

# Backhaul-Constrained Multi-Cell Cooperation Leveraging Sparsity and Spectral Clustering

Swayambhoo Jain, Seung-Jun Kim and Georgios B. Giannakis

**Abstract**—Multi-cell cooperative processing with limited backhaul traffic is studied for cellular uplinks. Aiming at reduced backhaul overhead, a sparse multi-cell linear receive-filter design problem is formulated. Both unstructured distributed cooperation as well as clustered cooperation, in which base station groups are formed for tight cooperation, are considered. Dynamic clustered cooperation, where the sparse equalizer and the cooperation clusters are jointly determined, is solved via alternating minimization based on spectral clustering and group-sparse regression. Furthermore, decentralized implementations of both unstructured and clustered cooperation schemes are developed for scalability, robustness and computational efficiency. Extensive numerical tests verify the efficacy of the proposed methods.

## I. INTRODUCTION

### A. Background

Rapid growth in wireless data traffic due to wide adoption of mobile devices has reinforced the research and development efforts to boost spectral efficiency of cellular networks. As the cell deployment becomes denser with a smaller cell size, it is recognized that inter-cell interference is a major bottleneck in provisioning desired link quality uniformly in the service area. Multi-cell cooperative processing (MCP), also referred to as network multi-input multi-output (MIMO), has shown promises for alleviating, and in many cases, exploiting, inter-cell interference. The idea is to have the base stations (BSs) collaborate in transmission and reception to/from the mobile stations (MSs), with the necessary coordination among the BSs taking place through the backhaul links [1]. The premise is that heavy interference experienced by the terminals residing at the cell boundaries is mitigated such that overall network throughput as well as fairness among users are improved. Field trials have verified that significant gain is indeed realizable [2], [3].

There are a number of approaches for implementing MCP [4], [3]. The BSs can loosely cooperate to avoid strong interference to/from the cell-edge users by exchanging channel state and scheduling information, and performing judicious

power control and spatial filtering. When a high-speed low-latency backhaul network is available, more proactive and tighter coordination can be employed, where the antennas located at different BSs are transformed into a virtual antenna array. In the downlink, collaborating BSs can share transmitted symbols to engage in cooperative MIMO transmission. In the uplink, received signal samples at different BSs can be processed together to jointly decode multiple MSs. Such cooperative transmission and reception allow exploiting inter-cell interference constructively to actually improve link quality. Sophisticated coding strategies may also be adopted to expand the achievable rate region. As the dominant computational burden is still placed on the BSs, MSs can enjoy performance improvement at minimal increase in complexity.

To achieve the promised gain of MCP, important practical issues such as the finite backhaul capacity, latency, and synchronization issues must be addressed [5]. Unlimited backhaul capacity assumed in many of the earlier theoretical works is impractical for large cellular systems [6], [7]. Rather, the backhaul traffic volume must be contained in practice. In the same token, it may be infeasible to connect a large number of BSs to a central coordinator due to limitations in long-range backhauling. Instead, more localized communication structure may be desirable [8], [9]. Also, the channel state information may contain uncertainties due to practical training, quantization, and latency [10]. Maintaining synchronization across multiple cells is challenging. Moreover, complexity of various MCP algorithms may become prohibitive as the size of the network grows, calling for scalable distributed solutions.

The main focus of this work is to incorporate the limited backhaul rates. For instance, in the recent 3GPP (3rd Generation Partnership Project) Long Term Evolution (LTE) and LTE-Advanced cellular standards, the BSs can coordinate their transmission and reception by communicating over the logical X2 interface, or the S1 interface (through the core network), in the Coordinated Multi-Point (CoMP) transmission/reception mode [11]. The backhaul rate requirements vary depending on the specific methods of coordination. Coordinated scheduling of multiple BSs in the sub-frame timescale requires backhauling on the order of several hundred kbps [12]. For joint decoding, the requirement goes up to several Mbps [2], [3], [5]. Since analog signal samples (albeit in quantized or compressed formats) received from distributed antennas must be pooled together for joint processing, the backhaul overhead is quite significant in the uplink, compared to the downlink case where digital signals need to be shared [13]. Our focus in this work is on the most backhaul-intensive uplink joint decoding scenario.

Different models have been considered to capture the

Submitted September 17, 2015.

This work was supported by NSF-ECCS grants 1202135 and 1343248, and NSF-AST 1247885.

S. Jain and G. B. Giannakis are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA. E-mail: {jainx174, georgios}@umn.edu. Seung-Jun Kim is with the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA. E-mail: sjkim@umbc.edu.

Part of this work was presented at the IEEE 13th Workshop on Signal Processing Advances in Wireless Communications (SPAWC) held in Cesme, Turkey during June 17th–20th, 2012.

backhaul rate constraints [14]. A *central* coordination model assumes that all BSs are connected to a central station through finite-capacity backhaul links [15], [10]. The model is simple and thus amenable to theoretical analysis, but may have scalability and robustness issues in practical implementation. A *distributed* backhaul model assumes that the BSs can communicate with one another via finite-capacity backhaul links, which may be confined to a neighborhood [16], [17]. The backhaul capacity may be shared among the BSs for successive interference cancellation to achieve rates within a constant number of bits to the theoretical limits [18]. A third approach is to consider *clusters* of BSs that cooperate tightly, possibly with coordination across the clusters as well to a lesser degree [19], [20], [21].

### B. Proposed Approach

Our approach is to design a linear combiner for joint decoding with the dual objectives of minimizing the mean-square error (MSE) of the combiner output, as well as the required backhaul rate. To perform joint decoding, a BS collects received signal samples at the neighboring BSs and combine them using a linear equalizer. The amount of the backhaul traffic<sup>1</sup> incurred in this process is proportional to the number of the neighboring BSs from which the samples are collected. The number of cooperating BSs, in turn, is the same as the number of non-zero entries in the equalizer coefficient vector. To determine which coefficients to allow to be non-zero under the minimization objective requires a combinatorial search in principle. Our pragmatic approach is to adopt a convex relaxation of the problem in the spirit of the recent compressive sensing literature in order to design *sparse* equalizers [22].

Both distributed and clustered cooperation models are considered. In the distributed cooperation setup, each BS communicates directly with the cooperating BSs to obtain the signal samples. In the clustered scenarios, the BSs are partitioned into a number of non-overlapping clusters, and the signal samples are fully shared among the BSs in a cluster for joint decoding of the served MSs [20], [21]. It is assumed that the inter-cluster backhaul traffic constitutes the bottleneck, rather than the intra-cluster backhaul traffic. When the clusters are given and fixed, the equalizer weights corresponding to the inter-cluster feedback are sparsified. To solve jointly for dynamic clusters and equalizer weights, a spectral clustering approach is derived [23].

