Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Decentralized Asymmetric Weighting Approach for Improved Convergence of Multi-Agent Systems with Undirected Interaction

Ann-Kathrin Schug * **Annika Eichler** * **Herbert Werner** *

* *Institute of Control Systems, Hamburg University of Technology, Germany (e-mail: {ann.schug},{annika.eichler}, {h.werner}@tuhh.de)*

Abstract: This work considers the convergence rate of multi-agent systems with discrete-time single-integrator dynamics and undirected interaction topologies. In recent work it has been proven that in case of lattice interaction topologies the convergence rate can be bounded away from zero, independent of the network size, using asymmetric weightings that give the interaction graph a preferred communication direction. Approximation methods for more general graphs, based on relative angles between agents, are presented, which suggest that the convergence rate is bounded away from zero as well. This work proposes alternative approximation methods, that improve the convergence rate compared to previous approximation methods. Furthermore it is shown that the improvement of the approximation methods degrade in comparison to other weighting approaches, the more the considered topology differs from a lattice graph. Therefore an iterative algorithm is proposed, that extends the idea of a preferred communication direction to general graphs, which are not similar to lattices or where the relative angles are not known.

Keywords: Multi-agent system, convergence, decentralized systems, graph theory, networks

1. INTRODUCTION

There is a wide range of problems, where communicating agents in a network or members of a group should agree on a common value, by iteratively aggregating information from their neighbors until consensus is reached. This task is called distributed consensus problem and has received considerable attention over recent years, due to its wide range of applications, e.g. in multi-vehicle rendezvous, data fusion in large sensor networks, cooperative or formation control of multi-agent systems, task assignment, etc. See Murray (2007); Ren and Cao (2011) and the references therein for an extensive list of applications.

One important issue in distributed consensus is the convergence rate, which determines the number of iterations, that are necessary until consensus is reached within a prescribed bound. A consensus protocol with a small convergence rate needs many iterations and is thus less useful in practice. Therefore much effort has been spent on analyzing and optimizing the convergence rate for various types of consensus protocols. Here discrete-time single-integrator systems are considered, where the convergence rate is determined by the choice of weights with which the information from different neighbors is weighted. Here one can distinguish between determining the weights centrally or in a distributed way. In the first case the interaction graph has to be fixed and known *a priori*. In Xiao and Boyd (2004) the weighting is optimized centrally for a fixed and undirected graph, i.e. symmetric interaction topology, by solving an LMI problem, which results in symmetric weighting factors. In Kim et al. (2009) the same has been done for directed graphs using qth-order spectral norm minimization and gradient sampling. There the auxiliary conclusion has been drawn that for symmetric interaction - which will be considered in this paper - symmetric weightings represent a local minimum of the convergence rate optimization, while asymmetric weighting can lead to better results. This is confirmed in Mangoubi et al. (2013), whose results on periodic gossip algorithms can be extended to the deterministic case considered here. The convergence rate of centralized and distributed weighting approaches is analyzed and bounded in Olshevsky and Tsitsiklis (2011). All the approaches discussed there for different graphs have in common that the upper and lower bounds of the convergence rate scales with $1/N^3$ or $1/N^2$, where N is the number of agents. That means that for large scale systems the convergence rate can be very small. Similar bounds, that scale with $1/N^{2/D}$ on the convergence rate for the special class of D-dimensional lattices are given in Hao and Barooah (2012), if symmetric weights are used. Motivated by that result, it is shown that with asymmetric weights, bounds independent of N can be found for lattices, such that even for infinitely large lattices the convergence rate can be bounded away from zero. Furthermore Hao and Barooah (2012) propose an distributed approximation method of the result for lattices to graphs, where the positions are known and the structure is similar to lattices. This approximation method is further examined in this work. In addition the method is extended to graphs, where no knowledge of the position is required.

1.1 Outline

In Section 2 basic concepts of graph theory are reviewed with a focus on lattices and the concept of asymmetric weightings. This concept is extended in Section 3 to general graphs; in Subsection 3.1 to graphs similar to lattices and in Subsection 3.3 to general graphs. Numerical examples are given for each case. Conclusions are drawn in Section 4.

