U.P.B. Sci. Bull., Series C, Vol. 79, Iss. 4, 2017 ISSN 2286-3540

INFLUENCE OF RANDOM LEADER APPOINTMENT ON CONVERGENCE RATE OF NETWORK SIZE ESTIMATION

Martin KENYERES[1], Jozef KENYERES[2]

The information about the network size is crucial for many real-life applications. It can be obtained by the distributed average consensus algorithm, whose implementation requires the proper leader appointment, which is an energy demanding process. The lack of the papers concerned with this aspect motivates us to verify the influence of a random leader appointment on the convergence rates of different weight models of average consensus. We examine the range of the achieved convergence rates in 30 randomly generated networks for different leaders and show the maximal possible deceleration of the algorithm due to an inappropriate leader appointment.

Keywords: Distributed computing, average consensus algorithm, network size estimation, leader appointment

1. Introduction

Typically, the estimations techniques can be divided into two main categories: the centralized and the decentralized ones.

The centralized estimation techniques are based on the presence of the fusion center to collect data measured and processed by the geographically distributed nodes [1]. This approach requires an energy-demanding communication within a large area or the implementation of a multi-hop routing protocol, causing a poor scalability of these networks. Other disadvantages of these techniques are a low robustness, low suitability, the necessity for the fusion center to know the measurement models, respectively, additional information about the nodes' parameters etc. [2].

In contrast to this solution, the decentralized estimation is based on the absence of the fusion center [3]. This approach does not require any node to be aware of the network topology and also the implementation of routing mechanisms is not necessary. Its principle lays in a neighbor-to-neighbor commutation among the nodes, which optimizes the aspects such as the energy consumption, the scalability, the natural robustness etc. This is probably the main reason of why these techniques are preferred to the centralized ones in modern

[1] PhD student, Dept.of Telecommunications, Brno University of Technology, Czech Republic, e-mail: kenyeres@phd.feec.vutbr.cz
[2] Software developer, Sipwise GmbH, Austria, e-mail: jkenyeres@sipwise.com

real-life applications. The decentralized optimization techniques can be divided into two subcategories. The first one is based on the transmission of the information in a sequential manner from one node to another one [3]. The principle of the other one lays in the diffusion of the local information into the network [4]. This approach is characterized by a higher robustness but requires a more complicated communication overhead. This subcategory involves consensus-based estimation mechanisms, which are based on the usage of distributed algorithms [5]. The goal of these algorithms is to make the states of all the nodes identical by local information exchanges [6].

In this paper, we focus on the average consensus algorithm, an iterative multifunctional distributed algorithm primarily proposed for an estimation of the average from all the initial values [7]. It is based on a mutual communication among the adjacent nodes, updating the local state according to the information from the adjacent area and the current inner state and the asymptotic convergence to the value of the estimated aggregated function [7]. As mentioned earlier, its main purpose is to estimate the average value, however, tiny modifications can ensure the change of its functionality [8]. In this paper, we focus our attention on an estimation of the network size. Compared with an average estimation, where the inner states are initialized by (for example) local measurements, the functionality of a network size estimation requires the appointment of the leader, whose initial value is initiated to 1, while the other nodes take 0 [8]. As discussed in [8], it is an energy-demanding process requiring the implementation of another complementary mechanism to appoint the most appropriate node as the leader. The lack of the papers focused on this aspect motivates us to compare the chosen weight models of the average consensus and examine how a random choice of the leader affects the converge rate of the algorithm. We choose and compare weight models with static weight matrices, i.e. the weight matrix is invariant over the iterations during the whole estimation process. As many real-life applications are proposed with minimal energy demands, our goal is to show how omitting the implementation of a complementary mechanism to determine the optimal leader can affect the convergence rate of the algorithm – we mutually compare the ranges of the convergence rates and the maximal possible decelerations of the algorithm caused by an inappropriate choice of the leader.

