Distributed Gram-Schmidt Orthogonalization based on Dynamic Consensus

Ondrej Slučiak*, Hana Straková†, Markus Rupp*, and Wilfried N. Gansterer†
*Vienna University of Technology, Institute of Telecommunications, (Ondrej.Sluciak@tuwien.ac.at)
†University of Vienna, Research Group Theory and Applications of Algorithms, (Hana.Strakova@univie.ac.at)

***Abstract*—We propose a novel distributed QR factorization algorithm for orthogonalizing a set of vectors in a wireless sensor network. The algorithm originates from the classical Gram-Schmidt orthogonalization which we formulate in a distributed way using the dynamic consensus algorithm. In contrast to existing distributed QR factorization algorithms, all elements of matrices Q and R are computed simultaneously and updated iteratively after each transmission. Assuming synchronous message broadcasting and communication only with neighboring nodes without any central computing unit (fusion center), we prove convergence of the algorithm. We investigate the algorithm in terms of numerical accuracy and we discuss the influence of the initial data distribution on the algorithm performance. Moreover, we provide a comparison with existing distributed QR algorithms in terms of communication cost and memory requirements, and we illustrate the comparison by simulations.**

***Index Terms*—distributed algorithms, Gram-Schmidt orthogonalization, QR factorization, dynamic consensus algorithm, wireless sensor network.**

I. Introduction

We consider the problem of orthogonalizing a set of vectors which are distributed over a network. We focus on standard algorithms for distributed QR factorization based on the Gram-Schmidt orthogonalization [1], a method which besides computing an orthonormal matrix obtains also an upper triangular matrix containing the set of projection coefficients. In contrast to parallel QR factorization algorithms (e.g., [2], [3]), *distributed* QR factorization algorithms designed for loosely coupled distributed systems with independently operating nodes and with possibly unreliable communication links have hardly been investigated [4], [5]. In the signal processing area, QR factorization is used widely in many applications, mostly involving linear least squares (LS) problems [4], [6].

The standard (centralized) approach using a fusion center has severe disadvantages in wireless sensor networks (WSN), such as uneven energy consumption, single point of failure and complex routing. Here, we develop a novel distributed QR factorization based on the dynamic consensus algorithm, which does not require any fusion center and assumes only local communication between neighboring nodes without any global knowledge of the topology.

A. Centralized Classical Gram-Schmidt Orthogonalization

Given a matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m) \in \mathbb{R}^{n\times m}$, $n \geq m$, Classical Gram-Schmidt orthogonalization (*CGS*) [1] computes the matrix $\mathbf{Q} \in \mathbb{R}^{n\times m}$ with orthonormal columns and the upper-triangular matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$, such that $\mathbf{A} = \mathbf{QR}$. Denoting

$$\mathbf{Q} = \begin{pmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \ldots & \mathbf{q}_m\end{pmatrix}$$
$$\mathbf{R} = \begin{pmatrix}\langle \mathbf{q}_1, \mathbf{a}_1\rangle & \langle \mathbf{q}_1, \mathbf{a}_2\rangle & \ldots & \langle \mathbf{q}_1, \mathbf{a}_m\rangle \\ 0 & \langle \mathbf{q}_2, \mathbf{a}_2\rangle & \langle \mathbf{q}_2, \mathbf{a}_3\rangle & \ldots \\ \vdots & \vdots & \ddots & \ldots \\ 0 & \ldots & 0 & \langle \mathbf{q}_m, \mathbf{a}_m\rangle\end{pmatrix} \quad (1)$$

we have

$$\mathbf{q}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}, \qquad i = 1, 2, \ldots, m, \quad (2)$$

and

$$\mathbf{u}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{q}_j, \mathbf{a}_i\rangle}{\langle \mathbf{q}_j, \mathbf{q}_j\rangle}\mathbf{q}_j, \quad i = 1, 2, \ldots, m, \quad (3)$$

where $\|\mathbf{u}\| = \sqrt{\sum_{j=1}^{n} u_j^2}$ and $\langle \mathbf{q}, \mathbf{a}\rangle = \sum_{j=1}^{n} q_j a_j$.

B. Existing distributed methods

A straightforward redefinition of the CGS suitable for WSN results from performing the summation in the norms (cf. (2)) and the inner products (cf. (3)) in a distributed way using any distributed data aggregation algorithm (e.g., average consensus [7], gossiping algorithms [8], etc.).

