MULTI-SENSOR DATA FUSION BY AVERAGE CONSENSUS ALGORITHM WITH FULLY-DISTRIBUTED STOPPING CRITERION: COMPARATIVE STUDY OF WEIGHT DESIGNS

Martin KENYERES[1], Jozef KENYERES[2]

The average consensus tends to be used as a complementary mechanism for multi-sensor data fusion in modern applications. This paper addresses a comparative study of its weight designs with a fully-distributed stopping criterion implementable into the wireless sensor networks. Two parameters of the examined stopping criterion, namely, accuracy and counter threshold, are changed to analyze which weight design achieves the best performance in terms of the precision (quantified by the MSE) and the convergence rate. It is shown how the values of both parameters affect the mentioned aspects as well as mutual comparison of the examined weight designs is provided.

Keywords: Distributed computing, wireless sensor networks, average consensus, stopping criterion

1. Introduction

Wireless sensor networks (WSNs) have attracted the attention of both the academy and the industry sector in the last years. WSNs are often formed by hundreds of geographically distributed entities (referred to as sensor nodes) for cooperative monitoring the adjacent environment and are assumed to work autonomously for long-lasting periods [1]. The sensor nodes, often deployed in large-scale areas, consist of hardware components such as a wireless transceiver, a sensor unit, the central processor, an energy source etc., which allows them to sense a particular environmental quantity (known as sensor reading), process the measured data, and mutually communicate in order to fulfill a specific functionality [2]. WSNs find an application in many areas such as agriculture, environmental monitoring, military surveillance, natural disaster detection, inventory tracking, pollution monitoring, medical systems, robotic exploration, acoustic detection, health care etc. ([3-4]). In many of these applications, the operation of WSNs is affected by negative factors (e.g. an electromagnetic noise, defectiveness of the nodes, radiation temperature, output correlations etc.) that can significantly decrease the precision of the sensor

[1] Post-doctoral research scientist, Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia, e-mail: martin.kenyeres@savba.sk

[2] Software developer, Sipwise GmbH, Austria, e-mail: jkenyeres@sipwise.com

readings [5]. Thus, the modern applications based on WSNs are often equipped with mechanisms for multi-sensor data fusion, whereby the negative impacts on the quality of these applications are minimized [5]. Consensus-based algorithms (primarily those for distributed averaging) find wide usage in WSNs as a technique for multi-sensor data fusion and have become an attractive research field in the signal processing during the last decades [6]. Their goal is to make the states at all the sensor nodes identical via a neighbor-to-neighbor communication and local updates [7]. Consensus-based algorithms find a wide usage not only in WSNs but also in other areas such as the blockchains, cloud computing, the Internet of Things etc. (they are assumed to be applied also in the integration of WSNs, IoT, and cloud computing) ([8-11]).

In this paper, the average consensus algorithm (AC), a distributed linear iterative algorithm asymptotically converging to the arithmetic average of all the inner states, is addressed [12]. The algorithm is multi-functional, i.e. it can estimate not only the average but also fulfills other functionalities [12]. Each sensor node is typically aware of its neighbors and has only limited information about the whole network [6]. At each iteration, it adapts its inner state according to the inner states collected from the adjacent area and the state from the previous iteration. As mentioned earlier, AC asymptotically convergences to the average and therefore, many real-life applications require the implementation of a stopping criterion, which ensures the consensus achievement in a finite time, however with a limited precision [14]. In this paper, our attention is focused on the stopping criterion from [13] and frequently discussed also in other papers ([3-4], [14]). In this paper, we deal with four frequently quoted weight designs (namely, the Maximum Degree, the Metropolis-Hastings, the Local Degree, and the Best Constant weights) and change the parameters of the examined stopping criterion (*accuracy* and *counter threshold*) to verify which design achieves the best performance using two metrics: the mean square error (MSE) [dB] and the number of the required iterations for the consensus.

In the next section of the paper, we provide a model of AC over WSNs, its basic properties, the convergence conditions etc. and introduce the implemented stopping criterion. Section 3 deals with the analyzed weight design of AC and their mathematical definitions. Section 4 is concerned with the experimentally obtained results and a discussion about them. Section Future research introduces our future plans and insight into the application of consensus-based algorithms in WSNs, IoT, and Cloud Computing and their integration.

2. Problem formulation

WSNs can be modeled as finite indirect graphs determined by two sets $G = (\mathbf{V}, \mathbf{E})$ [12]. The vertex/node set $\mathbf{V}$ gathers all the vertices, representing the sensor nodes in the network (the sensor nodes are identified by the unique identity number **V** = {v_1, v_2, …. v_N}, where N = |**V**| is the size of the network). The edge/link set **E** ⊂ **V**x**V** contains all the edges, whose existence indicates the direct connection between two corresponding sensor nodes {v_i, v_j}, and so, these two sensor nodes are one another's neighbor. Subsequently, a set gathering all the neighbors of v_i can be defined as $\boldsymbol{\mathcal{N}}_i$ = {v_j : {v_i, v_j} ∈ **E**}. It is assumed that each sensor node is allocated the initial scalar value $x_i(0)$ ∈ **R** in the beginning of the algorithm (in our case, binary AC is assumed, i.e. the initial states take either one or zero [14]). All the inner states at each iteration are gathered in the column vector variant over the iterations **x**(k) ∈ $\mathbf{R}^{\boldsymbol{N}\mathbf{x}\mathbf{1}}$.

As mentioned earlier, AC is a set of rules ensuring that each sensor node acquires an approximate value (referred to as an estimate) of the estimated aggregate function (in our case the arithmetic average). It is achieved by iterative exchanges of the local inner states among adjacent sensor nodes and updating the inner state for the next iteration using the linear update scheme modeled as follows [15]:

$$x_i(k+1) = [W]_{ii}.x_i(k) + \sum_j [W]_{ij}.x_j(k) \text{ for } i = 1,2,...N\,, \tag{1}$$

From a global point of view, it is possible to reformulate (1) as [12]:

$$\mathbf{x}(k+1) = \mathbf{W} \times \mathbf{x}(k)\,, \tag{2}$$

Here, **W** is the weight matrix, whose elements are determined by the used weight design. It affects several aspects such as the convergence rate, the robustness, the initial configuration etc. [12]. The limits of its sparsity pattern can be expressed as **W** ∈ **S**, where **S** is defined as follows [15]:

$$\mathbf{S} = \{\mathbf{W} \in \mathbf{R}^{N \times N} : [W]_{ij} = 0 \quad \text{if } \{v_i, v_j\} \notin \mathbf{E} \wedge i \neq j\}\,, \tag{3}$$