<sup>1</sup> In this work, the control signaling overhead is not explicitly captured in the backhaul overhead calculation, since it is negligible in comparison to the overhead due to sharing analog signal samples for joint decoding. The actual backhaul traffic rate, say in Mbps, depends on many implementation details such as the sampling frequency, quantization resolution, as well as various types of overhead. Therefore, in this work, what we term as the backhaul traffic captures the number of packets of samples transported from one BS to another BS through the backhaul network per unit time. Depending on how many samples (e.g. samples from multiple antennas at each BS) quantized with how many bits per sample are contained in one packet, the actual number in terms of Mbps will change. Also, due to this unspecified scaling, the backhaul traffic discussed in this work will not be represented in terms of particular units.

### C. Related Work

MCP in the downlink with limited backhaul rates has been investigated in the literature. Cooperative downlink transmission with per-BS power and Quality-of-Service (QoS) constraints was studied in [24]. A clustered cooperation scheme with linear processing in the downlink was proposed in [21]. In the context of multi-cell MIMO heterogeneous networks, joint BS clustering and beamformer design for downlink transmission were considered in [25], [26]. An energy-constrained beamformer design and BS-MS association for uplink and downlink were considered in [27]. Distributed precoder design and BS selection in a game theoretic framework were proposed in [28]. A semidefinite relaxation-based approach for backhaul-limited cooperation in the downlink was proposed in [29]. Particle swarm optimization was used for zero-forcing-type beamformer design in [30]. Our work focuses on the uplink and does not require coordination with MSs, with all MCP burden placed on the BSs.

Direct exchanges of the signal samples between the BSs were considered in the context of 3GPP LTE systems in [13], where distributed cooperation without central control was advocated. Overlapping clusters of BSs were elected based on proximity for uplink MCP in [19]. A greedy algorithm for dynamic clustering was proposed to maximize the uplink sum-rate in [20]. Successive interference cancellation was adopted under limited backhaul traffic in [18], and cooperative group decoding was considered in a similar setting in [31]. Here, our intention is to concentrate on simple linear processing but address the backhaul traffic volume issue in both distributed and clustered cooperation settings in a consistent framework, and also derive distributed algorithms for scalable implementation.

Compared to our conference precursor [32], decentralized implementation of the proposed distributed and clustered cooperation schemes is developed in the present work. Decentralized computation of spatial equalizers and clusters makes MCP scalable to large networks, robust to isolated points of failure, and more resource-efficient than centralized implementation, since pieces of the overall problem are solved concurrently at different BSs, coordinated by peer message exchanges. Decentralized implementation of the component algorithms, such as group-sparse regression, eigenvector computation for network Laplacians, and k-means clustering, is also discussed. In addition, extensive simulations are performed to verify the performance, including the use of multiple antennas, user fairness, and dynamic clustered cooperation scenarios.

The rest of the paper is organized as follows. Sec. II introduces the system model. Sec. III develops backhaul-constrained distributed cooperation, while Sec. IV presents static and dynamic clustered cooperation. Decentralized implementation is the subject of Sec. V. Numerical tests are performed in Sec. VI, and conclusions are provided in Sec. VII.

## II. SYSTEM MODEL

### A. Uplink Signal Model

Consider a multi-cell network uplink with  $N_B$  BSs and  $N_U$  MSs (users). Each BS  $b \in \mathcal{B} := \{1, 2, \dots, N_B\}$  is assumed to be equipped with  $A$  antennas. Thus, the total number of BS

antennas in the network is  $N_A := AN_B$ . The set of antennas that belong to BS  $b$  is denoted as  $\mathcal{A}_b$ . Each MS user  $u \in \mathcal{U} := \{1, 2, \dots, N_U\}$  possesses a single antenna. The serving BS of the  $u$ -th MS is denoted as  $b(u)$ . The set of MSs served by BS  $b$  is denoted as  $\mathcal{U}_b := \{u \in \mathcal{U} : b(u) = b\}$ .

Let  $x_u$  denote the symbol transmitted by the  $u$ -th MS in a particular time slot and (sub-)band. Define  $\mathbf{x} := [x_1, x_2, \dots, x_{N_U}]^T$ , where  $T$  stands for transposition. Similarly, the signal samples received by the  $N_A$  BS antennas are represented as  $\mathbf{y} := [y_1, y_2, \dots, y_{N_A}]^T$ , where  $y_a$  for  $a \in \mathcal{A} := \{1, 2, \dots, N_A\}$  denotes the sample taken by the  $a$ -th antenna. The flat-fading channel matrix  $\mathbf{H} \in \mathbb{C}^{N_A \times N_U}$  has its  $(a, u)$ -entry equal to the complex channel coefficient from the  $u$ -th MS to the  $a$ -th BS antenna. Thus, the input-output relationship for the uplink can be expressed compactly as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

where  $\mathbf{n}$  represents additive noise and possible interference from outside the network. Vectors  $\mathbf{x}$  and  $\mathbf{n}$  are assumed to be zero-mean complex random vectors with covariance  $\mathbf{I}$ , which can be justified through prewhitening and normalization, as well as by absorbing MS transmit-powers into  $\mathbf{H}$ . It is also assumed that  $\mathbf{x}$  and  $\mathbf{n}$  are uncorrelated.

With MCP, multiple BSs cooperate to decode  $\mathbf{x}$ . For simplicity of implementation, linear receiver processing is considered. If all  $N_A$  BS antennas in the network could fully cooperate, an estimate  $\hat{\mathbf{x}}$  of  $\mathbf{x}$  could be obtained using a linear equalizer  $\mathbf{W} \in \mathbb{C}^{N_U \times N_A}$  as

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y} \quad (2)$$

where  $\hat{\mathbf{x}} := [\hat{x}_1, \dots, \hat{x}_{N_U}]^T$ . Thus, in order to obtain the estimate  $\hat{x}_u$  of  $x_u$  of MS  $u \in \mathcal{U}$ , BS  $b(u)$  would need to collect samples received by other BS antennas, and compute  $\mathbf{w}_u^T \mathbf{y}$ , where  $\mathbf{w}_u^T$  denotes the  $u$ -th row of  $\mathbf{W}$ . A widely used linear equalizer, also adopted here, is the linear minimum mean-square error (LMMSE) one, given by

$$\mathbf{W}_{\text{lmmse}} = (\mathbf{H}^H \mathbf{H} + \mathbf{I})^{-1} \mathbf{H}^H \quad (3)$$

where  $^H$  denotes Hermitian transposition.

### B. Distributed Cooperation Model

In the simplest cooperation model, henceforth termed distributed cooperation, each BS collects from cooperating BSs the received samples necessary for decoding its served MSs [13]. The set of cooperating BSs may overlap, in the sense that the set of BSs helping BS  $b$  may have nonempty intersection with the set of BSs helping BS  $b'$  for  $b \neq b'$  [19], [33]. The BSs share the samples over the backhaul network. If unlimited backhaul traffic were allowed, every BS could collect samples from all other BSs to perform full cooperation. Since the backhaul capacity is limited in practice, each BS must judiciously choose the set of BSs whose samples are most helpful in decoding its intended MSs.