Nevertheless, to our knowledge, all existing distributed algorithms for orthogonalizing a set of vectors are based on the gossip-based *push-sum algorithm* [8]. Specifically, in [4] authors used distributed CGS based on gossiping for solving a distributed LS problem and in [5] a gossip-based distributed algorithm for *modified* Gram-Schmidt orthogonalization (*MGS*) was designed and analyzed. A slight modification of the latter algorithm was introduced in [9], which we use for comparison in this paper. We denote the two gossip-based distributed Gram-Schmidt orthogonalization algorithms as *G–CGS* [4] and *G–MGS* [9], respectively. Both methods compute columns of the matrix $\mathbf{Q}$ *sequentially*. In contrast, the distributed algorithm introduced in this paper approximates all elements of the orthonormal basis $\mathbf{Q}$ *simultaneously* and as the algorithm proceeds the values at all nodes are iteratively refined, eventually converging to the desired $\mathbf{Q}$ and $\mathbf{R}$ matrices. Therefore, it delivers estimates of $\mathbf{Q}$ and $\mathbf{R}$ at any intermediate stage.

In this paper, we first review the concept of static and dynamic consensus in Sec. II where we also discuss the convergence of dynamic consensus. In Sec. III we describe *DC–CGS*,

This work was supported by the Austrian Science Fund (FWF) under project grants S10608 and S10611-N13 within the National Research Network SISE.

the new distributed QR factorization algorithm based on the dynamic consensus and prove its convergence. We summarize the analysis and compare it to other distributed approaches in Sec. IV. Sec. V concludes the paper.

Notation and terminology. In what follows, N denotes the (known) number of nodes in the network, d_k the node degree of node k, $\bar{d}$ the average node degree of the network, and t a discrete time (iteration) index; $\mathbf{1}$ represents the vector of all ones; $\mathbf{X}_k$ represents the part of a matrix $\mathbf{X}$ stored at node k (it may be a matrix or a vector). For the sake of clarity, we abuse notation and denote element-wise division of two vectors as $\mathbf{z} = \frac{\mathbf{x}}{\mathbf{y}} \equiv \frac{x_i}{y_i}, \forall i$, element-wise multiplication of two vectors as $\mathbf{z} = \mathbf{x} \circ \mathbf{y} \equiv x_i y_i, \forall i$, and of two matrices as $\mathbf{Z} = \mathbf{X} \circ \mathbf{Y}$. The operation $\mathbf{X} \circledast \mathbf{Y} \equiv \mathbf{x}_{1:i} \circ \mathbf{y}_{i+1}$ ($\forall i = 1, \ldots, m-1$)[1], where $\mathbf{x}_{1:i}$ stands for a matrix of columns $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i]$.

II. Consensus algorithms

We model a WSN by synchronously working nodes which broadcast their data into their neighborhood. The WSN is considered to be static, connected, and with error-free transmissions. Although the practicality of synchronicity can be discussed[10], we note that it is not an unrealizable assumption [11].

Two consensus algorithms for computing an *average* of values distributed in a network are presented here. Note that both algorithms can be easily adapted to computing a *sum* in a distributed way, if the final average value is multiplied by the total number of nodes N in the network.

A. Static average consensus algorithm

The "*static*" distributed average consensus algorithm in a WSN computes at each node an estimate of the global *mean* (average) of distributed initial data $\mathbf{x}(0)$. In every iteration t each node updates its estimate using the weighted data received from its neighbors. From the global (network) point of view,

$$\mathbf{x}(t) = \mathbf{W}\mathbf{x}(t-1). \tag{4}$$

The selection of *weight matrix* $\mathbf{W}$, representing the connections in the network, crucially influences the convergence of the average consensus algorithm [7], [12]. The main condition for the algorithm to converge is that the largest eigenvalue of $\mathbf{W}$ is equal to 1, i.e., $\lambda_{\max} = 1$, with corresponding eigenvector $c\mathbf{1}$. According to the Perron-Frobenius theorem this condition is satisfied for any connected network [13].

B. Dynamic average consensus algorithm

In contrast to the static average consensus algorithm, which computes the mean of constant values, the *dynamic* consensus algorithm is able to track the mean of a time-varying signal. Similarly to [14], where a continuous version of the dynamic average consensus has been defined, we define a discrete dynamic average consensus algorithm[2] and prove its convergence.

[1]Operation $\mathbf{X} \circledast \mathbf{Y}$ represents the combinations of vectors, which represent the stacked approximated inner products (off-diagonal inner products of the matrix $\mathbf{R}$ (cf. (1))); e.g., $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \Rightarrow \mathbf{X} \circledast \mathbf{Y} = (\mathbf{x}_1 \circ \mathbf{y}_2, \mathbf{x}_1 \circ \mathbf{y}_3, \mathbf{x}_2 \circ \mathbf{y}_3)$.

[2]Generalizations and different types of dynamic average consensus algorithms can be found in [15]–[17].