2. Mathematical model of average consensus algorithm

In this section, we introduce mathematical tools used to model the average consensus algorithm [9]. A network is considered to be an indirect finite graph consisting of two sets G = (**V**,**E**). The set **V** contains all the vertices, which represent the particular nodes in a system. The nodes are allocated the unique identity v_i, used for the unambiguous identification. The set **E** consists of all the edges, whose existence indicates the direct connection between two nodes, and these edges are labeled as e_{ij}. We assume the homogeneity of the transmission range and therefore, the connectivity between every two nodes is always mutual.

Within the spectral graph theory, several descriptive tools for mutual connectivity are defined. One of the most frequently used ones is the Laplacian matrix defined as follows [10]:

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

As seen, it is a diagonally symmetric matrix for all the indirect graphs. The parameter d determines the degree of a node and therefore, the number of its neighbors.

In our experiments, we assume randomly generated topologies consisting of 200 nodes. We generate them as follows: each free position within a square area is allocated the probability of a node placement equaled to the reciprocal of the number of the free positions. Thus, the placement of a node is a random event of a uniform distribution. In order to ensure a different average connectivity of the topologies, the transmission range of the nodes varies. In this paper, we use three sets of the topologies: weakly, averagely and strongly connected. Representatives of these sets are shown in Fig. 1, Fig. 2 and Fig. 3.

Let us focus on modeling the average consensus algorithm in a network. The nodes update their inner states according to the states collected from the adjacent area and the inner state from the previous iteration. The described procedure can be modeled using the following difference equation [11]:

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

Here, $\mathbf{x}(k)$ is a column vector variant over the iterations containing the inner values of all the nodes at each iteration. $\mathbf{W}$ is the weight matrix, whose elements vary for different weight models. According to [12], for all the static systems, the following conditions have to hold:

$$\mathbf{W} \times \mathbf{1} = \mathbf{1} \tag{3}$$

$$\mathbf{1}^{\mathrm{T}} \times \mathbf{W} = \mathbf{1}^{\mathrm{T}} \tag{4}$$

$$\rho(\mathbf{W} - \frac{1}{N}.\mathbf{1} \times \mathbf{1}^{\mathrm{T}}) < 1 \tag{5}$$

Here, ρ is the spectral radius of the matrix determined as the difference of the weight matrix $\mathbf{W}$ and the matrix defined as $1/N.\mathbf{1}\times\mathbf{1}^{\mathrm{T}}$ [12]. Preserving condition (5) ensures the asymptotic convergence of the algorithm, while conditions (3–4) determine the convergence point as well as ensure that the weight matrix is bistochastic [12].

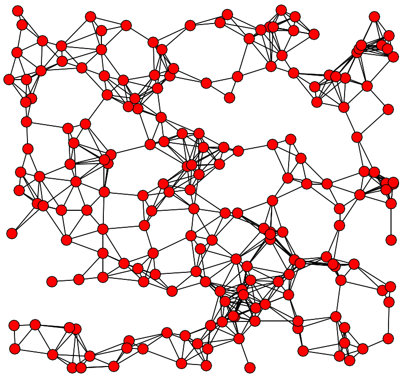

Fig. 1. Representative of networks with weak connectivity

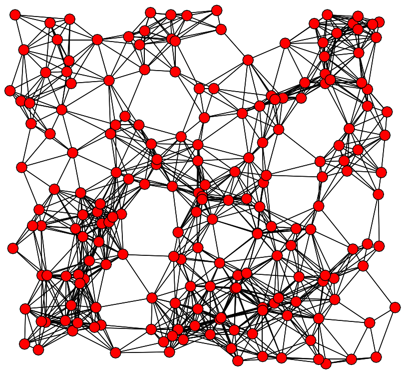

Fig. 2. Representative of networks with average connectivity

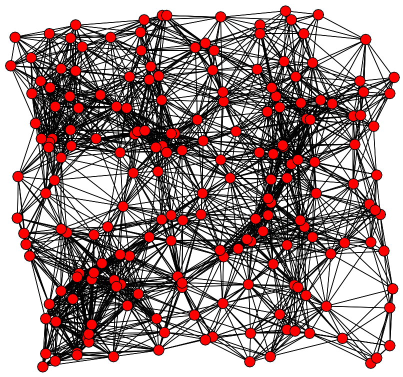

Fig. 3. Representative of networks with strong connectivity

As previously mentioned, the average consensus is an iterative algorithm that asymptotically converges to the value of an aggregate function. So, we can write the following [11]:

$$\lim_{k\to\infty}\mathbf{x}(k) = \lim_{k\to\infty}\mathbf{W}^{k-1}\times\mathbf{x}(1) = \frac{\mathbf{1}\times\mathbf{1}^{\mathrm{T}}}{N}\times\mathbf{x}(1) \tag{6}$$