Given the equalizer matrix  $\mathbf{W}$ , the amount of backhaul traffic can be characterized. To get the basic idea, consider again the case of decoding MS  $u$ 's signal by forming  $\mathbf{w}_u^T \mathbf{y}$ . Clearly, collecting the entries in  $\mathbf{y}$  that are measured by BSs

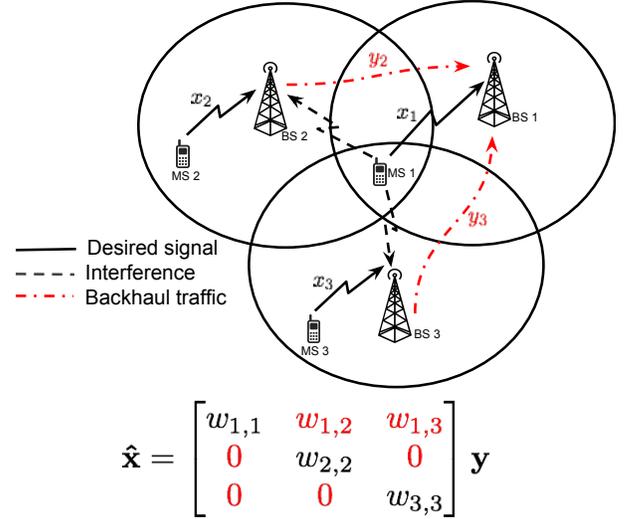


Fig. 1. An example MCP model with 3 single-antenna BSs. MS 1's transmission causes interference at BSs 2 and 3, but these BSs can cooperate by backhauling their received samples to BS 1, where joint decoding of MS 1 is performed. The backhaul traffic manifests itself as the non-zero off-diagonal entries in the first row of  $\mathbf{W}$ .

other than the serving BS  $b(u)$  must be conveyed to BS  $b(u)$  through the backhaul links. These are the samples  $\{y_a\}$  for  $a \in \mathcal{A} \setminus \mathcal{A}_{b(u)}$ . However, if the  $a$ -th entry  $w_{ua}$  of  $\mathbf{w}_u^T$  is zero, then the corresponding entry  $y_a$  need not be collected. Therefore, the actual backhaul traffic amount incurred is proportional to the number of nonzero entries in  $\{w_{ua}\}_{a \in \mathcal{A} \setminus \mathcal{A}_{b(u)}}$ .

The idea can be applied straightforwardly in the case of single-antenna BSs; that is when  $A = 1$ . Assume that  $N_U = N_A = N_B$  with  $b(u) = u$  for all  $u \in \mathcal{U}$ , and that the  $b$ -th antenna belongs to BS  $b$ ; that is  $\mathcal{A}_b = \{b\}$  for  $b \in \mathcal{B}$ . Then, under the distributed cooperation model, the amount of backhaul traffic is proportional to the number of nonzero off-diagonal entries of  $\mathbf{W} \in \mathbb{C}^{N_B \times N_B}$ . An illustration for the case with 3 BSs is provided in Fig. 1.

In the case of multi-antenna BSs with  $A > 1$ , each BS may well serve more than one MS. There are two considerations to take into account when estimating the amount of backhaul traffic in the case of multi-antenna cell sites.

- c1) Once BS  $b$  has collected samples from a cooperating BS  $b'$  to decode a user  $u \in \mathcal{U}_b$ , no additional backhaul traffic is incurred even if these samples are used for decoding other MSs  $u' \neq u$  that are in  $\mathcal{U}_b$ . In other words, once the  $(u, a')$ -entry of  $\mathbf{W}$  is determined to be nonzero, where  $u \in \mathcal{U}_b$  and  $a' \notin \mathcal{A}_b$ , one can improve the decoding performance without incurring additional backhaul traffic by allowing  $(u', a')$ -entries of  $\mathbf{W}$  to be nonzero for  $u' \in \mathcal{U}_b$ .
- c2) When a signal sample received from antenna  $a \in \mathcal{A}_b$  is collected, it may be desirable to collect together the samples from all the antennas in  $\mathcal{A}_b$  associated with that BS in order to economize the overhead due to backhaul

packet headers and to maximize the MCP benefit.<sup>2</sup>

A way to succinctly capture c1) and c2) is to consider the set

$$\left\{ \sum_{u \in \mathcal{U}_b} \sum_{a \in \mathcal{A}_{b'}} |w_{ua}|^2 \mid b, b' \in \mathcal{B}, b \neq b' \right\} \quad (4)$$

where it is worth noting that in order for an element to be zero, all squared terms in the sum must be zero. Thus, the backhaul traffic incurred is proportional to the number of nonzero elements in the set in (4).

### C. Clustered Cooperation Model

In the clustered cooperation, BSs are partitioned into a number of non-overlapping clusters, and the received samples are fully shared among the BSs belonging to a cluster so that the MSs are served jointly [20], [21]. Since the clusters do not overlap, the cluster-edge users may suffer from interference [34]. In some cases, natural clustering could emerge by the deployment constraints of the network, such as the hierarchical routing architecture of the backbone network, or the geographical proximity of the cell sites. In other cases, dynamic formation and adaptation of clusters may be beneficial. In the latter case, clusters themselves can be optimized based on channel gains or traffic patterns [20].

It is assumed that inter-cluster backhaul traffic is more costly to the backhaul network than intra-cluster traffic. For example, when multiple cells are controlled by a single cell tower, backhauling between these cell sites are virtually free. Similarly, the particular routing topology of the backhaul implementation may dictate certain backhauling paths are much cheaper than others. Inter-cluster backhaul traffic may pass through multiple layers of routing hierarchy, incurring higher signaling overhead. The precise mathematical backhaul traffic characterization will be presented in Sec. IV. Next, the two modes of MCP operation will be studied one by one.

## III. DISTRIBUTED COOPERATION

### A. Sparse LMMSE Equalizer

For simplicity of exposition, consider first the case with  $A = 1$ ,  $N_B = N_U = N_A$ ,  $\mathcal{U}_b = \{b\}$  and  $\mathcal{A}_b = \{b\}$  for all  $b \in \mathcal{B}$ , as in Sec. II-B. Since the amount of backhaul traffic is proportional to the number of nonzero off-diagonal entries of  $\mathbf{W}$ , a natural approach to obtain a backhaul-constrained MCP solution is to promote sparsity in the off-diagonal entries of  $\mathbf{W}$  (that is, favor many zero entries in the off-diagonal positions with minimal sacrifice in the MCP gain). Inspired by recent advances in compressive sensing techniques, the idea here is to optimize the cost functions related to the MMSE criterion, augmented with sparsity-promoting regularization terms [22], [35].

<sup>2</sup>As all-IP networking is advocated for cellular backhauling, the backhaul traffic is often routed using packet switching, which incurs header overhead. Assuming that the multiple antennas at a BS receive the MS signal at similar strengths, it is reasonable to collect all the samples from the multiple antennas together to maximize the MCP benefit. In this case, although the payload size increases linearly in the number of antennas, the header overhead is constant, as the samples may be aggregated into a single packet.