Theorem 1. *For a (strongly) connected network, assuming a bounded input signal* $\mathbf{s}(t) = (s_1(t), \ldots, s_k(t), \ldots, s_N(t))^\top$ *with decaying time differences* $\triangle\mathbf{s}(t) = \mathbf{s}(t) - \mathbf{s}(t-1)$, *the internal state* $x_k(t)$ *at node* k, *following the update equation*

$$\mathbf{x}(t) = \mathbf{W}\left[\mathbf{x}(t-1) + \triangle\mathbf{s}(t)\right], \tag{5}$$

with $\mathbf{x}(0) = \mathbf{s}(0)$, *converges to the instant average* $\bar{\mathbf{s}}(t)$ *of the input signal* $\mathbf{s}(t)$, *i.e.,*

$$\mathbf{x}(t) = \frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{s}(t) \equiv \bar{\mathbf{s}}(t), \text{ as } t \to \infty. \tag{6}$$

Proof: Let us define the error function:

$$\mathbf{e}(t) = \mathbf{x}(t) - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{s}(t). \tag{7}$$

Using $\mathcal{Z}$-transform, we can rewrite (7) in the $\mathcal{Z}$-domain as $\boldsymbol{E}(z) = \boldsymbol{X}(z) - 1/N\mathbf{1}\mathbf{1}^\top\boldsymbol{S}(z)$. Thus, transforming (5), we get

$$\begin{aligned}\boldsymbol{E}(z) &= \mathbf{W}(z^{-1}\boldsymbol{X}(z) + \mathbf{x}(0) + \boldsymbol{S}(z) - z^{-1}\boldsymbol{S}(z) - \mathbf{s}(0)) - \tfrac{1}{N}\mathbf{1}\mathbf{1}^\top\boldsymbol{S}(z) = \\ &= \left(\mathbf{I} - z^{-1}\mathbf{W}\right)^{-1}\mathbf{W}\left(\mathbf{I} - z^{-1}\mathbf{I}\right)\boldsymbol{S}(z) - \tfrac{1}{N}\mathbf{1}\mathbf{1}^\top\boldsymbol{S}(z).\end{aligned}$$

From the fact that the maximal eigenvalue of $\mathbf{W}$, $\lambda_{\max} = 1$, with eigenvector $\mathbf{v}_{\max} = 1/\sqrt{N}\mathbf{1}$ [13], we can see that the error transfer function is

$$\begin{aligned}H_{es}(z) &= \tfrac{\boldsymbol{E}(z)}{\boldsymbol{S}(z)} = \left(\mathbf{I} - z^{-1}\mathbf{W}\right)^{-1}\mathbf{W}\left(\mathbf{I} - z^{-1}\mathbf{I}\right) - \tfrac{1}{N}\mathbf{1}\mathbf{1}^\top \\ &= \mathbf{V}\begin{pmatrix} 0 & 0 & \cdots \\ 0 & \frac{\lambda_2(1-z^{-1})}{1-\lambda_2 z^{-1}} & \cdots \\ & & \ddots \end{pmatrix}\mathbf{V}^{-1},\end{aligned}$$

where $\mathbf{V}$ contains the eigenvectors of $\mathbf{W}$. Since $|\lambda_i| < 1$ for $i = 2, \ldots, N$, all poles of the transfer function are inside the unit circle, thus, the transfer function is stable. Using the final value theorem[3], for any bounded input signal $\mathbf{s}(t)$ (with at most one pole at 1 in the $\mathcal{Z}$-domain) we obtain $\lim_{z\to 1}(1-z^{-1})\boldsymbol{E}(z) = 0$. Thus, the error $\mathbf{e}(t) \to 0$ as $t \to \infty$. ■

III. Distributed QR factorization based on the dynamic consensus algorithm – DC–CGS

As mentioned in Sec. I-B, the Gram-Schmidt orthogonalization process can be computed in a distributed way by using any distributed aggregation algorithm. We refer to a CGS based on the static consensus as *SC–CGS*. SC–CGS as well as G–CGS [4] and G–MGS [9] have an important drawback. In Gram-Schmidt orthogonalization methods, the computation of the norms $\|\mathbf{u}_i\|$ and the inner products $\langle \mathbf{q}_j, \mathbf{a}_i \rangle, \langle \mathbf{q}_j, \mathbf{q}_j \rangle$ depends on the norms and inner products computed from the previous columns of the input matrix $\mathbf{A}$. Therefore, each node k must wait until the estimates of the previous norms $\|\mathbf{u}_j\|$ ($j < i$) have achieved an acceptable accuracy before processing the next norm $\|\mathbf{u}_i\|$.