2. PRELIMINARIES

Consider a network with N agents and m communication links, that is described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, ..., N\}$, representing the agents and edge set $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$, which describes the communication topology. For undirected graphs we have $\{i, j\} \in \mathcal{E}$ if and only if $\{j, i\} \in \mathcal{E}$. Here $\{i, j\} \in \mathcal{E}$ implies that the ith agent receives information from agent j. The set of neighbors of the ith agent is defined as $\mathcal{N}_i := \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$. The degree d_i of a vertex i is equal to the cardinality of $\mathcal{N}_i$. There exist different matrix representations for graphs. Well-known is the adjacency matrix A with $A_{i,j} = 1$ if $\{i, j\} \in \mathcal{E}$ and $A_{i,j} = 0$ otherwise. As alternative representation, the Laplacian matrix L can be determined form the adjacency matrix as $L = D - A$, where D is a diagonal matrix with $D_{i,i} = d_i$. The normalized adjacency is given as $A_n = \sqrt{D}A\sqrt{D}$ and the normalized Laplacian similarily.

The consensus protocol considered here is

$$x_i(k+1) = W_{i,i}x_i(k) + \sum_{j \in N_i} W_{i,j}x_j(k) \qquad (1)$$

for agent i, where x_i is ith agent's state at time step k and $W_{i,j}$ the associated weight of edge $\{i, j\} \in \mathcal{E}$. The consensus protocol (1) for the whole network is then

$$x(k+1) = Wx(k) \qquad (2)$$

with $x = [x_1, \ldots, x_N]^T$ and $W_{i,j} = 0$ if $\{i, j\} \notin \mathcal{E}$. Since $W_{i,j} \geq 0$ and $W\mathbf{1} = \mathbf{1}$, the weighting matrix W is stochastic. If the corresponding graph is connected, W is irreducible and primitive, such that (2) reaches consensus, and the eigenvalues of W satisfy $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq -1$. The convergence rate R is defined in Hao and Barooah (2012) as the spectral gap

$$R = 1 - \max(|\lambda_2|, |\lambda_N|). \qquad (3)$$

2.1 Lattice Graphs

In Hao and Barooah (2012) an asymmetric weight design for D-dimensional lattice graphs is proposed. Starting from one-dimensional lattices, i.e. line graphs, where each agent communicates only with its nearest neighbor, the weights are given as

$$W_{i;i+1} = \frac{1+\varepsilon}{2}, W_{i+1;i} = \frac{1-\varepsilon}{2}, \qquad (4)$$

where $\varepsilon \in (0, 1)$ is the factor of asymmetry. The diagonal entries result from $W\mathbf{1} = \mathbf{1}$. Using exact formulas for the eigenvalues of tridiagonal matrices (Yueh and Cheng, 2008), the convergence rate (3) can be lower bounded by an expression independent of N. This method can be applied to any D-dimensional lattice graph since they are the Cartesian product of one-dimensional lattices, which leads to

$$R \geq \frac{\varepsilon^2}{2D}. \qquad (5)$$

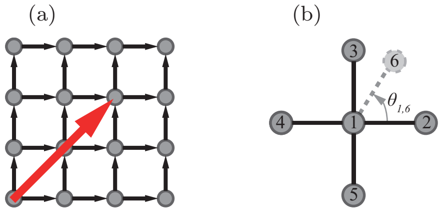

Fig. 1. (a) General idea of asymmetric weights and (b) its approximation to general graphs

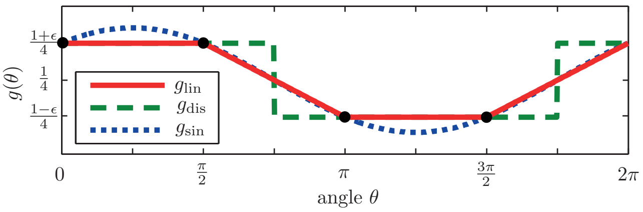

Fig. 2. Different approximationon functions

In the following, 2-dimensional lattices and general graphs in the in the 2-dimensional plane are considered.

3. GENERAL GRAPHS

We now discuss how to extend the results for lattices to general graphs. In the following, we will distinguish two cases: (i) graphs where the geometric positions of the agents correspond to the topology (e.g. proximity graphs) and which are thus "similar" to lattices, and (ii) graphs, where geometric positions of the agents and topology are unrelated and the position might even not be known.