Let  $\tilde{\mathbf{W}} \in \mathbb{R}^{N_B \times N_B}$  denote the matrix with zero diagonal entries, and the off-diagonal entries equal to the absolute values of the corresponding entries in  $\mathbf{W}$ .<sup>3</sup> The LMMSE equalizer seeks to minimize the mean-square error (MSE)  $\mathbb{E}\{\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2\} = \|\mathbf{I} - \mathbf{W}\mathbf{H}\|_F^2 + \|\mathbf{W}\|_F^2$ . Thus, one can penalize large values of  $\ell_0$ -“norm” of  $\tilde{\mathbf{W}}$  as in

$$\min_{\mathbf{W}} \|\mathbf{I} - \mathbf{W}\mathbf{H}\|_F^2 + \|\mathbf{W}\|_F^2 + \lambda \|\tilde{\mathbf{W}}\|_0 \quad (5)$$

where  $\|\tilde{\mathbf{W}}\|_0$  denotes the total number of nonzero entries in  $\tilde{\mathbf{W}}$ ,  $\|\cdot\|_F$  the Frobenius norm, and  $\lambda$  controls the amount of backhaul traffic incurred by the solution.

Since the criterion in (5) is nonconvex, to obtain a solution efficiently, we advocate replacing the  $\ell_0$ -“norm” with the  $\ell_1$ -norm, which offers the tightest convex relaxation of the  $\ell_0$  counterpart. Thus, the relevant optimization problem is

$$\min_{\mathbf{W}} \|\mathbf{I} - \mathbf{W}\mathbf{H}\|_F^2 + \|\mathbf{W}\|_F^2 + \lambda \|\tilde{\mathbf{W}}\|_1 \quad (6)$$

which is equivalent to

$$\min_{\tilde{\mathbf{W}}} \|\bar{\mathbf{I}} - \tilde{\mathbf{W}}\bar{\mathbf{H}}\|_F^2 + \lambda \|\tilde{\mathbf{W}}\|_1 \quad (7)$$

where  $\bar{\mathbf{I}} := [\mathbf{I} \ \mathbf{0}]$ ,  $\bar{\mathbf{H}} := [\mathbf{H} \ \mathbf{I}]$ , and  $\|\tilde{\mathbf{W}}\|_1$  is the sum of the (absolute) values of all entries of  $\tilde{\mathbf{W}}$ .

*Remark 1:* In practice, one must choose an appropriate value of  $\lambda$  to obtain the desired backhaul traffic amount. For a large enough value of  $\lambda$ , the entire matrix  $\mathbf{W}$  will be equal to zero. Thus, a viable strategy is to first compute an upper bound  $\lambda_{\max}$  for the  $\lambda$  value that yields the all-zero solution, and to search in the interval  $[0, \lambda_{\max}]$ .  $\lambda_{\max}$  can be computed in a closed form. The least angle regression (LARS) algorithm can be used to efficiently calculate the entire regularization path, which is the mapping between  $\lambda$  and the corresponding sparsity level (backhaul traffic amount) [37].

*Remark 2:* In our formulation, the total backhaul traffic amount is considered, and the backhaul traffic per BS is not individually controlled. However, the latter can be attained in the same framework by introducing different weights for individual rows of  $\mathbf{W}$ . For example, employing weights  $\{\lambda_b\}$ , one can consider

$$\min_{\{\mathbf{w}_b\}_{b=1}^{N_B}} \sum_{b=1}^{N_B} [\|\bar{\mathbf{I}}(b, \cdot) - \mathbf{w}_b^T \bar{\mathbf{H}}\|_F^2 + \lambda_b \|\tilde{\mathbf{w}}_b\|_1] \quad (8)$$

where  $\bar{\mathbf{I}}(b, \cdot)$  represents the  $b$ -th row of  $\bar{\mathbf{I}}$ ,  $\tilde{\mathbf{w}}_b^T$  the  $b$ -th row of  $\tilde{\mathbf{W}}$ , and  $\lambda_b$  captures the severity of the backhaul constraint for BS  $b$ .

### B. Multi-Antenna Case Using Group Sparsity

Considerations c1) and c2) discussed in Sec. II-B for characterizing the backhaul traffic naturally impose certain structural constraints in sparsifying  $\mathbf{W}$  in the multi-antenna case. That is, for any pair of BSs  $b$  and  $b'$  with  $b \neq b'$ , the

<sup>3</sup>Since the entries of  $\mathbf{W}$  are complex numbers, setting an entry to zero amounts to setting both the real and the imaginary parts of the entry to zero. Considering the absolute value of the complex entry satisfies this requirement, and leads later to the group sparsity formulation quite naturally [36]. More discussion on group sparsity will be provided in Sec. III-B.

weights  $w_{ua}$  for  $u \in \mathcal{U}_b$  and  $a \in \mathcal{A}_{b'}$  should be constrained to be either all zero or all nonzero.

Such a structure can be effected using penalty terms promoting *group sparsity* [36]. When a vector  $\mathbf{w} \in \mathbb{R}^N$  is partitioned to  $J$  subvectors as  $\mathbf{w} := [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_J^T]^T$  with  $\mathbf{w}_j \in \mathbb{R}^{N_j}$  and  $\sum_{j=1}^J N_j = N$ , rather than pursuing the sparsity of the individual entries in  $\mathbf{w}$  independently by using a penalty term of the form  $\lambda \|\mathbf{w}\|_1$ , one can encode group structures by using  $\lambda \sum_{j=1}^J \|\mathbf{w}_j\|_2$ . This gives rise to  $\mathbf{w}_j$ 's that are either entirely zero or nonzero.

Under these considerations, one can replace the penalty term in (7) with

$$\lambda \sum_{b,b' \in \mathcal{B}: b \neq b'} \sqrt{\sum_{u \in \mathcal{U}_b} \sum_{a \in \mathcal{A}_{b'}} |w_{ua}|^2} \quad (9)$$

where  $w_{ua}$  is the  $(u, a)$ -entry of  $\mathbf{W}$ . The first sum inside the square root collects the entries for all served MSs in BS  $b$ , and the second one for all the antennas in a cooperating BS  $b'$ . Another way of expressing this is by defining  $\tilde{\mathbf{W}} \in \mathbb{R}^{N_B \times N_B}$  with  $(b, b')$ -entry  $\tilde{w}_{bb'}$  for  $b, b' \in \mathcal{B}$  given as

$$\tilde{w}_{bb'} = \begin{cases} \sqrt{\sum_{u \in \mathcal{U}_b} \sum_{a \in \mathcal{A}_{b'}} |w_{ua}|^2} & \text{if } b \neq b' \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Then,  $\|\tilde{\mathbf{W}}\|_0$  represents the amount of backhaul traffic, and the relaxed problem (7) with this new  $\tilde{\mathbf{W}}$  can be solved to obtain a group-sparse LMMSE equalizer efficiently. Note that (10) coincides with the definition of  $\tilde{\mathbf{W}}$  in Sec. III-A when  $A = 1$ ,  $N_B = N_U$ ,  $\mathcal{U}_b = \{b\}$ , and  $\mathcal{A}_b = \{b\} \forall b \in \mathcal{B}$ .