Only the existence of this limit guarantees the proper functionality of the algorithm and is achieved by the preservation of (3-5). Since we assume the execution of the algorithm in a finite time, we define an indicator of the consensus achievement as follows:

$$\left|\max\{\mathbf{x}(k)\} - \min\{\mathbf{x}(k)\}\right| < \delta \tag{7}$$

Here, δ determines the precision of the final estimates as well as the convergence rate of the algorithm. We set this value to 0.0001 and kept it constant for all the experiments.

3. Examined weight models of average consensus algorithm

We choose the following six weight models for an examination: the Constant weight model (abbreviated as CW), the Maximum Degree weight model (abbreviated as MD), the Metropolis-Hasting weight model (abbreviated as MH), the Local Degree weight model (abbreviated as LD), the Best Constant weight model (abbreviated as BC), and the Biphasically configured Metropolis-Hasting weight model (abbreviated as BMH). All these models are characterized by a constant weight matrix, which simplifies the implementation into real-life applications.

The Constant weight model is characterized by uniform weights and is defined as follows [13]:

$$[W^{\mathrm{CW}}]_{ij} = \begin{cases} \varepsilon, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - d_i.\varepsilon, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

Here, ε is the mixing parameter, whose value determines the convergence rate of the algorithm. Its higher values ensure a higher convergence rate but too high values result in the divergence of the algorithm [14]. According to [13], the convergence is achieved for each ε from the interval:

$$0 < \varepsilon \leq \frac{1}{d_{\max}} \tag{9}$$

Note that the equality is possible only if the graph is neither regular nor bipartite. Here, d_{max} is the degree of the best-connected node. In this paper, we consider the Constant weight model to take the value of ε equaled to $0.5*1/d_{max}$.

Another model of our interest is the Maximum Degree weight model. Actually, it is the Constant weight model with the value of ε set to the maximal possible value and so, $\varepsilon = 1/d_{max}$. Thus, it poses the maximally optimized Constant weight model [15]. As we do not assume regular bipartite graphs in our experiments, we use this value in each network.

Both the Metropolis-Hasting [16] and the Local degree [12] weight models require only locally available information for their proper initial configuration and neither of them is characterized by uniform weights. Their weight matrices are defined as follows:

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

$$[W^{\mathrm{LD}}]_{ij} = \begin{cases} \max\{d_i, d_j\}^{-1}, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - \sum_{k=1, k \neq i}^{N} [W^{\mathrm{LD}}]_{ik}, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

The Best Constant weight model is considered to be the fastest model with uniform weights. In order to be optimally configured, it requires the knowledge about the largest and the second smallest eigenvalue of the Laplacian matrix corresponding to the graph. Its weight matrix is defined as [12]:

$$[W^{\mathrm{BC}}]_{ij} = \begin{cases} 2/(\lambda_1(\mathbf{L}) + \lambda_{N-1}(\mathbf{L})), & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - 2.d_i/(\lambda_1(\mathbf{L}) + \lambda_{N-1}(\mathbf{L})), & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

The last examined model is the Biphasically configured Metropolis-Hasting weight model [17]. It is derived from the Metropolis-Hasting weight model by adding an additional phase into the initial configuration. Its principle lays in an increase of the weights allocated to a node's neighbors at the cost of decreasing the weight of the current inner state. As discussed in [17], the convergence rate of this weight model depends on the distribution of the identity numbers. Therefore, we randomly shuffle the position of the identity numbers and repeat the execution of this weight model 100 times for each topology and for each leader. As a representative of the obtained set of the convergence rates, we choose the scenario with the widest range (and so, the worst case scenario).