3.1 General Graphs Similar to Lattices

In Hao and Barooah (2012) the approximation of lattice graphs to more general graphs has been studied. Here instead of the eigenvalue problem of the weighting matrix, the eigenvalue problem of the corresponding Laplacian $L = I - W$ is considered. The eigenvalues μ of L relate to those of W as $\lambda_i = 1 - \mu_i$. The eigenvalue problem for the one-dimensional case

$$L\Phi = \begin{bmatrix} \frac{1+\varepsilon}{2} & \frac{-1-\varepsilon}{2} & & & \\ \frac{-1+\varepsilon}{2} & 1 & \frac{-1-\varepsilon}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{-1+\varepsilon}{2} & 1 & \frac{-1-\varepsilon}{2} \\ & & & \frac{1+\varepsilon}{2} & \frac{-1-\varepsilon}{2} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N-1} \\ \Phi_N \end{bmatrix} = \mu \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N-1} \\ \Phi_N \end{bmatrix} = \mu\Phi \qquad (6)$$

can be written as the difference equation

$$-\frac{1}{2N}\frac{\Phi_{i-1} - 2\Phi_i + \Phi_{i+1}}{1/N^2} - \frac{\varepsilon}{N}\frac{\Phi_{i+1} - \Phi_{i-1}}{2/N} = \mu\Phi_i$$

for $i = 1, \ldots, N$ with $\Phi_0 = \Phi_1$ and $\Phi_N = \Phi_{N+1}$. This difference equation can be continuously approximated to

$$-\frac{1}{2N}\frac{\partial^2\Phi(x)}{\partial x^2} - \frac{\varepsilon}{N}\frac{\partial\Phi(x)}{\partial x} = \mu\Phi(x)$$

for a function $\Phi(x)$, that satisfies (6) at the discrete points $\Phi_i = \Phi(x)|_{x=i/(N+1)}$, which are meant to be the agents positions determined by the lattice topology. Note that here a fixed relationship between lattice topology and lattice structured positioning is assumed. This can be extended to any D-dimensional lattice. A 2-dimensional lattice is shown in Fig. 1(a), where one agent in the center with its neighbors is picked in Fig. 1(b). From the lattices it is known that the weights for the neighbors $2, 3, 4$ and 5 are fixed to $W_{1,2} = (1+\varepsilon)/4$, $W_{1,3} = (1+\varepsilon)/4$, $W_{1,4} = (1-\varepsilon)/4$ and $W_{1,5} = (1-\varepsilon)/4$. From the continuous approximation of the eigenvalue problem the idea is to introduce a continuous approximation of the weights to weight neighbors at any general position as agent 6 shown in Fig. 1(b). The weights of this approximation are determined by a function $g(\theta)$ that depends on the relative angle θ between two agents. In order to satisfy the weights in the discrete directions given by lattice graphs, $g(\theta)$ has to satisfy

$$\begin{aligned} g(0) &= \frac{1+\varepsilon}{4}, \quad g\left(\frac{\pi}{2}\right) = \frac{1+\varepsilon}{4}, \\ g(\pi) &= \frac{1-\varepsilon}{4}, \quad g\left(\frac{3\pi}{2}\right) = \frac{1-\varepsilon}{4}. \end{aligned} \tag{7}$$

In Hao and Barooah (2012) an approximation by linear interpolation between the discrete points is proposed, shown in Fig. 2. This is called g_{lin} in the following. Here with g_{dis}, a piecewise constant interpolation, and g_{sin}, an interpolation with a sinusoidal function, two additional functions are proposed, as shown in Fig. 2. The fixed directions by the lattice are marked by black dots. To ensure the stochastic property of W the weights are determined as

$$W_{i,j} = \frac{g(\theta_{i,j})}{\sum_{k\in N_i} g(\theta_{i,k})}. \tag{8}$$

Note that the sinusoidal function g_{sin} reaches negative values for $\varepsilon < 1/\sqrt{2}$, therefore the absolute value of g_{sin} for larger amounts of asymmetry are used. Note that for other methods as the optimal symmetric design by Xiao and Boyd (2004) negative values are not excluded, but here W is restricted to be stochastic.