We design a novel approach overcoming this drawback by using the dynamic consensus algorithm which processes a time-varying input values. If we rewrite equations (2) and (3)

[3]The final value theorem in the $\mathcal{Z}$-domain: $\lim_{t\to\infty}\mathbf{e}(t) = \lim_{z\to 1}(1-z^{-1})\boldsymbol{E}(z)$.

implicitly containing these dependencies, we obtain

$$\hat{\mathbf{q}}_i(t) = \frac{\hat{\mathbf{u}}_i(t)}{\sqrt{N\tilde{\mathbf{u}}_i(t-1)}}, \qquad i = 1, \ldots, m, \tag{8}$$

$$\hat{\mathbf{u}}_i(t) = \mathbf{a}_i - \mathbf{p}_i(t), \qquad i = 1, \ldots, m, \tag{9}$$

where $\tilde{\mathbf{u}}_i(t)$ is the approximation of $\|\mathbf{u}_i\|^2$ at time t and

$$\mathbf{p}_i(t) = \sum_{j=1}^{i-1} \frac{\tilde{\mathbf{p}}_j^{(2)}(t-1) \circ \hat{\mathbf{q}}_j(t-1)}{\tilde{\mathbf{q}}_j(t-1)}, \tag{10}$$

with $\tilde{\mathbf{p}}_j^{(2)}(t)$ being the approximation of the off-diagonal inner-products $\langle \mathbf{q}_j, \mathbf{a}_i \rangle$ of the matrix $\mathbf{R}$ (cf. (1)) and $\tilde{\mathbf{q}}_j(t)$ the approximation of $\langle \mathbf{q}_j, \mathbf{q}_j \rangle$ at time t. Similarly, we define $\tilde{\mathbf{p}}_i^{(1)}(t)$ to be the approximation of $\langle \mathbf{q}_i, \mathbf{a}_i \rangle$. Using the dynamic average consensus algorithm from Sec. II-B, we show that $\tilde{\mathbf{u}}_i(t), \tilde{\mathbf{q}}_i(t), \tilde{\mathbf{p}}_i^{(1)}(t)$, and $\tilde{\mathbf{p}}_i^{(2)}(t)$ converge to $\|\mathbf{u}_i\|^2, \langle \mathbf{q}_j, \mathbf{q}_j \rangle, \langle \mathbf{q}_i, \mathbf{a}_i \rangle$, and $\langle \mathbf{q}_j, \mathbf{a}_i \rangle$, respectively.

DISTRIBUTED QR FACTORIZATION BASED ON DYNAMIC AVERAGE CONSENSUS ALGORITHM (DC–CGS)

- Input matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m) \in \mathbb{R}^{n \times m}$ with $n \geq m$ is distributed row-wise across nodes. If $n > N$, some nodes store more than one row. Each node computes the rows of $\mathbf{Q}$ corresponding to the stored rows of $\mathbf{A}$ and an estimate of the whole matrix $\mathbf{R}$. (used indices i, l: $i = 1, \ldots, m$; $l(i) = l(i-1) + i, l(0) = 0$.)
- Initialization:

$$\begin{aligned} \tilde{\mathbf{U}}(0) &= \mathbf{A} \circ \mathbf{A}, & \hat{\mathbf{U}}(0) &= \mathbf{A}, \\ \tilde{\mathbf{Q}}(0) &= \mathbf{A} \circ \mathbf{A}, & \hat{\mathbf{Q}}(0) &= \mathbf{A}, \\ \tilde{\mathbf{P}}^{(1)}(0) &= \mathbf{A} \circ \mathbf{A}, & \tilde{\mathbf{P}}^{(2)}(0) &= \mathbf{A} \circledast \mathbf{A} \end{aligned}$$

- Repeat for $t = 1, \ldots$
  1) Compute locally at each node ($i = 1 \ldots m$)

$$\begin{aligned} \mathbf{p}_i(t) &= \textstyle\sum_{j=1}^{i-1} \frac{\tilde{\mathbf{p}}_j^{(2)}(t-1) \circ \hat{\mathbf{q}}_j(t-1)}{\tilde{\mathbf{q}}_j(t-1)} \\ \hat{\mathbf{u}}_i(t) &= \mathbf{a}_i - \mathbf{p}_i(t) \\ \hat{\mathbf{q}}_i(t) &= \frac{\hat{\mathbf{u}}_i(t)}{\sqrt{N\tilde{\mathbf{u}}_i(t-1)}} \end{aligned}$$

  2) At each node k store ($k = 1, \ldots, N$)

$$\mathbf{Q}_k(t) = (\hat{q}_{k,1}(t), \hat{q}_{k,2}(t), \ldots, \hat{q}_{k,m}(t)), \quad \text{and}$$

$$\mathbf{R}_k(t) = N \begin{pmatrix} \tilde{p}_{k,1}^{(1)}(t) & \tilde{p}_{k,1}^{(2)}(t) & \cdots & \tilde{p}_{k,l(m)-m+1}^{(2)}(t) \\ 0 & \tilde{p}_{k,2}^{(1)}(t) & \tilde{p}_{k,3}^{(2)}(t) & \cdots \\ \vdots & \vdots & \ddots & \cdots \\ 0 & \cdots & 0 & \tilde{p}_{k,m}^{(1)}(t) \end{pmatrix}$$

