Reduced Overhead Distributed Consensus-Based Estimation Algorithm

Ban-Sok Shin, Henning Paul, Dirk Wübben and Armin Dekorsy
Department of Communications Engineering
University of Bremen
Bremen, Germany
Email: {shin, paul, wuebben, dekorsy}@ant.uni-bremen.de

***Abstract*—We consider a cooperation among nodes in a network which aim to reconstruct a common broadcast message. Distributed estimation algorithms are highly suited for such a scenario. Nevertheless, communication overhead due to an exchange of variables among nodes can be problematic concerning energy efficiency and costs. In this paper, we present a Reduced Overhead Distributed Consensus-based Estimation (RO-DiCE) algorithm exhibiting a significantly reduced communication overhead in comparison to its unreduced version, the DiCE algorithm. However, by reducing the overhead a possible degradation in estimation performance of the algorithm is introduced. We investigate the RO-DiCE algorithm in terms of communication overhead, convergence behavior and error rate performance for different network topologies. We will show that for a full mesh topology the RO-DiCE algorithm is identical to the DiCE algorithm.**

I. Introduction

In a cooperative scenario, a network consisting of connected nodes aims to recover a common message broadcast by a detached source. One approach for cooperative reconstruction is a centralized scheme: A central node, also termed Fusion Center (FC), processes the received information of all nodes jointly in order to estimate the broadcast message [1], [2]. Obviously, this scheme lacks robustness since an outage of the FC would corrupt the whole estimation process. A more robust scheme can be achieved by a distributed reconstruction of a common message via In-Network-Processing (INP). Here, no central node is needed since each node calculates an estimate of the broadcast message by incorporating information from neighboring nodes in the network improving the robustness e.g., against link failures. In [3], we presented a distributed consensus-based estimation (DiCE) algorithm derived from a Least Squares (LS) optimization problem. This algorithm establishes a scheme where nodes in a network exchange information with each other in order to reach the centralized solution iteratively. Other distributed consensus-based algorithms were proposed, e.g., in [4], [5]. Compared to [4], the DiCE algorithm showed a faster convergence and higher robustness against link failures while keeping the same estimation performance.

The application of the DiCE algorithm for future mobile communication systems is topic of current research. In the EU FP7 ICT project iJOIN, dense networks consisting of small cells, termed iJOIN Small Cells (iSCs), are introduced. This deployment enables cooperation among small cells aiming to reconstruct the same message e.g., of one user equipment (UE). In [6], an application of the DiCE algorithm to Multi-User-Detection (MUD) in such a scenario is investigated. Further investigations of the DiCE algorithm concerning erroneous inter-node links were discussed in [7]. Both [4] and [7] describe the necessity of exchanging quantities in a unicast fashion between the nodes. Obviously, this exchange causes a high communication overhead in the network since each quantity depends on the transmitting and receiving node. In contrast, for a broadcast transmission quantities depend on the transmitting node only.

In this paper, we present a reduced overhead version of the DiCE algorithm in [3], the RO-DiCE, which avoids the exchange of quantities in a unicast fashion. We will compare the RO-DiCE algorithm to the DiCE algorithm in terms of communication overhead, convergence behavior and error rate performance.

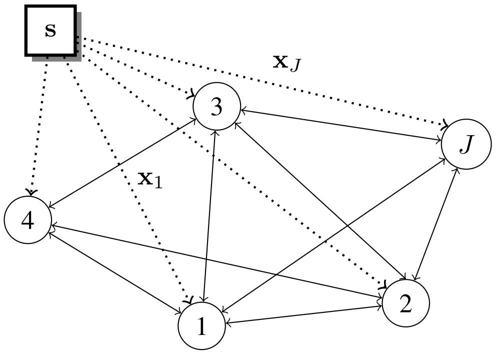

Fig. 1. Network consisting of J connected nodes, each receiving a different observation $\mathbf{x}_j$ of the original message $\mathbf{s}$. Dotted lines indicate an access link, solid lines an inter-node link.

