Distributed Aggregate Function Estimation by Biphasically Configured Metropolis-Hasting Weight Model

Martin KENYERES[1], *Jozef KENYERES*[2], *Vladislav SKORPIL*[1], *Radim BURGET*[1]

[1] Dept. of Telecommunications, Brno University of Technology, Technická 12, 612 00 Brno, Czech Republic
[2] Sipwise GmbH, Europaring F15, 2345 Brunn am Gebirge, Austria

kenyeres@phd.feec.vutbr.cz, jkenyeres@sipwise.com, skorpil@feec.vutbr.cz, burgetrm@feec.vutbr.cz

Submitted December 12, 2016 / Accepted April 7, 2017

Abstract. *An energy-efficient estimation of an aggregate function can significantly optimize a global event detection or monitoring in wireless sensor networks. This is probably the main reason why an optimization of the complementary consensus algorithms is one of the key challenges of the lifetime extension of the wireless sensor networks on which the attention of many scientists is paid. In this paper, we introduce an optimized weight model for the average consensus algorithm. It is called the Biphasically configured Metropolis-Hasting weight model and is based on a modification of the Metropolis-Hasting weight model by rephrasing the initial configuration into two parts. The first one is the default configuration of the Metropolis-Hasting weight model, while, the other one is based on a recalculation of the weights allocated to the adjacent nodes' incoming values at the cost of decreasing the value of the weights of the inner states. The whole initial configuration is executed in a fully-distributed manner. In the experimental section, it is proven that our optimized weight model significantly optimizes the Metropolis-Hasting weight model in several aspects and achieves better results compared with other concurrent weight models.*

Keywords

Distributed computing, aggregate function, average consensus algorithm, Metropolis-Hasting weight model, wireless sensor networks

1. Introduction

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are systems intended to perform a real-time detection of a stochastic event or to monitor physical quantities [1–2]. They are formed by battery-constrained nodes deployed in a geographical area where a phenomenon of interest is observed. These nodes are equipped with hardware components such as a wireless transceiver, a sensor to sense physical quantities, a central processor unit, a source of energy etc. [3–4] Thus, the nodes are able to obtain necessary information about the observed phenomenon, process it, mutually exchange data and make a meaningful decision on the examined physical quantity [4]. Due to their character, the WSNs find the application in various areas such as military surveillance, a natural disaster detection and its elimination, habitat monitoring, inventory tracking, an acoustic detection, pollution monitoring, medical systems, target tracking, a robotic exploration, a health care (especially, they find the usage in the scenarios considering monitoring elderly patients in a remote area), environment monitoring, a micro surgery, agriculture etc. [5–6]. In many applications, WSNs may be formed by hundreds of nodes potentially situated in inaccessible locations and therefore, a battery recharge or replacement may be complicated [7]. An exhausted battery results in a node death, which can decrease the quality of the final decisions or even prevent a whole system from fulfilling its functionality. As the results, the attention of many scientists has been focused on an optimization of the energy consumption aspect in the last years [8–11]. It is because an effective optimization can significantly increase the network lifetime of a WSN application [12].

In [13], the authors divide the architectures of a global event detection into three categories. The second and the third architecture require a complementary consensus algorithm to estimate aggregate functions in order to ensure a higher credibility of the measured outputs. These architectures do not assume the presence of a fusion center in a network. The implementation of this supplementary algorithm ensures a higher precision of the final decision on the observed phenomenon in many applications [14]. A decision made according to data obtained by independently-measuring nodes secures a more credible output than a decision made in terms of a single measurement and minimizes a change of an incorrect classification [14]. The importance of the consensus algorithm implementation for high-quality monitoring in WSNs is discussed in [15].

DOI: 10.13164/re.2017.0479

1.2 Average Consensus Algorithm

Due to the character of the WSNs, the modern applications are often based on the implementation of distributed mechanisms. The algorithms of distributed computing substitute the older frequently-implemented centralized manner of the computation [14]. Despite its reliability and high precision, the centralized algorithms do not pose the optimal solution for the implementation into the systems formed by battery-constrained devices.

One of the most appropriate distributed algorithms for WSN applications is average consensus, which is a fully distributed iterative algorithm primarily for estimation the average from the values of all the nodes present in a network [16]. This algorithm does not require the presence of any fusion center. The nodes are able to estimate the average by a mutual exchange of their inner states with the nodes situated in the adjacent area. The average consensus algorithm is characterized by a high flexibility because its execution is modifiable by the chosen weight model [17]. The weight models differ from each other in several aspects, for example, we can list the convergence rate of the algorithm, the process of the initial configuration, the information that is necessary for its proper functionality, the robustness etc. [17], [18]. In this paper, we focus on the Metropolis-Hasting model, which requires only the locally-available information for the initial configuration and therefore, it is one of the most preferred solutions for a real implementation into battery-constrained systems [19].

As mentioned earlier, the average consensus algorithm is primarily proposed for the estimation of the average value. However, tiny modifications can ensure that the algorithm is able to estimate other aggregate functions. One of the other frequently-used applications is the estimation of the network size. The information about the number of the nodes in a network is crucial for a proper functionality of many distributed systems [20]. In this case, the execution of the average consensus algorithm is modified in such a way that one of the nodes has the initial value set to the value equaling 1 (it is called the leader) [21]. The other nodes are set to 0. Subsequently, the nodes converge to the value equaling the reciprocal of the size of a network [21]. However, this modification causes several problems. One of the most significant ones is how to appoint the most suitable node as the leader. It often requires the implementation of other complementary mechanisms to determine this, which is not the optimal solution for battery-constrained devices [21]. As shown later in this paper, a bad choice of the leader can significantly decrease the convergence rate of the algorithm. Thus, an improvement of the leader selection can significantly optimize WSN applications by removing the necessity for other complementary algorithms.

1.3 Motivation

All the previously-discussed problems motive us to propose an optimization mechanism of the Metropolis-Hasting weight model that improves the convergence rates, the number of the necessary messages and minimize the negative effect caused by an inappropriate choice of the leader. An optimization of these mentioned aspects can significantly accelerate and simplify the computation process and therefore optimizes the real-life applications of WSNs.

The choice of this weight model is affected by an effort to improve a weight model of the average consensus algorithm that finds the wide usage thanks to its specific character. The Metropolis-Hasting weight model fulfills these criteria because it does not require any global information about the network for its proper initial configuration and so it works in a fully distributed manner. Thus, it is an appropriate solution for an implementation into the WSNs. Thus, this was the main reason that motivated us to focus our research on this weight model.

In this paper, we introduce an optimized weight model derived from the Metropolis-Hasting weight model that improves the discussed aspects of this model. The optimized model modifies the weight matrix of the Metropolis-Hasting weight model by an additional step during the initial configuration. Thus, the weight matrix is initially configured twice. The first configuration is the default one defined within the Metropolis-Hasting weight model and the other one poses the novelty proposed by us. The other phase is based on recalculating the weights allocated to the adjacent nodes' incoming states at cost of decreasing the value of the weight of the inner states in a distributed way.

1.4 Paper Organization

In Sec. 2, we turn our attention to the latest papers related to the average consensus optimization. In the next section, we provide the mathematical tool used to model the average consensus algorithm executed in the WSNs, present the main features of this algorithm and adduce important theorems defined within the spectral graph theory. We also introduce concurrent weight models which the optimized weight model is compared with. In Sec. 4, we introduce our optimized weight model, provide mathematical tools to model it and derive the convergence proof. In Sec. 5, we examine the optimization of chosen aspects ensured by the optimized weight model. We focus on an optimization of the average estimation, the network size estimation and the range of the convergence rates caused by the choice of the leader. We compare this optimization in three types of the networks – weakly, averagely and strongly connected. All of them consist of ten randomly generated networks. In Appendix section, we adduce the complete results obtained within our numerical experiments.

2. Related Work

This section is devoted to an insight into an optimization mechanism proposed for the average consensus algorithm. We introduce the latest papers dealing with optimization mechanisms.

In [19], [22–25], the authors' attention focuses on the Metropolis-Hasting weight model (also in various applications as a complementary mechanism), which our optimized weight model is derived from. It was developed by Metropolis, Rosenbluth and Teller in 1953 and generalized by Hastings in 1970. It was originally defined within Markov chain Monte Carlo methods and proposed to simulate complex, non-standard multivariate distributions [26]. Its modification for the consensus achievement problem finds the usage in many applications due to its character. For its proper initial configuration, only locally-available information is required, i.e. the number of the neighbors of a particular node and the number of neighbors of the node from its adjacent area. This significantly simplifies the initial configuration phase. Thus, this weight model finds the usage in many applications. Additionally, it also poses a robust solution against a quantization noise [18].

In [27], an average consensus optimization based on the usage of the opportunistic inter-agent communication to achieve the consensus is presented. Each node is endowed by a local criterion determining when to broadcast the inner state to the nodes situated in the adjacent area. In paper [28], a novel consensus protocol is presented that achieves the average state consensus for multi-agent systems in finite time. The protocol contains a linear and a non-linear term. The state consensus is achieved by the non-linear term, while the performance optimization is ensured by the linear term to some degree. The authors of paper [29] present an optimization mechanism based on a division of the computation process into two phases. The first one is the phase of reaching the local consensuses and the second one is the phase of reaching the global consensus. Within the first phase, a network is reorganized into geographically close areas so-called packs. Here, each node converges to the value equaling the average of all the nodes present in a pack. Subsequently, each pack appoints one of the nodes as the head, which communicates with the other heads and converges to the average value. The authors of [30] present a novel continuous-time dynamic average consensus algorithm for networks with the interaction that can be described by weight-balanced directed graphs of a strong connectivity. The nodes are able to track the average of the dynamic inputs with some non-zero steady-error. Its size is controlled by exploiting a design parameter. In [31], a distributed algorithm for average consensus that solves the discrete-time average consensus problem on strongly connected weighted digraphs is presented. Its principle lays in the computation of the average value using the estimation of the left eigenvector associated with the zero eigenvalue of the Laplacian matrix. The authors of [32] built their optimization mechanism on the exploiting of the second-order neighbors. They focus their attention on both the continuous-time case, where the edges are chosen by solving a convex optimization problem formed by utilizing the convex relaxation method, and the discrete-time case, when the edges are chosen using the brute force method. In [33], an optimization mechanism is presented that exploits the prediction of the future value of the inner states. This technique is based on the estimation of the states for the next iterations in terms of the values of the inner states from the previous iterations. The authors of [34] introduce IACA, which is a two-layer improved consensus algorithm of a multi-agent system. The authors propose a new distributed cost optimization method for loading shedding of an islanded microgrid considering cost. The technique solves distributed cost optimization of load shedding by exploiting the synchronization processing of IACA in the layer 2. The authors of [35] present an optimization mechanism minimizing the negative effects caused by a random packet loss. It is based on keeping track of the changes in the state variable, which the neighbors influence causes.