4. Experimental part

In the experimental part, we execute two experiments. As previously mentioned, we verify how a random appointment of the leader affects the convergence rates of the chosen weight models of the average consensus algorithm. We demonstrate this on three randomly generated sets of networks with the size of 200 nodes. For each weight model, we repeat the experiments 200 times - the executions differ from each other in the appointment of a different node as the leader (except for BMH, where the experiments are repeated 100 times for each leader appointment (20 000 repetitions) and subsequently the scenario with the widest range (and so, the worst result) is chosen as a representative of this data set). This procedure is repeated for each topology.

In the first experiment, we examine how a random appointment of the leader affects the range of the convergence rates of the chosen weight models. In Tab. 1, we show the ranges for all six models. We can see from the results that the narrowest range is achieved for the BMH (in seven weakly connected, seven averagely connected and five strongly connected networks), for LD (in three weakly connected, one averagely connected and three strongly connected networks) and in two averagely and two strongly connected networks, these two

Table 1

Comparison of influence of random leader appointment on range of convergence rates for examined weight models

Range of convergence rates						
	CW	MD	MH	LD	BC	BMH
W #1	2092 it.	1046 it.	1074 it.	947 it.	1177 it.	829 it.
W #2	2198 it.	1098 it.	580 it.	520 it.	988 it.	493 it.
W #3	2569 it.	1284 it.	774 it.	696 it.	1238 it.	579 it.
W #4	2036 it.	1018 it.	713 it.	639 it.	1047 it.	647 it.
W #5	1575 it.	787 it.	385 it.	329 it.	667 it.	293 it.
W #6	1316 it.	657 it.	464 it.	424 it.	910 it.	427 it.
W #7	2922 it.	1461 it.	779 it.	689 it.	1353 it.	677 it.
W #8	3685 it.	1841 it.	953 it.	800 it.	1668 it.	713 it.
W #9	12671 it.	6335 it.	3173 it.	2713 it.	4494 it.	2788 it.
W #10	5359 it.	2679 it.	1719 it.	1532 it.	2461 it.	1510 it.
A #1	792 it.	395 it.	208 it.	199 it.	285 it.	196 it.
A #2	692 it.	345 it.	205 it.	194 it.	267 it.	196 it.
A #3	374 it.	187 it.	122 it.	122 it.	228 it.	103 it.
A #4	829 it.	414 it.	162 it.	147 it.	299 it.	145 it.
A #5	462 it.	231 it.	177 it.	165 it.	231 it.	165 it.
A #6	1193 it.	595 it.	206 it.	190 it.	567 it.	180 it.
A #7	394 it.	196 it.	148 it.	138 it.	211 it.	138 it.
A #8	663 it.	331 it.	257 it.	239 it.	292 it.	232 it.
A #9	1056 it.	527 it.	262 it.	242 it.	573 it.	235 it.
A #10	652 it.	325 it.	112 it.	98 it.	147 it.	92 it.
S #1	394 it.	196 it.	126 it.	159 it.	121 it.	118 it.
S #2	316 it.	158 it.	99 it.	94 it.	156 it.	95 it.
S #3	486 it.	242 it.	126 it.	119 it.	189 it.	114 it.
S #4	423 it.	211 it.	143 it.	139 it.	211 it.	138 it.
S #5	317 it.	158 it.	129 it.	123 it.	179 it.	124 it.
S #6	269 it.	134 it.	83 it.	79 it.	129 it.	80 it.
S #7	263 it.	131 it.	64 it.	60 it.	135 it.	60 it.
S #8	239 it.	119 it.	73 it.	69 it.	99 it.	69 it.
S #9	417 it.	207 it.	147 it.	143 it.	284 it.	135 it.
S #10	277 it.	138 it.	98 it.	94 it.	140 it.	89 it.

weight models have the identical range. Regarding the widest range (and so, the most negatively affected weight model), CW achieves worse results compared with the other examined weight models in all 30 networks.