II. System Model

Fig. 1 depicts the basic transmission scenario. A message $\mathbf{s} \in \mathbb{R}^{N \times 1}$ is broadcast by a source and received by J connected nodes which are set up to a network. The network is described by means of a graph $\mathcal{G} := \{\mathcal{E}, \mathcal{J}\}$ consisting of a set of nodes $\mathcal{J} := \{1, \ldots, J\}$ and a set of undirected edges $\mathcal{E} \subset \mathcal{J} \times \mathcal{J}$ assuming that inter-node links are symmetric. A set of neighbors $\mathcal{N}_j$ is assigned to each node j containing all nodes which are connected to node j. Furthermore, the graph is assumed to be connected, i.e., each node can be reached by every other node in the network by one or more hops. Regarding the transmission, each node j observes a different quantity $\mathbf{x}_j \in \mathbb{R}^{M\times 1}$ of the original message $\mathbf{s}$ due to distortions by an individual source-to-node channel matrix $\mathbf{H}_j \in \mathbb{R}^{M\times N}$ and white Gaussian noise $\mathbf{n}_j \in \mathbb{R}^{M\times 1}$. Hence, the transmission scenario can be formulated by

$$\mathbf{x}_j = \mathbf{H}_j \mathbf{s} + \mathbf{n}_j, \quad \forall j \in \mathcal{J}. \tag{1}$$

The channel matrix $\mathbf{H}_j$ can be interpreted in terms of a Multiple-Input-Multiple-Output (MIMO) channel model with N antennas at the transmitter side and M antennas at each receiving node j. Using the set of all observations $\{\mathbf{x}_j\}_{j=1}^J$ in the network, an estimation $\hat{\mathbf{s}}$ of the original message $\mathbf{s}$ is desired. This can be achieved e.g., by a centralized Least Squares (LS) estimation which incorporates all observations $\{\mathbf{x}_j\}_{j=1}^J$ and channel matrices $\{\mathbf{H}_j\}_{j=1}^J$. Using (1), we can summarize the overall transmission to the network by

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{2}$$

with the stacked channel matrix $\mathbf{H} = [\mathbf{H}_1^T, \ldots, \mathbf{H}_J^T]^T$ and the stacked vectors $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_J^T]^T$ and $\mathbf{n} = [\mathbf{n}_1^T, \ldots, \mathbf{n}_J^T]^T$. Using (2), a centralized LS estimation problem can be formulated by

$$\hat{\mathbf{s}}_{\text{LS}} = \arg \min_{\mathbf{s}' \in \mathbb{R}^N} ||\mathbf{x} - \mathbf{H}\mathbf{s}'||^2. \tag{3}$$

In order to solve this estimation problem different approaches exist which will be described in the following section.

III. Detection Algorithms

A. Central LS Solution

A straightforward solution of (3) would be the central LS solution which is given by the observation vector $\mathbf{x}$ multiplied by the Moore-Penrose pseudoinverse of $\mathbf{H}$ [8]:

$$\hat{\mathbf{s}}_{\text{LS}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x}. \tag{4}$$

In order to calculate the estimate $\hat{\mathbf{s}}_{\text{LS}}$, knowledge of all individual observations $\{\mathbf{x}_j\}_{j=1}^J$ and channel matrices $\{\mathbf{H}_j\}_{j=1}^J$ is needed. For this purpose, a central node or FC can be used which collects these quantities and processes them jointly in order to estimate the message $\mathbf{s}$. As already mentioned, this scheme lacks robustness since the estimation process depends on the functionality of a single node. Moreover, calculation of the pseudoinverse of $\mathbf{H}$ with dimensions $\mathbb{R}^{JM\times N}$ causes high computational complexity at the central node. A more robust scheme without any central node can be achieved by a distributed calculation of the centralized solution in (4). This is done through information exchange among neighboring nodes which also distributes the complexity over all nodes in the network. In [3], we proposed such an iterative distributed algorithm based on the Augmented Lagrangian method [9].

This distributed algorithm, which will be called DiCE algorithm, is described in the following subsection.

B. DiCE Algorithm

For a distributed calculation of the central LS solution in (4), we introduce local estimates $\mathbf{s}_j$ per node j and reformulate the LS problem in (3) as a set of local optimization problems as follows:

$$\{\hat{\mathbf{s}}_j\}_{j=1}^J = \underset{\{\mathbf{s}_j \in \mathbb{R}^N | j \in \mathcal{J}\}}{\text{argmin}} \sum_{j=1}^J ||\mathbf{x}_j - \mathbf{H}_j\mathbf{s}_j||^2 \tag{5a}$$

$$\text{s.t.} \quad \mathbf{s}_j = \mathbf{s}_i \quad \forall j \in \mathcal{J},\, i \in \mathcal{N}_j, \tag{5b}$$