  3) Aggregate data for the dynamic consensus (cf. (5))

$$\begin{aligned} \Psi^{(1)} &= \tilde{\mathbf{U}}(t-1) &&+ \hat{\mathbf{U}}(t) \circ \hat{\mathbf{U}}(t) - \hat{\mathbf{U}}(t-1) \circ \hat{\mathbf{U}}(t-1) \\ \Psi^{(2)} &= \tilde{\mathbf{Q}}(t-1) &&+ \hat{\mathbf{Q}}(t) \circ \hat{\mathbf{Q}}(t) - \hat{\mathbf{Q}}(t-1) \circ \hat{\mathbf{Q}}(t-1) \\ \Psi^{(3)} &= \tilde{\mathbf{P}}^{(1)}(t-1) &&+ \hat{\mathbf{Q}}(t) \circ \mathbf{A} - \hat{\mathbf{Q}}(t-1) \circ \mathbf{A} \\ \underbrace{\Psi^{(4)}} &= \underbrace{\tilde{\mathbf{P}}^{(2)}(t-1)} &&+ \underbrace{\hat{\mathbf{Q}}(t) \circledast \mathbf{A} - \hat{\mathbf{Q}}(t-1) \circledast \mathbf{A}} \\ \Psi(t) &= \mathbf{X}(t-1) &&+ \Delta\mathbf{S}(t) \end{aligned}$$

  4) If $n > N$, sum $\Psi(t)$ locally at the nodes and broadcast $\Psi(t) = \left(\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}, \Psi^{(4)}\right)$ to the neighbors, i.e.,

$$\left(\tilde{\mathbf{U}}(t), \tilde{\mathbf{Q}}(t), \tilde{\mathbf{P}}^{(1)}(t), \tilde{\mathbf{P}}^{(2)}(t)\right) = \mathbf{W}\Psi(t).$$

Proof of convergence: For $i = 1$, we obtain $\hat{\mathbf{u}}_1(t) = \mathbf{a}_1$, and thus, as $t \to \infty$, $\tilde{\mathbf{u}}_1(t) = 1/N\|\mathbf{a}_1\|^2$ and thus also, $\hat{\mathbf{q}}_1(t) = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}$, $\tilde{\mathbf{q}}_1(t) = 1/N\mathbf{1}$, $\tilde{\mathbf{p}}_1^{(1)}(t) = 1/N\|\mathbf{a}_1\|$, and $\tilde{\mathbf{p}}_1^{(2)}(t) = 1/N\langle \mathbf{a}_1, \mathbf{a}_2 \rangle$.

Furthermore, for all $i > 1$, we see that the elements depend only on the first column ($i = 1$), e.g., Eq. (9), $\hat{\mathbf{u}}_2(t) = \mathbf{a}_2 - \frac{\tilde{\mathbf{p}}_1^{(2)}(t-1) \circ \hat{\mathbf{q}}_1(t-1)}{\tilde{\mathbf{q}}_1(t-1)}$. Thus, since the first elements, $\mathbf{s}_1(t)$ (see step 3 of the algorithm), are bounded (constant), all subsequent elements $\mathbf{s}_i$ are bounded as well, $\lim_{t\to\infty} \Delta\mathbf{S}(t) \to 0$, and thus, the conditions of Theorem 1 are fulfilled. Therefore, as the elements $\tilde{\mathbf{u}}_i(t)$, $\tilde{\mathbf{q}}_i(t)$, $\tilde{\mathbf{p}}_i^{(1)}(t)$, $\tilde{\mathbf{p}}_i^{(2)}$ converge, the algorithm converges. ■

IV. SIMULATIONS

In the simulations we consider a connected WSN with $N = 30$ nodes. We explore the behavior for various topologies: *fully connected* (each node is connected to every other node), *regular* (each node has the same degree d) and *geometric* (each (randomly deployed) node is connected to all nodes in some radius ρ (WSN model)). The random input matrix $\mathbf{A} \in \mathbb{R}^{300\times100}$ has uniformly distributed elements from the interval $[0, 1]$ and a low condition number $\kappa(\mathbf{A}) = 35.7$.