The choice of the weight matrix is crucial for the convergence conservation, i.e. the vector of the inner states **x**(0) converges to the value of the arithmetic average, i.e. [16]

$$\lim_{k\to\infty} \mathbf{x}(k) = \lim_{k\to\infty} \mathbf{W}^k \times \mathbf{x}(0) = \bar{\mathbf{x}} = \frac{\mathbf{1}\times\mathbf{1}^{\mathrm{T}}}{N} \times \mathbf{x}(0)\,, \tag{4}$$

Here, **1** is a column vector whose all coefficients are equal to one (known as an all-ones vector) [12]. Equivalently to (4), one can write the followings [15]:

$$\lim_{k\to\infty} \mathbf{W}^k = \frac{\mathbf{1}\times\mathbf{1}^{\mathrm{T}}}{N}\,, \tag{5}$$

Then, we can define the asymptotic convergence factor and its associated convergence time, measures for performance evaluation, as follows [15]:

$$r_{asym}(\mathbf{W}) = \sup_{\mathbf{x}(0)\neq\bar{\mathbf{x}}} \lim_{k\to\infty} \left(\frac{\left\|\mathbf{x}(k)-\bar{\mathbf{x}}\right\|_2}{\left\|\mathbf{x}(0)-\bar{\mathbf{x}}\right\|_2} \right)^{1/k}, \ \tau_{asym}(\mathbf{W}) = \frac{1}{\log(1/r_{asym})}, \tag{6}$$

As discussed in [15], the convergence of the algorithm is ensured if and only if the limits in (4) exists. Its existence is conditioned by holding these necessary and sufficient conditions (see [17] for a proof):

$$\mathbf{W}\times\mathbf{1}=\mathbf{1},\ \mathbf{1}^{\mathrm{T}}\times\mathbf{W}=\mathbf{1}^{\mathrm{T}},\ \rho(\mathbf{W}-\frac{1}{N}.\mathbf{1}\times\mathbf{1}^{\mathrm{T}})<1, \tag{7}$$

Here, $\rho(\cdot)$ is the spectral radius of the corresponding matrix. The implementation of the stopping criterion proposed in [13] is assumed in this paper, which guarantees a finite time execution of AC. The examined stopping criterion is fully-distributed (thereby finds the application in WSNs) and requires that the sensor nodes store two constants: *accuracy* and *counter threshold* (both have to be pre-set before the beginning of AC). Each sensor node has its own *counter* that is incremented by one at the iterations when the difference between two subsequent inner states is smaller than the pre-set *accuracy*. When not, its value is set to zero. If *y* subsequent comparisons of the inner states are smaller than *accuracy*, AC is considered to be completed at the corresponding sensor node - it does not participate in AC any longer and does not update its inner state. The value *y* is determined by the pre-set *counter threshold*. The iteration when the last sensor node completes the algorithm is labeled as k_l – it represents the number of the iterations for the consensus. Eventually, a formalization of asymptotic AC (Algorithm 1) and AC with the stopping criterion (Algorithm 2) are provided.

Algorithm 1: Distributed Linear Average Consensus Algorithm

In the beginning, each sensor node v_i initiates its inner state with a scalar value (1 TRUE/0 FALSE in our case) labeled as $x_i(0)$.

At each iteration

1. Each sensor node $v_i \in \mathbf{V}$ sends a broadcast message containing its current inner state (i.e. $x_i(k)$ to $\forall v_j \in \boldsymbol{N}_i$)
2. Each node sensor $v_i \in \mathbf{V}$ receives the inner states from $\forall v_j \in \boldsymbol{N}_i$
3. Each sensor node $v_i \in \mathbf{V}$ multiplies all the states with the corresponding weight $[W]_{ij}$
4. Each sensor node $v_i \in \mathbf{V}$ adapts its current inner state using a linear update scheme as follows: $x_i(k+1) = [W]_{ii} . x_i(k) + \Sigma_j [W]_{ij} . x_j(k)$

Algorithm 2: Distributed Linear Average Consensus Algorithm with Stopping Criterion [13]

In the beginning, each sensor node v_i initiates its inner state with a scalar value $x_i(0)$ (1 TRUE / 0 FALSE in our case) and *counter* with zero. (a node is active until it completes AC)

At each iteration as long as $k \neq k_l$

1. Each active sensor node $v_i \in \mathbf{V}$ sends a broadcast message containing its current inner state (i.e. $x_i(k)$ to $\forall v_j \in \boldsymbol{N}_i$)
2. Each active node sensor $v_i \in \mathbf{V}$ receives the inner states from $\forall v_j \in \boldsymbol{N}_i$
3. Each active sensor node $v_i \in \mathbf{V}$ multiplies all the states with the weight $[W]_{ij}$
4. Each active sensor node $v_i \in \mathbf{V}$ adapts its current inner state using a linear update scheme

as follows: $x_i(k+1) = [W]_{ii} \,.\, x_i(k) + \Sigma_j [W]_{ij} \,.\, x_j(k)$
5. Each active sensor node $v_i \in \mathbf{V}$ computes $|\Delta x_i(k)| = |x_i(k+1) - x_i(k)|$ and increments *counter* by one if $|\Delta x_i(k)| <$ *accuracy*, otherwise, sets *counter* to zero
6. Each active sensor node $v_i \in \mathbf{V}$ verifies whether *counter* = *counter threshold* and considers the algorithm to be completed if the condition is valid (and becomes inactive)

3. Examined weight designs

This section addresses four frequently cited weight designs for AC further examined in this paper. The first one is the Maximum Degree weight design (abbreviated as MD), which requires the information about the degree of the best-connected sensor node (this information can be assessed for example by the distributed max-consensus algorithm or pre-set in some networks). Its weight matrix is defined as follows [12]:

$$[W^{\mathrm{MD}}]_{ij} = \begin{cases} \dfrac{1}{\max\limits_{l}\{d_l\}}, & \text{if } e_{ij} \in \mathbf{E} \\ 0, & \text{if } e_{ij} \notin \mathbf{E} \wedge i \neq j \text{ ,} \\ 1 - \dfrac{d_i}{\max\limits_{l}\{d_l\}} & \text{if } i = j \end{cases} \tag{8}$$