with the set $\{\hat{\mathbf{s}}_j\}_{j=1}^J$ containing the converged estimates of all nodes $j \in \mathcal{J}$. The consensus constraint (5b) directly couples estimates of neighboring nodes in order to guarantee that the estimates of all nodes converge to the same solution, namely the central LS solution. Due to the direct coupling among estimates of neighboring nodes a parallel processing of the optimization problem is not possible. Thus, we introduce auxiliary variables per node j denoted by $\mathbf{z}_j$ and decompose the constraint (5b) equivalently into the constraints

$$\mathbf{s}_j = \mathbf{z}_j, \quad \mathbf{z}_j = \mathbf{s}_i. \tag{6}$$

This enables a distributed calculation of the local optimization problems in (5a). To solve these problems we apply the Augmented Lagrangian method [9] and the Alternating Direction Method of Multipliers [10], [11] resulting in the following update equations for each node in the network:

$$\tilde{\mathbf{s}}_j(k+1) = \left(\mathbf{H}_j^T\mathbf{H}_j + \frac{|\mathcal{N}_j|+1}{\mu}\mathbf{I}_N\right)^{-1} \cdot \left[\mathbf{H}_j^T\mathbf{x}_j + \sum_{i\in\mathcal{N}_j\cup\{j\}} \left(\boldsymbol{\lambda}_{ji}(k) + \frac{\tilde{\mathbf{z}}_i(k)}{\mu}\right)\right], \tag{7a}$$

$$\tilde{\mathbf{z}}_j(k+1) = \frac{\mu}{|\mathcal{N}_j|+1} \sum_{i\in\mathcal{N}_j\cup\{j\}} \left(-\boldsymbol{\lambda}_{ij}(k) + \frac{1}{\mu}\tilde{\mathbf{s}}_i(k+1)\right), \tag{7b}$$

$$\boldsymbol{\lambda}_{ij}(k+1) = \boldsymbol{\lambda}_{ij}(k) - \frac{1}{\mu}(\tilde{\mathbf{s}}_i(k+1) - \tilde{\mathbf{z}}_j(k+1)). \tag{7c}$$

These equations describe the DiCE algorithm which is derived in detail in [3]. The parameter μ determines the step size of the algorithm whereas $\boldsymbol{\lambda}_{ij} \in \mathbb{R}^{N\times 1}$ are Lagrange multipliers stemming from the optimization method. The multipliers are specific to one edge from node i to node j. $\tilde{\mathbf{s}}_j(k)$ represents the instantaneous local estimate of message $\mathbf{s}$ at node j at iteration k. By using the Alternating Direction Method of Multipliers it is guaranteed that all estimates $\tilde{\mathbf{s}}_j(k)$ converge to the central solution (4) so that after convergence all nodes will have the same estimate fulfilling the consensus constraint in (5b). In order to update the current estimates, auxiliary variables $\tilde{\mathbf{z}}_i(k)$ from neighboring nodes and Lagrange multipliers $\boldsymbol{\lambda}_{ji}(k)$ are required. The multipliers $\boldsymbol{\lambda}_{ji}(k)$ can be calculated locally at each node if the variables $\tilde{\mathbf{z}}_i(k)$ are received from neighboring nodes $i \in \mathcal{N}_j$. This avoids a transmission of these Lagrange multipliers in the network. Similarly, for the calculation of $\tilde{\mathbf{z}}_j(k+1)$ the updated estimates $\tilde{\mathbf{s}}_i(k+1)$ as well as the

Lagrange multipliers $\boldsymbol{\lambda}_{ij}(k)$ of neighboring nodes need to be known. Consequently, each node j transmits its local quantities $\tilde{\mathbf{s}}_j$, $\tilde{\mathbf{z}}_j$ and $\boldsymbol{\lambda}_{ji}$ to its neighboring nodes and receives the quantities $\tilde{\mathbf{s}}_i$, $\tilde{\mathbf{z}}_i$ and $\boldsymbol{\lambda}_{ij}$ from its neighboring nodes after they have been updated. Earlier investigations for error-prone inter-node links showed, that the multipliers $\boldsymbol{\lambda}_{ji}$ can be calculated locally while the multipliers $\boldsymbol{\lambda}_{ij}$ need to be exchanged among the nodes [7]. It is important to note, that both the instantaneous estimates $\tilde{\mathbf{s}}_j$ and the auxiliary variables $\tilde{\mathbf{z}}_j$ are only specific to their originating node j and thus can be transmitted in a broadcast fashion. In contrast, the Lagrange multipliers $\boldsymbol{\lambda}_{ij}$ are specific to one edge and one direction from node i to node j. Hence, these variables need to be transmitted in a unicast fashion and therefore cause a significantly higher communication overhead. Since a higher data exchange is related to higher costs and higher energy consumption, a reduction of the overall communication effort in the network is desirable leading to the following considerations.