The papers [36–38] focus on the Maximum Degree weight model, which is a modification of the Constant weight model, and its applications. The Constant weight model is characterized by the parameter ε, which affects the convergence rate as well as the interval of the convergence. The higher value it takes, the faster the algorithm is. However, a too high value can cause the divergence of the algorithm. The divergence is a type of a failure when the convergence is not reached. Instead of it, the nodes diverge to infinite large values [39]. This error poses a serious problem that stunts a whole network [39]. The Maximum Degree weight model is based on the setting of the parameter ε to the value equaled to the reciprocal of the number of the neighbors of the best-connected node in a network. The initial configuration requires the knowledge about this value and therefore, it is necessary to implement a supplementary mechanism to determine it [19].

In [18], [40], the Best Constant weight model is discussed. Its optimized version is based on the utilization of the knowledge about the second smallest and the largest eigenvalue of the Laplacian matrix [18]. To compose it, it is necessary to know the information about the complete network topology. Thus, this weight model requires a particular centralization for its optimization.

3. Modeling of Average Consensus Algorithm in WSNs

In Sec. 3.1, we introduce the used mathematical model of the WSNs executing the average consensus algorithm and the main features of this algorithm. In Sec. 3.2, we discuss and mathematically describe the concurrent weight models of average consensus with which our optimized weight model is compared.

3.1 Used Mathematical Model

In order to model the WSNs, a mathematical tool defined within the spectral graph theory is used [41], [42]. A WSN is considered to be an indirect finite graph defined as $G = (\mathbf{V},\mathbf{E})$. The set **V** is formed by all the vertexes, which are representatives of the particular nodes. Each node is labeled by the unique identity number v_i. We assume that the nodes are labeled with the numbers 1, 2,...,N, where N is the size of a network and therefore $|\mathbf{V}| = N$. The mutual connectivity between the nodes is indicated by the existence of an edge. The set $\mathbf{E}\subset\mathbf{V}\mathrm{x}\mathbf{V}$ consists of all the edges present in a graph. The edge is labeled as (v_i,v_j) or e_{ij}. We assume the range homogeneity of the nodes and therefore the following statement holds:

$$e_{ij} \in \mathbf{E} \Leftrightarrow e_{ji} \in \mathbf{E}\,. \tag{1}$$

There are several tools to describe a network topology within the spectral graph theory. One of them is the Laplacian matrix for a description of the mutual connectivity among the nodes. It is a square symmetric matrix defined for all the indirect graphs as follows [43]:

$$[L]_{ij} = \begin{cases} -1, & \text{if } e_{ij} \in \mathbf{E} \\ d_i, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

Here, d_i is the degree of a vertex v_i and so, the number of the corresponding node's neighbors. Except for the mutual connectivity, the Laplacian matrix provides other useful information about the topology. Let us focus on the following sentence [44]:

Lemma 1: Let $G = (\mathbf{V},\ \mathbf{E})$ *be a graph and let* $0 = \mu_1(\mathbf{L}) \leq \mu_2(\mathbf{L}) \leq \mu_3(\mathbf{L}) \leq \dots\dots \leq \mu_N(\mathbf{L})$ *be the ascendingly-ordered eigenvalues of the Laplacian matrix of this graph. Then, G is not connected if* $\mu_2(\mathbf{L}) = 0$.

According to Lemma 1, only the networks with the Laplacian matrix whose second smallest eigenvalue is not equaled to 0 are connected. In the case when $\mu_2(\mathbf{L})$ equals 0, the average consensus algorithm does not estimate the average from the values of all the nodes but estimates the set of the local averages in each connected subpart of a graph. Therefore, we assume only topologies whose second smallest eigenvalue of the Laplacian matrix is not equaled to 0. The knowledge about the exact value of $\mu_2(\mathbf{L})$ and $\mu_N(\mathbf{L})$ is necessary for the optimized initial configuration of the Best Constant weight model [26]. A configuration of this model with a smaller positive value ensures the convergence but the execution of the algorithm is slower [18].

As mentioned above, the average consensus algorithm is an iterative distributed algorithm based on a mutual exchange of the current states among the nodes. The algorithm is modeled by the difference equation defined as follows [45]:

$$\mathbf{x}(k+1) = \mathbf{W} \times \mathbf{x}(k)\,. \tag{3}$$

Here, $\mathbf{W} \subset \mathbf{L}$ is a weight matrix of the algorithm and the time-variant vector $\mathbf{x}(k) \in \mathrm{R}^{N \times 1}$ gathers all the inner states at kth iteration. We assume that the initial states are labeled as $k = 1$. The elements of **W** depend on the used weight model. This matrix also provides useful information about the network topology. The following lemma says about the connectivity of the topology [44]:

Lemma 2: Let $G = (\mathbf{V},\ \mathbf{E})$ *be a graph and let* $1 = \lambda_1(\mathbf{W}) \geq \lambda_2(\mathbf{W}) \geq \lambda_3(\mathbf{W}) \geq \dots\dots \geq \lambda_N(\mathbf{W}) \geq -1$ *be the descendingly-ordered eigenvalues of the weight matrix. Then, G is not connected if* $max\{\lambda_2(\mathbf{W}),\!-\lambda_N(\mathbf{W})\} = 1$.

Within the spectral theory, it is defined that the value $max\{\lambda_2(\mathbf{W}),\ -\lambda_N(\mathbf{W})\}$ equals the spectral radius ρ of the matrix determined as the difference between the matrix **W** and matrix defined as $1/N\cdot\mathbf{1}\times\mathbf{1}^{\mathrm{T}}$ [46]. Thus, in terms of the previous statement, we can write as follows:

$$\rho(\mathbf{W} - \frac{1}{N}\cdot\mathbf{1}\times\mathbf{1}^{\mathrm{T}}) = \max\{\lambda_2(\mathbf{W}), -\lambda_N(\mathbf{W})\}\,. \tag{4}$$

Let us focus on the features of the weight matrix **W**. According to [47], the weight matrix **W** is required to hold the following conditions:

$$\mathbf{W}\times\mathbf{1} = \mathbf{1}, \tag{5}$$

$$\mathbf{1}^{\mathrm{T}} \times \mathbf{W} = \mathbf{1}^{\mathrm{T}}, \tag{6}$$

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

Here, the vector **1** is a column vector whose all elements are equaled to 1 (its size is implicitly assessable from the previous context). Fulfilling the formula (7) ensures the convergence of the average consensus algorithm, meanwhile, the formulae (5) and (6) imply that the weight matrix is doubly stochastic (sometimes, labeled as bistochastic) and determines the convergence point [47]. These two formulas also implicate the following statement:

$$\mathbf{W} = \mathbf{W}^{\mathrm{T}}. \tag{8}$$

Within our analysis, we use also another descriptive tool defined within the spectral graph theory. It is called the adjacency matrix $\mathbf{A} \in \{0,\ 1\}^{N \times N}$ and contains the information about the mutual connectivity between the pairs of the nodes. It is a diagonally symmetric matrix of a square shape for all the indirect graphs. The direct connection (i.e. the existence of an edge) is indicated by the presence of 1 in the corresponding position. Thus, the presence of 0 is an indicator that two nodes are not directly connected to one another. Mathematically, the adjacency matrix is defined as follows [48]:

$$[A]_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Another useful tool is the identity matrix $\mathbf{I} \in \{0,1\}^{N \times N}$ defined as (10) [49]. In literature, also the other notation $\mathbf{I} = \mathrm{diag}(1,1,\dots.1)$ can be found.

$$[I]_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Usually, a lower index is allocated to indicate its size (the label $\mathbf{I}_N$ indicates that the underlying matrix has the size equaled to the number of the size a network).

As mentioned earlier, the average consensus algorithm is an iterative algorithm (regardless of the used weight model) executed in such a way that the nodes converge to the value determined as the average calculated from all the initial values [50]. Therefore, this behavior can be described as follows [45]:

$$\lim_{k\to\infty}\mathbf{x}(k)=\lim_{k\to\infty}\mathbf{W}^{k-1}\times\mathbf{x}(1)=\frac{\mathbf{1}\times\mathbf{1}^{\mathrm{T}}}{N}\times\mathbf{x}(1)\,. \tag{11}$$

Only the existence of this limit ensures the convergence of the average consensus algorithm. As mentioned, it is achieved by using such a weight matrix that holds the conditions (5-7) [46].

As the algorithm convergences to the value in the infinite [51], it is necessary to implement a mechanism indicating the consensus. We use the mechanism defined as follows:

$$\left|\max\{\mathbf{x}(k)\}-\min\{\mathbf{x}(k)\}\right|<\delta\,. \tag{12}$$

Here, the parameter δ determines the precision. Its higher values ensure a higher precision at the cost of a slower convergence rate. In our experiments, we assume that its value is equaled to 0.0001.

3.2 Concurrent Weight Models

As mentioned, we compare our optimized weight model with other three concurrent ones. In order to distinguish these models from each other, the weight matrix of a particular model has an upper index with the abbreviated name of the model. We use the following abbreviations:

- Metropolis-Hasting weight model – MH
- Maximum Degree weight model – MD
- Best Constant weight model – BC
- Biphasically configured Metropolis-Hasting weight model – BMH

The first examined model is the Metropolis-Hasting weight model whose weight matrix is defined as follows [19], [22-25]:

$$[W^{\mathrm{MH}}]_{ij}=\begin{cases}(1+\max\{d_i,d_j\})^{-1}, & \text{if } (v_i,v_j)\in\mathbf{E}\\ 1-\sum_{k=1,k\neq i}^{N}[W^{\mathrm{MH}}]_{ik}, & \text{if } i=j\\ 0, & \text{otherwise}\end{cases} \tag{13}$$

As mentioned, our contribution optimizes this model.

The second model is the Maximum Degree weight model. It is derived from the Constant weight model in such a way that the weighting parameter ε is set to the value equaling the reciprocal of the degree of the best-connected node. Therefore, it is defined as follows [36–38]:

$$[W^{\mathrm{MD}}]_{ij}=\begin{cases}1/d_{\max}, & \text{if } (v_i,v_j)\in\mathbf{E}\\ 1-d_i/d_{\max}, & \text{if } i=j\\ 0, & \text{otherwise}\end{cases} \tag{14}$$

Within the initial configuration phase, this model requires each node in a network to be aware of the number of the neighbors of the best-connected node. In order to get this information in a distributed manner, it is necessary to implement a complementary algorithm [19].

The last examined model is called the Best Constant weight model. We assume its optimized variant even though it requires the information about the second smallest and the largest eigenvalue of the Laplacian matrix. It is defined as follows [40]:

$$[W^{\mathrm{BC}}]_{ij}=\begin{cases}2/(\mu_2(\mathbf{L})+\mu_N(\mathbf{L})), & \text{if } (v_i,v_j)\in\mathbf{E}\\ 1-2\cdot d_i/(\mu_2(\mathbf{L})+\mu_N(\mathbf{L})), & \text{if } i=j\\ 0, & \text{otherwise}\end{cases} \tag{15}$$

4. Biphasically Configured Metropolis-Hasting Weight Model

In this section, we introduce our Biphasically configured Metropolis-Hasting weight model. In Sec. 4.1, we explain the model mechanics and provide the mathematical description. In Sec. 4.2, the convergence conditions are presented.