In the next experiment, we examine the maximal possible deceleration caused by an inappropriate appointment of the leader, i.e. we express the relative difference between the fastest and the slowest execution (for example: 100 % deceleration means that the slowest execution needs twice as many iterations as the fastest one in the corresponding network and for corresponding weight model). The results have been depicted in Fig. 4, Fig. 5, Fig. 6 and Tab. 2.

In the weakly connected networks, we can see that CW is the least decelerated from all the examined weight models in four networks, MD in two, LD in one and BMH in three. In the averagely connected networks, CW achieves the best result in one network, MD in two, MH in two, LD in two and the BMH also in two. In one network, CW and MD equally achieve the smallest maximal relative deceleration of the algorithm. In the strongly connected networks, CW achieves the best result in two networks, MD in one, MH in one, LD in three, BC in one and the BMH in two.

Regarding the models with the worst results: in the weakly connected networks, MD achieves the largest deceleration in two networks, MH in one, LD

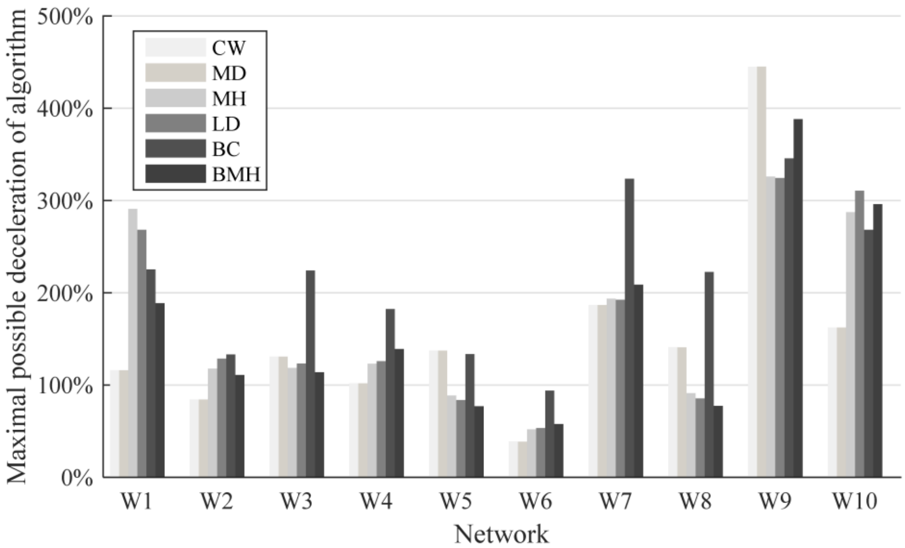

Fig. 4. Comparison of maximal possible deceleration – weakly connected networks

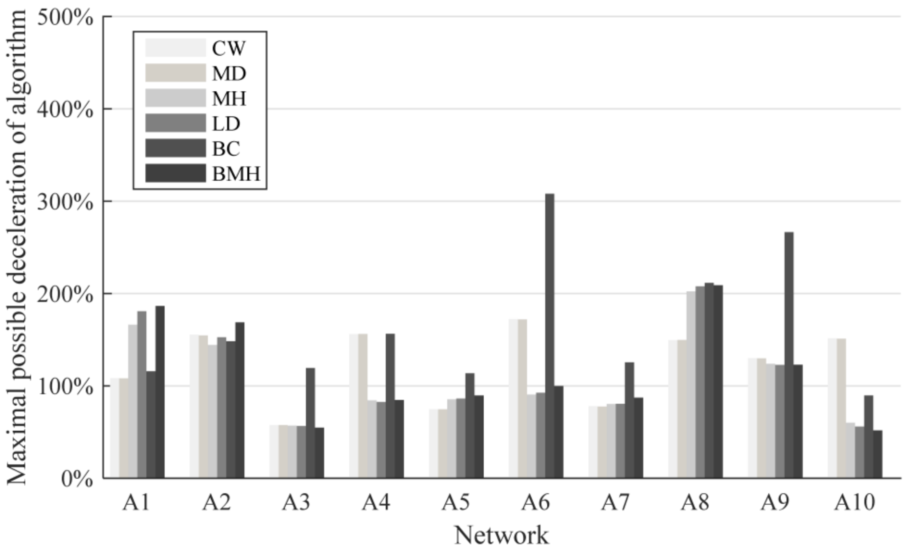

Fig. 5. Comparison of maximal possible deceleration – averagely connected networks