In general, according to (5), a larger convergence rate can be expected for a larger factor of asymmetry. With increasing asymmetry the information exchange in the 0– and $\pi/2$–direction is preferred, such that the main information flow direction is in the $\pi/4$–direction as shown in Fig. 1(a). This is confirmed in Fig. 4(a), where the convergence rate, achieved with the different approximation functions from Fig. 2 for increasing ε is shown. The considered graphs with $N = 100$ vertices were generated by perturbing the positions of a $\left(\sqrt{N}\times\sqrt{N}\right)$ 2-dimensional lattice, with vertices positioned at intervals of $1/\sqrt{N}$, by Gaussian zero mean noise with a standard deviation of $\sigma = 1/\left(4\sqrt{N}\right)$. Two vertices are connected if their distance is $\leq r = 2/\sqrt{N}$ as in Hao and Barooah (2012). In total 100 sample graphs were considered and the average is shown in Fig. 4(a). As expected, the convergence rate increases with rising ε up to $\varepsilon = 0.8$, whereby the sinusoidal approximation outperforms the others, followed by g_{dis}. The inflection point at $\varepsilon \approx 0.7$ can be explained with the sinusoidal function getting negative for $\varepsilon < 1/\sqrt{2} = 0.707$, such the the absolute value has to be taken and thus the weighting function is not continuously differentiable. The different curves of the convergence rate with different approximations show that not only ε influences the convergence rate, but also the shape of the approximation function, since it also determines, to what degree the $\pi/4$–direction is preferred. In Fig. 4(b) the center of line of the approximation functions in the $\pi/4$–direction, with respect to which they are symmetric, is plotted. For one value $\varepsilon = 0.6$ this is shown exemplarily in Fig. 3, where the approximation functions from Fig. 2 are plotted in the xy–plane and the center of line of the respective curves is shown, located on the $\pi/4$–axis. Based on this in Fig. 4(b) the distance from the center of line to the origin is then shown for increasing ε. Here as well for small ε the sinusoidal function leads to largest asymmetry, which increases with ε increasing up to $\varepsilon = 1/\sqrt{2}$, and confirms the shape of Fig. 4(a). With the conjectured dependence of the center of line on the approximation function and the increase of the convergence rate, the shape of the approximation function can be further optimized.

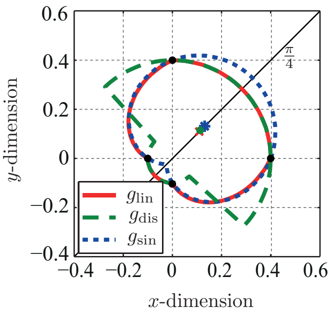

Fig. 3. Approximation function in the xy–plane with center of line on the $\pi/4$–direction for $\varepsilon = 0.6$

Numerical Comparison with Other Approaches The weighting approach using the approximation functions presented in Fig. 2 is now compared with the corresponding two weighting approaches in Hao and Barooah (2012), first with the optimal symmetric weighting (OS) by Xiao and Boyd (2004). In this centralized weighting approach the optimal symmetric weightings are determined by solving an LMI problem. The second approach is the decentralized Metropolis-Hastings method (MH), where the weights are determined as $W_{i,j} = 1/d_i$, if $\{ij\} \in \mathcal{E}$. Note that this leads in general to non-symmetric weights unless the considered graph is balanced.

The considered graphs were generated as explained in the previous subsection for different numbers of agents N. For each value of N, 100 sample graphs were considered and the results were averaged. In the following the asymmetry factor is set to $\varepsilon = 0.5$. The logarithmic plot of the convergence rate R over the number of agents N with a constant $\varepsilon = 0.5$ is shown in Fig. 5(a). For a more detailed comparison Fig. 5(b) illustrates the stochastic distribution of the convergence rates for $N = 100$. It describes the percentage of samples which have at least a certain convergence rate, thus the more a distribution function is located to the right the faster the convergence. The non-symmetric MH design leads to the slowest convergence rates, which is decreasing with increasing N. This suggests that asymmetric weights in general do not improve convergence rate or lead to bounds away from zero, unless the asymmetry is not randomly distributed but leads to a preferred direction in the network. The asymmetric weight design dependent

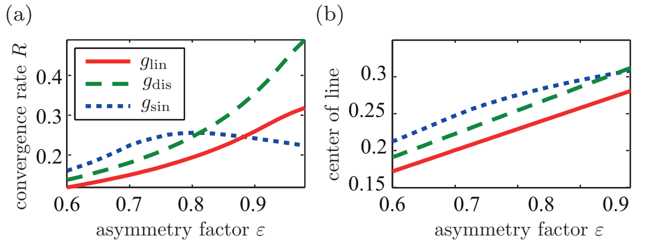

Fig. 4. In dependence on ε in (a) the convergence rate and in (b) the center of line of $g(\theta)$ on the $\pi/4$-axis averaged over 100 samples