4.1 Principle of Biphasically Configured Metropolis-Hasting Weight Model

This subsection focuses on an introduction of the main features of our optimized weight model. As mentioned above, it is called the Biphasically configured Metropolis-Hasting weight model and is derived from (as its name implies) the Metropolis-Hasting weight model. As already discussed, the Metropolis-Hasting model is appropriate for the implementation into real-life applications thanks to its simplified demands for the initial configuration. In order to correctly fulfill its functionality, each node has to be aware of the number of its neighbors as well as the number of the neighbors of the adjacent nodes. Thus, only locally available information is necessary for the correct initial configuration. There are several approaches to obtain this information (centralized one, distributed one assuming a phase when this information is distributed in the adjacent area, a manual configuration etc.). The most appropriate solution depends on a particular application.

Our optimized weight model is based on rephrasing the initial configuration process of the Metropolis-Hasting weight model into two phases. The first phase is identical to the default configuration of the Metropolis-Hasting weight model. The other phase consists of a recalculation of the weights allocated to the incoming values from the adjacent nodes. Within this phase, each node determines when to do this recalculation according to its unique identity. Thus, each node has to be additionally aware of the diagonal value of the weight matrix $\mathbf{W}^{\mathrm{BMH}}$ corresponding to all its neighbors – this value is locally available. The recalculation of $\mathbf{W}^{\mathrm{BMH}}$ has to be executed sequentially, i.e. the node with the identity number equaled to 1 initiates the whole process, updates the matrix $\mathbf{W}^{\mathrm{BMH}}$ (the active node updates its inner updating rules as well as informs its neighbors about new weights) and only then the node 2 can start the recalculation. Thus, we assume the variability of the weight matrix $\mathbf{W}^{\mathrm{BMH}}(a)$ during the second phase of the configuration process. The parameter a takes the values from $1, 2,\ldots,N$ and labels the active node (i.e. the one that is allowed to make the recalculation) as well as the round of the recalculation process (we assume that $\mathbf{W}^{\mathrm{BMH}}(0) = \mathbf{W}^{\mathrm{MH}}$ and each label of the round corresponds to the unique number of a node). Thus, we label the node currently allowed to make the update as v_a. The length of the recalculation process is determined by the size of a network (i.e. the process lasts N rounds).

Let us define the set $\mathbf{N}_a$ gathering all the nodes from the adjacent area of v_a. So, we can write as follows:

$$\mathbf{N}_a = \{v_i | \ [A]_{ai} = 1\}. \tag{16}$$

At the round when a node v_a is allowed to update $\mathbf{W}^{\mathrm{BMH}}$, it has to be aware of the current value in the diagonal corresponding to it and all its neighbors. These values represent the weights of the current inner state. Subsequently, v_a calculates the growth coefficient defined as follows:

$$\chi(v_a) = \frac{\min\{[W^{\mathrm{BMH}}(a-1)]_{aa}, \sum_{j=1}^{N}[W^{\mathrm{BMH}}(a-1)]_{jj}.[A]_{aj}\}}{\sum_{j=1}^{N}[W^{\mathrm{BMH}}(a-1)]_{jj}.[A]_{aj}}. \tag{17}$$

The choice of the minimal value from the weight of the inner state of the active node and the sum of weights of the inner states of its neighbors ensures that the growth coefficient is never greater than 1. Thus, the convergence of the weight model regardless of the underlying topology (see Sec. 4.2) is guaranteed.

Subsequently, the node currently making the recalculation decreases the weight of its inner state to the minimal possible value (the ideal scenario is when this weight is equaled to 0 after finishing this procedure) and distributes this value among the neighbors in terms of the weights of the inner states of these nodes. From the central view, this procedure is described according to the following rules:

$$[W^{\mathrm{BMH}}(a)]_{ia} = [W^{\mathrm{BMH}}(a-1)]_{ia} + \chi(v_a)\cdot[W^{\mathrm{BMH}}(a-1)]_{ii} \quad \text{for } \forall v_i \in \mathbf{N}_a. \tag{18}$$

This formula describes an increase of the weights of the incoming values of the adjacent nodes. Their values are increased with the value equaled to the diagonal value of the adjacent nodes (i.e. the weight of their inner states) weighted by the growth coefficient. The Metropolis-Hasting weight model assumes a doubly stochastic matrix (i.e. an edge e_{ij} is allocated only one weight – the incoming state of v_a and v_i is weighted with the same value) and therefore, it is necessary to preserve this condition:

$$[W^{\mathrm{BMH}}(a)]_{ai} = [W^{\mathrm{BMH}}(a-1)]_{ai} + \chi(v_a)\cdot[W^{\mathrm{BMH}}(a-1)]_{ii} \quad \text{for } \forall v_i \in \mathbf{N}_a. \tag{19}$$

After all the neighbors are allocated a new weight, the nodes with an increased weight have to decrease their diagonal value by the increase of their incoming value weight. Otherwise, the convergence conditions may not be preserved. From the central view, it is possible to describe the previous procedure using tools defined within the spectral graph theory as follows:

$$[W^{\mathrm{BMH}}(a)]_{ii} = 1 - \sum\nolimits_{j=1, j\neq i}^{N}[W^{\mathrm{BMH}}(a)]_{ij}, \quad \text{for } v_i \in \mathbf{N}_a, v_a. \tag{20}$$

After all the nodes execute the recalculation described above (this procedure is repeated for all N nodes), the weight matrix for Biphasically configured Metropolis-Hasting weight model is completed and the average consensus algorithm can be executed according to (3).

4.2 Convergence Proof

In the following subsection, we provide the sufficient conditions for the convergence of average consensus algorithm. The average consensus algorithm whose weights are symmetric can be described using a weighted graph [52]. The non-zero elements of its adjacency matrix $\mathbf{A}^{\mathrm{WG}}$ are allocated a strictly positive weight $[A^{\mathrm{WG}}]_{ij} = w_{ij}$. Subsequently, it is possible to derive the weighted Laplacian matrix as [52]:

$$\mathbf{L}^{\mathrm{WG}} = \mathrm{diag}\{\mathbf{d}^{\mathrm{WG}}\} - \mathbf{A}^{\mathrm{WG}}. \tag{21}$$

Here, the $\mathbf{d}^{\mathrm{WG}}$ is a weighted degree vector formed by the value of the degrees of the nodes. The weighted degree vector is defined as follows [52]:

$$\mathbf{d}^{\mathrm{WG}} = \mathbf{A}^{\mathrm{WG}} \times \mathbf{1}. \tag{22}$$

Firstly, we show that the weights of the Metropolis-Hasting weight model (13) ensure the convergence of the algorithm regardless of the underlying topology. Within the spectral graph theory, it is defined [52] that $\mathbf{W} = \mathbf{I} - \mathbf{L}^{\mathrm{WG}}$ and therefore, $\mathbf{W}$ is doubly stochastic with an eigenvalue with the magnitude equaled to 1 and associated to the eigenvector with the values $N^{-1/2}$. Furthermore, the matrix $\mathbf{L}^{\mathrm{WG}}$ is semidefinite and fulfills the following statement [52]:

$$\lambda_k(\mathbf{W}) = 1 - \lambda_k(\mathbf{L}^{\mathrm{WG}}) \quad \text{for } \forall k. \tag{23}$$

Thus, the convergence condition (7) can be reformulated as follows [52]:

- $\rho(\mathbf{L}^{\mathrm{WG}}) < 2$,
- Multiplicity of the zero eigenvalue of $\mathbf{L}^{\mathrm{WG}}$ has multiplicity one.

The second statement is satisfied for all the connected graphs. It can be confirmed using the quadratic form defined as follows [52]:

$$\mathbf{v}^{\mathrm{T}} \times \mathbf{L}^{\mathrm{WG}} \times \mathbf{v} = \sum_{e_{ij}\in\mathbf{E}} w_{ij}\cdot(v_i - v_j)^2. \tag{24}$$

The vector **v** is an eigenvector associated to the weighted Laplacian matrix $\mathbf{L}^{\mathrm{WG}}$. As seen, the quadratic form is equaled to the zero value if and only if $v_i = v_j$. This statement is valid for the weights of the positive values. This requirement is met by the unique normalized vector $\mathbf{v} = N^{-1/2}$. Let us focus on the first constraint. As (21) and (22) hold, the eigenvalues λ^{WG} satisfy the following condition according to Gershgorin circle theorem, which is defined as follows [53]:

$$\left|\lambda(\mathbf{L}^{\mathrm{WG}}) - d_i^{\mathrm{WG}}\right| \leq d_i^{\mathbf{WG}}. \tag{25}$$

The parameter d_i^{WG} is the weighted degree of the node i defined as:

$$d_i^{\mathrm{WG}} = \sum_{j=1}^{N} w_{ij}. \tag{26}$$

The value also presents the ith row of the vector $\mathbf{d}_i^{\mathrm{WG}}$. In particular, the following statement is valid [52]:

$$\lambda(\mathbf{L}^{\mathrm{WG}}) \leq 2 \cdot d_i^{\mathrm{WG}}. \tag{27}$$

Thus, $d_i^{\mathrm{WG}} \leq 1$ is a sufficient condition for the average consensus algorithm to convergence for all i [52] (when a graph is neither bipartite nor regular). Since it is improbable that the graph describing a WSN is bipartite regular [52], we do not deal with these critical graph topologies. From (13), it is clear that this condition holds for each topology. Thanks to the expression in the numerator in (17), our recalculation always ensures $d_i^{\mathrm{WG}} \leq 1$. It secures that the sum in both the rows and the columns does not change despite the recalculation. This guarantees the convergence conditions for our mechanism.

Let us analyze the functionality of our optimized weight model. The minimal value from two parameters in the numerator of (17) is chosen to preserve the convergence conditions. The value of $[\boldsymbol{W}^{\mathrm{BMH}}(a-1)]_{aa}$ poses the maximal possible value with which $\boldsymbol{v}_a$ can decrease the weight of its inner state and distribute it among its adjacent nodes. A decrease with a value greater than $[\boldsymbol{W}^{\mathrm{BMH}}(a-1)]_{aa}$ always causes $d_i^{\mathrm{WG}} \leq 1$ not to hold. Thus, in such a scenario, the convergence conditions are not fulfilled. Regarding all the positive values smaller than $[\boldsymbol{W}^{\mathrm{BMH}}(a-1)]_{aa}$: the convergence conditions are preserved but the optimization is less significant. Now, let us focus on $\Sigma_j[\boldsymbol{W}^{\mathrm{BMH}}(a-1)_{jj}\cdot[\boldsymbol{A}]_{aj}$. There can be a scenario when $\boldsymbol{v}_a$ can increase the weights of the incoming values of its adjacent nodes with a value that causes some of its neighbors to achieve a negative value in the diagonal after (20). In this scenario, the growth coefficient χ is greater than 1, which results in $d_i^{\mathrm{WG}} > 1$. Therefore, parameter $\Sigma_j[\boldsymbol{W}^{\mathrm{BMH}}(a-1)_{jj}\cdot[\boldsymbol{A}]_{aj}$ ensures that the growth coefficient never exceed the value 1 and so, the convergence conditions are always preserved. The procedure in (20) must be executed in order to keep the weight matrix **W** doubly-stochastic. This step secures that (5) and (6) always hold.