Here, d_i is the degree of a vertex and so, the number of neighbors of the corresponding sensor node. Another weight design of our interest is the Metropolis-Hastings weight design (abbreviated as MH). Its initial setup requires only locally available information, i.e. the own degree and the degrees of the sensor nodes from the adjacent area (i.e. for v_i, the degrees of $\forall v_j \in \mathcal{N}_i$). Subsequently, its weight matrix is composed as follows [12]:

$$[W^{\mathrm{MH}}]_{ij} = \begin{cases} (1 + \max\{d_i, d_j\})^{-1}, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - \sum_{k=1, k \neq i}^{N} [W^{\mathrm{MH}}]_{ik}, & \text{if } i = j \quad , \\ 0, & \text{otherwise ,} \end{cases} \tag{9}$$

Another analyzed design is the Local Degree weight design (abbreviated as LD), which is optimized MH in such a way that the weights for the edges are increased by omitting one in the denominator. Its weight matrix is defined as [15]:

$$[W^{\mathrm{LD}}]_{ij} = \begin{cases} (\max\{d_i, d_j\})^{-1}, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - \sum_{k=1, k \neq i}^{N} [W^{\mathrm{LD}}]_{ik}, & \text{if } i = j \quad , \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

The last design of our interest is the Best Constant weight design (abbreviated as BC), which requires the exact values of the largest (λ_1) and the second smallest ($\lambda_{N\text{-}1}$) eigenvalue of the corresponding Laplacian matrix **L** (a mathematical tool for a description of network topologies (see [12] for a definition)) for its optimized variant. This weight design is considered to be the most efficient among the uniform-weight designs [15].

$$[W^{\mathrm{BC}}]_{ij} = \begin{cases} 2/(\lambda_1(\mathbf{L}) + \lambda_{N-1}(\mathbf{L})), & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - 2.d_i/(\lambda_1(\mathbf{L}) + \lambda_{N-1}(\mathbf{L})), & \text{if } i = j \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

4. Experiments and discussion

This section is concerned with the results obtained from numerical experiments (the simulations are carried out in Matlab 2016a) and a discussion about the observed phenomena. As mentioned above, this paper addresses a comparative study of four AC weight designs (namely, MD, MH, LD, and BC) with the fully-distributed stopping criterion for WSNs presented in [13], ensuring a finite execution time of the algorithm. The parameters of the stopping criterion are varied: *accuracy* (takes these values: 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}) and *counter threshold* (takes these values: 3, 5, 7, 10, 20, 40, 60, 80, 100) to examine their impact on the precision of the final states quantified using an MSE-metric (mean square error) and the convergence rate expressed in the number of the iterations necessary for the consensus (a lower value means a higher rate). Thus, the main goal of this paper is to find the most appropriate weight designs for the examined stopping criterion (for the varied values of the mentioned parameters) in terms of the precision and the convergence rate. As shown in ([16-18]), the MSE is a reasonable metric frequently used for an analysis (not only) of consensus-based algorithms and is defined as follows:

$$MSE = \frac{1}{N} . \sum_i \left(x_i(k_l) - \frac{\mathbf{1}^{\mathrm{T}} \times \mathbf{x}(0)}{N} \right)^2, \tag{12}$$

Our intention is demonstrated on 60 random geometric graphs (30 sparsely and 30 densely connected) whose sensor nodes have randomly generated initial inner states of the Bernoulli distribution with Pr(x = 1) = 0.5 and Pr(x = 0) = 0.5. The size of all the networks is the same - 200 sensor nodes. No negatives factors are assumed. In all the experiments, only the averaged MSE over 30 networks of the same connectivity is shown. See Appendix for detailed information.

In the first experiment, the asymptotic convergence factor and the associated convergence time for each weight design are examined. As mentioned above, due to the limited range of the paper, only the average value over all 30 networks for the densely and the sparsely connected networks is shown separately (see Fig. 1).

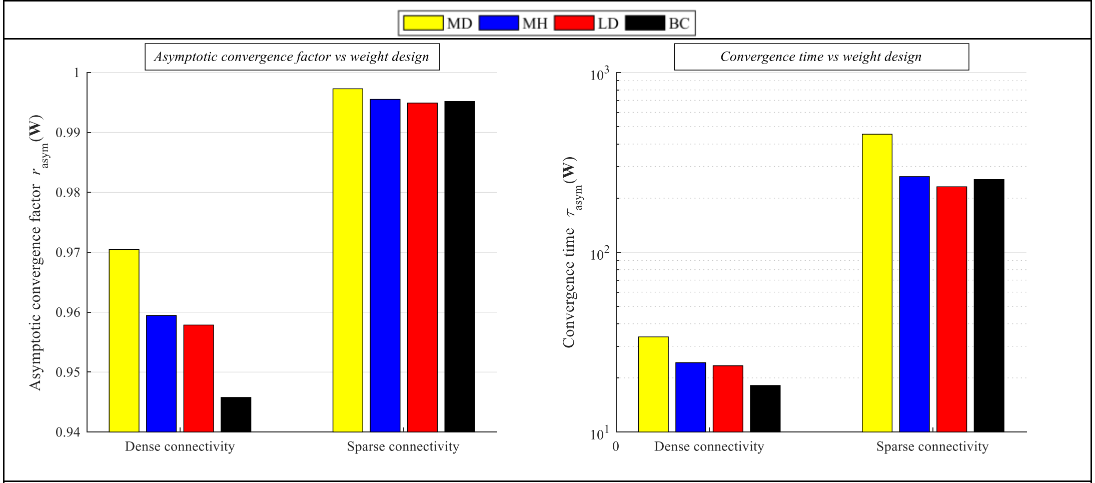

Fig. 1. Asymptotic convergence factor and convergence time averaged over 30 randomly generated densely and sparsely connected networks for each weight design

In the densely connected networks, MD takes the highest values of both parameters and so, is the worst among the examined weight designs (a smaller value means a higher convergence rate). The second worst is MH, the third one is LD, and the best results are observed for BC. In the sparsely connected network, the highest values of both parameters are taken by MD, the second highest one by MH, the third one by BC, and the lowest values are taken by LD (thus, LD outperforms BC compared to the densely connected networks). Generally, it can be seen that the algorithm is slower in the sparsely connected networks than in the densely connected ones regardless of the used weight design. Higher performance is caused by the fact that the second largest eigenvalue of **W** in magnitude is generally lower in graphs with more edges.