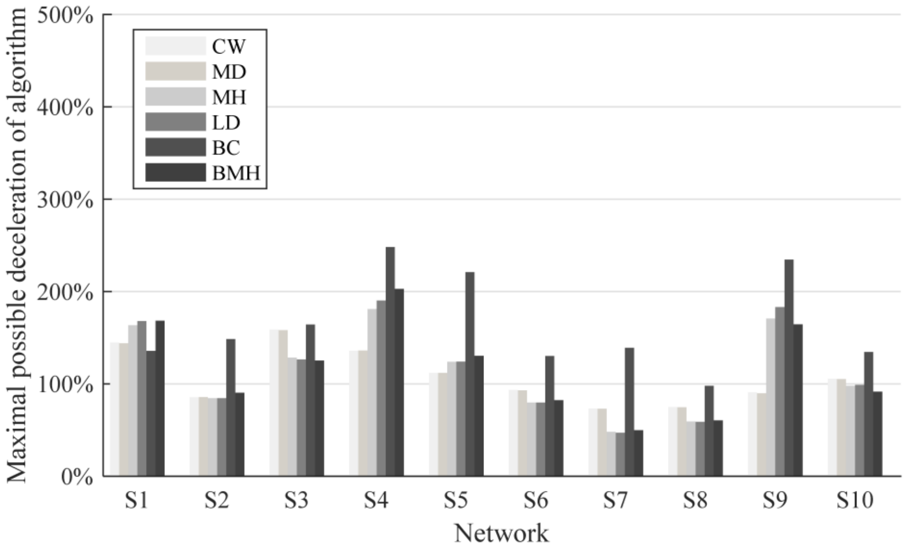

Fig. 6. Comparison of maximal possible deceleration – strongly connected networks

in one, and BC in six. In the averagely connected, CW achieves the worst result in one network, BC in seven and BMH in two. In the strongly connected, BC is the worst in nine networks and BMH in one.

In Fig. 7, Fig. 8, and Fig. 9 and Fig. 10 (the most and the least maximal possible deceleration of the average consensus algorithm for each topology), we express in percentages in how many networks the weight models achieve the best and the worst result within both experiments.

5. Future research

The average consensus algorithm is frequently used as a distributed estimation technique in many real-life applications, especially in those based on wireless sensor devices. The design of these devices is significantly affected by cost constraints, which often results in limited energy sources. Thus, the modern algorithms for wireless sensor networks have to take into account this aspect. Our research shows that a random appointment of the leader can significantly affect the convergence rate of the algorithm (a slower rate causes also an increased energy consumption), but also a complementary mechanism for its best determination is a significant energy redundancy. Our future work is going to be focused on an optimization of this aspect and the results presented in this paper reason the relevance of this planned research.

Table 2

Comparison of maximal possible deceleration of algorithm caused by inappropriate leader appointment