5. Numerical Experiments and Discussion

In this section, we present the results of the numerical experiments executed in Matlab R2015a. All the used software was designed by the authors of this paper. In our experiments, three sets of networks are assumed with randomly generated topologies. We assume weakly, averagely and strongly connected networks. Each set consists of 10 unique topologies with the size of 200 nodes. Due to the limited range of the paper, only one representative of these sets is shown in Fig. 1, Fig. 2 and Fig. 3 respectively. The networks were generated as follows: each free position within the working area of a square shape was allocated the probability equaled to the reciprocal of the number of free positions. Thus, the choice of the position where a node was placed had a uniform distribution. Subsequently, the nodes situated in the transmission range of each node were labeled as its neighbors (i.e. there is an edge between them). In order to ensure a various average connectivity, the transmission range was changed.

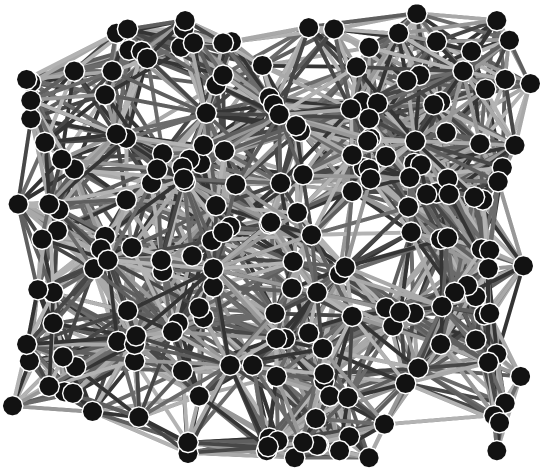

Fig. 1. Example representative of strongly connected topologies

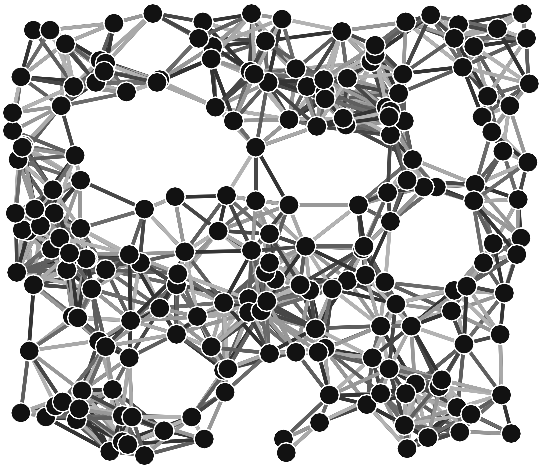

Fig. 2. Example representative of averagely connected topologies

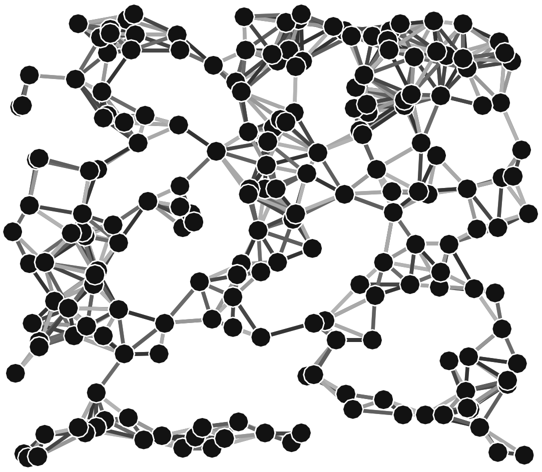

Fig. 3. Example representative of weakly connected topologies.

In order to evaluate the achieved optimization, we compare our mechanism with other discussed weight models. We compare it with the Metropolis-Hasting weight model, the Maximum-degree weight model, and the Best Constant weight model. These models were chosen because all of them are classified as constant weight models [25], frequently-used and discussed in other papers and optimize average consensus with a similar principle as our mechanism.

5.1 Estimation of Average Value

Within the first experiment, we examine the convergence rates and the number of the required messages. In this section, we draw our attention to the average estimation, compare our mechanism with the concurrent weight models and show the difference in the number of the messages that are necessary for the average consensus algorithm to achieve the consensus for each model.

As the initial configuration of our optimized weight model depends on the placement of the identity numbers, we use *randperm*, which is a built-in function in Matlab to generate a vector with a random position of the numbers. Therefore, in order to ensure a higher credibility of our conclusions, our mechanism was executed 100 times (each execution is characterized by a shuffled set of the identity numbers) in each topology.

In Tab. 1, our optimized weight model is compared with three other examined weight models. A positive value means an improvement ensured by our mechanism, meanwhile, a negative value indicates that a concurrent model achieves a faster convergence rate. We label these scenarios as positive, respectively, negative optimization. In Tab. 1, we have shown the average (calculated as the average value of the optimization of all ten networks within one set of the networks), the maximal (the optimization achieved in the network where the algorithm is optimized most significantly) and the minimal optimization (the optimization achieved in the network where the algorithm is optimized worse or even negatively) as well as the range of the optimization (all are expressed in %). Within this comparison, the average calculated from 100 convergence rates obtained within these executions is chosen as a representative of 100 executions of the Biphasically configured Metropolis-Hasting weight model. The complete results are shown in Appendix A. The column labeled as CR contains the convergence rates expressed as the number of the iterations. The column OPT [%] is formed by the relative optimizations [%] of our mechanism compared with the concurrent weight models.

We can see from the results that our optimized weight model achieves a faster average convergence rate compared with all the concurrent models in all the sets of the networks. However, the Best Constant weight model achieves a faster rate in one of ten strongly connected topologies. The optimization is the most significant compared with the Maximum Degree weight model. Its average value ranges from 40.08% to 54.60%. The optimization of the Metropolis-Hasting model is from the range 8.19% to 15.45%. In this case, the most important fact is that our optimized weight model achieves a faster convergence rate in all the networks. The Best Constant weight model is optimized in the range 14.24% to 23.25%. Furthermore, we can see that (for all the weight models) the less connected the networks are, the higher average optimization our mechanism ensures. Let us focus on the range of the optimization (calculated as the difference between the maximal and the minimal optimization within one set). We can observe that this parameter achieves the highest value for the Best Constant weight model in all the sets. In Appendix A, we have also shown the number of the messages necessary for the average consensus algorithm to be completed. Since this parameter is closely related to the convergence rate, we do not provide a separate analysis.

Consequently, we analyze the effect of a random shuffle of the identity numbers on the convergence rates.

Convergence rate optimization of average estimation [%] in Weakly connected networks			
	MD	MH	BC
Average	54.60 %	15.45 %	23.25 %
Maximum	64.60 %	21.28 %	35.95 %
Minimum	47.38 %	12.65 %	8.77 %
Range	17.22 %	8.63 %	27.18 %
Convergence rate optimization of average estimation [%] in Averagely connected networks			
	MD	MH	BC
Average	46.84 %	11.15 %	15.36 %
Maximum	60.04 %	15.64 %	36.21 %
Minimum	39.71 %	7.80 %	1.42 %
Range	20.33 %	7.84 %	34.79 %
Convergence rate optimization of average estimation [%] in Strongly connected networks			
	MD	MH	BC
Average	40.08 %	8.19 %	14.24 %
Maximum	69.30 %	16.93 %	49.29 %
Minimum	31.47 %	4.29 %	-3.42 %
Range	37.83 %	12.64 %	52.62 %

Tab. 1. Comparison of BMH with others – average est.

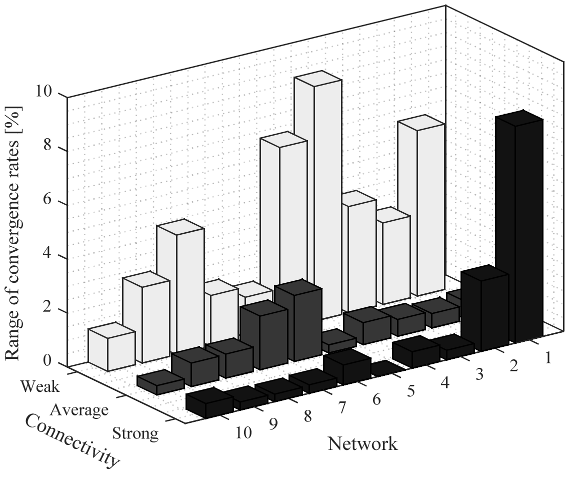

Fig. 4. Percentage range of convergence rates caused by shuffle of identity numbers (average est.).

The position of the identity numbers can affect only our optimized weight model – the other ones do not assume this value during the configuration process. In Fig. 4, we have depicted the range of the convergence rates calculated from the convergence rates obtained within 100 reparations (we depict the value of the ratio: the range/the average convergence rate expressed in % because the convergence rates differ from each other in different topologies).

We can see that the range does not exceed 9 % of the average value in any case. This primarily affects the convergence rate in the weakly connected topologies. In the averagely and the strongly connected networks, its impact is negligible except for one topology of a strong connectivity.

In the following part, we examine whether this shuffle can cause that there is an execution whose convergence rate is slower than the convergence rate of one of the concurrent models. Thus, we depict the slowest scenario within the Biphasically configured Metropolis-Hasting weight model with the fastest concurrent model in order to show that a shuffle does not cause a negative optimization when the average optimization is positive. The mentioned comparison for each set of the networks is depicted in Fig. 5 (weakly connected networks), Fig. 6 (averagely connected networks) and Fig. 7 (strongly connected networks).

The white column represents the slowest convergence rate within the Biphasically configured Metropolis-Hasting weight model, while, the black one represents the fastest concurrent weight model.

We can see that a positive optimization is preserved in all the cases when a positive average optimization is. Thus, a random allocation of the identity numbers has only a minimal impact on the convergence rate.

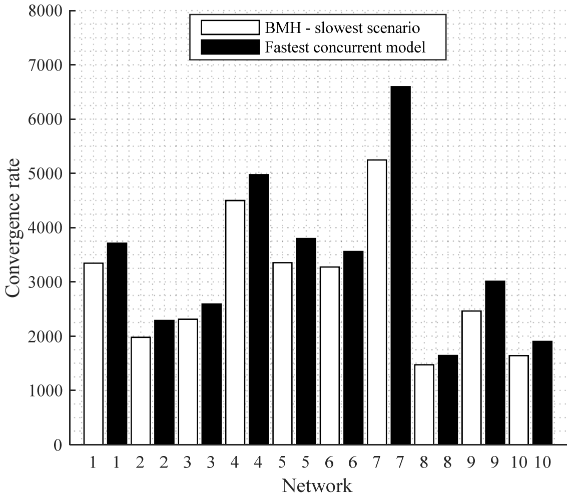

Fig. 5. Comparison of slowest scenario of BMH with fastest concurrent model – weak connectivity – average est.