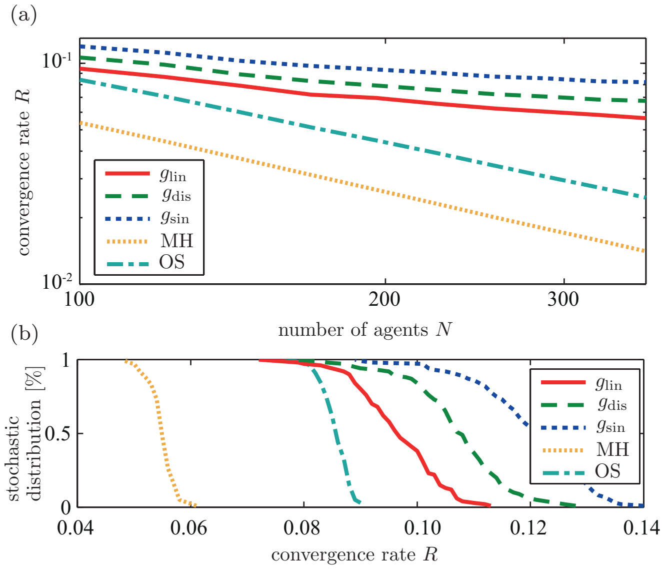

Fig. 5. Comparison of the convergence rate for $\varepsilon = 0.5$ in (a) for increasing number of agents and in (b) by the stochastic distribution for $N = 100$

on the angular position is in fact better than the optimized symmetric design. Recall that optimized symmetric design is a centralized approach and requires knowledge of the graph topology, in contrast to the the weight design approaches presented here. For the asymmetric weighting approach, it appears that the convergence rate is bounded away from zero independent of the size of the network. As expected from the previous results, the sinusoidal function outperforms the other asymmetric weighting methods.

3.2 Performance Dependence on Similarity to Lattice Graphs

It can be expected that the asymmetric weighting methods, based on a generalization of the results for lattice graphs to general graphs, shows best results when the considered general graph is similar to a lattice. This is discussed heuristically in the following. The considered graphs with $N = 100$ are generated by perturbing the positions of a general $\left(\sqrt{N}\times\sqrt{N}\right)$ $2D$-lattice with vertices positioned at intervals of $1/\sqrt{N}$, with Gaussian zero mean noise with standard deviations $\sigma = 0, 0.005, \ldots, 0.05$. The nodes are connected if their distance is $\leq r$, where the radii $r = 0.2, 0.21, \ldots, 0.31$ are considered. For each value of (σ, r), 100 samples were considered and the average taken as result. In Fig. 6(a) the ratio $R_{\text{sin}}/R_{\text{MH}}$is shown depending on σ and r, where R_{sin} is the resulting convergence rate using asymmetric weighting with g_{sin} and R_{MH} obtained using the Metropolis-Hastings method. We consider that ratio, since it is clear that for both methods the absolute value of the convergence rate increases with the number of edges, which we obtain with increasing radius, while the ratio shows the improvement gained with the asymmetric weightings. It can be seen that the asymmetric weighting outperforms the MH most for small r and decreases with rising r, while the influence of σ is hardly visible. The *edit distance* (Gao et al., 2010) is used in graph theory to measure the distance between graphs. In Wilson and Zhu (2008) another quantity, based on the graph spectra, is introduced to measure the similarity of graphs, given a graph $\mathcal{G}$ with the graph spectra $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_N$. Here the graph spectra can be the spectra of respective (normalized) Laplacian matrix, (normalized) adjacency matrix or any other matrix used to represent graphs. The distance of $\mathcal{G}$ to another graph $\bar{\mathcal{G}}$ with $\bar{\mu}_1 \leq \bar{\mu}_2 \leq \ldots \leq \bar{\mu}_N$ is expressed by

$$\left\| \left[\, \mu_1 \; \mu_2 \; \cdots \; \mu_N \,\right]^T - \left[\, \bar{\mu}_1 \; \bar{\mu}_2 \; \cdots \; \bar{\mu}_N \,\right]^T \right\|. \qquad (9)$$

It has been shown in Wilson and Zhu (2008) that this quantity shows linear dependence on the edit distance and can thus be used to quantify the notion of similarity of graphs. Since the edit distance is cost-intensive to calculate, we use the quantity (9). For the graphs considered in Fig. 6(a), their distance to a lattice graph is shown in Fig. 6(b) using (9). Here the spectra of the normalized adjacency matrix are used. Small values imply strong similarity to lattices. The analogy of the two plots confirms the expectation that the performance of the asymmetric weighting depends on the similarity of the respective graph with a lattice.