	Maximal possible deceleration of algorithm					
	CW	MD	MH	LD	BC	BMH
W #1	116.03 %	116.09 %	291.06 %	268.27 %	225.48 %	188.84 %
W #2	84.38 %	84.33 %	117.89 %	128.71 %	133.15 %	111.04 %
W #3	130.87 %	130.89 %	118.71 %	123.40 %	224.28 %	113.98 %
W #4	101.85 %	101.90 %	123.14 %	126.04 %	182.40 %	139.14 %
W #5	137.31 %	137.35 %	88.71 %	83.72 %	133.67 %	77.11 %
W #6	38.82 %	38.76 %	52.08 %	53.54 %	94.01 %	57.86 %
W #7	186.71 %	186.83 %	193.78 %	192.46 %	323.68 %	208.95 %
W #8	141.03 %	140.86 %	91.28 %	85.56 %	222.70 %	77.50 %
W #9	445.06 %	445.19 %	326.10 %	324.52 %	345.69 %	388.30 %
W #10	162.34 %	162.36 %	287.46 %	310.75 %	268.38 %	296.08 %
A #1	108.49 %	108.22 %	166.40 %	180.91 %	115.85 %	186.67 %
A #2	155.51 %	154.71 %	144.37 %	152.76 %	148.33 %	168.97 %
A #3	57.63 %	57.72 %	57.01 %	56.57 %	119.37 %	54.79 %
A #4	156.12 %	156.23 %	84.38 %	82.58 %	156.54 %	84.80 %
A #5	74.76 %	74.76 %	85.51 %	86.39 %	113.79 %	89.67 %
A #6	172.40 %	171.97 %	90.75 %	92.68 %	308.15 %	100.00 %
A #7	78.17 %	77.78 %	80.43 %	80.70 %	125.60 %	87.34 %
A #8	149.66 %	149.77 %	202.36 %	207.83 %	211.59 %	209.01 %
A #9	130.05 %	129.80 %	124.06 %	122.84 %	266.51 %	123.04 %
A #10	151.63 %	151.16 %	60.22 %	56.00 %	89.63 %	51.98 %
S #1	144.85 %	144.12 %	163.64 %	168.06 %	135.90 %	168.57 %
S #2	85.64 %	85.87 %	84.62 %	84.68 %	148.57 %	90.48 %
S #3	158.82 %	158.17 %	128.57 %	126.60 %	164.35 %	125.27 %
S #4	136.01 %	136.13 %	181.01 %	190.41 %	248.24 %	202.94 %
S #5	112.01 %	112.06 %	124.04 %	124.24 %	220.99 %	130.53 %
S #6	93.40 %	93.06 %	79.81 %	79.80 %	130.30 %	82.47 %
S #7	73.46 %	73.18 %	48.12 %	47.24 %	139.18 %	50.00 %
S #8	74.92 %	74.84 %	59.35 %	58.97 %	98.02 %	60.53 %
S #9	90.85 %	90.00 %	170.93 %	183.33 %	234.71 %	164.63 %
S #10	105.73 %	105.34 %	98.00 %	98.95 %	134.62 %	91.75 %

6. Conclusion

We experimentally verified the influence of a random leader appointment on the convergence rates of CW, MD, MH, LD, BC, and BMH weight model of average consensus. In the first part, we changed the leader in a network and showed the range of the gained convergence rates. We could have seen from the results that the LD (in 32 % networks) and the BMH (in 68 % networks) weight model achieved the narrowest range from all the examined weight models. So, the range of these two weight models was the least affected by an inappropriate appointment of the leader. The most negatively affected weight model was CW, which reached the worst results in all 30 networks.

