Distributed Augmented Lagrangian Method for Cooperative Estimation in Small Cell Networks

Guang Xu, Henning Paul, Dirk Wübben, Armin Dekorsy
Department of Communications Engineering
University of Bremen, 28359 Bremen, Germany
Email:{xu, paul, wuebben, dekorsy}@ant.uni-bremen.de

***Abstract*—In a dense small-cell (SC) network with several users to be served, a multi-user detection (MUD) can be employed across SCs, and distributed estimation is a promising technique for such a scenario. Nevertheless, large communication overhead due to frequently exchange of variables among SCs will cause high energy consumption and processing latency. This paper is focused on the reduction of communication overhead for the distributed processing. To this end, two algorithms, Augmented Lagrangian based Cooperative Estimation (ALCE) and Priority-aided ALCE (PALCE) will be presented. In ALCE a new efficient approach is adopted to achieve parallel processing among all SCs, which needs fewer variables to be exchanged. Thus, a considerable amount of overhead will be saved. However, the ALCE algorithm is not robust when applied to a network with erroneous backhaul (BH) links, therefore a variant of this approach termed PALCE is proposed using a priority oriented principle to enhance the robustness and maintain low amount of information exchange. The proposed algorithms are investigated by means of error rate and communication overhead showing significant improvement in estimation performance compared to state of the art algorithms.**

I. Introduction

Recently, extensive deployment of small cells is becoming a growing trend to extend wireless service coverage and to increase network capacity for mobile communication systems. In such a dense deployment, the coverage areas of neighboring SCs might be overlapping within the same frequency resources and, correspondingly, interference needs to be coped with. Therefore, a joint detection of messages from several users (UEs) can be performed in these dense SC networks for uplink (UL) transmission. A possible way for a MUD is that each SC forwards the raw base-band receive signals to a central entity, e.g., a C-RAN deployment [1] to accomplish the centralized processing. However, the central processing usually requires long distance transmission to the central node. Thus, in order to overcome this weakness, an alternative is to apply distributed processing exploiting cooperation among SCs by sharing information through rather short low-cost wireless transmission links. Several algorithms like the Distributed Consensus on Demodulated Symbols (DCDS) [2] and the Distributed Consensus-based Estimation (DiCE) [3] adapt the Alternating Direction Method of Multipliers (ADMM) [4] technique to accomplish the distributed estimation, and in [5] both algorithms have been applied to the scenario of erroneous inter-node links. Furthermore, considering the high inter-node communication overhead produced during the iterative processing, some algorithms [6], [7] based on [3] have been proposed to reduce the communication effort.

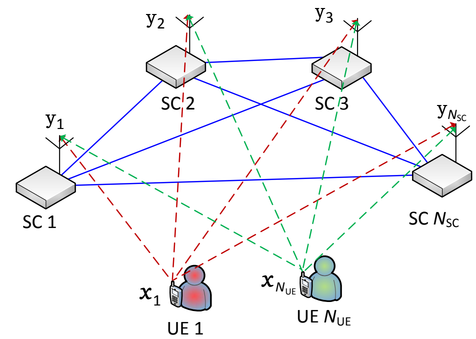

Fig. 1. Small-cell network where N_{SC} SCs detecting N_{UE} mobile users. Each SC receives messages $\boldsymbol{x}_u$ and exchanges information with neighboring SCs.

In this paper, we aim to improve the communication efficiency and reduce the computation complexity for the distributed estimation. Thus, we propose the new algorithm ALCE by adapting another numerical approach, i.e., the Augmented Lagrangian Multipliers (ALM) method for distributed processing, which needs fewer variables to be exchanged compared to the ADMM-based algorithms [2]-[8]. Additionally, we also consider the erroneous inter-node link scenario, and correspondingly we further propose variant approach PALCE to enhance the system robustness by increasing the communication effort a little, but keeping the improvement in overall performance. Principally, both algorithms can be applied to a general cooperative network, but here we apply the algorithms to a small cell network.

The remainder of this paper is structured as follows: The system model and the investigated scenario are introduced in Section II. The considered algorithms ALCE and PALCE are derived and discussed in Section III. In the subsequent Section IV, simulation results of the proposed algorithms are analyzed considering different BH topologies and link conditions. The paper is concluded in Section V.

II. System Model

Fig. 1 shows a small cell network where N_{SC} SCs are serving N_{UE} users. This network of nodes is described by a geometric graph $\mathcal{G} := \{\mathcal{J}, \mathcal{E}\}$, in which $\mathcal{J} := \{1, ..., N_{\rm SC}\}$ denotes the set of nodes and $\mathcal{E}$ represents the set of (directional or bidirectional) edges for the linked nodes. For exchanging information, each node $j \in \mathcal{J}$ can only communicate with its neighboring nodes $i \in \mathcal{N}_j \subseteq \mathcal{J}$ through inter-node links, where both ideal and erroneous links are considered in this paper.