#### IV. CLUSTERED COOPERATION

##### A. Static Clustering

In the clustered cooperation scenario, the set of BSs is partitioned into  $N_C$  non-overlapping clusters  $\{\mathcal{B}_c\}_{c=1}^{N_C}$ , i.e.,

$$\bigcup_{c=1}^{N_C} \mathcal{B}_c = \mathcal{B} \quad \text{and} \quad \mathcal{B}_c \cap \mathcal{B}_{c'} = \emptyset \text{ for } c \neq c'. \quad (11)$$

The BSs belonging to the same cluster fully cooperate by sharing the received signal samples. This can be done in a distributed fashion via broadcasting the samples within each cluster through the backhaul network, or by having a central unit per cluster such as the cluster head, which collects the samples from the cluster members for further processing. In the latter case, with a BS in each cluster elected as the cluster head, the intra-cluster traffic of cluster  $\mathcal{B}_c$  can be modeled as proportional to  $|\mathcal{B}_c| - 1$ , where  $|\cdot|$  denotes the set cardinality. In the former case, the intra-cluster backhaul traffic is proportional to  $|\mathcal{B}_c|(|\mathcal{B}_c| - 1)$ , since each BS in a cluster needs samples from all the other BSs in the same cluster. Note that electing cluster heads may incur additional control overhead and implementation issues.

In *static* clustered cooperation, the clusters  $\{\mathcal{B}_c\}_{c=1}^{N_C}$  are given and fixed, regardless of the system operating conditions such as the channel realizations. Static clustering is appealing thanks to its simplicity of implementation. Clustering based on the geographical proximity of cells or the network architectural

constraints is natural in this scenario. Cells that are within a fixed radius were clustered together in [19]. The distributed antenna system (DAS) connected to a central control unit makes a straightforward cooperating cluster [38].

Unlike the distributed cooperation discussed in Sec. III, clustered cooperation can have an edge effect. That is, while the users located in the center of a cluster can clearly benefit from the MCP, the users at the cluster edges can suffer from excessive interference [34]. Judicious inter-cluster cooperation can help mitigate such a drawback. To this end, it is necessary to assess the inter-cluster traffic overhead.

The inter-cluster backhaul traffic is proportional to the number of nonzero  $(b, b')$ -entries of  $\tilde{\mathbf{W}}$ , when BSs  $b$  and  $b'$  belong to different clusters, namely,  $b \in \mathcal{B}_c$  and  $b' \in \mathcal{B}_{c'}$  with  $c \neq c'$ . By relaxing this cardinality-based characterization using the  $\ell_1$ -norm, a suitable regularizer is

$$\lambda \sum_{(c,c'): c \neq c'} s(\mathcal{B}_c, \mathcal{B}_{c'}) \quad (12)$$

where  $s(\mathcal{B}_c, \mathcal{B}_{c'}) := \sum_{b \in \mathcal{B}_c} \sum_{b' \in \mathcal{B}_{c'}} \tilde{w}_{bb'}$ . Note that in the special case of singleton clusters, i.e., if  $N_C = N_B$  and  $\mathcal{B}_b = \{b\}$  for all  $b \in \mathcal{B}$ , then (12) reduces to (9).

It is useful to view the backhaul penalty in (12) from the perspective of graph cuts.<sup>4</sup> Consider a directed graph with the set of vertices comprising the BSs  $\mathcal{B}$ , and the set of directed edges  $\mathcal{E} := \{(b, b') : b, b' \in \mathcal{B}\}$  connecting the vertices. The edge weight associated with edge  $(b, b')$  is given by  $\tilde{w}_{bb'}$  for  $b, b' \in \mathcal{B}$ ; i.e.,  $\tilde{\mathbf{W}}$  is the affinity matrix. Then,  $s(\mathcal{B}_c, \mathcal{B}_{c'})$  is the sum of edge weights that connect the vertices in  $\mathcal{B}_c$  to the vertices in  $\mathcal{B}_{c'}$ . Thus, (12) is proportional to the graph cut on  $\mathcal{G} := (\mathcal{B}, \mathcal{E}, \tilde{\mathbf{W}})$  induced by the set of clusters  $\{\mathcal{B}_c\}$ , defined as

$$\text{cut}(\{\mathcal{B}_c\}) := \sum_{c=1}^{N_C} s(\mathcal{B}_c, \bar{\mathcal{B}}_c) = \sum_{(c,c'): c \neq c'} s(\mathcal{B}_c, \mathcal{B}_{c'}) \quad (13)$$

where  $\bar{\mathcal{B}}_c := \cup_{c' \neq c} \mathcal{B}_{c'}$ . Thus, static clustered MCP solves

$$\min_{\mathbf{W}} \|\bar{\mathbf{I}} - \mathbf{W}\bar{\mathbf{H}}\|_F^2 + \lambda \cdot \text{cut}(\{\mathcal{B}_c\}) \quad (14)$$

which is a convex optimization problem. Note that in the special case of singleton clusters, i.e., if  $N_C = N_B$  and  $\mathcal{B}_b = \{b\}$  for all  $b \in \mathcal{B}$ , then (14) reduces to the distributed cooperation problem in (7).

##### B. Dynamic Partitioning Using Spectral Clustering

When the clusters are *not* given a priori, or when one is allowed to adapt the clusters dynamically, it is of interest to determine the partitioning that provides the best tradeoff between the system performance and the feedback overhead, possibly with limited inter-cluster coordination. However, finding the best  $\{\mathcal{B}_c\}$  using the regularizer (12) leads to clusters of severely unbalanced sizes. For instance, taking  $\mathcal{B}_1 = \mathcal{B}$  and  $\mathcal{B}_c = \emptyset$  for  $c = 2, \dots, N_C$  is the degenerate optimum

<sup>4</sup>A graph is an ordered pair of a set of vertices and a set of edges. A weighted graph is a graph, whose edges are assigned with real numbers (weights). A cut of a weighted graph is the sum of the weights associated with the edges, whose endpoints lie in different partitions of vertices.

for (12) with full cooperation and zero inter-cluster traffic. Obviously, such unbalanced cluster sizes are undesirable as the intra-cluster traffic can become excessive.