The next experiment is concerned with a performance analysis of the mentioned weight designs with the implemented examined stopping criterion quantified by an MSE-metric (the MSE is converted into dB). The results obtained in the densely connected networks are shown in Fig. 2 (the y-axis is reversed). It can be seen from the results that an increase in *counter threshold* and a decrease in the value of *accuracy* result in lower values of the MSE and so, a higher precision of the final estimates regardless of the used weight design. Moreover, for each *accuracy* with each *counter threshold*, BC achieves the highest performance, LD is the second most precise, MH is the third one, and the lowest precision is achieved by MD.

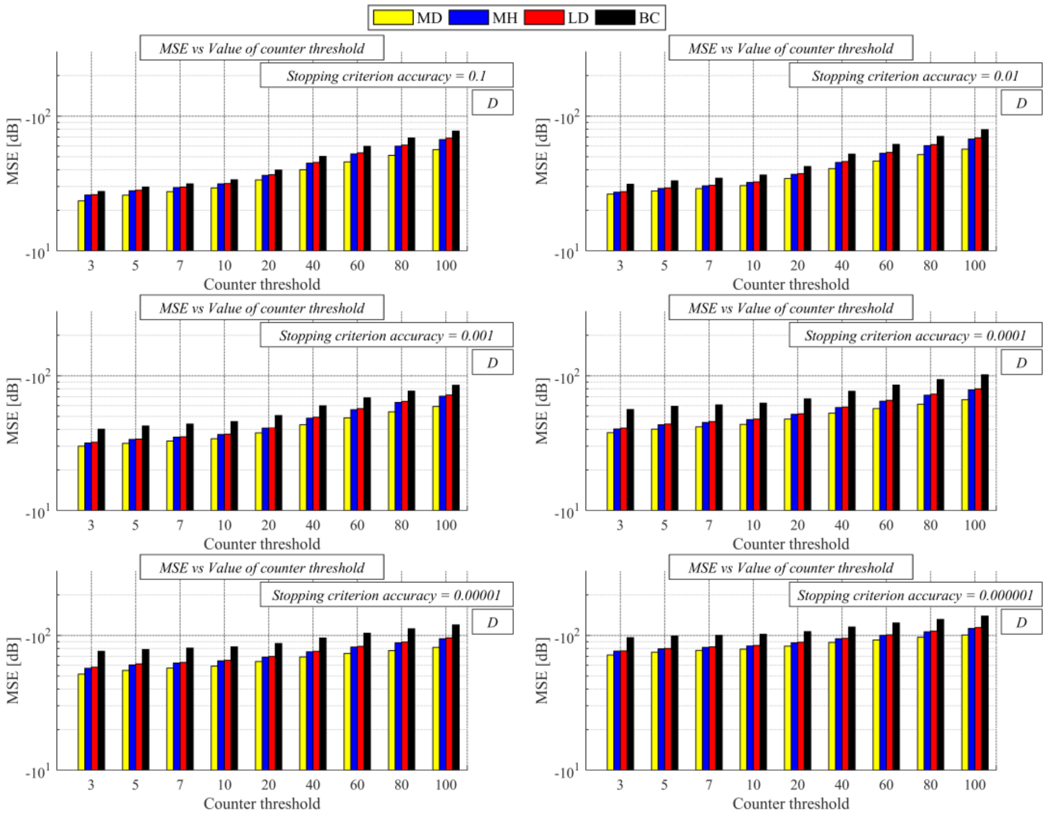

Fig. 2. MSE in decibels averaged over 30 randomly generated densely connected networks as function of *counter threshold* for each *accuracy*

In the sparsely connected networks (Fig. 3), an increase in *counter threshold* and a decrease in *accuracy* cause that lower values of the MSE can be observed again. For *accuracy* = 10^{-1} with each value of *counter threshold*, the highest precision is achieved by LD, the second highest one by BC, the third one by MH, and the lowest precision is achieved by MD. For *accuracy* = 10^{-2} – 10^{-5} with *counter threshold* = 3 – 40, BC achieves the highest precision, LD the second highest one, MH the third one, and the lowest one is achieved by MD. For the other values of *counter threshold* (i.e. {60, 80, 100}), LD outperforms BC and so, is the most precise (BC is the second one), MH is the third most precise, and MD is the most imprecise. For *accuracy* = 10^{-6} with each *counter threshold*, the order of the weight designs sorted according to the precision is the same as in the densely connected networks, i.e. BC is the best one, LD is the second, MH is the third one, and MD is the worst. Moreover, it is seen that the precision is higher in the densely connected networks in general.