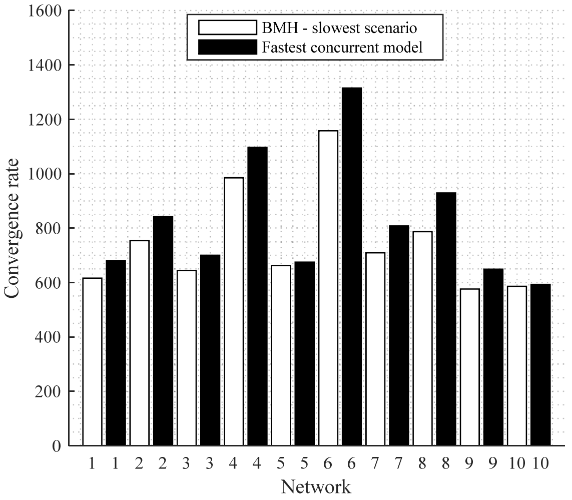

Fig. 6. Comparison of slowest scenario of BMH with fastest concurrent model – average connectivity – average est.

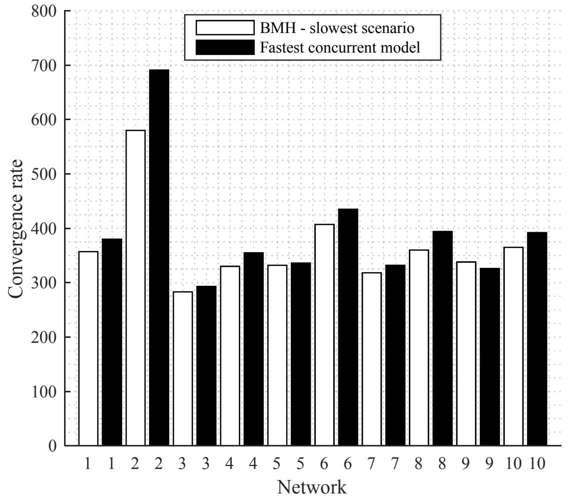

Fig. 7. Comparison of slowest scenario of BMH with fastest concurrent model – strong connectivity – average est.

5.2 Estimation of the Network Size: Convergence Rate Optimization

The second experiment is focused on the network size estimation. We examine the optimization ensured by our optimized weight model in the same topologies used in the previous experiment. In the first part of this section, an examination of the convergence rate optimization achieved by our optimized weight model is presented. As discussed in Sec. 1, one of the aspects by which the network size estimation differs from the average estimation is the necessity of the choice of the leader. Thus, in order to ensure a generality of the simulation results, the average consensus was repeated 200 times (i.e. executions use different leaders – 200 times = 200 leaders). Consequently, the convergence rates of the Maximum Degree, the Metropolis-Hasting and the Best Constant were examined. Within our first analysis, we choose the average of these values as a representative of the convergence rates. When the Biphasically configured Metropolis-Hasting was examined, the experiment was repeated 200 times for each shuffle. Therefore, within one topology, we made $200 \cdot 100 = 20\,000$ executions. Here, the average was again chosen as a representative. In Tab. 2, we have shown the optimization of our optimized weight model compared with the concurrent ones. In Appendix B, the complete results are depicted. As in the previous experiment, also the number of the messages is depicted in Appendix B. We can see from the results that our mechanism again achieves a positive average optimization in all the cases. However, the Best Constant achieves a faster convergence rate in two strongly connected networks. The Maximum Degree (the averages are in the range 41.80% – 55.05%) and the Metropolis-Hasting (within the range 7.87% – 16.62%) are optimized similarly as in the previous experiment (in the strongly connected networks, the MD achieves a small positive deviation) and with the same character compared with the average estimation. Like in the first experiment, the Metropolis-Hasting weight model is positively optimized for each network. Regarding the Best Constant weight model, the optimization is not as significant for the averagely and strongly connected networks as in the previous experiment (8.19% – 23.50%). The worst average results are obtained for the averagely connected networks in contrast to the first experiment, where a higher connectivity results in a higher optimization. Like in the first experiment, the optimization range is the widest for this model.

In the next part, we examine the impact of a random shuffle on the range of the convergence rates like in the first experiment. The character of this phenomenon is similar to the one from the previous experiment but the values for some networks are higher (see Fig. 8).

Furthermore, in order to show that there is no slower rate (when the average positive optimization is achieved) than the fastest concurrent model, Fig. 9 (weak connectivity), Fig. 10 (average connectivity) and Fig. 11 (strong connectivity) are shown.

Convergence rate optimization of network size estimation [%] in Weakly connected networks			
	MD	MH	BC
Average	55.05	16.62	23.50
Maximum	64.46	20.86	36.47
Minimum	47.75	13.95	13.94
Range	16.71	6.91	22.53
Convergence rate optimization of network size estimation [%] in Averagely **connected networks**			
	MD	MH	BC
Average	45.99	10.56	8.19
Maximum	60.00	13.21	32.14
Minimum	36.90	8.29	0.78
Range	23.10	4.92	31.36
Convergence rate optimization of network size estimation [%] in Strongly connected networks			
	MD	MH	BC
Average	41.80	7.87	9.23
Maximum	68.16	15.21	47.41
Minimum	30.76	5.25	-11.10
Range	37.40	9.96	58.51

Tab. 2. Comparison of BMH with others – network size est.

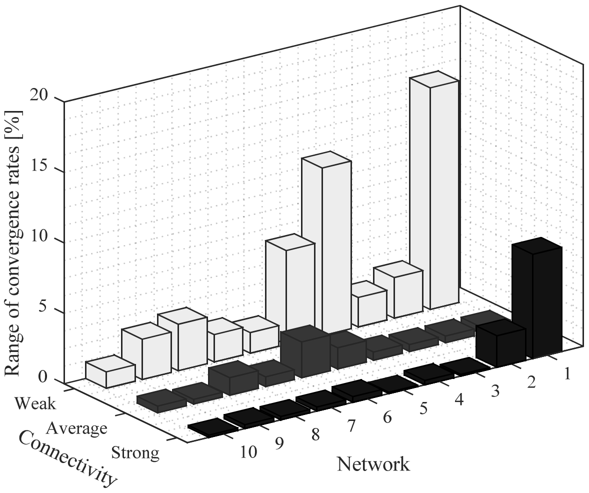

Fig. 8. Percentage range of convergence rates caused by shuffle of identity numbers (network size est.).

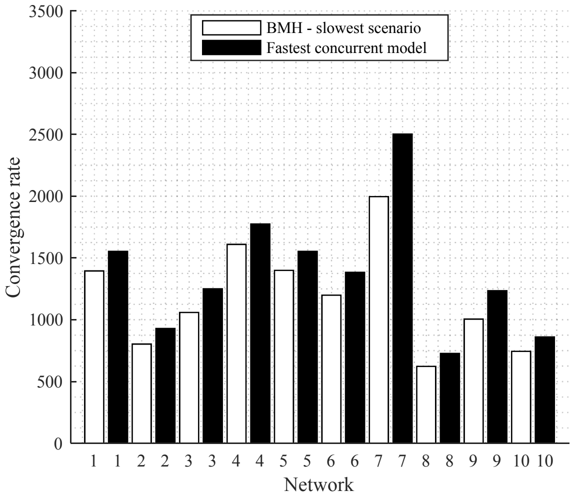

Fig. 9. Comparison of slowest scenario of BMH with fastest concurrent model – weak connectivity – size est.

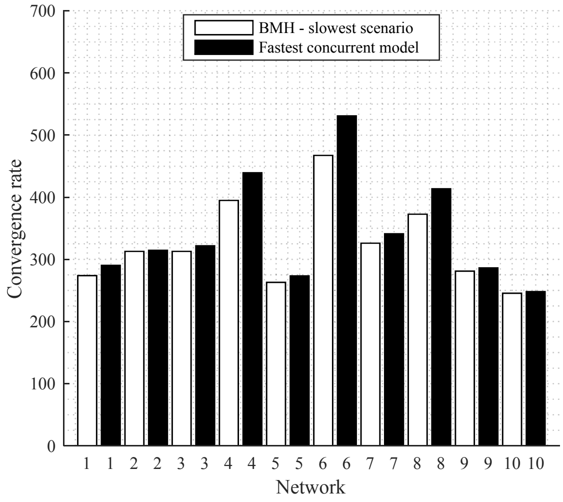

Fig. 10. Comparison of slowest scenario of BMH with fastest concurrent model – average connectivity – size est.

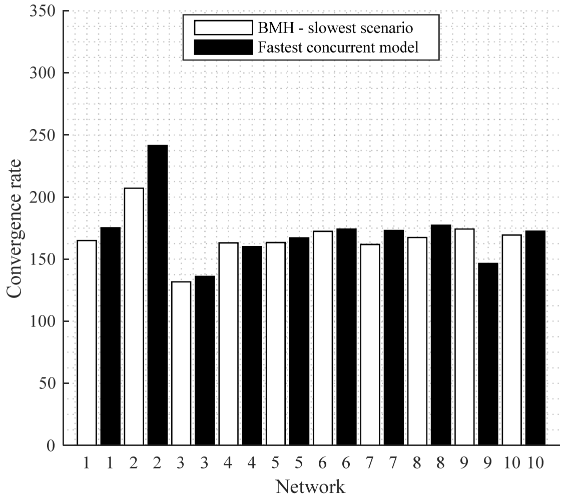

Fig. 11. Comparison of slowest scenario of BMH with fastest concurrent model – strong connectivity – size est.

5.3 Estimation of the Network Size: Impact of the Leader Choice

In the last experiment, it is examined how the choice of the leader affects the convergence rates of the algorithm. We choose the range of the obtained convergence rates as a quality indicator of this aspect. The ideal scenario would be the range equaled to 0. However, this state is unreachable for complicated structures and therefore, we compare the value of the range of our optimized weight model with the concurrent weight models. Its smaller value means that the choice of the leader has a smaller impact on the convergence rate – which is the desired outcome. The experiment is again executed on our optimized weight model, the Maximum Degree weight model, the Metropolis-Hasting weight model, and the Best Constant weight model.

In Tab. 3 and Appendix C, we show the optimization of our mechanism for all the topologies. We can see that our optimized weight model achieves a positive optimization for all 30 networks.

Convergence rate optimization of network size estimation [%] in Weakly connected networks			
	MD	MH	BC
Average	53.08 %	15.41 %	52.02 %
Maximum	65.26 %	28.55 %	63.80 %
Minimum	38.19 %	4.84 %	37.66 %
Range	27.07 %	23.71 %	26.14 %
Convergence rate optimization of network size estimation [%] in Averagely connected networks			
	MD	MH	BC
Average	45.61 %	8.51 %	46.80 %
Maximum	68.08 %	16.28 %	66.57 %
Minimum	25.21 %	4.00 %	15.87 %
Range	42.87 %	12.28 %	50.70 %
Convergence rate optimization of network size estimation [%] in Strongly connected networks			
	MD	MH	BC
Average	52.55 %	11.92 %	51.36 %
Maximum	83.19 %	30.79 %	66.74 %
Minimum	22.72 %	1.64 %	19.47 %
Range	60.47 %	29.15 %	47.27 %

Tab. 3. Comparison of BMH with others – choice of the leader.