To obtain clusters with balanced sizes, one may consider the *ratio cut* cost used in spectral graph partitioning [39]:

$$\text{rcut}(\{\mathcal{B}_c\}) := \sum_{c=1}^{N_C} \frac{s(\mathcal{B}_c, \bar{\mathcal{B}}_c)}{|\mathcal{B}_c|}. \quad (15)$$

Here, the clusters of small sizes are penalized as  $|\mathcal{B}_c|$  is in the denominator. Thus, a joint sparse LMMSE equalization and clustering problem is formulated as

$$\min_{\mathbf{W}, \{\mathcal{B}_c\}_{c=1}^{N_C}} \|\bar{\mathbf{I}} - \mathbf{W}\bar{\mathbf{H}}\|_F^2 + \lambda \cdot \text{rcut}(\{\mathcal{B}_c\}) \quad (16)$$

where  $\{\mathcal{B}_c\}$  must satisfy (11).

Even with fixed  $\mathbf{W}$ , problem (16) is known to be NP-complete [39]. To obtain an approximate solution, it is first noted that with fixed  $\mathbf{W}$ , optimization w.r.t.  $\{\mathcal{B}_c\}$  is precisely the minimum cut problem on graph  $\mathcal{G}$ , whose relaxation is what spectral clustering algorithms aim to solve [23], [40]. On the other hand, given  $\{\mathcal{B}_c\}$ , optimization w.r.t.  $\mathbf{W}$  is a convex problem, as noted in Sec. IV-A. Thus, an iterative approach is pursued, where one alternately optimizes over  $\mathbf{W}$  and  $\{\mathcal{B}_c\}$  until no more reduction in the cost is obtained.

Spectral clustering has been mostly applied to undirected graphs [23]. To perform spectral clustering on directed graphs, some care must be taken [40]. Define matrix  $\mathbf{D} \in \mathbb{R}^{N_B \times N_B}$  to be a diagonal matrix with  $b$ -th diagonal entry  $\sum_{b' \in \mathcal{B}} \tilde{w}_{bb'}$ . Define also  $\mathbf{L} := \mathbf{D} - \tilde{\mathbf{W}}$ . Consider an  $N_B \times N_C$  cluster indicator matrix  $\Phi$  whose  $(b, c)$ -entry is defined as

$$\phi_{bc} = \begin{cases} \frac{1}{\sqrt{|\mathcal{B}_c|}} & \text{if } b \in \mathcal{B}_c \\ 0 & \text{otherwise} \end{cases}. \quad (17)$$

As clusters are non-overlapping, it can be seen that  $\Phi^T \Phi = \mathbf{I}$ . Then, it follows that [cf. (15)]

$$\text{rcut}(\{\mathcal{B}_c\}) = \text{tr}(\Phi^T \mathbf{L} \Phi). \quad (18)$$

Thus, a minimum cut graph clustering problem based on the ratio cut objective can be stated as

$$\min_{\Phi} \text{tr}(\Phi^T \mathbf{L} \Phi) \text{ subject to } \Phi^T \Phi = \mathbf{I} \text{ and (17)}. \quad (19)$$

Spectral clustering drops constraint (17) such that  $\Phi$  takes real-valued (rather than binary) entries. However, since  $\mathbf{L}$  is not symmetric, one cannot apply the Rayleigh-Ritz theorem to the resulting relaxed problem directly. Noting that  $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{A}^T)$ , one can obtain a formulation equivalent to (19) (except for the dropped constraint (17)) as

$$\min_{\Phi} \text{tr}(\Phi^T \tilde{\mathbf{L}} \Phi) \text{ subject to } \Phi^T \Phi = \mathbf{I} \quad (20)$$

where  $\tilde{\mathbf{L}} := (\mathbf{L} + \mathbf{L}^T)/2$  is symmetric. The optimal solution to (20) is given by setting the columns of  $\Phi$  as the eigenvectors of  $\tilde{\mathbf{L}}$  corresponding to the  $N_C$  smallest eigenvalues. Clusters  $\{\mathcal{B}_c\}$  can then be found by running the k-means algorithm on the rows of  $\Phi$ . The overall algorithm is listed in Table I.

Input: $\lambda$ , number of clusters $N_C$ , initial equalizer $\mathbf{W}_0$ , small $\varepsilon \geq 0$
Output: $\{\mathcal{B}_c\}_{c=1}^{N_C}$ , $\mathbf{W}$
1: Initialize $\tilde{\mathbf{W}} \leftarrow \mathbf{W}_0$
2: Repeat:
3:     Compute $\tilde{\mathbf{W}}$ per (10).
4:     Set $\mathbf{D} \leftarrow \text{diag}(\tilde{\mathbf{W}}\mathbf{1})$ ( $\mathbf{1}$ is the all-one vector)
5:     Set $\mathbf{L} \leftarrow \mathbf{D} - \tilde{\mathbf{W}}$ and $\tilde{\mathbf{L}} \leftarrow (\mathbf{L} + \mathbf{L}^T)/2$
6:     Let $\phi_c$ be the eigenvector of $\tilde{\mathbf{L}}$ corresponding to the $c$ -th smallest eigenvalue for $c = 1, \dots, N_C$ .
7:     Run the k-means algorithm on the rows of $\Phi := [\phi_1, \dots, \phi_{N_C}]$ to obtain clusters $\{\tilde{\mathcal{B}}_c\}$ .
8: $\tilde{\mathbf{W}} = \arg \min_{\tilde{\mathbf{W}}} \ \bar{\mathbf{I}} - \tilde{\mathbf{W}}\bar{\mathbf{H}}\ _F^2 + \lambda \cdot \text{rcut}(\{\tilde{\mathcal{B}}_c\})$
9:     If the decrease in the objective in line 8 is less than $\varepsilon$ , stop.
10:    Set $\mathbf{W} \leftarrow \tilde{\mathbf{W}}$ and $\{\mathcal{B}_c\} \leftarrow \{\tilde{\mathcal{B}}_c\}$ .

TABLE I

JOINT MCP AND DYNAMIC PARTITIONING.

## V. DECENTRALIZED IMPLEMENTATION

The algorithms developed so far require knowledge of the full channel matrix  $\mathbf{H}$ . In practice, a particular BS might not have access to the full  $\mathbf{H}$ , but only to the local channel gains, such as the gains between itself and the nearby MSs including the ones served by the BS. This essentially provides one or part of a few rows of  $\mathbf{H}$ . A centralized implementation would need to collect all the rows of  $\mathbf{H}$  from individual BSs at a central processor, solve the appropriate MCP problem, and feed back the solution to the relevant BSs. However, such an implementation may not be very scalable, but rather vulnerable as the failure of the central processor can affect the entire system. Furthermore, the feedback overhead can be significant. A naive decentralized implementation would simply broadcast  $\mathbf{H}(b, \cdot)$  across all BSs, whereupon each BS can independently solve the relevant MCP problems for the entire system. Clearly, however, the computational resources are wasted. Moreover, in the case of dynamic clustering, the solution to the spectral clustering may not be unique when  $\tilde{\mathbf{L}}$  has repeated eigenvalues, and the results of the k-means algorithm may be different from BS to BS.

In view of these considerations, the following features are desired for the decentralized implementation.