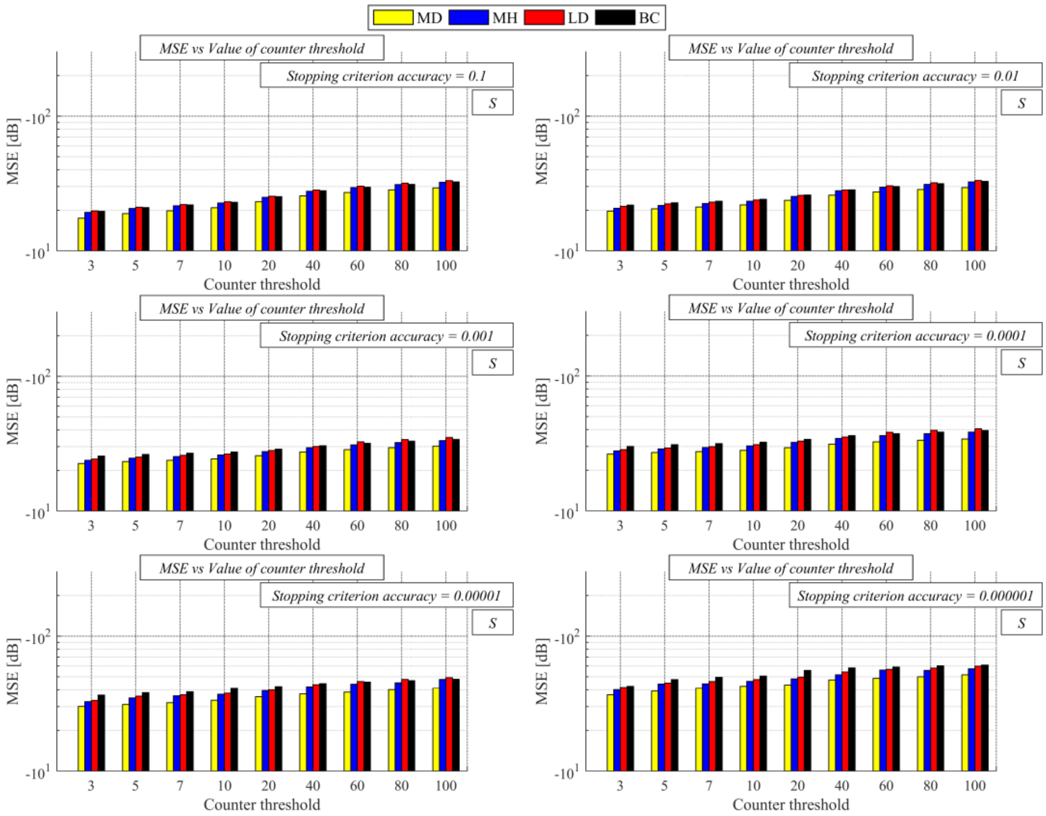

Fig. 3. MSE in decibels averaged over 30 randomly generated sparsely connected networks as function of *counter threshold* for each *accuracy*

The following paragraphs address the convergence rate of the examined weight designs in both sets of the networks. The convergence rates of all the examined weight designs in the densely connected networks are shown in Fig. 4. An increase in *counter threshold* and a decrease in *accuracy* (and so, factors ensuring a higher precision of the final estimates) result in a deceleration of the algorithm regardless of the used weight design. For *accuracy* = 10^{-1} with each *counter threshold*, the highest convergence rate is achieved by MH and LD (both achieve the same average convergence rate), the third fastest is MD, and the lowest convergence rate is observed for BC. For *accuracy* = 10^{-2} – 10^{-4} with each *counter threshold*, LD is the fastest, MH is the second, MD is the third, and BC is the slowest again. For *accuracy* = 10^{-5}, BC outperforms MD for each value of *counter threshold* and is thus the third fastest (MD is the slowest), MH is the second fastest, and LD the fastest one. For *accuracy* = 10^{-6} with *counter threshold* = 3 – 20, the highest convergence rate is achieved by BC, the second highest one by

LD, the third one by MH, and the lowest one by MD. For *counter threshold* = 40 – 100, BC is outperformed by LD (the fastest weight designs in this interval) and MH (the second fastest) and MD is slowest again.

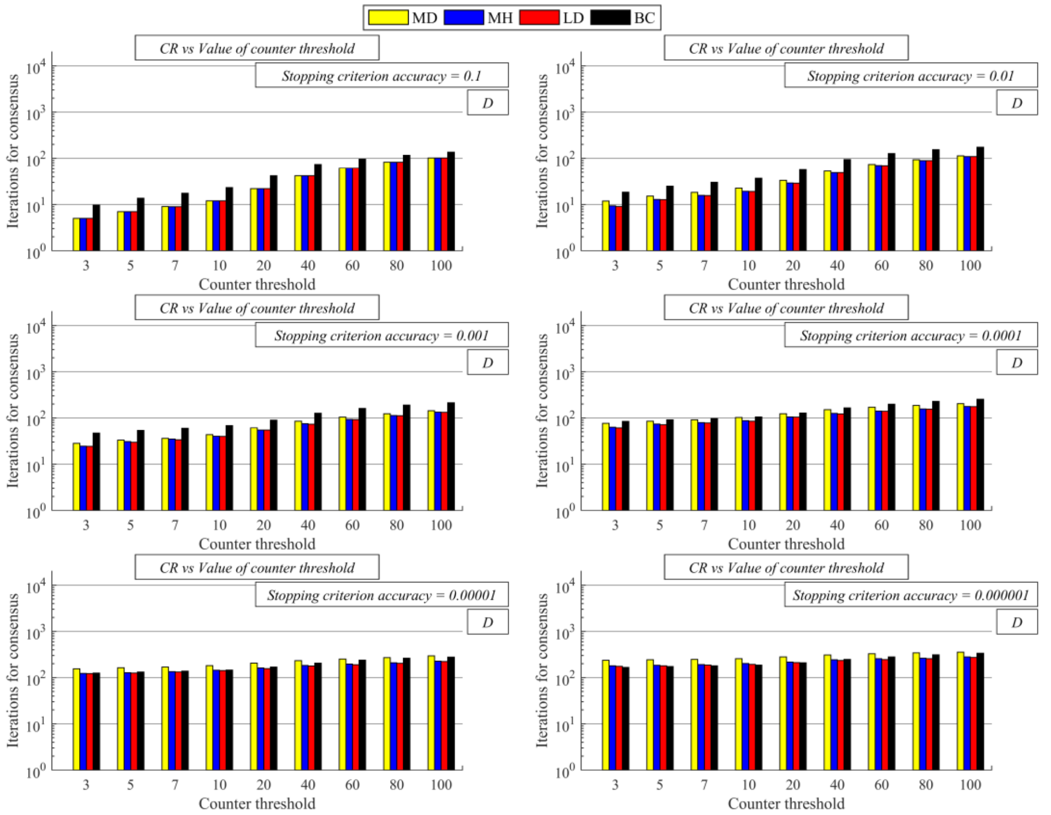

Fig.4. Convergence rate (expressed as iterations for consensus) averaged over 30 randomly generated densely connected networks as function of *counter threshold* for each *accuracy*

In the sparsely connected networks, an increase in *counter threshold* and a decrease in *accuracy* cause a deceleration of the algorithm regardless of the used weight design like in the densely connected networks. Furthermore, for *accuracy* = 10^{-1} – 10^{-2} with each *counter threshold*, MH achieves the highest performance, LD is the second, MD is the third, and BC is the slowest. For all other examined values of *accuracy* with each *counter threshold*, LD is the fastest, MH is the second fastest, MD is the third, and BC is the slowest. In general, the convergence rate is higher in the densely connected networks.

Furthermore, we prove the unpredictability of BC, whose convergence rate and precision are also the most significantly affected by factors such as the network topology, the connectivity, the distribution of the initial states ([14], [19]), also when the algorithm is bounded by the stopping criterion from [13]. The weight design is the slowest for most of the stopping criterion parameters, however, we can see that it is also the fastest one for some configuration of the parameters in the densely connected networks.