In this scenario, each UE is assumed to use $N_{\rm T}$ antennas to transmit a data vector $\boldsymbol{x}_u \in \mathcal{A}^{N_{\rm T}\times 1}$ containing symbols from a modulation alphabet $\mathcal{A}$. The transmitted data per time instant from all $N_{\rm UE}$ users can then be constructed into a stacked vector $\mathbf{x} = \left[\boldsymbol{x}_1^{\rm T}, \cdots, \boldsymbol{x}_{N_{\rm UE}}^{\rm T}\right]^{\rm T}$ containing $N_{\rm I} = N_{\rm T} \cdot N_{\rm UE}$ system input components. Those messages are received by individual SCs equipped with $N_{\rm R}$ receive antennas, and the corresponding observation $\mathbf{y}_j$ at SC j can be formulated as

$$\mathbf{y}_j = \sum_{u=1}^{N_{\rm UE}} \mathbf{H}_{ju}\boldsymbol{x}_u + \mathbf{n}_j = \mathbf{H}_j\mathbf{x} + \mathbf{n}_j \tag{1}$$

where $\mathbf{H}_{ju} \in \mathbb{C}^{N_{\rm R}\times N_{\rm T}}$ is the channel matrix between UE u and SC j, and $\mathbf{H}_j \in \mathbb{C}^{N_{\rm R}\times N_{\rm I}}$ is a stacked channel matrix $\mathbf{H}_j = \left[\mathbf{H}_{j1}^{\rm T}, .., \mathbf{H}_{ju}^{\rm T}, .., \mathbf{H}_{jN_{\rm UE}}^{\rm T}\right]^{\rm T}$. The complex additive Gaussian white noise vector $\mathbf{n}_j$ contains elements with zero mean and variance σ_n^2. In general, each SC can perform local estimation for $\mathbf{x}$ based on the local observation. However, the local estimations may vary from SC to SC because of different transmission conditions, and it is possible that the message cannot be properly estimated due to a badly conditioned system of linear equations (e.g., $N_{\rm I} > N_{\rm R}$) [7]. Therefore, a joint estimation based on global knowledge over the whole network can achieve a better estimation performance.

In general, one alternative way to recover the UE messages $\mathbf{x}$ in (1) is that each SC j forwards its local observation $\mathbf{y}_j$ and channel information $\mathbf{H}_j$ to a central node performing centralized least square (LS) estimation

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}'\in\mathbb{C}^{N_{\rm I}}} \|\mathbf{y} - \mathbf{H}\mathbf{x}'\|^2 \tag{2}$$

with the stacked observation vector $\mathbf{y} = \left[\mathbf{y}_1^{\rm T}, ..., \mathbf{y}_{N_{\rm SC}}^{\rm T}\right]^{\rm T}$ and the stacked channel matrix $\mathbf{H} = \left[\mathbf{H}_1^{\rm T}, ..., \mathbf{H}_{N_{\rm SC}}^{\rm T}\right]^{\rm T}$ achieving an overall $N_{\rm O} \times N_{\rm I}$ MIMO system with $N_{\rm O} = N_{\rm R} \cdot N_{\rm SC}$ output signals per time instant. The solution is given by the Zero-Forcing (ZF) linear equalizer

$$\tilde{\mathbf{x}}_{\rm ZF} = (\mathbf{H}^{\rm H}\mathbf{H})^{-1}\mathbf{H}^{\rm H}\mathbf{y} = \mathbf{H}^{+}\mathbf{y}\ , \tag{3}$$

obtaining the central solution of estimate $\tilde{\mathbf{x}}_{\rm ZF}$ by filtering the observation vector $\mathbf{y}$ with the Moore-Penrose pseudo inverse of the stacked channel matrix $\mathbf{H}^{+} = (\mathbf{H}^{\rm H}\mathbf{H})^{-1}\mathbf{H}^{\rm H}$.

Although it is beneficial to achieve the joint estimate at a central entity instead of local estimations at separate SCs, the local observation $\mathbf{y}_j$ and channel information $\mathbf{H}_j$ have to be delivered to a central node through BH links that are likely to span a long distance. To avoid the long distance BH transmission, the centralized LS problem can also be solved in a distributed fashion by reformulating the problem (2) into set of separate LS problems over $\mathbf{x}_j$ parallelized among nodes with pairwise consensus constraints:

$$\begin{aligned}\{\mathbf{x}_j, j \in \mathcal{J}\} = \arg\min_{\mathbf{x}'_j\in\mathbb{C}^{N_{\rm I}}} \sum_{j=1}^{N_{\rm SC}} \left\|\mathbf{y}_j - \mathbf{H}_j\mathbf{x}'_j\right\|^2 \\ \text{s.t.} \quad \mathbf{x}_j = \mathbf{x}_i, \quad \forall i \in \mathcal{N}_j, \forall j \in \mathcal{J}\end{aligned} \tag{4}$$

In such way, the joint estimation over all SCs can be achieved distributedly, and the system becomes more flexible compared to the central node processing.

III. Distributed Estimation Schemes

A. ADMM-based Algorithms

The ADMM technique was applied, e.g., in [2] and [3] to solve the LS problem (4) in a distributed fashion, fulfilling the consensus constraint that the estimate $\mathbf{x}_j$ at SC j equals the estimate $\mathbf{x}_i$ at SC i, i.e., $\mathbf{x}_j = \mathbf{x}_i$. In [2], the DCDS algorithm uses auxiliary variables $\mathbf{z}_{ji}$ and $\mathbf{z}'_{ji}$ to decouple the constraint directionally, i.e., $\mathbf{x}_j = \mathbf{z}_{ji}, \mathbf{x}_i = -\mathbf{z}'_{ji}$, and $\mathbf{z}_{ji} = -\mathbf{z}'_{ji}$, enabling the SCs to update estimates in parallel. In [3] the DiCE algorithm was proposed, showing improved performance. It uses only non-directional auxiliary variables $\mathbf{z}_j$ at SC j to decouple the constraint as $\mathbf{z}_j = \mathbf{x}_j$, and $\mathbf{x}_i = \mathbf{z}_j$. Both algorithms follow the ADMM update sequence that can be implemented distributedly among SCs. Considering DiCE as an example, the update equations for estimate $\mathbf{x}_j$, auxiliary variable $\mathbf{z}_j$ and Lagrangian multipliers $\boldsymbol{\lambda}_{ji}$ at SC j in iteration k are given by [8]