A. Orthogonality and factorization error

As performance metrics we use the *relative factorization error*, $\|\mathbf{A}-\mathbf{QR}\|/\|\mathbf{A}\|$, which measures the accuracy of the QR decomposition, and the *orthogonality error*, $\|\mathbf{I}-\mathbf{Q}^\top\mathbf{Q}\|$, which measures the orthogonality of the matrix $\mathbf{Q}$. The simulation results for a geometric topology with an average node degree $\bar{d} = 8.533$ are depicted in Fig. 1. Since both errors behave almost identically and since each node k can compute a local factorization error $\|\mathbf{A}_k - \mathbf{Q}_k\mathbf{R}_k\|/\|\mathbf{A}_k\|$ from its local data,

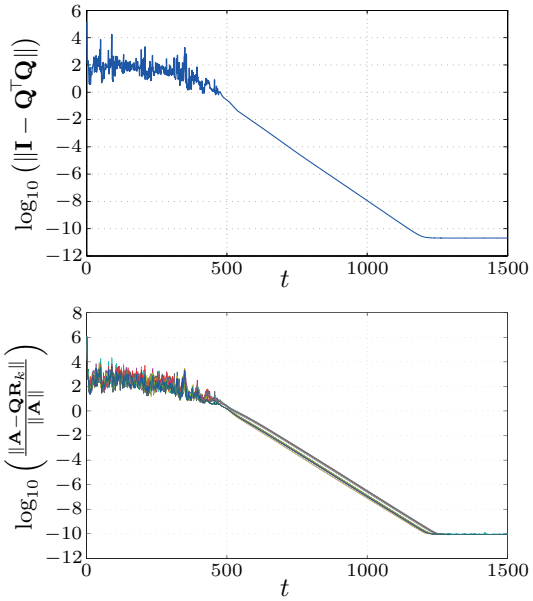


Fig. 1: Orthogonality and factorization error for each node k for a geometric topology with $\bar{d} = 8.533$.

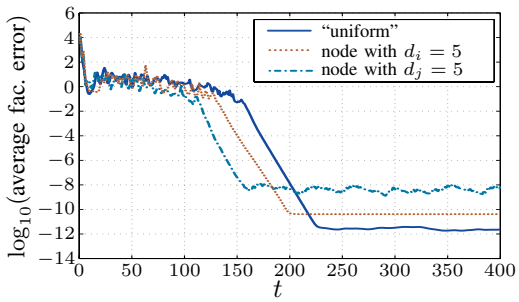
we conjecture that such local error evaluation might be used also as a local convergence criterion in practice.

B. Initial data distribution

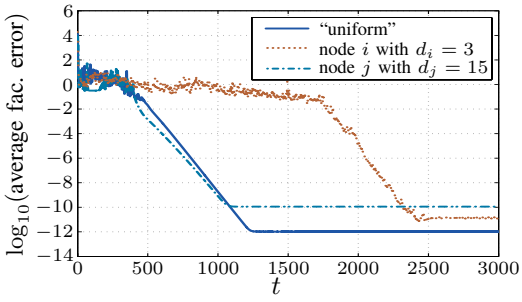
If $n > N$, some nodes store more than one row of $\mathbf{A}$. Thus, before doing distributed summation, every node has to locally sum the values from its local rows.

Simulations show that the convergence behavior of DC–CGS strongly depends on the distribution of the rows across the network (see Fig. 2). We investigate the following cases: (1) each node stores ten rows of $\mathbf{A}$ ("uniform"); (2) 271 rows are stored in a node with the lowest degree, the other 29 rows in the remaining 29 nodes; (3) 271 rows are stored in a node with the highest degree, the rest in the remaining 29 nodes. Fig. 2 shows the relative factorization error averaged over all nodes.

We observe that not only the initial distribution of the data influences the convergence behavior but also the topology of the underlying network. In the case of a regular topology, Fig. 2(a), the influence of the distribution is small and we can see that the difference between the nodes comes only from the variance of the values in the input matrix $\mathbf{A}$. On the other hand, in case of a highly irregular geometric topology, see Fig. 2(b), where the node with most neighbors stores most of the data, the algorithm converges much faster than in the case when the most of the data are stored in a node with only few neighbors.



(a) Regular topology ($\bar{d} = 5$).



(b) Geometric topology ($\bar{d} = 8.533$).

Fig. 2: Convergence for networks with different topology and initial data distribution: either all nodes store the same amount of data ("uniform") or most of the data is stored in one node.