C. RO-DiCE Algorithm

For a reduction in communication effort of the DiCE algorithm, we propose the use of local Lagrange multipliers $\boldsymbol{\lambda}_{jj}$ only instead of the multipliers $\boldsymbol{\lambda}_{ij}$ and $\boldsymbol{\lambda}_{ji}$, respectively. The multipliers $\boldsymbol{\lambda}_{jj}$ are locally available at each node j since they consist of local variables $\mathbf{s}_j$ and $\mathbf{z}_j$ only. Thus, these quantities need not to be exchanged among nodes. In contrast, the multipliers $\boldsymbol{\lambda}_{ij}$ and $\boldsymbol{\lambda}_{ji}$ require variables $\tilde{\mathbf{s}}_i$ and $\tilde{\mathbf{z}}_i$, respectively, from neighboring nodes. By substituting both $\boldsymbol{\lambda}_{ij}$ and $\boldsymbol{\lambda}_{ji}$ by $\boldsymbol{\lambda}_{jj}$, we approximate the exchangeable multipliers by local multipliers. Hence, for equations (7a)-(7c) we achieve

$$\tilde{\mathbf{s}}_j(k+1) = \left(\mathbf{H}_j^T\mathbf{H}_j + \frac{|\mathcal{N}_j|+1}{\mu}\mathbf{I}_N\right)^{-1} \cdot \Bigg[\mathbf{H}_j^T\mathbf{x}_j + (|\mathcal{N}_j|+1)\cdot\boldsymbol{\lambda}_{jj}(k) + \sum_{i\in\mathcal{N}_j\cup\{j\}} \frac{\tilde{\mathbf{z}}_i(k)}{\mu}\Bigg], \quad (8a)$$

$$\tilde{\mathbf{z}}_j(k+1) = -\mu\boldsymbol{\lambda}_{jj}(k) + \frac{1}{|\mathcal{N}_j|+1}\sum_{i\in\mathcal{N}_j\cup\{j\}} \tilde{\mathbf{s}}_i(k+1), \quad (8b)$$

$$\boldsymbol{\lambda}_{jj}(k+1) = \boldsymbol{\lambda}_{jj}(k) - \frac{1}{\mu}(\tilde{\mathbf{s}}_j(k+1) - \tilde{\mathbf{z}}_j(k+1)). \quad (8c)$$

In this way, an exchange of multipliers is not necessary anymore because every node simply uses its local multiplier only. The communication overhead is reduced significantly since an exchange of multipliers would have to be done in a unicast fashion. Now, only local estimates $\tilde{\mathbf{s}}_j$ and auxiliary variables $\tilde{\mathbf{z}}_j$ need to be exchanged among neighboring nodes.

Since the multipliers indicate how well the constraints are fulfilled, constraints between local quantities and quantities from neighboring nodes are not taken into account if only local multipliers are used for each node. This means that each node approximates the constraints to each neighbor by its local constraint. A consensus among all nodes can still be achieved since quantities from neighboring nodes are incorporated into the calculation of $\tilde{\mathbf{s}}_j$ and $\tilde{\mathbf{z}}_j$. However, it is expected that a strict convergence to the central LS solution is not given anymore.

IV. Performance Evaluation

A. Communication Overhead

In the following, the communication overhead[1] of the DiCE and the RO-DiCE algorithm will be investigated. The communication overhead can be quantified by means of the number of edges $|\mathcal{E}|$ and the number of nodes J in the network. For the DiCE algorithm, where each node j needs to transmit $\tilde{\mathbf{s}}_j$, $\tilde{\mathbf{z}}_j$ and $\boldsymbol{\lambda}_{ji}$ to its neighbors, the total communication overhead in the network per iteration is equal to $2|\mathcal{E}|+2J$. The multipliers $\boldsymbol{\lambda}_{ji}$ are edge specific and belong to one direction from node j to node i causing 2 transmissions per edge while $\tilde{\mathbf{s}}_j$ and $\tilde{\mathbf{z}}_j$ are only node specific adding $2J$ more transmissions. In comparison, for the RO-DiCE without an exchange of multipliers the communication overhead per iteration is always equal to $2J$ since for each node the two quantities $\tilde{\mathbf{s}}_j$ and $\tilde{\mathbf{z}}_j$ need to be exchanged only.