$$\mathbf{z}_j^k = \arg\min_{\mathbf{z}_j} \mathcal{L}_j(\mathbf{z}_j; \mathbf{x}_i^{k-1}, \boldsymbol{\lambda}_{ij}^{k-1}) \tag{5a}$$

$$\boldsymbol{\lambda}_{ji}^k = \boldsymbol{\lambda}_{ji}^{k-1} - \frac{1}{\mu}(\mathbf{x}_j^{k-1} - \mathbf{z}_i^k) \tag{5b}$$

$$\mathbf{x}_j^k = \arg\min_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j; \mathbf{z}_i^k, \boldsymbol{\lambda}_{ji}^k),\ \ i \in \mathcal{N}_j \cup \{j\} \tag{5c}$$

where $\mathcal{L}_j$ [1] is the Lagrangian function of local SC j detailed in [3]. Due to the iterative processing, those variables are required to be exchanged between neighboring SCs for each iteration, leading to high communication overhead [8]. Correspondingly, in order to reduce the overall communication effort, the exchange of some variables can be avoided. Thus, in the following we will apply the ALM method [9] to solve the LS problem (4), leading to our novel algorithms ALCE and PALCE, in which no auxiliary variable is required to decouple the constraint leading to the reduction of the communication overhead and improvement of the estimation performance.

B. ALCE Algorithm

Consider a general convex optimization problem:

$$\text{minimize} \quad f(\mathbf{s}) \tag{6a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{s} = \mathbf{b} \tag{6b}$$

where $f : \mathbb{C}^{n\times 1} \rightarrow \mathbb{C}$ is a convex function and vector $\mathbf{s} \in \mathbb{C}^{n\times 1}$ follows the constraint (6b) in a relation with matrix $\mathbf{A} \in$

[1] In this paper, we define that the variable before the semicolon in a Lagrangian cost function is under optimization, e.g., $\mathbf{x}_j$ in $\mathcal{L}_j(\mathbf{x}_j; \mathbf{z}_i^k, \boldsymbol{\lambda}_{ji}^k), j \in \mathcal{J}$, while the arguments after the semicolon are fixed, e.g., $\mathbf{z}_i^k, \boldsymbol{\lambda}_{ji}^k$.

$\mathbb{C}^{p\times n}$ and vector $\mathbf{b} \in \mathbb{C}^{p\times 1}$. The constrained minimization problem can be solved by ALM method setting the derivative of the Augmented Lagrangian (AL) function [10] w.r.t. $\mathbf{s}$ to zero and solving for $\mathbf{s}$. The AL function is given by

$$\mathcal{L}(\mathbf{s},\boldsymbol{\lambda}) = f(\mathbf{s}) - \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{A}\mathbf{s}-\mathbf{b}) + \frac{1}{2\mu}\|\mathbf{A}\mathbf{s}-\mathbf{b}\|^2 \tag{7}$$

with the Lagrangian multipliers $\boldsymbol{\lambda} \in \mathbb{C}^{p\times 1}$ and penalty parameter μ which can be chosen properly according to the object function (the choice depends on e.g., $\mathbf{H}_j$ in (4)). Here, without loss of generality, we set $\mu = 1$ in following algorithms.

The ALM method is then applied to solve the constrained LS problem (4), where the objective function still maintains the convexity [11]. Considering the constraint $\mathbf{x}_j = \mathbf{x}_i$ in (4), the general term $\boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{A}\mathbf{s}-\mathbf{b})$ in (7) can be expressed as $\sum_{j=1}^{N_{\mathrm{SC}}}\sum_{i\in\mathcal{N}_j}\boldsymbol{\lambda}_{ji}^{\mathrm{T}}(\mathbf{x}_j-\mathbf{x}_i)$. Correspondingly, the AL function can be rewritten as

$$\begin{aligned}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}) =& \frac{1}{2}\sum_{j=1}^{N_{\mathrm{SC}}}\|\mathbf{y}_j-\mathbf{H}_j\mathbf{x}_j\|^2 - \sum_{j=1}^{N_{\mathrm{SC}}}\sum_{i\in\mathcal{N}_j}\boldsymbol{\lambda}_{ji}^{\mathrm{T}}(\mathbf{x}_j-\mathbf{x}_i)\\ &+\frac{1}{2\mu}\sum_{j=1}^{N_{\mathrm{SC}}}\sum_{i\in\mathcal{N}_j}\|\mathbf{x}_j-\mathbf{x}_i\|^2 \qquad (8a)\\ =&\sum_{j=1}^{N_{\mathrm{SC}}}\Bigg(\frac{1}{2}\|\mathbf{y}_j-\mathbf{H}_j\mathbf{x}_j\|^2 - \sum_{i\in\mathcal{N}_j}\boldsymbol{\lambda}_{ji}^{\mathrm{T}}(\mathbf{x}_j-\mathbf{x}_i)\\ &+\frac{1}{2\mu}\sum_{i\in\mathcal{N}_j}\|\mathbf{x}_j-\mathbf{x}_i\|^2\Bigg) \qquad (8b)\\ =&\sum_{j=1}^{N_{\mathrm{SC}}}\mathcal{L}_j(\mathbf{x},\boldsymbol{\lambda}) \qquad (8c)\end{aligned}$$