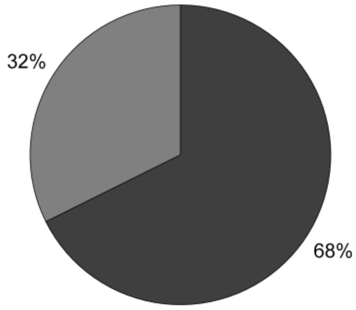

Fig. 7. Range of convergence rates – weight models with narrowest range

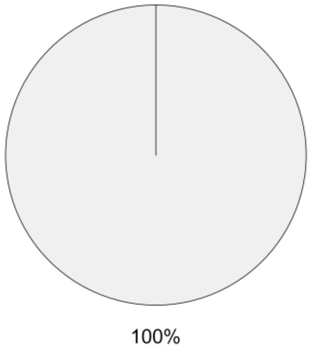

Fig. 8. Range of convergence rates – weight models with widest range

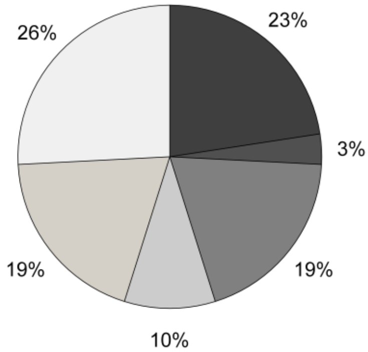

Fig. 9. Maximal possible deceleration – least negatively affected weight models

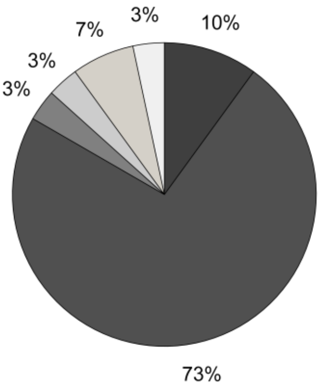

Fig. 10. Maximal possible deceleration – most negatively affected weight models

In the next experiment, we examined the maximal possible deceleration of the algorithm caused by an inappropriate leader appointment, i.e. we compared the difference between the fastest and the slowest execution of the algorithm for each weight model. Here, CW achieved the best result, which was the least decelerated in 26 % networks. The second best was BMH with 23 %, then LD and MD equally with 19 %, MH with 10 %, and BC with 3 %. The worst one was the BC, which was the most decelerated in 73 % networks. According to the obtained results, we can conclude that BMH is the least affected by an inappropriate leader appointment. Nevertheless, it was decelerated in the range 47.24% - 24.52 % and so, we experimentally proved the necessity of a complementary mechanism to determine the most appropriate node as the leader.

Acknowledgment

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

R E F E R E N C E S

[1]. *F. Zhao, L. J. Guibas*, Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, Amsterdam, 2004.

[2]. *A. Nayak, I. Stojmenovic*, Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication. John Wiley & Sons, New Jersey, 2010

[3]. *T. Zhao, A. Nehorai,* Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks, in IEEE Trans. Signal Process., **vol. 55**, no. 4, Mar. 2007, pp. 1511–1524.

[4]. *R. G. Baraniuk, V. Delouille, and R. Neelamani*, Robust distributed estimation using the embedded subgraphs algorithm, in IEEE Trans. Signal Process., **vol. 54**, no. 8, July 2006, pp. 2998-3010,

[5]. *A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione,* Gossip algorithms for distributed signal processing, in Proc. IEEE, **vol. 98**, no. 11, Aug. 2010, pp. 1847–1864.

[6]. *D. Merezeanu, M. Nicolae,* Consensus control of discrete-time multi-agent systems, in University Politehnica of Bucharest Scientific Bulletin-Series A-Applied Mathematics and Physics, **vol. 79**, no. 1, Aug. 2017, pp. 1223–7027.

[7]. *Y. Mo, R. M. Murray,* Privacy preserving average consensus, in IEEE Trans. Autom. Control, **vol. 62**, no. 2, May 2017, pp. 753–765.

[8]. *I. Shames, et al.*, Distributed Network Size Estimation and Average Degree Estimation and Control in Networks Isomorphic to Directed Graphs, in 50th Annual Allerton Conference on Communication, Control, and Computing, 2012.

[9]. *A. Benjamin, G. Chartrand and G. Zhang,* The Fascinating World of Graph Theory, Princeton University Press, New Jersey, 2015.

[10]. *W.N. Anderson, T.D. Morley,* Eigenvalues of the Laplacian of a graph, in Linear and multilinear algebra, **vol. 18**, no. 2, May 2007, pp. 141–145.

[11]. *L. Xiao, S. Boyd, and S.J. Kim,* Distributed average consensus with least-mean-square deviation, in J Parallel Distrib Comput., **vol. 67**, no. 1, Jan. 2007, pp. 33–46.

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

[13]. *V. Schwarz, G. Hannak and G. Matz,* On the convergence of average consensus with generalized metropolis-hasting weights. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 5442–5446.

[14]. *M. Kenyeres, J. Kenyeres, and V. Skorpil,* The distributed convergence classifier using the finite difference, in Radioengineering, **vol. 25**, no. 1, Apr. 2016, pp. 148–155.

[15]. *W. Li, Y. Jia,* Consensus-based distributed multiple model UKF for jump Markov nonlinear systems, in IEEE Trans. Autom. Control, 2012, **vol. 57**, no. 1, July 2012, pp. 227–233.

[16]. *B.Y. Kim, H. Hur and H.S. Ahn,* Coordination and control for freeway traffic network. In Proceeding of the IEEE International Systems Conference (SysCon), 2013, pp. 159–163.

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