Table I summarizes the overall communication overhead per iteration for different topologies. It can be seen that the overhead can be halved for a ring topology if the RO-DiCE algorithm is used. For a full mesh network the reduction is even higher, e.g., for $J = 6$ nodes 42 transmissions/iteration with the DiCE algorithm and only 12 transmissions/iteration with the RO-DiCE algorithm would be required. Regarding the instantaneous local estimates of the nodes, it is expected that these, in general, will not converge to the central LS solution. This is due to the fact that the Lagrange multipliers, which indicate how well the consensus constraint is matched, are approximated by local variables $\tilde{\mathbf{s}}_j$ and $\tilde{\mathbf{z}}_j$ only. Thus, deviations among instantaneous estimates and auxiliary variables of neighboring nodes are not taken into account. Nevertheless, for a full mesh network we will see that the RO-DiCE and the DiCE algorithm show an identical behavior.

TABLE I
Communication overhead per iteration for different topologies.

Topology	No. of Edges $\lvert\mathcal{E}\rvert$	Transmissions DiCE	Transmissions RO-DiCE
Line	$J-1$	$4J-1$	$2J$
Ring	J	$4J$	$2J$
Full mesh	$J(J-1)/2$	$J(J+1)$	$2J$

B. Estimation Performance

For a comparison between the DiCE and the RO-DiCE algorithm regarding their estimation performance, we investigate the overall square error (OSE) per iteration k

$$\mathrm{OSE}(k) = \sum_{j=1}^{J} ||\mathbf{s} - \tilde{\mathbf{s}}_j(k)||^2 \quad (9)$$

[1] By communication overhead, we mean the total number of exchanged vectors with a dimension of N.

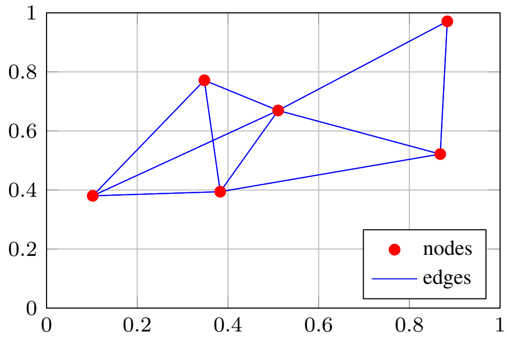

Fig. 2. Random topology with $J = 6$ nodes. Nodes are connected via an edge if they lie within a radius of $r < 0.3$ to each other.

between the instantaneous local estimates of all nodes and the true value. We choose the dimensions of $\mathbf{s}$ and $\mathbf{x}_j$ to be $N = 2$ and $M = 5$, respectively, and the step size to be $\mu = 1$. The channel matrices $\mathbf{H}_j$ contain real valued, i.i.d. and zero mean normal distributed coefficients. The same is true for the entries of the noise vectors $\mathbf{n}_j$ and the message vector $\mathbf{s}$ while the variance of the noise is set according to a channel SNR of $10\,\text{dB}$. Regarding the network topologies we consider a line, a full mesh and a random topology each with $J = 6$ nodes placed on a unit square. For a random topology, nodes are sharing a connection if they lie within a radius of $r < 0.3$ to each other. The resulting topology which has been used for simulations in a random network is depicted in Fig. 2.

Fig. 3 to 5 show the OSE averaged over 100 random realizations of the message $\mathbf{s}$, the channels $\mathbf{H}_j$ and the noise $\mathbf{n}_j$ over the iteration k for the considered networks. For all networks it can be observed that the DiCE algorithm is converging to the central LS solution while the RO-DiCE algorithm is showing performance degradation depending on the network topology. In a line topology the RO-DiCE algorithm is not converging to the central LS solution anymore but approaches a higher OSE value (Fig. 3). This confirms the aforementioned expectation that for the RO-DiCE algorithm a convergence to the central solution is not guaranteed anymore.