where $\mathbf{x}$ and $\boldsymbol{\lambda}$ denote the sets of all variables $\mathbf{x}_j$ and $\boldsymbol{\lambda}_{ji}$ respectively. Thus the AL function $\mathcal{L}(\mathbf{x},\boldsymbol{\lambda})$ in (8a) is decomposed into several AL sub-functions $\mathcal{L}_j(\mathbf{x},\boldsymbol{\lambda})$ in (8c) that can be implemented separately at each individual SC j. Note that, the consensus constraint still needs to be decoupled for distributed parallel processing. But instead of using auxiliary variables like in [2] or [3], the decoupling can be done by matching the local estimate $\mathbf{x}_j^{k+1}$ of SC j at iteration $k+1$ with the neighboring estimate $\mathbf{x}_i^k$ from the last iteration k, i.e., $\mathbf{x}_j^{k+1} = \mathbf{x}_i^k, \forall i \in \mathcal{N}_j$, since for $k\to\infty$ the constraint $\mathbf{x}_j = \mathbf{x}_i$ will also be fulfilled throughout the network. Then, the local solution $\mathbf{x}_j^{k+1}$ of SC j can be derived by minimizing the convex local AL function $\mathcal{L}_j(\mathbf{x}_j;\mathbf{x}_i^k,\boldsymbol{\lambda}_{ji}^k)$, $i\in\mathcal{N}_j$ w.r.t. $\mathbf{x}_j$, and the Lagrangian multiplier $\boldsymbol{\lambda}_{ji}^{k+1}$ is updated using the gradient descent method for $\mathcal{L}_j(\boldsymbol{\lambda}_{ji};\mathbf{x}_j^k,\mathbf{x}_i^k,\boldsymbol{\lambda}_{ji}^k)$ w.r.t. $\boldsymbol{\lambda}_{ji}$. Thus, the update equations of ALCE are given by:

$$\begin{aligned}\mathbf{x}_j^{k+1} =& \arg\min_{\mathbf{x}_j}\mathcal{L}_j(\mathbf{x}_j;\mathbf{x}_i^k,\boldsymbol{\lambda}_{ji}^k)\\ =&\left(\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j+\frac{|\mathcal{N}_j|}{\mu}\mathbf{I}_{N_{\mathrm{I}}}\right)^{-1}\left(\mathbf{H}_j^{\mathrm{H}}\mathbf{y}_j+\sum_{i\in\mathcal{N}_j}\left(\frac{\mathbf{x}_i^k}{\mu}+\boldsymbol{\lambda}_{ji}^k\right)\right)\end{aligned} \tag{9a}$$

$$\boldsymbol{\lambda}_{ji}^{k+1} = \boldsymbol{\lambda}_{ji}^k - \frac{1}{\mu}\left(\mathbf{x}_j^k-\mathbf{x}_i^k\right), \forall i\in\mathcal{N}_j \tag{9b}$$

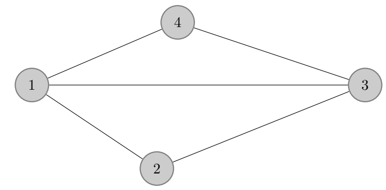

Fig. 2. An example of BH topology for a small cell network composed of SCs $j \in \{1,..,4\}$

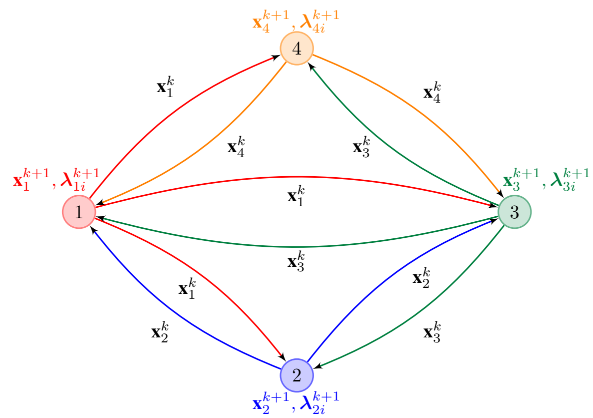

Fig. 3. Update and exchange of estimates $\mathbf{x}_j$ as well as multipliers $\boldsymbol{\lambda}_{ji}$ for ALCE, i denotes the index of neighboring node

Assuming $\boldsymbol{\lambda}_{ji}^0 = \mathbf{0}$ and $\mathbf{x}_i^0 = \mathbf{0}$, we find that the first ($k=1$) estimate $\mathbf{x}_j^1$ is a regularized LS estimate:

$$\mathbf{x}_j^1 = \left(\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j+\frac{|\mathcal{N}_j|}{\mu}\mathbf{I}_{N_{\mathrm{I}}}\right)^{-1}\mathbf{H}_j^{\mathrm{H}}\mathbf{y}_j \tag{10}$$

After the updates in iteration k, each SC j exchanges its local estimate $\mathbf{x}_j$ with neighboring SCs and prepares for the new updates (9a) and (9b) in $k+1$ based on the received information in k. For a simple exemplary illustration, a network of 4 SCs $j\in\{1,..,4\}$ is connected in a certain topology as shown in Fig. 2. Based on that network, the general progress of variable updates and exchanges for ALCE has been summarized in Fig. 3. It can be seen that the local estimate $\mathbf{x}_j^{k+1}$ and multipliers $\boldsymbol{\lambda}_{ji}^{k+1}$ of SC j are updated in $k+1$ based on the received estimate $\mathbf{x}_i^k, i\in\mathcal{N}_j$ from the last iteration k. Afterwards, only the local estimates $\mathbf{x}_j$ are exchanged between neighboring SCs. Comparing to ADMM-based algorithms, where the estimates $\mathbf{x}_j$, the auxiliary variables $\mathbf{z}_j$ as well as multipliers $\boldsymbol{\lambda}_{ji}$ need to be exchanged [3], ALCE shows great advantage in reducing communication overhead.