C. Performance comparison with existing algorithms

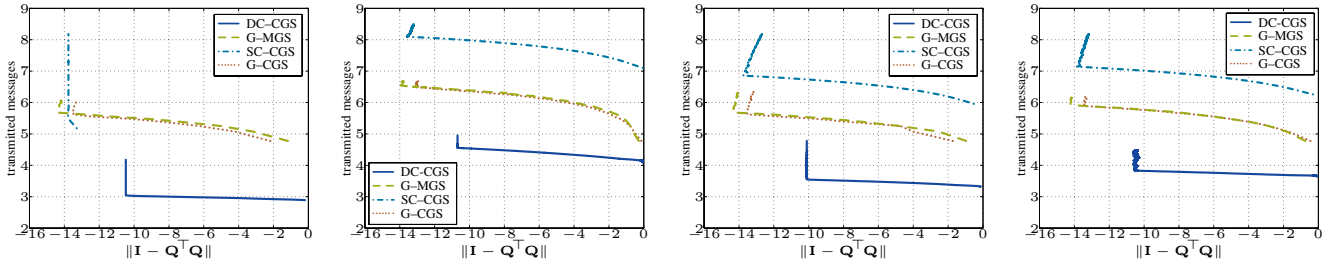
We compare our new DC–CGS algorithm with SC–CGS, G–CGS and G–MGS introduced in Sec. I-B. Although all approaches have iterative aspects, the cost per iteration strongly differs for each algorithm. Thus, instead of providing a comparison in terms of number of iterations to converge, we compare the communication cost needed for achieving a certain accuracy of the result. We investigate the total number of messages sent as well as the total amount of data (real numbers) exchanged.

Simulation results for various topologies are shown in Figs. 3 and 4. We can see that the gossip-based approaches exchange, in general, less data (Fig. 4), but since their message size is much smaller than in DC–CGS, the total number of messages sent is higher (Fig. 3). Because the message size of SC–CGS is even smaller than in the gossip-based approaches, it sends the highest number of messages. Since the energy consumption in a WSN is mostly influenced by the number of transmissions, thus, it is better to transmit as few messages as possible (with any payload size), therefore, DC–CGS is the most suitable method for a WSN scenario. However, we notice that in many cases DC–CGS does not achieve the same final accuracy of the result as the other methods. Investigation and improvement of this aspect is a work in progress. Note that in a fully connected network SC–CGS delivers a highly accurate result from the beginning, because within the first iterations all nodes exchange the required information with all other nodes.

In Tab. I we summarize the communication cost and local memory requirements of the algorithms. Due to considered wireless setting, the communication cost represents the number of messages (data) each node needs to transmit. However, due to different parameters it is difficult to rank the approaches in a general case. The requirements depend especially on the topology of the underlying network, the number of iterations I to converge in consensus-based algorithms, or the number of rounds R needed for convergence of push-sum in the gossip-based approaches. For example, in a *fully connected* network $R = O(\log N)$ [8], $I = 1$. Thus, the SC–CGS requires $O(m^2 N)$ messages sent as well as data exchanged, whereas gossip-based approaches need $O(mN \log N)$ messages and $O(m^2 N \log N)$ data. Note that, G–CGS and G–MGS have theoretically identical communication cost, however, G–MGS is numerically more stable and achieves a higher accuracy (see Figs. 3 and 4). In case of the DC–CGS and a fully connected network, we can view the DC–CGS as to be m consequent static consensus algorithms (one for each column), thus, $I = O(m)$, and the number of transmitted messages is $O(mN)$ and data $O(m^3 N)$. Nevertheless, theoretical convergence properties of the DC–CGS have to be investigated further.

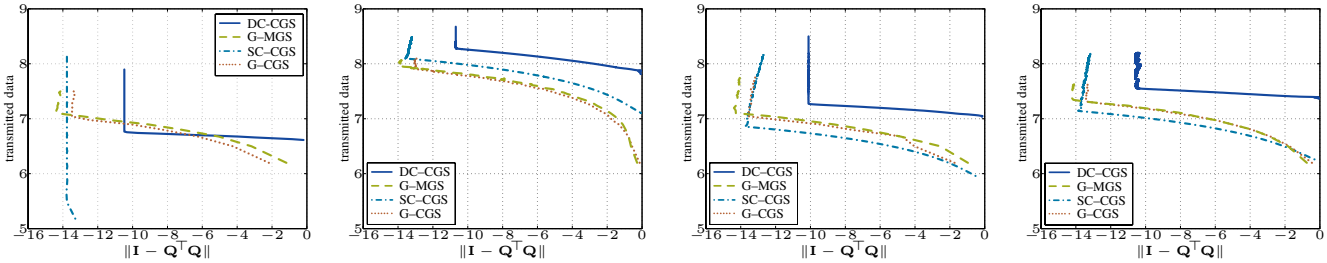
	Number of messages	Amount of data (real numbers)	Local memory requirements
DC–CGS	$N \cdot I$	$N \cdot I \cdot \frac{m^2+5m}{2}$	$O(mn/N + m^2)$
SC–CGS	$N \cdot I \cdot \frac{(m+1)m}{2}$	$N \cdot I \cdot \frac{(m+1)m}{2}$	$O(mn/N + m^2)$
G–CGS [4]	$N \cdot R \cdot (2m-1)$	$N \cdot R \cdot \frac{m^2+5m-2}{2}$	$O(nm/N)$
G–MGS [9]	$N \cdot R \cdot (2m-1)$	$N \cdot R \cdot \frac{m^2+5m-2}{2}$	$O(nm/N)$

TABLE I: Comparison of various distributed QR factorization algorithms. I – number of iterations, R – number of rounds per push-sum, N – number of nodes, m – number of columns of the input matrix.