The novelty of our paper compared to other papers: we analyze four frequently weight designs of AC bounded by the stopping criterion from [13] proposed for WSNs. Only in [13], this stopping criterion is experimentally analyzed, however, this paper is concerned with the Constant weight design and also a significantly different research methodology is applied. Thus, the other papers focused on the same problems either theoretically discuss the implemented stopping criterion ([3], [14], [26-33]) or a comparative study is carried out using a different research methodology (e.g. no/another stopping criterion is implemented) ([19], [34-35]).

As mentioned earlier, the only paper focused on an experimental analysis of the implemented stopping criterion is [13]. The other papers ([3], [14], [26-27], [29-33]) citing the paper where this stopping criterion is proposed only theoretically discuss its advantages/disadvantages without any simulation evaluation. In [13], the stopping criterion is implemented into a hardware platform from Memsic. The authors focus their attention on the Constant weight design

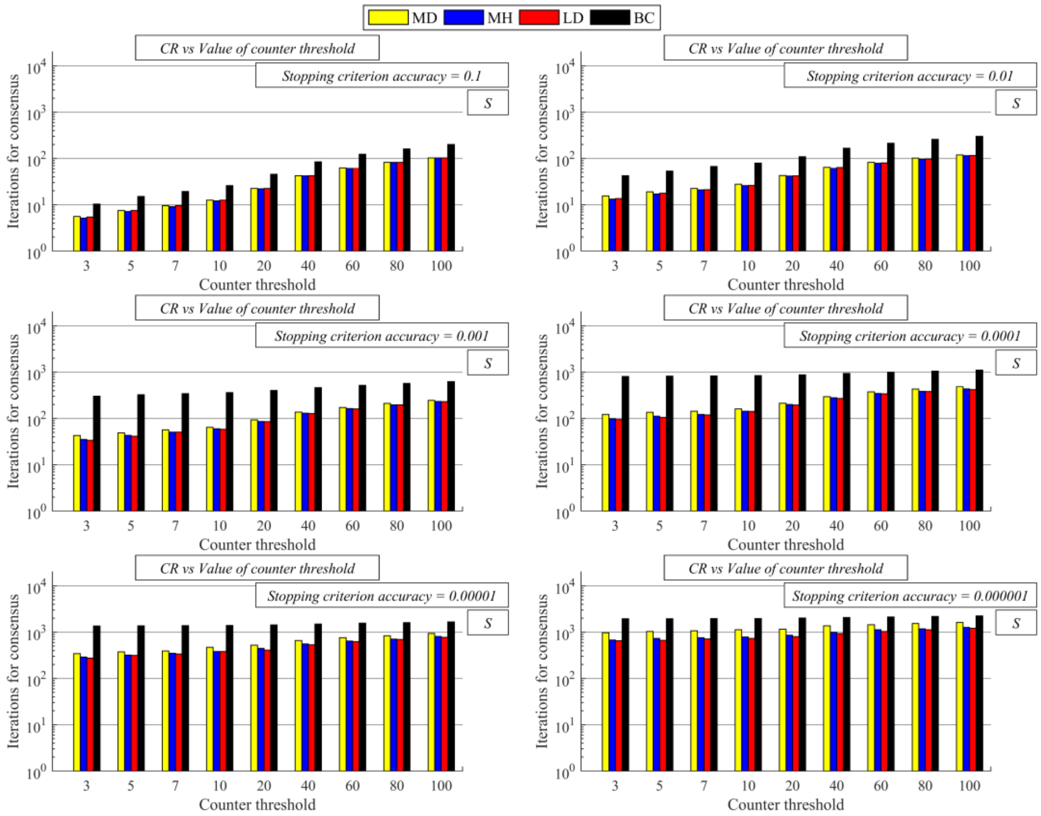

Fig.5. Convergence rate (expressed as iterations for consensus) averaged over 30 randomly generated sparsely connected networks as function of *counter threshold* for each *accuracy*

with a modifiable mixing parameter. They show that an increase in the mixing parameter ensures a higher convergence rate until it reaches the threshold value determined by the size of a network - then, the convergence rate decreases. Moreover, they prove that lower values of the mixing parameter result in higher robustness of the algorithm to collisions (however, the character is, in general, the exact opposite when the link failures are assumed). It is also shown that lower values of *accuracy* decelerate the algorithm (this statement is proven also in our paper). In the paper [28], the authors confirm the same results as in [13].

5. Future research

Our future research related to the examined stopping criterion is going to be focused on finding the most appropriate weight design also in terms of other aspects (the robustness, other functionalities etc.) and its optimization. Furthermore, our plans also include experimental verification of whether the examined stopping criterion is suitable for the distributed gossip-based algorithms – this research can provide a sufficient background for a mutual comparison.

As mentioned above, consensus-based algorithms find the application in WSNs, IoT, and Cloud Computing and their integration. In this paragraph, we provide a brief overview of this topic. There are many papers concerned with these technologies and their mutual integration in the literature ([8], [20-25]). In [20], an approach based on Timed Colored Petri Net and Ontology. The authors of the paper address the controlling of the logical correctness of the context-aware services, which is considered to be one of the most important challenges in the IoT technology. Some approaches are based on merging of artificial intelligence with IoT. The authors of [21] propose and present a hybrid model consisting of IoT and artificial neural networks taught by the back-propagation algorithm. It allows heterogeneous technologies to act as intelligent entities that are able to make independent decisions and interact with human beings or other smart devices. In paper [22], the premise that the residential houses will evolve into modern households with own solar panels and wind turbines able to sell or buy energy to or from the smart power grid is addressed. The authors propose a holistic framework for the integration of smart home objects in a cloud-centric IoT solution is proposed. This hybrid serves not only for collecting and storing the data but works as a gateway to third-parties that develop applications. In [23], the authors propose a low-cost automation system based on WSNs incorporating IoT. This mechanism provides a cost-effective solution to Home Automation. The authors of [8] integrate WSNs, IoT, and Cloud Computing and propose a concept for controlling and monitoring an irrigation system that is connected to an IoT platform. The authors of [24] propose a cloud computing and fog computing architecture for effectively processing IoT data. A classification mechanism for

IoT data types is presented. The data placement problem is cast by the authors as an optimization problem so that the latency in accessing the data can be minimized. The authors of [25] conclude that the current trends in WSNs are to embrace IP-based sensor network using standards such as 6LoWAN and IPv6. Moreover, the authors propose a framework to harmonize new installations and non-IP based ones and preserve the possibility to migrate to an all-IP environment. Our future goals will also include an effort to find the applicability of the average consensus algorithm in the integration of WSNs, IoT, and Cloud Computing in order to increase QoS (Quality of Service) of the executed applications.