Although the ALCE algorithm achieves a significant reduction in communication overhead, the robustness of the algorithm is quite low for the case of erroneous inter-node links as shown in Section IV. As can be seen in Fig. 3, only the estimates $\mathbf{x}_j^k$ are delivered via inter-node links, but multipliers which enforce the constraint are not exchanged among SCs. If some disturbance exists on the channel, the exchanged variables will not be received correctly and the error will be accumulated after each iteration, since no multiplier from neighboring SC is received to maintain the constraint. This behavior is similar to the algorithm [6], which also experiences deteriorated performance by avoiding the exchange of the multipliers to reduce overhead.

C. PALCE Algorithm

In order to improve the robustness while keeping the low communication overhead, a variant of ALCE method entitled "PALCE" is proposed. This algorithm is inspired by the idea of using directional transmissions in [12] where the links are oriented for the information exchange in a cooperative network. Thus, differing from ALCE, the PALCE algorithm adopts a priority mechanism, i.e., each SC has a different level of priority which is defined by the index number: the lower index of the SC is, the higher priority it has. Note that without loss of generality, the indices of SCs are randomly chosen. Then, the set of constraints $\mathbf{x}_j = \mathbf{x}_i$ in (4) can be divided into two sub-sets as $\mathbf{x}_i = \mathbf{x}_j$ for $i < j$ and $\mathbf{x}_j = \mathbf{x}_i$ for $i > j$ respectively. Considering the same LS estimation problem (2) with the newly defined constraints, the local Lagrangian cost function is obtained:

$$\begin{aligned}\mathcal{L}_j(\mathbf{x}, \boldsymbol{\lambda}) =& \frac{1}{2}\|\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_j\|^2 \\ &- \Bigg(\sum_{i\in\mathcal{N}_j^-} \boldsymbol{\lambda}_{ij}(\mathbf{x}_i - \mathbf{x}_j) + \sum_{i\in\mathcal{N}_j^+} \boldsymbol{\lambda}_{ji}(\mathbf{x}_j - \mathbf{x}_i)\Bigg) \\ &+ \frac{1}{2\mu}\Bigg(\sum_{i\in\mathcal{N}_j^-}\|\mathbf{x}_i - \mathbf{x}_j\|^2 + \sum_{i\in\mathcal{N}_j^+}\|\mathbf{x}_j - \mathbf{x}_i\|^2\Bigg) \quad (11)\end{aligned}$$

where $\mathcal{N}_j^+$ is the set of neighboring SCs i of SC $j, i > j$, and $\mathcal{N}_j^-$ is the set of neighboring SCs $i, i < j$. Similar to the ALCE algorithm, the decoupling for parallel processing among SCs follows $\mathbf{x}_j^{k+1} = \mathbf{x}_i^k, \forall i \in \mathcal{N}_j$. Therefore, the iterative update of estimates $\mathbf{x}_j^{k+1}$ can be derived by minimizing $\mathcal{L}_j(\mathbf{x}_j; \mathbf{x}_i^k, \boldsymbol{\lambda}_{ij}^k, \boldsymbol{\lambda}_{ji}^k)$ w.r.t. $\mathbf{x}_j$, and the multiplier $\boldsymbol{\lambda}_{ji}^{k+1}$ is updated based on the newest local $\mathbf{x}_j^{k+1}$ and the received $\mathbf{x}_i^k$ from neighboring SCs. The update equations are given by

$$\begin{aligned}\mathbf{x}_j^{k+1} &= \arg\min_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j; \mathbf{x}_i^k, \boldsymbol{\lambda}_{ij}^k, \boldsymbol{\lambda}_{ji}^k) \\ &= \left(\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j + \frac{|\mathcal{N}_j^-| + |\mathcal{N}_j^+|}{\mu}\mathbf{I}_{N_\mathrm{I}}\right)^{-1}\Bigg(\mathbf{H}_j^{\mathrm{H}}\mathbf{y}_j \\ &+ \sum_{i\in\mathcal{N}_j^-}\left(\frac{1}{\mu}\mathbf{x}_i^k - \boldsymbol{\lambda}_{ij}^k\right) + \sum_{i\in\mathcal{N}_j^+}\left(\frac{1}{\mu}\mathbf{x}_i^k + \boldsymbol{\lambda}_{ji}^k\right)\Bigg) \quad \text{(12a)}\end{aligned}$$

$$\boldsymbol{\lambda}_{ji}^{k+1} = \boldsymbol{\lambda}_{ji}^k - \frac{1}{\mu}\left(\mathbf{x}_j^{k+1} - \mathbf{x}_i^k\right), \ i > j \quad \text{(12b)}$$

Similarly, by initializing the estimate $\mathbf{x}_i^0 = \mathbf{0}$ and multipliers $\boldsymbol{\lambda}_{ji}^0 = \boldsymbol{\lambda}_{ij}^0 = \mathbf{0}$, the first local estimate $\mathbf{x}_j^1$ is identical to (10). After that, the latest estimates $\mathbf{x}_j^{k+1}$ have to be exchanged among the neighboring SCs, and multipliers $\boldsymbol{\lambda}_{ji}^{k+1}$ will be updated subsequently. Based on the network in Fig. 2, the general progress of update and exchange for PALCE is summarized in Fig. 4, where it should be noticed that the exchange of local estimates $\mathbf{x}_j$ among SCs is similar to ALCE, i.e., using bidirectional transmission (both neighboring SCs exchange their estimates). But in PALCE, the Lagrangian multipliers $\boldsymbol{\lambda}_{ij}, i \in \mathcal{N}_j^-$ need to be exchanged additionally, meaning that only the SC j with higher priority ($j < i$) shares its multipliers $\boldsymbol{\lambda}_{ji}$ with lower priority neighboring SC i, i.e., uni-directional transmission is used. Thus only the multipliers $\boldsymbol{\lambda}_{ji}^{k+1}$ need to be computed at SC j, while $\boldsymbol{\lambda}_{ij}^k$ in (12a) just needs to be received by SC j for $i < j$, e.g., SC 4 only needs to update the estimate $\mathbf{x}_4^{k+1}$ in Fig. 4. Compared to ALCE, the robustness of the PALCE algorithm will be stronger due to the additional exchange of multipliers even for the erroneous inter-node link, since the fulfillment of the constraint is affected by the multipliers. Besides, with more information being exchanged, the convergence behavior for PALCE is faster than ALCE, which in turn can further reduce the communication overhead, since fewer iterations are required for the same performance.