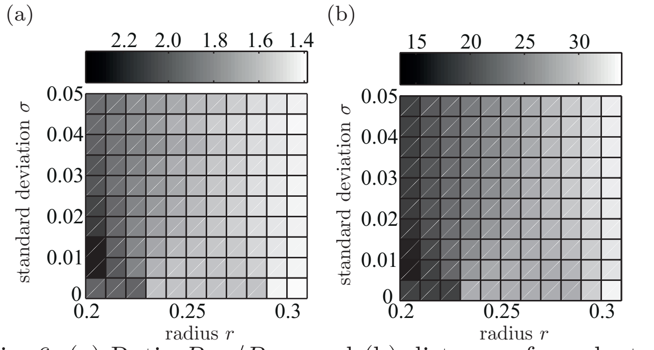

Fig. 6. (a) Ratio $R_{\text{sin}}/R_{\text{MH}}$ and (b) distance of graphs to lattices using (9) averaged over 100 samples

3.3 General Graphs Differing from Lattices

In the previous section it has been shown that the more a graph differs from a lattice, the smaller is the performance increase obtained by the asymmetric weighting method, compared to other approaches as the Metropolis-Hastings method. Furthermore, the geometric graphs considered so far were always related to the position of agents. If the position is not related to the topology or is not known at all, the approach is not useful. Therefore in the following an approach, referred to as *boss algorithm*, is proposed, where without any knowledge of positions we can make use of the obvious advantage of the asymmetric weights, if they give the network a preferred direction and are not distributed randomly.

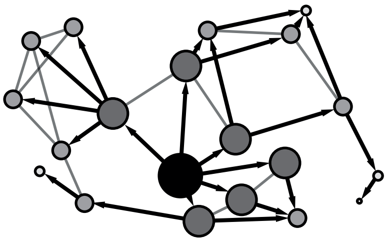

Fig. 7. Example for boss algorithm

The idea is to select a starting node, referred to as *boss* in the following. The direction originating from the boss along the spanning tree, which would be gained in a breadth-first search, is the resulting preferred one, the way back the lowly weighted one. Directions between nodes, within the same distance from the boss, are weighted equally without preference. This is shown exemplarily in Fig. 7, where the large black dot is selected as boss. The preferred directions are shown in black with arcs and the edges without preference in gray. With increasing distance to the boss the nodes get smaller and lighter. We introduce a vector $B(k)$ signifying with $B_i(k) = 1$ that agent i has been assigned as boss. For the preselected boss agent v_b, we initialize $B_b(0) = 1$, while all others start with zero. The values $B_i(k)$ have to be exchanged between neighbors. The weights are updated iteratively, thus (2) changes to $x(k+1) = W(k)x(k)$. Details are shown in Algorithm 1.

Lemma 1. Algorithm (1) leads to asymptotic convergence in x and the weighting matrix W converges in finite time after $\epsilon(v_b)$-steps, where $\epsilon(v_b)$ is the eccentricity of the boss vertex, which is the maximum graph distance between the boss vertex v_b and any other vertices $v_i \in \mathcal{V}$.

Proof. With respect to the convergence of the weighting matrix, it is clear from the algorithm (line 7,8), that if all vertices have been assigned as boss, W does not change anymore, thus has converged. In the first step all vertices with graph distance one from the boss are assigned, in the second step, those with distance two, etc. Thus after $\epsilon(v_b)$ steps, all vertices have been assigned. If the underlying graph, which does not change with k is connected, then $W(k)$ with $W(k)\mathbf{1} = \mathbf{1}$ and $W_{i,j} \geq 0$ is primitive for every k, as well as $\prod_{i=1}^{k} W(i)$, thus x converges. See e.g. Ren and Beard (2005) Theorem 3.10.