(a) Fully connected topology. (b) Geometric topology ($\bar{d} = 8.53$). (c) Geometric topology ($\bar{d} = 24.46$). (d) Regular topology ($\bar{d} = 5$).

Fig. 3: Number of transmitted messages vs. orthogonality error (both axes are in logarithmic scale $\log_{10}$).



(a) Fully connected topology. (b) Geometric topology ($\bar{d} = 8.53$). (c) Geometric topology ($\bar{d} = 24.46$). (d) Regular topology ($\bar{d} = 5$).

Fig. 4: Number of transmitted real numbers (data) vs. orthogonality error (both axes are in logarithmic scale $\log_{10}$).

V. Conclusion

We presented a novel distributed algorithm for QR decomposition based on dynamic consensus and proved its convergence. In contrast to existing methods, which compute the columns of the resulting matrix $\mathbf{Q}$ consecutively, our method iteratively refines all elements at once. Thus, in any moment the algorithm can deliver an estimate of both matrices $\mathbf{Q}$ and $\mathbf{R}$. Simulations showed that by using this approach we were able to significantly reduce the number of transmitted messages for achieving a certain accuracy compared to state-of-the-art distributed QR factorization algorithms, thus making it suitable for energy-constrained WSNs. Based on empirical observation, we argue that the evaluation of the local factorization error at each node might lead to a stopping criterion for the algorithm. We also observed that the numerical accuracy of the result is influenced by several factors, such as topology of the network, selected weights and the elements of the input matrix. Optimization of these parameters, together with theoretical bounds on the convergence rate, are topics of future work.

References

[1] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[2] A. Buttari, J. Langou, J. Kurzak, and J. Dongarra, "Parallel tiled QR factorization for multicore architectures," in *Proceedings of the 7th International Conference on Parallel Processing and Applied Mathematics*. Springer-Verlag, 2008, pp. 639–648.

[3] J. Demmel, L. Grigori, M. F. Hoemmen, and J. Langou, "Communication-optimal parallel and sequential QR and LU factorizations," no. UCB/EECS-2008-89, EECS Department, University of California, Berkeley, Tech. Rep., 2008.

[4] C. Dumard and E. Riegler, "Distributed sphere decoding," in *International Conference on Telecommunications ICT '09*, 2009, pp. 172–177.

[5] H. Straková, W. N. Gansterer, and T. Zemen, "Distributed QR factorization based on randomized algorithms," in *Proceedings of the 9th International Conference on Parallel Processing and Applied Mathematics, Part I*, ser. Lecture Notes in Computer Science, vol. 7203, 2012, pp. 235–244.

[6] G. Tauböck, M. Hampejs, P. Švač, G. Matz, F. Hlawatsch, and K. Gröchenig, "Low-Complexity ICI/ISI Equalization in Doubly Dispersive Multicarrier Systems Using a Decision-Feedback LSQR Algorithm," *Signal Processing, IEEE Trans. on*, vol. 59, no. 5, pp. 2432–2436, 2011.

[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

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

[9] H. Straková and W. N. Gansterer, "A distributed eigensolver for loosely coupled networks," in *21st Euromicro Int. Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, Belfast, N. Ireland, Mar. 2013, [Online] Available: http://eprints.cs.univie.ac.at/3483/.

[10] O. Slučiak and M. Rupp, "Reaching consensus in asynchronous WSNs: Algebraic approach," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Rep., May 2011, pp. 3300–3303.

[11] F. Sivrikaya and B. Yener, "Time Synchronization in Sensor Networks: A Survey," *IEEE Network Mag.–spec. issue on 'Ad Hoc Networking: Data Communications & Topology Control'*, vol. 18, no. 4, pp. 45–50, 2004.

[12] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, pp. 65–78, 2004.

[13] C. Godsil and G. F. Royle, *Algebraic Graph Theory*. Springer, 2001.

[14] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, Jul. 2005.

[15] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[16] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. Int. Conf. Inf. Fusion (FUSION 2008)*, Cologne, Germany, Jun.–Jul. 2008, pp. 152–157.

[17] O. Slučiak, O. Hlinka, M. Rupp, F. Hlawatsch, and P. M. Djurić, "Sequential likelihood consensus and its application to distributed particle filtering with reduced communications and latency," in *Proc. 45th Asilomar Conf. on Sig., Syst., Comp.*, Pacific Grove, CA, USA, Nov. 2011, pp. 1766–1770.