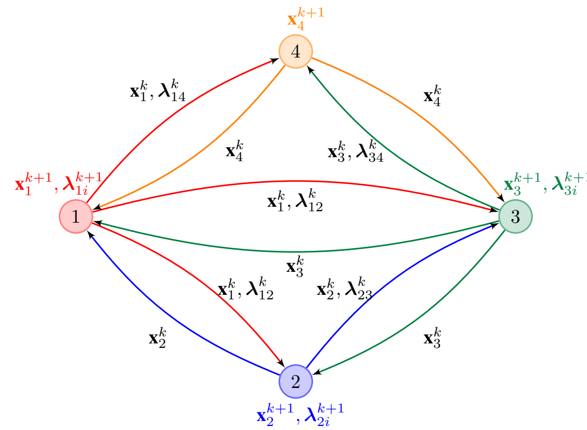

Fig. 4. Update and exchange of local estimates $\mathbf{x}$ as well as multipliers $\boldsymbol{\lambda}$ for PALCE, i denotes the index of SC that has higher priority

IV. Performance Evaluation

A. Bit Error Rate

Both ALCE and PALCE algorithms have been evaluated by means of Monte Carlo simulations in a small-cell scenario. Two different topologies, full-meshed type and ring type, have been simulated for a network consisting of $N_{\mathrm{SC}} = 4$ small cells, each equipped with $N_{\mathrm{R}} = 2$ receive antennas serving $N_{\mathrm{UE}} = 2$ UEs with $N_{\mathrm{T}} = 2$ transmit antennas in an UL transmission, where uncoded QPSK symbols are transmitted.

We perform LS estimation using the proposed distributed algorithms to investigate the bit error rate (BER) over different Signal-to-Noise Ratios (SNRs). Fig. 5 shows the averaged BER over all SCs in a full-meshed network with ideal BH links. As can be seen, all distributed algorithms show the same performance as the central ZF solution at low SNRs, while error floors appear at high SNRs when the number of iterations is not sufficient for a satisfying convergence. Nevertheless, better performance can be achieved by these distributed algorithms with more iterations, as the error floors decrease from iteration $k = 15$ to $k = 30$. Comparing the

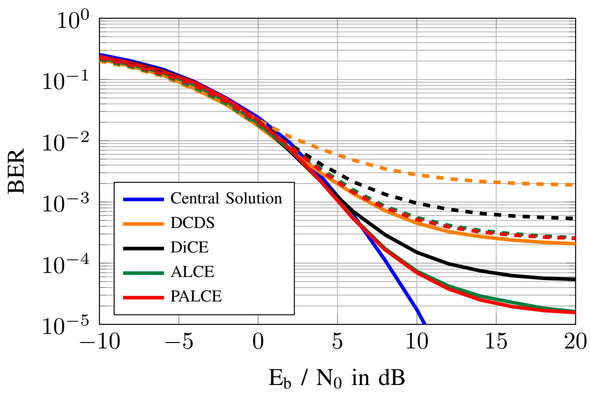

Fig. 5. BER performance over SNRs for considered distributed algorithms, $N_{\mathrm{UE}} = 2, N_{\mathrm{T}} = 2, N_{\mathrm{SC}} = 4, N_{\mathrm{R}} = 2$, fully meshed topology with ideal BH, 15 iterations (····), 30 iterations (—)

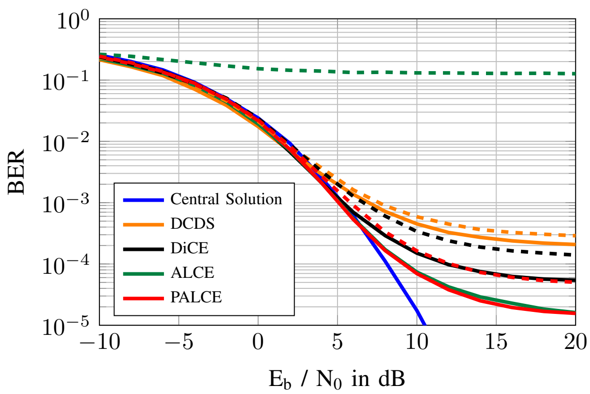

Fig. 6. BER performance over SNRs for considered distributed algorithms, $N_{\mathrm{UE}} = 2, N_{\mathrm{T}} = 2, N_{\mathrm{SC}} = 4, N_{\mathrm{R}} = 2$, fully meshed topology at iteration $k = 30$, ideal BH links (—), noisy BH links (····)

performance of those algorithms, significant improvement has been achieved by ALCE and PALCE, since much lower BER error floors are achieved compared to DCDS [2] and DiCE [3].

