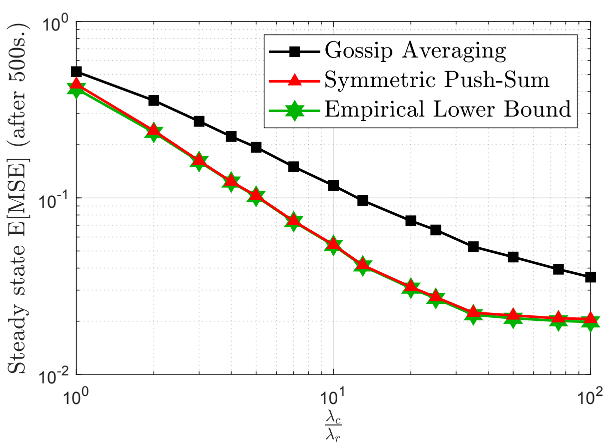
We observe that the steady-state value of the expected MSE for the SPS algorithm (in red) is always better than the one of the gossip averaging algorithm (in black) and approaches closely the empirical lower bound (in green) in most system configurations.

A possible improvement for this SPS algorithm seems therefore related to the speed at which it approaches the lower bound. It can be relatively slow for large systems, such as in Fig. 1. This can be explained by the perturbation due to the arrival of a new agent, which is proportional to its initial weight. This latter decreases with a rate of $\frac{n-1}{n}$, which is slower when $n$ is large.

To solve this issue related to the perturbation introduced by new agents, we can consider an extension for this SPS algorithm: the difference tracking. When entering the system, the new agent pretends to have the same values (for $x$ and $w$) as the first agent it meets. But the new agent keeps track of the differences between its true initial values and its pretended values to add them progressively into the system. This allows the information from new agents to be progressively taken into account without disturbing too much the estimates of other agents. Details are presented in [7], Chap. 6. Applying this extension to SPS allows obtaining results that approach more quickly and as closely (within 1%) the performance limitations, as shown in Fig. 1.




(a) Evolution with system size $n$, for $\frac{\lambda_c}{\lambda_r} = 5$.




(b) Evolution with rate ratio $\frac{\lambda_c}{\lambda_r}$, for $n = 20$.

Fig. 2: Steady state expected mean squared error (after 500 seconds) of the symmetric push-sum algorithm in comparison with the one of the gossip averaging algorithm and the optimal algorithm (empirical lower bound); for a variety of different systems (with constant global communication rate $n\lambda_c = 10$). Expectations have been computed empirically with 500 realization simulations, by considering a normal distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$.

## VI. Conclusion

We considered the problem of estimating the mean of a distribution from which measurements are distributed over an open multi-agent system, subject to replacements of agents. We showed that the classical gossip led to relatively poor performance, which could be explained by the disproportionate importance it gives to recently arrived agents. Our new push-sum based algorithm SPS addresses this problem by controlling the weight of the incoming agents: they are set in a decreasing manner, with a rate of $\frac{n-1}{n}$. This allows the SPS algorithm to have a better expected performance than the gossip averaging and to approaches the empirical lower bound, defined by the optimal algorithm. An important remaining challenge for this work is to formally prove these observations, by deducing a theoretical description of the expected performance of SPS.

## References


[1] J. M. Hendrickx and S. Martin, “Open multi-agent systems: Gossiping with random arrivals and departures,” September 2017.

[2] C. Monnoyer de Galland, S. Martin, and J. M. Hendrickx, “Open multi-agent systems with variable size: the case of gossiping,” 2020.

[3] C. Monnoyer de Galland and J. M. Hendrickx, “Fundamental performance limitations for average consensus in open multi-agent systems,” 2020.

[4] M. Abdelrahim, J. M. Hendrickx, and W. P. M. H. Heemels, “Max-consensus in open multi-agent systems with gossip interactions,” in *IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 4753–4758.

[5] M. Franceschelli and P. Frasca, “Stability of open multi-agent systems and applications to dynamic consensus,” *IEEE Transactions on Automatic Control*, 2020.

[6] Z. A. Z. Sanai Dashti, C. Seatzu, and M. Franceschelli, “Dynamic consensus on the median value in open multi-agent systems,” in *IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 3691–3697.

[7] S. Colla, “Data fusion in open multi-agent systems for decentralized estimation,” Master’s thesis, UCLouvain - Ecole Polytechnique de Louvain, 2020, supervised by J. Hendrickx, http://hdl.handle.net/2078.1/thesis:25154.

[8] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, p. 2508–2530, Jun. 2006.

[9] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information,” in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, 2003, pp. 482–491.

[10] B. Gerencsér and J. M. Hendrickx, “Push-sum with transmission failures,” *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1019–1033, 2019.

[11] I. Shames, T. Charalambous, C. N. Hadjicostis, and M. Johansson, “Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs,” in *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012, pp. 1885–1892.

[12] R. Lucchese, D. Varagnolo, J. Delvenne, and J. Hendrickx, “Network cardinality estimation using max consensus: The case of bernoulli trials,” in *54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 895–901.