In the random topology the difference in the OSE between both algorithms is significantly lower in comparison to the case of a line topology. Both algorithms show a similar convergence behavior while the RO-DiCE algorithm is approaching an OSE value which is slightly higher than the OSE of the central LS solution (Fig. 4). For a full mesh topology it is remarkable that both algorithms even perform identically (Fig. 5). This improved performance of the RO-DiCE algorithm is due to the high connectivity in both topologies. In the random network of Fig. 2 most of the nodes have at least three neighbors. There is only one node which has two neighbors. In a full mesh network each node is connected with every other node in the network establishing an even higher connectivity. Obviously, the RO-DiCE algorithm performs worse in a line topology caused by the low connectivity since each node has maximum two neighbors. Thus, only a limited amount of information from neighboring nodes is available at each node making it difficult for a compensation of the error introduced by the approximation of the edge specific Lagrange multipliers $\boldsymbol{\lambda}_{ij}$ and $\boldsymbol{\lambda}_{ji}$. In fact, if we compare equations (7a) and (7b) of the DiCE algorithm to the equations (8a) and (8b) of the RO-DiCE algorithm, the following conditions have to be fulfilled for an equivalence between both algorithms:

$$\tilde{\mathbf{s}}_j(k) = \frac{1}{|\mathcal{N}_j| + 1} \sum_{i \in \mathcal{N}_j \cup \{j\}} \tilde{\mathbf{s}}_i(k), \tag{10a}$$

$$\tilde{\mathbf{z}}_j(k) = \frac{1}{|\mathcal{N}_j| + 1} \sum_{i \in \mathcal{N}_j \cup \{j\}} \tilde{\mathbf{z}}_i(k). \tag{10b}$$

These conditions are always fulfilled for a full mesh network since here the set $\mathcal{N}_j \cup \{j\}$ is always equal to the set of nodes $\mathcal{J}$. Thus, the right-hand side of above equations is always equal to the same value independent of the node j explaining why both algorithms show the same behavior in a full mesh network. This is remarkable, since the communication overhead in a full mesh network can be more than halved while keeping the performance of the algorithm. Furthermore, equations (10a) and (10b) state that with a higher connectivity the RO-DiCE algorithm is approaching the DiCE algorithm.

Interestingly, Fig. 5 shows that for some iterations the OSE of the DiCE algorithm can be even lower than the OSE of the central LS solution before it is approaching it. This is possible since the update equation of the local instantaneous estimate $\tilde{\mathbf{s}}_j(k+1)$ contains a regularized pseudoinverse term. E.g. for the first iteration equation (7a) will be

$$\tilde{\mathbf{s}}_j(k+1) = \left(\mathbf{H}_j^T \mathbf{H}_j + \frac{|\mathcal{N}_j| + 1}{\mu} \mathbf{I}_N\right)^{-1} \mathbf{H}_j^T \mathbf{x}_j, \tag{11}$$

which is a modified expression of the central LS solution in (3) containing a pseudoinverse regularized by the term $\frac{|\mathcal{N}_j|+1}{\mu}$. It is known that the MMSE solution also contains a regularized pseudoinverse and outperforms the LS solution in the presence of noise. Therefore, the DiCE algorithm can temporarily have a lower OSE than the central LS solution.

C. Bit Error Rate Performance

With regard to the bit error rate (BER) performance of the algorithms we consider uncoded BPSK transmissions where the message $\mathbf{s}$ contains symbols out of the alphabet $\mathcal{A} = \{-1, 1\}$. Both algorithms are stopped after $k = 10$ iterations followed by a hard decision on the estimates $\tilde{\mathbf{s}}_j(k = 10)$ of all nodes in the network. Bit errors are then counted and averaged over all nodes in the network. Fig. 6 depicts the BER for a line topology. Here, an SNR loss of roughly $2\,\text{dB}$ can be observed for the RO-DiCE algorithm corresponding to the OSE performance in Fig. 3. Furthermore, it can be seen that the curve of the DiCE algorithm achieves the performance of the central solution indicating a satisfying convergence after only 10 iterations. Fig. 7 and 8 show the BER performance for a random and a full mesh topology, respectively. Due to a higher connectivity in the random topology the RO-DiCE algorithm performs better than in a line topology. Thus, there is only a minimal degradation in the BER performance compared