6. Conclusion

This paper addresses a comparative study of four weight designs of AC (MD, MH, LD, BC) with a fully-distributed stopping criterion proposed for WSNs and finds out which weight design achieves the highest/the lowest precision (using an MSE-metric) and the highest/the lowest convergence rate for various *accuracy* and *counter threshold* over 30 densely and 30 sparsely connected random geometric graphs.

It is seen that the precision of the final estimates and the convergence rate is higher in the densely connected networks for each weight design (when the results for same *accuracy* and same *counter threshold* are mutually compared). Thus, it is shown that the theoretical assumptions (that higher performance is achieved in networks with a higher connectivity - demonstrated among others by Fig. 1 in this paper, i.e. a smaller asymptotic convergence time and associated time are achieved in the densely connected networks) valid for AC, whose execution is not bounded, are valid also for AC with the implemented stopping criterion. Moreover, it can be seen that an increase in *counter threshold* and a decrease in *accuracy* ensure a lower MSE (and so, a higher precision) at a cost of a deceleration of the algorithm in both sets of the networks and regardless of the used weight design. Moreover, it can be seen that BC is the most precise in the densely connected networks (it maximally outperforms MD by approx. 39 dB, MH by approx. 27 dB, and LD by approx. 25 dB). This design has also a significantly lower r_{asym} and τ_{asym} in these networks than the concurrent ones. The lowest precision is achieved by MD, which has also the highest r_{asym} and $\tau_{\text{asym.}}$ In terms of the convergence rate, the best performance is in general achieved by LD (it maximally outperforms MD by approx. 86 iterations, MH by approx. 11 iterations, and BC by approx. 82 iterations) and the worst one by BC. Paradoxically, BC outperforms all the concurrent weight designs for lowest examined *accuracy* with lower *counter threshold* (it maximally outperforms fastest LD by approx. 10).

In the sparse networks, the highest precision is achieved by BC (primarily, either for lower values of *counter threshold* or lower values of *accuracy* - it maximally outperforms MD by approx. 12 dB, MH by approx. 7.5 dB, and LD by approx. 6 dB) and LD (primarily, either for higher values of *counter threshold* or higher values of *accuracy* - it maximally outperforms MD by approx. 8 dB, MH by approx. 3 dB, and LD by approx. 1 dB). Regarding r_{asym} and τ_{asym}, LD has the lowest values of these parameters in the sparsely connected networks, meanwhile, BC the second lowest. Like in the densely connected networks, MD (the highest r_{asym} and τ_{asym}) has the lowest precision again. The highest convergence rate is achieved by LD except for lower values of *accuracy*, when it is slightly (less than 2 dB) outperformed by MH (the second highest r_{asym} and τ_{asym}) (LD maximally outperforms MD by approx. 441 iterations, MH by approx. 102 iterations, and BC by approx. 1296 iterations). Furthermore, BC is significantly slower in the sparsely connected networks than the concurrent weight designs. So, BC achieves generally the highest performance in terms of the precision, meanwhile, LD is the fastest in general among the examined weight designs. The unpredictable character of BC, i.e. this weight design is more significantly affected by the network topology, the initial states, the connectivity etc., is confirmed also in this paper. In our work, we show that this weight design is the slowest in the most of the cases, however, also the fastest one for some configuration of the stopping criterion parameters in the densely connected networks.

Acknowledgment

This work was supported by VEGA agency under the contract No. 2/0155/19 and by the Slovak Research and Development Agency under the contract No. APVV-17-0116.

REFERENCES

[1]. *J. Cota-Ruiz, J. G. Rosiles, P. Rivas-Perea, and E. Sifuentes,* A distributed localization algorithm for wireless sensor networks based on the solutions of spatially-constrained local problems, in IEEE Sens. J., **vol. 55**, no. 6, Mar. 2013, pp. 2181–2191.

[2]. *N. Zhao, X. Chi, L. Zhao, and Y. Zhu,* A Spectrum Efficient Self-Admission Framework for Coexisting IEEE 802.15. 4, in Radioengineering, **vol. 27**, no. 1, Apr. 2018, pp. 326–334.

[3]. *F. G. Zacchigna and A. Lutenberg,* A novel consensus algorithm proposal: Measurement estimation by silent agreement (MESA), in Proceedings of the 2014 Fifth Argentine Symposium and Conference on Embedded Systems SASE/CASE 2014, pp. 7–12.

[4]. *R. Mishra, V. Jha, R. K. Tripathi, and A. K. Sharma,* Design of Probability Density Function Targeting Energy Efficient Network for Coalition Based WSNs, in Wirel. Pers. Commun., **vol. 99**, no. 2, Mar. 2018, pp. 651–680.

[5]. *D. Izadi, J. H. Abawajy, S. Ghanavati, and T. Herawan,* A data fusion method in wireless sensor networks, in Sensors, **vol. 15**, no. 2, Jan. 2015, pp. 2964–2979.

[6]. *G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson,* Event-based broadcasting for multi-agent average consensus, in Automatica, **vol. 49**, no. 1, Jan. 2013, pp. 245–252.

[7]. *D. Merezeanu and M. Nicolae,* Consensus control of discrete-time multi-agent systems, in U. Politeh. Buch. Ser. A, **vol. 79**, no. 1, Aug. 2017, pp. 167–174.

[8]. *D. Merezeanu, G. Vasilescu, and R. Dobrescu,* in Context-aware control platform for sensor network integration, in IoT and Cloud, in Stud. Inform. Control, **vol. 25**, no. 4, Dec. 2016, pp. 489–498.

[9]. *G. Suciu, V. Suciu, V. C. A. Focsa, S. Halunga, O. A. Mohamad, S. C. Arseni, and C. Butca* Integrating telemetry sensors with cloud computing, in Proceedings of the 14th RoEduNet International Conference - Networking in Education and Research (RoEduNet NER) 2015, pp. 218–222.

[10]. *V. Pilloni and L. Atzori,* Consensus-based resource allocation among objects in the internet of things, in Ann. Telecommun., **vol. 72**, no.7-8, Aug. 2017, pp. 415–429.