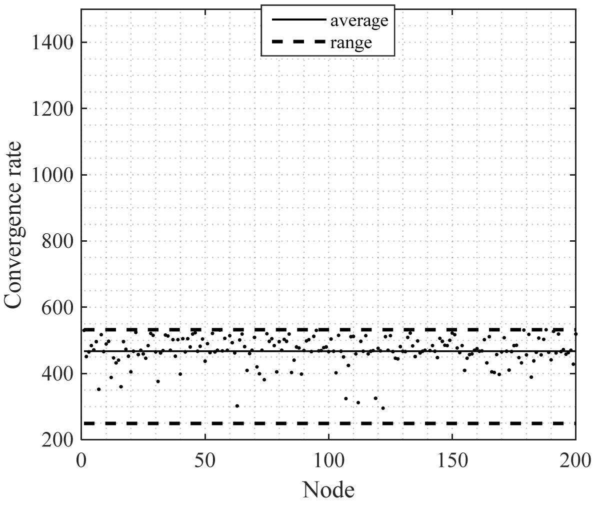

Fig. 12. Biphasically configured Metropolis-Hasting weight model – range of convergence rates.

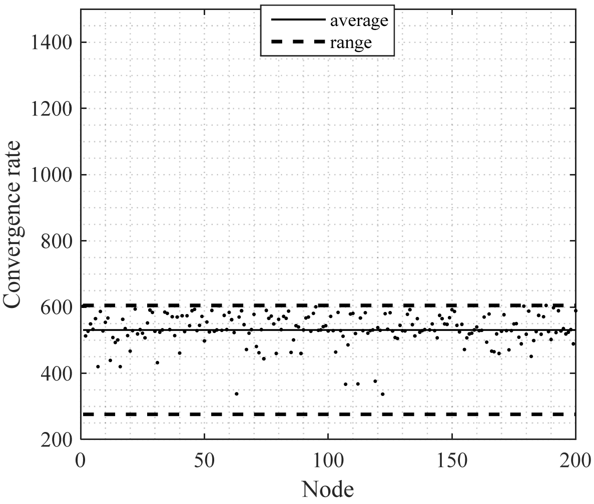

Fig. 13. Metropolis-Hasting weight model – range of convergence rates.

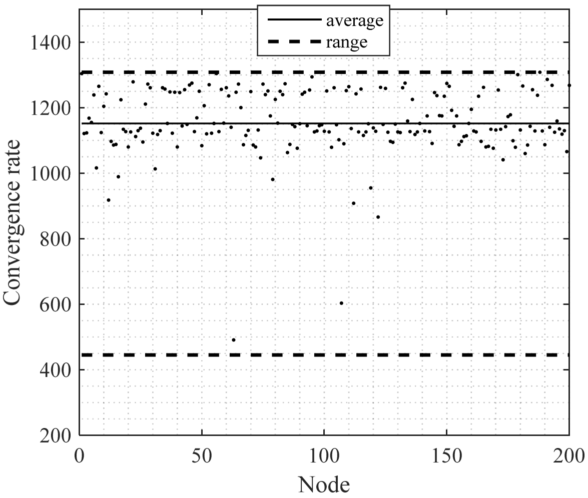

Fig. 14. Maximum Degree weight model – range of convergence rates.

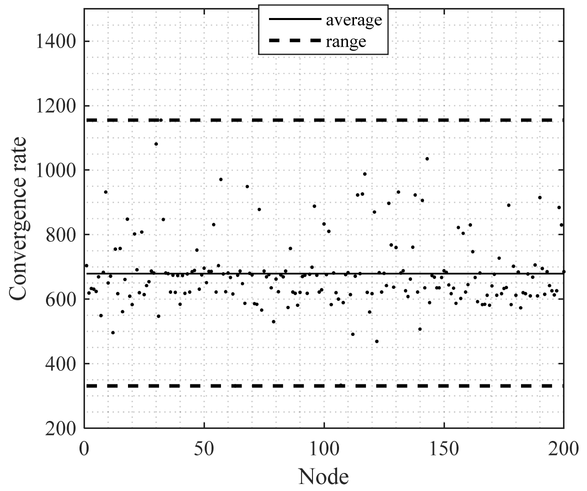

Fig. 15. Best Constant weight model – range of convergence rates.

In Fig. 12 (Biphasically configured Metropolis-Hasting weight model), Fig. 13 (Metropolis-Hasting weight model), Fig. 14 (Maximum Degree weight model) and Fig. 15 (Best Constant weight model), we have shown the results of one of the examined topologies in order to illustrate this problem. We have depicted the convergence rates for each node and highlighted the range calculated as the difference between the slowest and the fastest convergence rate. We can see from the figures that our optimized weight model achieves the smallest range and therefore, it is the best also in this aspect for the examined topology. Regarding the results from 30 networks, the best optimization is achieved in the weakly connected networks, meanwhile, the worst one is observed in the network of the average connectivity regardless of the examined model. The Maximum Degree weight model is optimized in the range (45.61% – 53.08%), the Metropolis-Hasting weight model in the range (8.51% – 15.41%), and the Best Constant weight model in (46.80% – 52.02%). Thus, as seen from the results of all the executed experiments, the best optimization within all three optimized aspects is achieved for the optimization of the impact of the choice of the leader.

6. Conclusion

In this paper, we present an optimized version of the Metropolis-Hasting weight model called the Biphasically configured Metropolis-Hasting weight model. We examined the achieved optimization compared with other constant concurrent weight models (Maximum degree weight model, Metropolis-Hasting weight model, Best Constant weight model). In our analysis, we focused on the estimation of the average value, the estimation of the network size and the impact of the choice of the leader. The improvement ensured by our optimized weight model was demonstrated in randomly generated networks with a weak, an average, and a strong connectivity. According to the depicted results, the optimization achieved by our optimized weight model poses a significant improvement of the computation process of the average consensus algorithm.

Acknowledgments

The research described in this paper was financed by the Czech Ministry of Education in frame of the National Sustainability Program under grant LO1401. For research, infrastructure of the SIX Center was used.

References

[1] BUTUN, I., MORGERA, S. D., SANKAR, R. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Survey & Tutorials,* 2014, vol. 16, no. 1, p. 266–282. DOI: 10.1109/SURV.2013.050113.00191

[2] BAHREPOUR, M., MERATNIA, N., POEL, M., et al. Distributed event detection in wireless sensor networks for disaster management. In *Proceeding of the 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS).* Thessaloniki (Greece), 2010, p. 507–512. DOI: 10.1109/INCOS.2010.24

[3] OTHMAN, M. F., SHAZALI, K. Wireless sensor network applications: A study in environment monitoring system. *Procedia Engineering,* 2012, vol. 41, p. 1204–1210. DOI: 10.1016/j.proeng.2012.07.302

[4] YICK, J., MUKHERJEE, B., GHOSAL, D. Wireless sensor network survey. *Computer Networks,* 2008, vol. 52, no. 12, p. 2292–2330. DOI: 10.1016/j.comnet.2008.04.002

[5] SHERLY PUSPHA, L., MURUGAN, K. Energy-efficient quorum-based MAC protocol for wireless sensor networks. *ETRI Journal,* 2015, vol. 37, no. 3, p. 480–490. DOI: 10.4218/etrij.15.0114.0688

[6] HATAMIAN, M., ALMASI BARDMILY, M., ASADBOLAND, M., el al. Congestion-aware Routing and fuzzy-based rate controller for wireless sensor networks. *Radioengineering,* 2016, vol. 25, no. 1, p. 114–123. DOI: 10.13164/re.2016.0114

[7] XIE, L., SHI, Y., HOU, Y.T., et al. Wireless power transfer and applications to sensor networks. *IEEE Wireless Communications,* 2013, vol. 20, no. 4, p. 140–145. DOI: 10.1109/MWC.2013.6590061

[8] KUILA, P., JANA, P. K. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm

optimization approach. *Engineering Applications of Artificial Intelligence,* 2014, vol. 33, p. 127–140. DOI: 10.1016/j.engappai.2014.04.009

[9] RAULT, T., BOUABDALLAH, A., CHALLAL, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks,* 2014, vol. 67, p. 104–122. DOI: 10.1016/j.comnet.2014.03.027

[10] VISSER, H. J., VULLERS, R. J. RF energy harvesting and transport for wireless sensor network applications: Principles and requirements. *Proceedings of the IEEE,* 2013, vol. 101, no. 6, p. 1410–1423. DOI: 10.1109/JPROC.2013.2250891

[11] SHELTAMI, T. An enhanced energy saving approach for WSNs. *Procedia Computer Science,* 2013, vol. 21, p. 199–206. DOI: 10.1016/j.procs.2013.09.027

[12] KARABOGA, D., OKDEM, S., OZTURK, C. Cluster based wireless sensor network routing using artificial bee colony algorithm. *Wireless Networks,* 2012, vol. 18, no. 7, p. 847–860. DOI: 10.1007/s11276-012-0438-z

[13] KAR, S., TANDON, R., POOR, H. V., et al. Distributed detection in noisy sensor networks. In *Proceeding of the IEEE International Symposium on Information Theory Proceedings (ISIT).* St. Petersburg (Russia), 2011, p. 2856–2860. DOI: 10.1109/ISIT.2011.6034097

[14] KEMPE, D., DOBRA, A., GEHRKE, J. Gossip-based computation of aggregate information. In *Proceeding of the 44th Annual IEEE Symposium on Foundations of Computer Science.* Cambridge (Massachusetts, USA), 2003, p. 482–491. DOI: 10.1109/SFCS.2003.1238221

[15] BRACA, P., MARANO, S., MATTA, V. Enforcing consensus while monitoring the environment in wireless sensor networks. *IEEE Transactions on Signal Processing,* 2008, vol. 56, no. 7, p. 3375–3380. DOI: 10.1109/TSP.2008.917855

[16] SLUCIAK, O., RUPP, M. Reaching consensus in asynchronous WSNs: Algebraic approach. In *Proceeding of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP).* Prague (Czech Republic), 2011, p. 3300 to 3303. DOI: 10.1109/ICASSP.2011.5946727

[17] XIONG, G., KISHORE, S. Linear high-order distributed average consensus algorithm in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing,* 2010, vol. 1, no. 1, p. 1–6. DOI: 10.1155/2010/373604

[18] JAFARIZADEH, S., JAMALIPOUR, A. Weight optimization for distributed average consensus algorithm in symmetric, CCS & KCS star networks. *arXiv preprint arXiv:1001.4278.*

[19] SLUCIAK, O., RUPP, M. Network size estimation using distributed orthogonalization. *IEEE Signal Processing Letters,* 2013, vol. 20, no. 4, p. 347–350. DOI: 10.1109/lsp.2013.2247756

[20] OLFATI-SABER, R., FAX, J. A., MURRAY, R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE,* 2007, vol. 95, no. 1, p. 215–233. DOI: 10.1109/JPROC.2006.887293

[21] SHAMES, I., CHARALAMBOUS, T., HADJICOSTIS, C.N., et al. Distributed network size estimation and average degree estimation and control in networks isomorphic to directed graphs. In *Proceeding of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton).* Monticello (Illinois, USA), 2012, p. 1885–1892. DOI: 10.1109/Allerton.2012.6483452