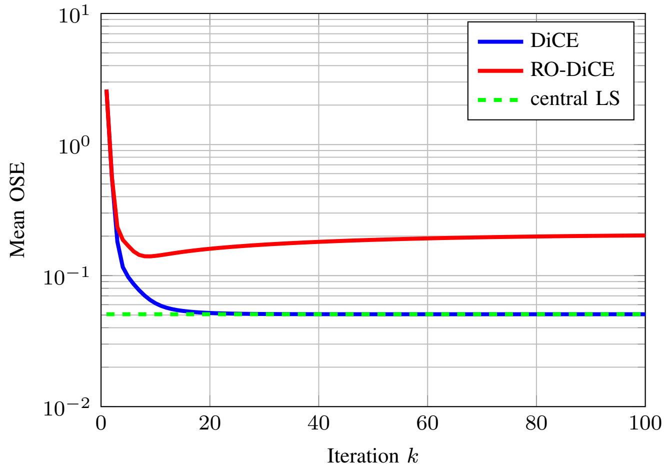

Fig. 3. Line topology with $J = 6$ nodes: Mean OSE of the DiCE and the RO-DiCE algorithm.

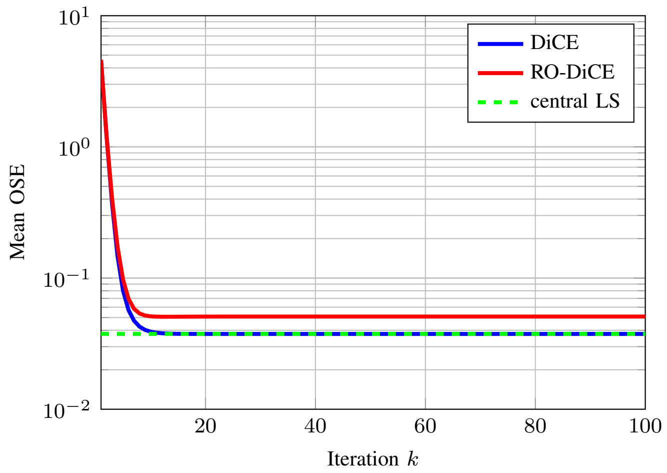

Fig. 4. Random topology with $J = 6$ nodes: Mean OSE of the DiCE and the RO-DiCE algorithm.

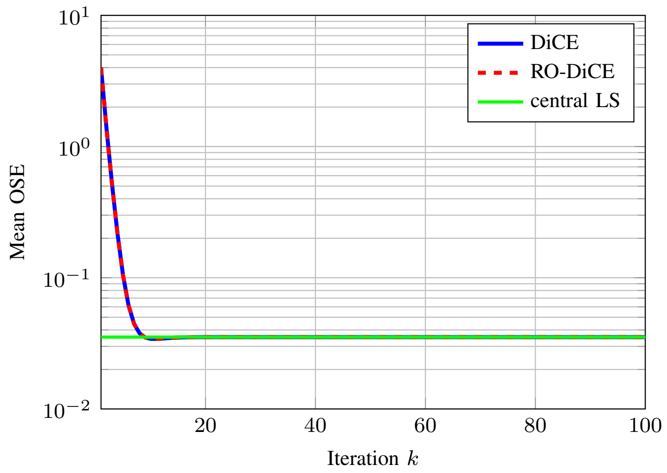

Fig. 5. Full mesh topology with $J = 6$ nodes: Mean OSE of the DiCE and the RO-DiCE algorithm.

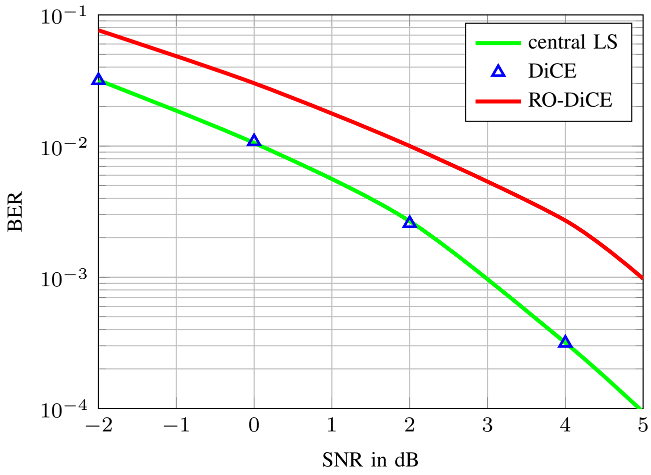

Fig. 6. Line topology with $J = 6$ nodes: BER of the DiCE and the RO-DiCE algorithm after $k = 10$ iterations.