- Only locally available channel state information should be used.
- The computation should be efficiently done by having individual BSs solve only the subproblems of the entire problem.
- There must be a consensus on the solution across the BSs at the end.

Next, decentralized algorithms for static and dynamic versions of MCP are derived, satisfying the aforementioned requirements.<sup>5</sup>

### A. Decentralized Algorithm for Statically Clustered MCP

In this section, a decentralized algorithm for solving (14) is developed. First, consider a communication graph  $\mathcal{G}_c := (\mathcal{B}, \mathcal{E}_c)$ , where the BSs  $b$  and  $b'$  can communicate whenever an undirected edge  $(b, b') \in \mathcal{E}_c$ . Not to count the same edge twice,

<sup>5</sup>The decentralized algorithm for static clustered MCP covers the case of distributed cooperation as well. This is because distributed MCP is a special case of static clustered MCP, where each cluster contains a single BS and the number of clusters is equal to number of BSs.

assume without loss of generality that  $b < b'$  for all  $(b, b') \in \mathcal{E}_c$ . It is assumed that  $\mathcal{G}_c$  is connected, which means that there is always a path from any node to any other node. To simplify the exposition, single-antenna BSs ( $A = 1$ ) are assumed in the sequel. However, it is straightforward to extend the algorithm to the case with  $A > 1$ . Finally, it is postulated that each BS knows the column of  $\mathbf{H}$  corresponding to its served MS. This means that the BS knows the channel coefficients from the MS to all other BSs in the network. This is not a very restrictive assumption as the MSs keep track of the channel coefficients of the neighboring BSs to perform tasks like the handoff.

With these assumptions, a decentralized algorithm for (14) can be obtained using the alternating direction method of multipliers (ADMM) [41], [42], [43]. First, it is noted that (14) can be equivalently written as

$$\min_{\mathbf{W}} \sum_{b=1}^{N_B} \left\{ \|\bar{\mathbf{I}}(:, b) - \mathbf{W}\bar{\mathbf{H}}(:, b)\|_F^2 + \frac{\lambda}{N_B} \sum_{(c, c'): c \neq c'} s(\mathcal{B}_c, \mathcal{B}_{c'}) \right\} \quad (21)$$

where  $\mathbf{M}(:, b)$  denotes the  $b$ -th column of matrix  $\mathbf{M}$ . Then, a key step in deriving a decentralized solution is to introduce local auxiliary variables that decouple the problem across different BSs. Specifically, consider  $\bar{\mathbf{W}} := \{\mathbf{W}_b\}_{b \in \mathcal{B}}$  and  $\mathbf{U} = \{\{\mathbf{U}_b\}_{b \in \mathcal{B}}, \{\mathbf{U}_e\}_{e \in \mathcal{E}_c}\}$ . Then, (21) can be re-written as

$$\min_{\bar{\mathbf{W}}, \mathbf{U}} \sum_{b=1}^{N_B} \left[ \|\bar{\mathbf{I}}(:, b) - \mathbf{U}_b \bar{\mathbf{H}}(:, b)\|_F^2 + \frac{\lambda}{N_B} \sum_{(c, c'): c \neq c'} s_b(\mathcal{B}_c, \mathcal{B}_{c'}) \right] \quad (22)$$

$$\text{subject to } \mathbf{W}_b = \mathbf{U}_b, \quad \forall b \in \mathcal{B} \quad (23)$$

$$\mathbf{W}_b = \mathbf{U}_{(b, b')} = \mathbf{W}_{b'}, \quad \forall (b, b') \in \mathcal{E}_c \quad (24)$$

where  $s_b(\mathcal{B}_c, \mathcal{B}_{c'}) := \sum_{\bar{b} \in \mathcal{B}_c} \sum_{\bar{b}' \in \mathcal{B}_{c'}} \tilde{w}_{b, \bar{b}\bar{b}'}$  and  $\{\tilde{w}_{b, \bar{b}\bar{b}'}\}$  are defined in the same way as  $\{\tilde{w}_{\bar{b}\bar{b}'}\}$  are generated, but from  $\mathbf{W}_b$ , not  $\mathbf{W}$ . The idea is that each BS  $b$  keeps a local copy  $\mathbf{W}_b$  of  $\mathbf{W}$ , and (24) enforces that the local copies of the neighboring BSs coincide. Thus, under the assumption that  $\mathcal{G}_c$  is connected,  $\mathbf{W}_b$  for all  $b \in \mathcal{B}$  are identical across the network. Also, at each BS  $b$ ,  $\mathbf{U}_b$  and  $\mathbf{W}_b$  are constrained to be the same by (23). Using separate  $\mathbf{W}_b$  and  $\mathbf{U}_b$  leads to decoupling of the MSE and the graph cut terms in (22), yielding closed-form local update rules.

To apply the ADMM technique, the Lagrangian is formed using the Lagrange multipliers  $\mathbf{V} := \{\{\mathbf{V}_b\}_{b \in \mathcal{B}}, \{\mathbf{V}_e, \bar{\mathbf{V}}_e\}_{e \in \mathcal{E}_c}\}$  as

$$\begin{aligned} & \mathcal{L}_\rho(\bar{\mathbf{W}}, \mathbf{U}, \mathbf{V}) \\ & := \sum_{b=1}^{N_B} \left[ \|\bar{\mathbf{I}}(:, b) - \mathbf{U}_b \bar{\mathbf{H}}(:, b)\|_F^2 + \frac{\lambda}{N_B} \sum_{(c, c'): c \neq c'} s_b(\mathcal{B}_c, \mathcal{B}_{c'}) \right] \\ & + \sum_{b=1}^{N_B} \rho \left[ \langle \mathbf{V}_b, \mathbf{W}_b - \mathbf{U}_b \rangle + \frac{1}{2} \|\mathbf{W}_b - \mathbf{U}_b\|_F^2 \right] \\ & + \sum_{(b, b') \in \mathcal{E}_c} \rho \left[ \langle \mathbf{V}_{(b, b')}, \mathbf{W}_b - \mathbf{U}_{(b, b')} \rangle + \frac{1}{2} \|\mathbf{W}_b - \mathbf{U}_{(b, b')}\|_F^2 \right. \\ & \left. + \langle \bar{\mathbf{V}}_{(b, b')}, \mathbf{W}_{b'} - \mathbf{U}_{(b, b')} \rangle + \frac{1}{2} \|\mathbf{W}_{b'} - \mathbf{U}_{(b, b')}\|_F^2 \right] \quad (25) \end{aligned}$$