[22] BIANCHIN, G., CENEDESE, A., LUVISOTTO, M., et al. Distributed faults detection in sensor networks via clustering and consensus. In *Proceeding of the IEEE 54th Conference on Decision and Control (CDC).* Osaka (Japan), 2015, p. 3828–3833. DOI: 10.1109/cdc.2015.7402814

[23] KIM, B. Y., HUR, H., AHN, H. S. Coordination and control for freeway traffic network. In *Proceeding of the IEEE International Systems Conference (SysCon).* Orlando (Florida, USA), 2013, p. 159–163. DOI: 10.1109/syscon.2013.6549875

[24] ZENG, F., LI, C., TIAN, Z. Distributed compressive spectrum sensing in cooperative multihop cognitive networks. *IEEE Journal of Selected Topics in Signal Processing,* 2011, vol. 5, no. 1, p. 37 to 48. DOI: 10.1109/vtcfall.2013.6692163

[25] TAHBAZ-SALEHI, A., JADBABAIE, A. A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times. In *45th IEEE Conference on Decision and Control.* San Diego (CA, USA), 2006, p. 4664–4669. DOI: 10.1109/cdc.2006.377308

[26] CHIB, S., GREENBERG, E. Understanding the Metropolis-Hastings algorithm. *The American Statistician,* 1995, vol. 49, no. 4, p. 327–335. DOI: 10.2307/2684568

[27] KIA, S. S., CORTES, J., MARTINEZ, S. Distributed event-triggered communication for dynamic average consensus in networked systems. *Automatica,* 2015, vol. 59, p. 112–119. DOI: 10.1016/j.automatica.2015.06.011

[28] WANG, X., LI, J., XING, J., et al. A novel finite-time average consensus protocol for multi-agent systems with switching topology. *Transaction of the Institute of Measurement and Control,* 2016, p. 1–9. DOI: 10.1177/0142331216663617

[29] KENYERES, M., KENYERES, J., SKORPIL, V. Split distributed computing in wireless sensor networks. *Radioengineering*, 2015, vol. 24, no. 3, p. 749–756. DOI: 10.13164/re.2015.0749

[30] KIA, S. S., CORTES, J., MARTINEZ, S. Dynamic average consensus under limited control authority and privacy requirements. *International Journal of Robust and Nonlinear Control*, 2015, vol. 25, no. 13. p. 1941–1966. DOI: 10.1002/rnc.3178

[31] PRIOLO, A., GASPARRI, A., MONTIJANO, E., et al. A distributed algorithm for average consensus on strongly connected weighted digraphs. *Automatica,* 2014, vol. 50, no. 3, p. 946–951. DOI: 10.1016/j.automatica.2013.12.026

[32] YUAN, D., XU, S., ZHAO, H., et al. Accelerating distributed average consensus by exploring the information of second-order neighbors. *Physics Letters A,* 2010, vol. 374, no. 24. p. 2438–2445. DOI: 10.1016/j.physleta.2010.03.053

[33] AYSAL, T. C., ORESHKIN, B. N., COATES, M. J. Accelerated distributed average consensus via localized node state prediction. *IEEE Transactions on Signal Processing,* 2009, vol. 57, no. 3. p. 1563–1576. DOI: 10.1109/TSP.2008.2010376

[34] LIU, W., GU, W., XU, Y., et al. Improved average consensus algorithm based distributed cost optimization for loading shedding of autonomous microgrids. *International Journal of Electrical Power & Energy Systems,* 2015, vol. 73, p. 89–96. DOI: 10.1016/j.ijepes.2015.04.006

[35] CHEN, Y., TRON, R., TERZIS, A., et al. Corrective consensus: Converging to the exact average. In *Proceeding of the 49th IEEE Conference on Decision and Control (CDC).* Atlanta (Georgia, USA), 2010, p. 1221–1228. DOI: 10.1109/CDC.2010.5717925

[36] LI, W., JIA, Y. Consensus-based distributed multiple model UKF for jump Markov nonlinear systems. *IEEE Transactions on Automatic Control,* 2012, vol. 57, no. 1, p. 227–233. DOI: 10.1109/TAC.2011.2161838

[37] LI, Z., YU, F.R., HUANG, M. A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios. *IEEE Transactions on Vehicular Technology,* 2010, vol. 59, no. 1, p. 383–393. DOI: 10.1109/TVT.2009.2031181

[38] FRASCA, P., CARLI, R., FAGNANI, F., et al. Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control,* 2009, vol. 19, no. 16, p. 1797 to 1816. DOI: 10.1109/TVT.2009.2031181

[39] KENYERES, M., KENYERES, J., SKORPIL, V. The distributed convergence classifier using the finite difference. *Radioengineering,* 2016, vol. 25, no. 1, p. 148–155. DOI: 10.13164/re.2016.0148

[40] TOULOUSE, M., MINH, B. Q., CURTIS, P. A consensus based network intrusion detection system. In *Proceeding of the 5th International Conference on IT Convergence and Security (ICITCS).* Kuala Lumpur (Malaysia), 2015, p. 1–6. DOI: 10.1109/ICITCS.2015.7292913

[41] BENJAMIN, A., CHARTRAND, G., ZHANG, P. *The Fascinating World of Graph Theory.* Princeton (NJ, USA): Princeton University Press, 2015. ISBN: 9780691163819

[42] SCHWARZ, V., MATZ, G. Average consensus in wireless sensor networks: Will it blend? In *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Vancouver (British Columbia, Canada), 2013, p. 4584–4588. DOI: 10.1109/ICASSP.2013.6638528

[43] ANDERSON, W. N., MORLEY, T. D. Eigenvalues of the Laplacian of a graph. *Linear and Multilinear Algebra,* vol. 18, no. 2, 1985, p. 141–145. DOI: 10.1080/03081088508817681

[44] SPIELMAN, D. A. *The Laplacian (Lecture 2).* 6 pages. [Online] Cited 2016-10-20. Available at: http://www.cs.yale.edu/homes/spielman/561/2009/lect02-09.pdf

[45] XIAO, L., BOYD, S., KIM, S. J. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing,* 2007, vol. 67, no. 1. p. 33–46. DOI: 10.1016/j.jpdc.2006.08.010

[46] XIAO, L., BOYD, S. Fast linear iterations for distributed averaging. *Systems & Control Letters,* 2004, vol. 53, no. 1, p. 65 to 78. DOI: 10.1016/j.sysconle.2004.02.022

[47] MACUA, S. V., LEON, C. M., ROMERO, J. S., et al. How to implement doubly-stochastic matrices for consensus-based distributed algorithms. In *Proceeding of the IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM).* A Coruña, (Spain), 2014, p. 333–336. DOI: 10.1109/SAM.2014.6882409

[48] BAPAT, R. B. *Graphs and Matrices.* New York (NY): Springer, 2010. DOI: 10.1007/978-1-84882-981-7

[49] LI, Z., REN, W., LIU, X., et al. Consensus of multi-agent systems with general linear and Lipschitz nonlinear dynamics using distributed adaptive protocols. *IEEE Transactions on Automatic Control,* 2013, vol. 58, no. 7, p. 1786–1791. DOI: 10.1109/TAC.2012.2235715

[50] NOWZARI, C., CORTÉS, J. Zeno-free, distributed event-triggered communication and control for multi-agent average consensus. In *Proceeding of the American Control Conference (ACC).* Portland (Oregon, USA), 2014, p. 2148–2153. DOI: 10.1109/ACC.2014.6859495

[51] AL-NAKHALA, N., RILEY, R. ELFOULY, T. Distributed algorithms in wireless sensor networks: An approach for applying binary consensus in a real testbed. *Computer Networks,* vol. 79, p. 30–38. DOI: 10.1016/j.comnet.2014.12.011

[52] SCHWARZ, V., HANNAK, G., MATZ, G. On the convergence of the average consensus with generalized Metropolis-Hasting weights. In *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Florence (Italy), 2014, p. 5442–5446, DOI: 10.1109/icassp.2014.6854643

[53] VARGA, R. S. *Gersgorin and His Circles.* Berlin (GE): Springer-Verlag, 2004. DOI: 10.1007/978-3-642-17798-9

About the Authors ...

Martin KENYERES was born in Bratislava, Slovakia. He received his M.Sc. from the Slovak University of Technology in Bratislava in 2013. His research interests include distributed computing and wireless sensor networks. In 2013, he was with the Vienna University of Technology, Austria, where he participated in NFN SISE project under Professor Markus Rupp's supervision. He dealt with the implementation of distributed algorithms for an estimation of aggregate functions into wireless sensor networks. Since 2014, he has been with Brno University of Technology (BUT), where he works towards his PhD thesis on an analysis and an optimization of distributed systems.

Jozef KENYERES was born in Bratislava, Slovakia. He received his Ph.D. from the Slovak University of Technology in Bratislava in 2014. His research interests include embedded systems, wireless sensor networks and VoIP. From 2006 to 2009, he worked as a technician at Slovak Telecom, a. s., Bratislava, Slovakia. From 2009 to 2013, he was a project assistant at the Vienna University of Technology, Austria and from 2014 to 2015 he was with Zelisko GmbH, where he worked as a software developer. Since 2015, he has been working as a software developer at Sipwise GmbH, Austria.

Vladislav ŠKORPIL was born in Brno, Czech Republic. He graduated from the BUT, Faculty of Electrical Engineering, Dept. of Telecommunications in 1980. From 1985 to 1989 he was a doctoral student in the same Department. From 1980 to 1982 he worked as a designer for the telecommunication design office. He again entered the Dept. of Telecommunications, BUT in 1982 as a university teacher and he has been working there since that time (1984 Associate Professor). From 1994 to 2013, he was a vice-head of this department. He takes a keen interest in modern telecommunication systems. He has taught in courses on transmission systems from analogue through all categories of digital up to special applications. He is the author of more than 100 international scientific papers and some manuals. He has complemented his theoretical knowledge by co-operation with a lot of firms and institutions. He has co-operated on telecommunication projects such as digital transmission and switching systems, telecommunication broadband networks, ISDN, ATM, data networks LAN and MAN, on structured cabling design, neural networks, wavelet transform, Quality of Service QoS, data bit rate compression, etc. He is a member of international organizations IEEE and WSEAS.

Radim BURGET is an Associated Professor (2014) at the Dept. of Telecommunications, Faculty of Electrical Engineering, BUT, Czech Republic. He obtained his MSc. in 2006 (Information Systems) and finished his Ph.D. in 2010. He is interested in image processing, data mining, genetic programming and optimization.