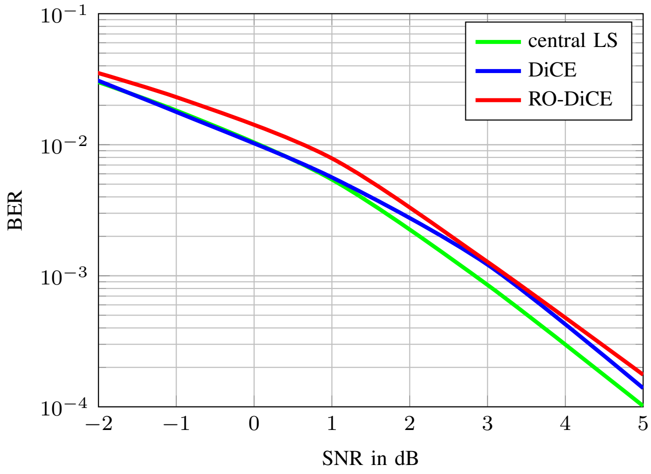

Fig. 7. Random topology with $J = 6$ nodes: BER of the DiCE and the RO-DiCE algorithm after $k = 10$ iterations.

to the DiCE algorithm. As already described, the performance of both algorithms in a full mesh topology is identical leading to the same BER performance, as can be seen in Fig. 8. These results correspond to the OSE behavior of both algorithms in Fig. 3 to 5.

V. Conclusion and Future Work

In this paper, we presented the RO-DiCE algorithm which exhibits a significantly reduced communication overhead in comparison to the unreduced DiCE algorithm. In order to reduce the overhead of the DiCE algorithm, we proposed the use of local Lagrange multipliers only in order to save a unicast exchange of multipliers among nodes. We investigated the RO-DiCE algorithm regarding its convergence behavior and its BER performance for different network topologies. We showed that it is equivalent to the DiCE algorithm in a full mesh topology. In future work, the RO-DiCE algorithm should be investigated in detail for error-prone inter-node links such

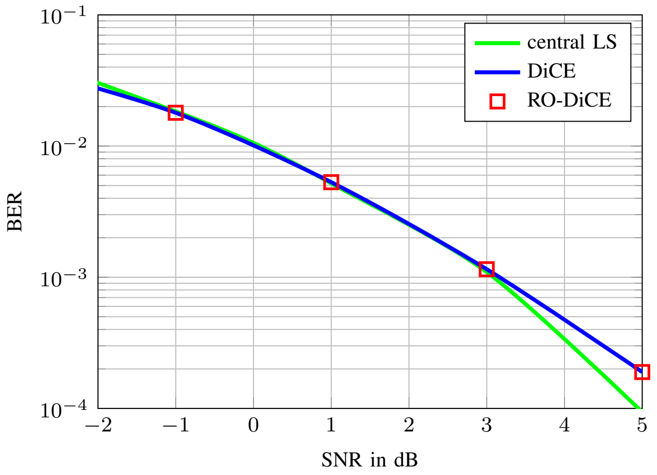

Fig. 8. Full mesh topology with $J = 6$ nodes: BER of the DiCE and the RO-DiCE algorithm after $k = 10$ iterations.

as noisy links and link failures. A mathematical convergence analysis of the RO-DiCE algorithm is also of interest.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 317941. The authors would like to acknowledge the contributions of their colleagues in iJOIN, although the views expressed are those of the authors and do not necessarily represent the project.

References

[1] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, pp. 27–41, Jul. 2006.

[2] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, no. 1, pp. 98–101, Jan. 1986.

[3] H. Paul, J. Fliege, and A. Dekorsy, "In-network-processing: Distributed consensus-based linear estimation," *IEEE Commun. Lett.*, vol. 17, no. 1, pp. 59–62, Jan. 2013.

[4] H. Zhu, A. Cano, and G. Giannakis, "Distributed consensus-based demodulation: algorithms and error analysis," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 2044–2054, Jun. 2010.

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

[6] H. Paul, B.-S. Shin, D. Wübben, and A. Dekorsy, "In-network-processing for small cell cooperation in dense networks," in *IEEE VTC2013-Fall Workshop (CLEEN 2013)*, Las Vegas, USA, Sept. 2013.

[7] H. Paul, B.-S. Shin, and A. Dekorsy, "Distributed consensus-based linear estimation with erroneous links," in *17th International ITG Workshop on Smart Antennas (WSA) 2013*, Stuttgart, Germany, Mar. 2013.

[8] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, Inc., 2008.

[9] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, USA: Springer Science+Business Media, 2006.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, USA: Athena Scientific, 1997.