[11]. *D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei,, and C. Qijun,* A review on consensus algorithm of blockchain, in Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2567–2572.

[12]. *M. Kenyeres, J. Kenyeres, V. Skorpil, and R. Burget,* Distributed aggregate function estimation by Biphasically configured Metropolis-Hastings weight model, in Radioengineering, **vol. 26**, no. 2, June 2017, pp. 148–155.

[13]. *M. Kenyeres, J. Kenyeres, M. Rupp, and P. Farkas*, WSN implementation of the average consensus algorithm, in Proceedings of the 17th European Wireless 2011 – Sustainable Wireless Technologies, pp. 139–146.

[14]. *N. Al-Nakhala, R. Riley, and T. Elfouly,* Distributed algorithms in wireless sensor networks: An approach for applying binary consensus in a real testbed, in Comput. Netw., **vol. 79**, Mar. 2015, pp. 30–38.

[15]. *L. Xiao and S. Boyd*, Fast linear iterations for distributed averaging, in Syst. Control Lett., **vol. 53**, no. 1, Sept. 2004, pp. 65–78.

[16]. *S. S. Pereira and A. Pagés-Zamora,* Mean square convergence of consensus algorithms in random WSNs, in IEEE Trans. Signal Process., **vol. 58**, no. 5, Sept. 2011, pp. 2866–2874.

[17]. *V. Skorpil and J. Stastny,* Back-propagation and k-means algorithms comparison, in Proceedings of the 2006 8th international Conference on Signal Processing, pp. 1871–1874.

[18]. *T. C. Aysal, B. N. Oreshkin, and M. J. Coates*, Accelerated distributed average consensus via localized node state prediction, in IEEE Trans. Signal Process., **vol. 57**, no. 4, Dec. 2008, pp. 1563–1576.

[19]. *M. Kenyeres, J. Kenyeres, and R. Burget,* Evaluation of Natural Robustness of Best Constant Weights to Random Communication Breakdowsn, in Journal of Communications Software and Systems, **vol. 14**, no. 3, Dec. 2018, pp. 201-210.

[20]. *M. Davoudpour, A. Sadeghian, and H. Rahnama,* CANthings (Context Aware Network for the Design of Connected Things) service modeling based on Timed CPN, in Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing, IEEE ICSC 2015, pp. 127–130.

[21]. *A. Poniszewska-Maranda and D. Kaczmarek,* Selected methods of artificial intelligence for internet of things conception, in Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, pp. 1343–1348.

[22]. *B. L. Risteska Stojkoska and K. V. Trivodaliev,* A review of Internet of Things for smart home: Challenges and solutions, in J Clean. Prod., **vol. 140**, Jan. 2017, pp. 1454–1464.

[23]. *N. Vikram, K. S. Harish, M. S. Nihaal, R. Umesh, A. Shetty, and A. Kumar,* A low cost home automation system using wi-fi based wireless sensor network incorporating Internet of Things (IoT), in Proceedings of the 7th IEEE International Advanced Computing Conference, IACC 2017, pp. 174–178.

[24]. *F. Karatas and I. Korpeoglu,* Fog-Based Data Distribution Service (F-DAD) for Internet of Things (IoT) applications, in Future Gener. Comput. Syst., **vol. 93**, Apr. 2019, pp. 159-169.

[25]. *L. Mainetti, L. Patrono, and A. Vilei,* Evolution of wireless sensor networks towards the Internet of Things: A survey, in Proceedings of the 2011 International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2011, pp. 16–21.

[26]. *G. Stamatescu, I. Stamatescu, and D. Popescu,* Consensus-based Data Aggregation for Wireless Sensor Networks, in Journal of Control Engineering and Applied Informatics, **vol. 19**, no. 2, June 2017, pp. 43–50.

[27]. *O. Sluciak and M. Rupp, Reaching* consensus in asynchronous WSNs: Algebraic approach, in Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3300-3303.

[28]. *T. Sheltami,* An enhanced energy saving approach for WSNs, in Procedia Comput. Sci., **vol. 21**, Oct. 2013, pp. 199-206.

[29]. *N. Al-Nakhala, R. Riley, and T. Elfouly,* Clustered binary consensus in sensor motes, in Proceedings of the 20th European Wireless Conference, EW 2014, pp. 1-7.

[30]. *V. Jelicic, D. Tolic, and V. Bilas,* Consensus-based decentralized resource sharing between co-located wireless sensor networks, in Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 1-6.

[31]. *S. V. Macua, C. et al.,* How to implement doubly-stochastic matrices for consensus-based distributed algorithms, in Proceedings of the 2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 333-336.

[32]. *N. Goddemeier, D. Behnke, and C. Wietfeld,* Impact of communication constraints on consensus finding in multi-agent systems, in Proceedings of the 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 2464-2468.

[33]. *N. Al-Nakhala, R. Riley, and M. T. Elfouly,* Binary consensus in sensor motes, in Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1337-1342.

[34]. *S. Jafarizadeh and A. Jamalipour*, Weight optimization for distributed average consensus algorithm in symmetric, CCS & KCS star networks, arXiv preprint arXiv:1001.4278.

[35]. *V. Schwarz and G. Matz,* Average consensus in wireless sensor networks: Will it blend?, in Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4584-4588.

Appendix

Table 1

Summarization of simulation setup parameters

SIMULATION PARAMATERS	
Simulation tool	
Matlab 2016a (all the used simulation software was developed by the authors of this paper)	
Used Environment	
Random Geometric Graphs	
Graph Size	200 vertices
Connectivity	Dense / Sparse
Number of graphs in each graph set	30 unique graphs
Topology	Time-invariant
Implemented stopping criterion	
Character	Fully-distributed
Values of *accuracy*	10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}
Values of *counter threshold*	3, 5, 7, 10, 20, 40, 60, 80, 100
Initial inner states	
Random variables of Bernoulli distribution $x_i(0) \sim B(1, 0.5)$	
Analyzed parameters	
Asymptotic convergence factor averaged over 30 densely/sparsely connected networks Associated convergence time averaged over 30 densely/sparsely connected networks Mean square error averaged over 30 densely/sparsely connected networks Convergence rate expressed as the number of the iterations for the consensus achievement averaged over 30 densely/sparsely connected networks	
Restrictions and presumptions	
The algorithm execution is affected by no negative factors such as communication interference, potential link/node failures, communication delays, noises etc. Moreover, the sensor nodes are homogenous in all the aspects and synchronized.	