Appendix A

The convergence rates of the average estimation

		Maximum Degree		Metropolis-Hasting		Best Constant		BMH
		CR	OPT [%]	CR / OPT [%]		CR / OPT [%]		CR
Weakly connected	Network #1	6928	53.18	3713	12.65	4366	25.71	3243.47
	Network #2	4259	54.20	2288	14.75	2431	19.76	1950.56
	Network #3	5128	55.98	2591	12.88	2775	18.66	2257.28
	Network #4	9554	55.98	4975	15.47	6268	32.91	4205.35
	Network #5	7011	53.39	3798	13.95	4502	27.41	3268.15
	Network #6	6172	47.38	3794	14.40	3560	8.77	3247.65
	Network #7	13059	60.23	6597	21.28	7854	33.88	5193.18
	Network #8	2881	49.93	1701	15.20	1643	12.20	1442.52
	Network #9	6837	64.60	3011	19.61	3779	35.95	2420.40
	Network #10	3335	51.09	1903	14.29	1970	17.20	1631.07
Averagely connected	Network #1	1064	42.29	680	9.70	706	13.03	614.02
	Network #2	1370	45.11	845	11.01	842	10.69	751.97
	Network #3	1226	47.64	700	8.29	869	26.13	641.95
	Network #4	2085	52.94	1097	10.56	1114	11.92	981.20
	Network #5	1181	44.02	727	9.06	675	2.05	661.14
	Network #6	2864	60.04	1315	12.98	1794	36.21	1144.37
	Network #7	1309	46.37	824	14.80	808	13.11	702.06
	Network #8	1568	50.02	929	15.64	1014	22.71	783.75
	Network #9	951	39.71	649	11.66	685	16.30	573.35
	Network #10	979	40.29	634	7.80	593	1.42	584.55
Strongly connected	Network #1	520	35.56	380	11.82	629	46.72	335.10
	Network #2	1870	69.30	691	16.93	1130	49.20	574.01
	Network #3	412	31.47	297	4.9400	293	3.64	282.33
	Network #4	610	46.06	355	7.3100	389	15.41	329.05
	Network #5	497	33.20	352	5.6800	336	1.19	332.00
	Network #6	622	34.82	435	6.8000	448	9.51	405.40
	Network #7	492	35.41	332	4.2900	351	9.47	317.77
	Network #8	620	42.09	394	8.8800	413	13.07	359,03
	Network #9	502	32.84	363	7.1200	326	-3.42	337.15
	Network #10	607	40.03	396	8.0800	392	7.15	363.99

The number of the messages necessary for the average estimation

		Maximum Degree	Metropolis-Hasting	Best Constant	BMH
		MN	MN	MN	MN
Weakly connected	Network #1	1378473	738688	868635	645251.53
	Network #2	847342	455113	483570	387962.44
	Network #3	1020273	515410	552026	448999.72
	Network #4	1901047	989826	1247133	836665.65
	Network #5	1378473	738688	868635	644988.85
	Network #6	1228029	754807	708241	646083.35
	Network #7	2598542	1312604	1562747	1033243.82
	Network #8	573120	338300	326758	286862.48
	Network #9	1360364	598990	751822	481460.60
	Network #10	663466	378498	391831	324383.93
Averagely connected	Network #1	211537	135121	140295	121990.98
	Network #2	272431	167956	167359	149443.03
	Network #3	243775	139101	172732	127549.05
	Network #4	414716	218104	221487	195059.80
	Network #5	234820	144474	134126	131367.86
	Network #6	569737	261486	356807	227530.63
	Network #7	260292	163777	160593	139510.94
	Network #8	311833	184672	201587	155767.25
	Network #9	189050	128952	136116	113897.65
	Network #10	194622	125967	117808	116126.45
Strongly connected	Network #1	103480	75620	125171	66485.90
	Network #2	372130	137509	224870	114028.99
	Network #3	81988	59103	58307	55984.67
	Network #4	121390	70645	77411	65281.95
	Network #5	98903	70048	66864	65869.00
	Network #6	123778	86565	89152	80475.60
	Network #7	97908	66068	69849	63037.23
	Network #8	123380	78406	82187	71247.97
	Network #9	99898	72237	64874	66893.85
	Network #10	120793	78804	78008	72235.01

Appendix B

The convergence rates of the network size estimation

		Maximum Degree		Metropolis-Hasting		Best Constant		BMH
		CR	OPT [%]	CR	OPT [%]	CR	OPT [%]	CR
Weakly connected	Network #1	2930.50	56.19	1553.11	17.35	1733.61	25.95	1283.72
	Network #2	1733.00	54.28	928.87	14.70	955.4	17.08	792.30
	Network #3	2399.74	56.36	1250.58	16.26	1334.13	21.50	1047.27
	Network #4	3435.97	56.15	1775.21	15.13	2159.40	30.23	1506.69
	Network #5	3104.41	56.44	1605.87	15.79	1812.49	25.39	1352.31
	Network #6	2278.28	47.75	1402.14	15.10	1383.30	13.94	1190.48
	Network #7	4828.82	58.99	2502.52	20.86	3117.06	36.47	1980.42
	Network #8	1206.68	49.27	740.11	17.29	726.44	15.73	612.15
	Network #9	2788.35	64.46	1234.56	19.74	1516.88	34.67	990.91
	Network #10	1498.38	50.58	860.51	13.95	861.25	14.03	740.43
Averagely connected	Network #1	486.48	43.85	304.95	10.43	290.33	5.92	273.14
	Network #2	563.86	44.62	350.46	10.90	314.70	0.78	312.26
	Network #3	591.57	47.23	342.56	8.87	321.78	2.98	312.18
	Network #4	815.70	51.73	439.52	10.41	475.28	17.15	393.76
	Network #5	481.54	45.82	288.13	9.44	273.42	4.57	260.92
	Network #6	1151.97	60.00	530.98	13.21	679.12	32.14	460.84
	Network #7	562.69	42.26	365.90	11.20	341.12	4.75	324.91
	Network #8	682.59	45.68	426.31	13.03	413.60	10.36	370.76
	Network #9	444.69	36.90	311.02	9.78	286.20	1.96	280.60
	Network #10	421.22	41.85	267.08	8.29	248.19	1.31	244.94
Strongly connected	Network #1	339.23	53.51	175.28	10.02	231.73	31.94	157.72
	Network #2	643.01	68.16	241.43	15.21	389.26	47.41	204.72
	Network #3	208.51	36.87	138.93	5.25	136.07	3.26	131.64
	Network #4	295.72	44.93	175.35	7.13	159.99	-1.78	162.84
	Network #5	240.14	32.02	172.34	5.27	167.13	2.32	163.25
	Network #6	284.81	39.61	183.94	6.49	174.17	1.25	172.00
	Network #7	248.03	34.88	173.06	6.67	182.39	11.44	161.52
	Network #8	279.66	40.23	182.12	8.21	177.24	5.69	167.16
	Network #9	251.15	30.76	187.15	7.09	156.52	-11.1	173.89
	Network #10	268.61	36.99	182.65	7.34	172.51	1.90	169.24

The number of the messages necessary for the network size estimation

		Maximum Degree	Metropolis-Hasting	Best Constant	BMH
		MN	MN	MN	MN
Weakly connected	Network #1	583169.50	309068.89	344988.39	255460.28
	Network #2	344872.97	184845.13	190140.52	157667.70
	Network #3	477548.26	248865.42	265491.87	208406.73
	Network #4	683758.03	353266.79	429720.60	299831.31
	Network #5	617777.59	319568.13	360685.51	269109.69
	Network #6	453377.72	279025.86	275276.70	236905.52
	Network #7	960935.18	498001.48	620294.94	394103.58
	Network #8	240129.32	147281.89	144561.56	121817.85
	Network #9	554881.65	245677.44	301859.12	197191.09
	Network #10	298177.62	171241.49	171388.75	147345.57
Averagely connected	Network #1	96809.52	60685.05	57775.67	54354.86
	Network #2	112208.14	69741.54	62625.30	62139.74
	Network #3	117722.43	68169.44	64034.22	62123.82
	Network #4	162324.30	87464.48	94580.72	78358.24
	Network #5	95826.46	57337.87	54410.58	51923.08
	Network #6	229242.03	105665,02	135144.88	91707.16
	Network #7	111975.31	72814.10	67882.88	64657.09
	Network #8	135835.41	84835.69	82306.40	73781.24
	Network #9	88493.31	61892.98	56953.80	55839.40
	Network #10	83822.78	53148.92	49389.81	48743.06
Strongly connected	Network #1	67506.77	34880.72	46114.27	31386.28
	Network #2	127958.99	48044.57	77462.74	40739.28
	Network #3	41493.49	27647.07	27077.93	26196.36
	Network #4	58848.28	34894.65	31838.01	32405.16
	Network #5	47787.86	34295.66	33258.87	32486.75
	Network #6	56677.19	36604.06	34659.83	34228.00
	Network #7	49357.97	34438.94	36295.61	32142.48
	Network #8	55652.34	36241.88	35270.76	33264.84
	Network #9	49978.85	37242.85	31147.48	34604.11
	Network #10	53453.39	36347.35	34329.49	33678.76

Appendix C

The range of the convergence rates caused by the change of the leader

		Maximum Degree		Metropolis-Hasting		Best Constant		BMH
		CR	**OPT [%]**	**CR**	**OPT [%]**	**CR**	**OPT [%]**	**CR**
Weakly connected	Network #1	2700	50.77	1574	15.54	2518	47.21	1329.34
	Network #2	1441	54.44	767	14.40	1343	51.11	656.53
	Network #3	1560	60.75	714	14.24	1638	62.62	612.36
	Network #4	3637	58.69	1822	17.54	3255	53.84	1502.48
	Network #5	2814	49.48	1612	11.81	2625	45.84	1421.58
	Network #6	1949	40.85	1267	9.01	1874	38.48	1152.87
	Network #7	4996	65.26	2429	28.55	4795	63.80	1735.61
	Network #8	512	49.63	271	4.84	630	59.07	257.88
	Network #9	2158	62.78	1028	21.86	2036	60.55	803.26
	Network #10	937	38.19	692	16.31	929	37.66	579.16
Averagely connected	Network #1	345	43.09	205	4.22	267	26.46	196.34
	Network #2	261	34.31	181	5.28	348	50.74	171.44
	Network #3	335	48.96	185	7.58	386	55.70	170.98
	Network #4	522	56.52	253	10.28	486	53.29	226.99
	Network #5	379	57.87	176	9.28	273	41.51	159.67
	Network #6	863	68.08	329	16.28	824	66.57	275.44
	Network #7	248	50.77	138	11.54	259	52.86	122.08
	Network #8	239	25.21	199	10.17	444	59.74	178.76
	Network #9	207	31.83	147	4.00	258	45.30	141.12
	Network #10	289	39.45	187	6.42	208	15.87	175.00
Strongly connected	Network #1	560	83.19	136	30.79	283	66.74	94.13
	Network #2	834	74.68	293	27.93	615	65.67	211.16
	Network #3	119	52.16	60	5.12	92	38.12	56.93
	Network #4	141	51.09	75	8.05	158	56.35	68.96
	Network #5	863	68.08	329	16.28	824	66.57	275.44
	Network #6	245	49.68	130	5.17	199	38.05	123.28
	Network #7	196	41.24	126	8.60	143	19.47	115.16
	Network #8	130	47.85	78	13.09	168	59.65	67.79
	Network #9	111	22.72	88	2.52	144	40.43	85.78
	Network #10	92	34.78	61	1.64	160	62.50	60.00