Fig. 6 shows the BER performance for ideal BH links as well as noisy BH links based on a fully meshed topology. Here, all iterative algorithms are terminated at iteration $k = 30$, leading to error floors at high SNRs. For the implementation with noisy BH links, the variables are additionally distributed by noise with variance of 0.01 during the inter-node exchange. Thus, the algorithms implemented with noisy BH links show lower performance than the implementations with ideal BH links. Moreover, the ALCE algorithm implemented with noisy BH links shows extremely low performance as discussed in Section III, noticing that the error floor of ALCE with noisy BH links is much higher than other algorithms. Nevertheless, the PALCE algorithm still shows higher performance compared to the DCDS and DiCE algorithms for both BH conditions.

TABLE I
COMMUNICATION OVERHEAD PER ITERATION FOR VARIOUS TOPOLOGIES

	$\mathrm{D_{DCDS}}$	$\mathrm{D_{DiCE}}$
Arbitrary	$(4\lvert\mathcal{E}\rvert + N_{\mathrm{SC}})N_{\mathrm{I}}L$	$2(\lvert\mathcal{E}\rvert + N_{\mathrm{SC}})N_{\mathrm{I}}L$
Full Mesh	$(2N_{\mathrm{SC}}^2 - N_{\mathrm{SC}})N_{\mathrm{I}}L$	$N_{\mathrm{SC}}(N_{\mathrm{SC}} + 1)N_{\mathrm{I}}L$
Ring	$5N_{\mathrm{SC}}N_{\mathrm{I}}L$	$4N_{\mathrm{SC}}N_{\mathrm{I}}L$

	$\mathrm{D_{ALCE}}$	$\mathrm{D_{PALCE}}$
Arbitrary	$N_{\mathrm{SC}}N_{\mathrm{I}}L$	$(0.5\lvert\mathcal{E}\rvert + N_{\mathrm{SC}})N_{\mathrm{I}}L$
Full Mesh	$N_{\mathrm{SC}}N_{\mathrm{I}}L$	$N_{\mathrm{SC}}(N_{\mathrm{SC}} + 3)/4N_{\mathrm{I}}L$
Ring	$N_{\mathrm{SC}}N_{\mathrm{I}}L$	$1.5N_{\mathrm{SC}}N_{\mathrm{I}}L$

B. Communication Overhead

In the following, the issue of communication overhead for all algorithms is investigated. As discussed above, information needs to be exchanged among neighboring SCs via BH links during the iterative processing. In each iteration, $N_{\mathrm{I}}L$ variables are transmitted per link within frame length L. Moreover, point to multi-point wireless BH links are assumed here, such that the local estimates $\mathbf{x}_j$ in ALCE only need to be broadcasted once from SC j to its neighboring SCs. Thus, the total amount of exchanged variables per iteration for ALCE $\mathrm{D_{ALCE}}$ depends on the number of SCs N_{SC}. However, compared to ALCE, PALCE needs to transmit extra multipliers $\boldsymbol{\lambda}_{ji}$ from SC j to its neighboring SC i, which are directed transmissions. Consequently, the total amount of exchanged variables per iteration for PALCE $\mathrm{D_{PALCE}}$ depends on not only the number of SCs N_{SC} but also the number of edges $\lvert\mathcal{E}\rvert$. To this end, we consider an arbitrary topology and two particular topologies i.e., full mesh and ring for investigating the communication overhead. As a comparison, the overhead per iteration for the ADMM-based algorithms DCDS $\mathrm{D_{DCDS}}$ and DiCE $\mathrm{D_{DiCE}}$ [8] are also taken into account. Table I lists for ADMM-based and ALM-based algorithms the total amount of exchanged variables per iteration considering various topologies. As can be seen, ALCE needs the same amount of exchanged variables per iteration for each topology, which is the lowest among these algorithms. For the fully meshed network, the number of edges is counted as $\lvert\mathcal{E}\rvert = N_{\mathrm{SC}}(N_{\mathrm{SC}} - 1)/2$. Thus, $N_{\mathrm{SC}}(N_{\mathrm{SC}} - 1)/4$ additional variables are transmitted due to the priority based transmission for multipliers $\boldsymbol{\lambda}_{ji}$ in PALCE. Nevertheless, compared to the $\mathrm{D_{DCDS}}$ and $\mathrm{D_{DiCE}}$ indicated in the Table I, both ALCE and PALCE can significantly reduce the amount of exchange variables per iteration.

Hence, considering both metrics discussed above, Fig. 7 shows the averaged BER versus the total number of exchanged variables based on a fully meshed network for an SNR of 10 dB. Additionally, the error rates after $k = \{1, 10, 20, .., 50\}$ iterations are labeled with markers. Depicted are the results of distributed algorithms based on a) ideal BH links and b) erroneous BH links with noise variance of 0.01. In Fig. 7 a), ALCE shows a great advantage in reducing the overhead compared to PALCE as well as DCDS and DiCE. However, since no multiplier $\boldsymbol{\lambda}_{ji}$ is exchanged during the iterative processing of ALCE, the performance becomes unreliable if the BH links are noisy as shown in Fig. 7 b), where the PALCE and ADMM-based algorithms are more robust. Moreover, by avoiding exchange of auxiliary variables $\mathbf{z}_j$ and