Boss-Selection It can be expected that the performance of the boss algorithm crucially depends on the choice of the boss vertex. In the following different possibilities due to different properties are presented and compared. All leader selection methods have in common, that the boss selection has to be centrally done in advance. It seems intuitive, that an *important* vertex in the network should be selected as boss, which reminds of the leader selection problem (Clark et al., 2012, 2013). Thus one possible boss selection is the relaxed leader selection approach for one fixed leader presented in Clark et al. (2012). This method optimizes the choice of an leader with respect to the convergence rate, for a given weighting matrix W by an LMI. Since for the boss algorithm, the resulting weighting matrix W (after it has converged) depends on the choice of the boss vertex and is thus not known in the boss selection step, the Laplacian matrix is thus taken instead. Boss selection according to the maximum and the average degree, as used for comparison of the leader selection in Clark et al. (2012), are also considered. As shown in Lemma 1 W in the boss algorithm has converged after $\epsilon(v_b)$-steps. To consider both extreme cases for $\epsilon(v_b)$, the central vertex with minimum eccentricity in the network and the peripheral vertex with maximum eccentricity are compared as possible boss vertices as well. For small networks up to 100 vertices, hardly a difference between the different boss selection approaches is recognizable. With increasing network size the boss selection according to Clark et al. (2012) and the boss selection to to the maximum degree outperform the others, followed by central vertex choice and the mean degree, while the peripheral vertex choice shows the poorest result. For $N = 450$ the stochastic distribution of the convergence rate, after the weighting matrix has converged, for 100 samples is shown in Fig. 8, where curve labeled "opt. LMI" is the choice due to Clark et al. (2012). In the following we will only consider the boss selection based on Clark et al. (2012) and the boss selection based on the maximum degree, since they led to best results. Note that while the weighting scheme itself is decentralized, the choice of the boss as presented here is made centrally. While the vertex with the maximum degree can easily be determined by solving a certain consensus problem, this is not possible for the solution of an LMI as needed in Clark et al. (2012).

Algorithm 1 Boss Algorithm

```
 1: Initialization: $B_b = 1$, $B_i = 0 \, \forall i \in \{1, \dots, N \setminus b\}$

 2: Iteration for time step $k$:
 3: update $g$
 4: for all $\{ij\} \in \mathcal{E}$ do
 5:     if $B_i(k) = 0 \wedge B_j(k) = 1$ then
 6:         $g_{ij}(k) = \frac{1+\epsilon}{4}$, $g_{ji}(k) = \frac{1-\epsilon}{4}$, $B_i(k+1) = 1$
 7:     else if $B_i(k) = 1 \wedge B_j(k) = 1$ then
 8:         $g_{ij}(k) = g_{ij}(k-1)$, $g_{ji}(k) = g_{ji}(k-1)$
 9:     else $g_{ij}(k) = g_{ji}(k) = \frac{1}{4}$
10:     end if
11: end for
12: update $W$
13: for all $i, j \in \{1, \dots, N\}$ do $W_{ij}(k) = \frac{g_{ij}(k)}{\sum_{l \in \mathcal{N}_i} g_{il}(k)}$
14: end for
15: update $x$
16: $x(k+1) = W(k)x(k)$
```

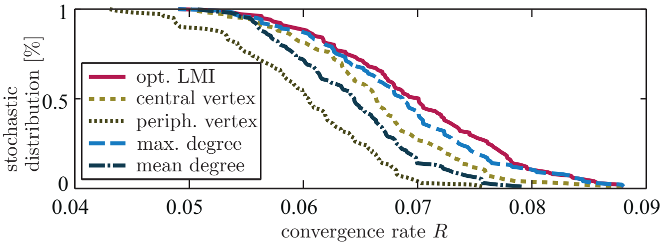

Fig. 8. Boss selection: stochastic distribution of convergence rate

Numerical Comparison with Other Approaches Fig. 9 shows the performance rate for the boss algorithm in comparison with previously presented methods for different values of N. For each value of N, 100 sample graphs were considered. To generate these graphs, the vertices for each

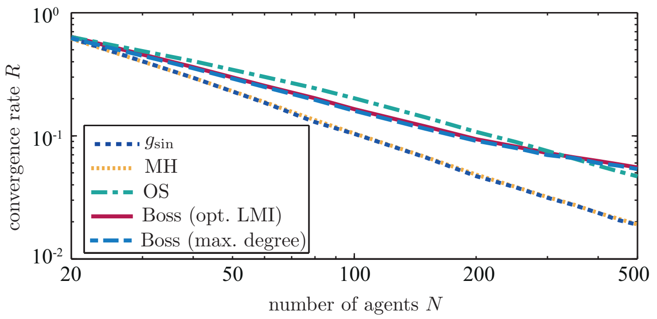

Fig. 9. Comparison of boss-algorithm with other approaches

sample were randomly positioned inside the unit square, and they were connected, if their distance was less than $3/\sqrt{N}$. Notice that the convergence rate achieved with the MH method and the asymmetric weighting with the approximation function $g_{\sin}$ with $\varepsilon = 0.5$ hardly differ, in contrast to Fig. 5(a). This is not surprising in the light of Section 3.2, since the graphs considered here were created completely randomly without any similarity to lattices. The boss-algorithm with the boss selection as in Clark et al. (2012) and the boss selection according to the maximum degree hardly differ as well. For small N the optimal symmetric weighting (OS) outperforms all other considered methods. But with rising N the decrease of the convergence rate (in logarithmic representation) obtained with the the boss algorithm diminishes, while it keeps constant for the optimal symmetric weighting. Thus for $N > 300$ the boss-algorithm leads to better results. One should keep in mind that while the optimal symmetric weights are optimized centrally, with the boss-algorithm the weights are determined in a decentralized manner. For the boss algorithm the convergence rate for the converged W is shown in Fig. 9. The number of iterations to achieve this, depends, as shown in Lemma 1, on the eccentricity of the initial boss-vertex. For both considered boss-selection approaches, the average eccentricity over all samples hardly differs: for $N = 20$, $\epsilon(v_b) = 1,4$, and it increases almost linearly with $\sqrt{N}$, resulting in $\epsilon(v_b) = 8.7$ for $N = 500$.

4. CONCLUSION

In Hao and Barooah (2012) it has been proven that for the consensus protocol for discrete-time single-integrators, for lattice interaction graphs the convergence rate can be bounded away from zero using asymmetric weights. Approximation methods are given to adapt this idea to more general graphs and are confirmed for the example of perturbed lattice graphs. Here new approximation methods are proposed, that outperform the previous one. With the center of line, a quality measure is given that is expected to determine the performance of different approximation functions. Furthermore the conjecture that the approximation degrades in performance if the considered graph differs significantly from a lattice graph, is confirmed when the euclidean norm of the difference of the graph spectra is used as measure of distance. To adapt the idea of asymmetric weighting to enforce a preferred communication direction for general graphs, the boss algorithm is proposed, where the weightings are iteratively updated. All approaches are compared with Metropolis-Hastings weights and optimal symmetric weightings, where the weightings are determined centrally, and, at least for large networks the boss-algorithm outperforms both, while having the advantage that the weights can be determined decentralized without global knowledge of the topology.

REFERENCES

Clark, A., Alomair, B., Bushnell, L., and a. R. Poovendran (2012). Leader selection in multi-agent systems for smooth convergence via fast mixing. In *Proceedings of the 51st IEEE Conference on Decision and Control*, 818–824.

Clark, A., Bushnell, L., and Poovendran, R. (2013). Joint leader and link weight selection for fast convergence in multi-agent systems. In *Proceedings of the American Control Conference 2013*, 3814–3820.

Gao, X., Xiao, B., Tao, D., and Li, X. (2010). A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1), 113–129.

Hao, H. and Barooah, P. (2012). Improving convergence rate of distributed consensus through asymmetric weights. In *Proceedings of the American Control Conference 2012*, 787–792.

Kim, Y., Gu, D., and Postlethwaite, I. (2009). Spectral radius minimization for optimal average consensus and output feedback stabilization. *Automatica*, 45(6), 1379–1386.

Mangoubi, O., Mou, S., Liu, J., and Morse, A.S. (2013). Towards optimal convex combination rules for gossiping. In *Proceedings of the American Control Conference 2013*, 1263–1267.

Murray, R.M. (2007). Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5), 571–583.

Olshevsky, A. and Tsitsiklis, J. (2011). Convergence speed in distributed consensus and averaging. *SIAM Review*, 53(4), 747–772.

Ren, W. and Beard, R.W. (2005). Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5), 655–661.

Ren, W. and Cao, Y. (2011). *Distributed Coordination of Multi-agent Networks*. Springer-Verlag London Limited.

Wilson, R.C. and Zhu, P. (2008). A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9), 2833–2841.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *System and Control Letters*, 53, 65–78.

Yueh, W. and Cheng, S.S. (2008). Explicit eigenvalues and inverses of tridiagonal toeplitz matrices with four perturbed corners. *Anziam Journal*, 49(3), 361–388.