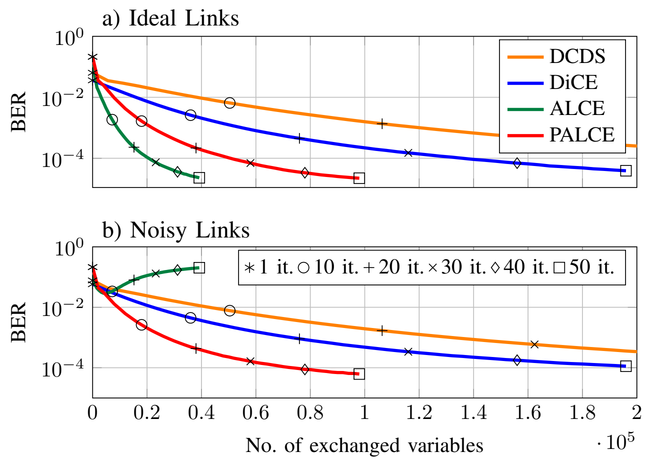

Fig. 7. BER vs. communication overhead, $N_{\mathrm{UE}} = 2, N_{\mathrm{T}} = 2, N_{\mathrm{SC}} = 4, N_{\mathrm{R}} = 2$. Full mesh, a) ideal BH links, b) noisy BH links

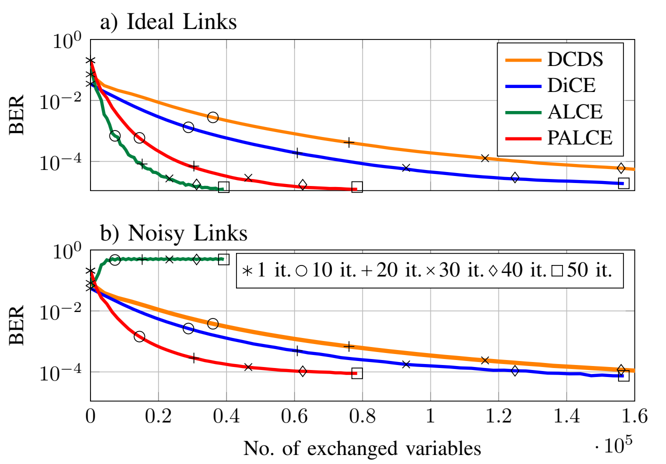

Fig. 8. BER vs. communication overhead, $N_{\mathrm{UE}} = 2, N_{\mathrm{T}} = 2, N_{\mathrm{SC}} = 4, N_{\mathrm{R}} = 2$. Ring, a) ideal BH links, b) noisy BH links

reducing exchange of multipliers $\boldsymbol{\lambda}_{ji}$ due to the priority based transmission, PALCE needs lower overhead for achieving the same performance compared to DCDS and DiCE. Fig. 8 depicts the results of these algorithms implemented in a ring connected network assuming the same system configuration. Compared to Fig. 7, fewer variables are exchanged for the same number of iterations, due to the lower connectivity of the network. Nevertheless, similar results as shown in Fig. 8 can be observed. ALCE can still reduce the overhead significantly compared to the other algorithms in a network with ideal BH links. Similarly, the performance of ALCE implemented with noisy BH links is deteriorated as shown in Fig. 8 b). In general, the PALCE algorithm can achieve more satisfying performance than the ADMM-based algorithms, since extra auxiliary variables $\mathbf{z}_j$ are required to be broadcasted in DiCE [8] and extra variables $\mathbf{z}_{ji}$ and $\mathbf{z}'_{ji}$ need to be transmitted directionally in DCDS [2]. Thus, DCDS produces the highest overhead for the same performance among all distributed algorithms.

V. Conclusion

For cooperative processing in small-cell networks, we proposed the ALCE and PALCE algorithms to improve the performance compared to the ADMM-based algorithms from the state of the art. Both algorithms are presented regarding the performance, reliability as well as the communication overhead. According to the analysis and simulations, improved performance has been achieved by both algorithms compared to the ADMM-based algorithms in reducing the overhead for the iterative processing considering a network with ideal BH links. Furthermore, PALCE has enhanced the robustness for the application in a network with noisy inter-node links compared to ALCE, and still outperforms the ADMM-based algorithms. In the future work, both algorithms can be further optimized by numerical optimization approaches, and the robustness of ALCE might to be improved.

VI. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Program (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. Wu, S. Rangan, and H. Zhang, *Green Communications: Theoretical Fundamentals, Algorithms and Applications*. Florida, USA: CRC Press, 2012.

[2] 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.

[3] H. Paul, J. Fliege, and A. Dekorsy, "In-Network-Processing: Distributed Consensus-Based Linear Estimation," *IEEE Communications Letters*, vol. 17, no. 1, pp. 59–62, Jan. 2013.

[4] 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, Jan. 2010.

[5] 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, Sep. 2013.

[6] B.-S. Shin, H. Paul, D. Wübben, and A. Dekorsy, "Reduced Overhead Distributed Consensus-Based Estimation Algorithm," in *IEEE Globecom Workshop (IWCPM 2013)*, Atlanta, USA, Dec. 2013.

[7] G. Xu, H. Paul, D. Wübben, and A. Dekorsy, "Fast Distributed Consensus-based Estimation for Cooperative Wireless Sensor Networks," in *18th International ITG Workshop on Smart Antennas (WSA 2014)*, Erlangen, Germany, Mar. 2014.

[8] D. Wübben, H. Paul, B.-S. Shin, G. Xu, and A. Dekorsy, "Distributed consensus-based estimation for small cell cooperative networks," in *10th International Workshop on Broadband Wireless Access (BWA 2014), co-located with IEEE Globecom 2014*, Austin, TX, USA, Dec. 2014.

[9] D. Jakovetic, J. M. F. Moura, and J. M. F. Xavier, "Linear Convergence Rate of a Class of Distributed Augmented Lagrangian Algorithms," *ArXiv e-prints*, July 2013.

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

[11] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[12] D. Jakovetic, J. Xavier, and J. Moura, "Cooperative Convex Optimization in Networked Systems: Augmented Lagrangian Algorithms With Directed Gossip Communication," *IEEE Trans. Signal Processing*, vol. 59, no. 8, pp. 3889–3902, Aug 2011.