Input: $\lambda, \rho$ , clusters $\{\mathcal{B}_c\}_{c=1}^{N_C}$ Output: $\{\mathbf{W}_b\}_{b=1}^{N_B}$ 1: Initialize $\{\mathbf{W}_b^{(0)}\}_{b \in \mathcal{B}}, \{\mathbf{V}_b^{(0)}\}_{b \in \mathcal{B}}, \{\mathbf{V}_e^{(0)}\}_{e \in \mathcal{E}_c}$ , and $\{\mathbf{U}_b^{(0)}\}_{b \in \mathcal{B}}$ as zero matrices. 2: Repeat for $k = 0, 1, \dots$ at each BS $b \in \mathcal{B}$ 3: Transmit $\mathbf{W}_b^{(k)}$ to neighbors. 4: Update $\mathbf{V}_b^{(k+1)}$ and $\mathbf{V}_e^{(k+1)}$ via (26) and (34). 5: Update $\mathbf{W}_b^{(k+1)}$ via (35). 6: Update $\mathbf{U}_b^{(k+1)}$ via (31). 7: $k \leftarrow k + 1$ 8: Until convergence 9: Output $\{\mathbf{W}_b^{(k)}\}$ .
--

TABLE II

DECENTRALIZED ALGORITHM FOR STATIC CLUSTERED MCP.

where  $\rho > 0$  is a constant parameter and inner-product  $\langle \mathbf{M}_1, \mathbf{M}_2 \rangle$  for complex matrices  $\mathbf{M}_1$  and  $\mathbf{M}_2$  is defined as  $\langle \text{Re}\{\mathbf{M}_1\}, \text{Re}\{\mathbf{M}_2\} \rangle + \langle \text{Im}\{\mathbf{M}_1\}, \text{Im}\{\mathbf{M}_2\} \rangle$  with  $\text{Re}\{\cdot\}$  and  $\text{Im}\{\cdot\}$  denoting the real and the imaginary parts, respectively.

The ADMM consists of three steps: the  $\mathbf{V}$ -update, the  $\bar{\mathbf{W}}$ -update and the  $\mathbf{U}$ -update. The update for the Lagrange multipliers  $\mathbf{V}$  can be done as

$$\mathbf{V}_b^{(k+1)} = \mathbf{V}_b^{(k)} + [\mathbf{W}_b^{(k)} - \mathbf{U}_b^{(k)}], \quad \forall b \quad (26)$$

$$\mathbf{V}_{(b, b')}^{(k+1)} = \mathbf{V}_{(b, b')}^{(k)} + [\mathbf{W}_b^{(k)} - \mathbf{U}_{(b, b')}^{(k)}], \quad \forall (b, b') \in \mathcal{E}_c \quad (27)$$

$$\bar{\mathbf{V}}_{(b, b')}^{(k+1)} = \bar{\mathbf{V}}_{(b, b')}^{(k)} + [\mathbf{W}_{b'}^{(k)} - \mathbf{U}_{(b, b')}^{(k)}], \quad \forall (b, b') \in \mathcal{E}_c \quad (28)$$

where superscript  $(k)$  signifies that the quantity corresponds to the  $k$ -th iteration.

The update for  $\bar{\mathbf{W}}$  is given by  $\bar{\mathbf{W}}^{(k+1)} = \arg \min_{\bar{\mathbf{W}}} \mathcal{L}_\rho(\bar{\mathbf{W}}, \mathbf{U}^{(k)}, \mathbf{V}^{(k+1)})$ , which can be decomposed into individual BSs as

$$\begin{aligned} \mathbf{W}_b^{(k+1)} = \arg \min_{\mathbf{W}_b} & \left[ \frac{\lambda}{N_B} \sum_{(c, c'): c \neq c'} s_b(\mathcal{B}_c, \mathcal{B}_{c'}) \right. \\ & + \frac{\rho}{2} \|\mathbf{W}_b - \mathbf{U}_b^{(k)} + \mathbf{V}_b^{(k+1)}\|_F^2 \\ & + \frac{\rho}{2} \sum_{b': (b, b') \in \mathcal{E}_c} \|\mathbf{W}_b - \mathbf{U}_{(b, b')}^{(k)} + \mathbf{V}_{(b, b')}^{(k+1)}\|_F^2 \\ & \left. + \frac{\rho}{2} \sum_{b': (b', b) \in \mathcal{E}_c} \|\mathbf{W}_b - \mathbf{U}_{(b', b)}^{(k)} + \bar{\mathbf{V}}_{(b', b)}^{(k+1)}\|_F^2 \right] \quad (29) \end{aligned}$$

which is essentially a group Lasso problem. The update for  $\mathbf{U}$  is done similarly as

$$\mathbf{U}^{(k+1)} = \arg \min_{\mathbf{U}} \mathcal{L}_\rho(\bar{\mathbf{W}}^{(k+1)}, \mathbf{U}, \mathbf{V}^{(k+1)}) \quad (30)$$

which is equivalent to

$$\begin{aligned} \mathbf{U}_b^{(k+1)} = \arg \min_{\mathbf{U}_b} & \left[ \|\bar{\mathbf{I}}(:, b) - \mathbf{U}_b \bar{\mathbf{H}}(:, b)\|_F^2 \right. \\ & \left. + \frac{\rho}{2} \|\mathbf{W}_b^{(k+1)} - \mathbf{U}_b + \mathbf{V}_b^{(k+1)}\|_F^2 \right], \quad \forall b \in \mathcal{B} \quad (31) \end{aligned}$$

$$\begin{aligned} \mathbf{U}_{(b, b')}^{(k+1)} = \arg \min_{\mathbf{U}_{(b, b')}} & \frac{\rho}{2} \left( \|\mathbf{W}_b^{(k+1)} - \mathbf{U}_{(b, b')} + \mathbf{V}_{(b, b')}^{(k+1)}\|_F^2 \right. \\ & \left. + \|\mathbf{W}_{b'}^{(k+1)} - \mathbf{U}_{(b, b')} + \bar{\mathbf{V}}_{(b, b')}^{(k+1)}\|_F^2 \right) \\ = \frac{1}{2} & \left( \mathbf{W}_b^{(k+1)} + \mathbf{W}_{b'}^{(k+1)} + \mathbf{V}_{(b, b')}^{(k+1)} + \bar{\mathbf{V}}_{(b, b')}^{(k+1)} \right), \\ & \forall (b, b') \in \mathcal{E}_c. \quad (32) \end{aligned}$$

Now, it is observed that by summing up (27)–(28) and plugging in (32), one obtains

$$\begin{aligned} & \mathbf{V}_{(b,b')}^{(k+1)} + \bar{\mathbf{V}}_{(b,b')}^{(k+1)} \\ &= \mathbf{V}_{(b,b')}^{(k)} + \bar{\mathbf{V}}_{(b,b')}^{(k)} + \mathbf{W}_b^{(k)} + \mathbf{W}_{b'}^{(k)} - 2\mathbf{U}_{(b,b')}^{(k)} = \mathbf{0} \end{aligned} \quad (33)$$

which can simplify the update equations. Specifically, (27) becomes

$$\mathbf{V}_{(b,b')}^{(k+1)} = \mathbf{V}_{(b,b')}^{(k)} + \frac{1}{2} \left( \mathbf{W}_b^{(k)} - \mathbf{W}_{b'}^{(k)} \right), \quad \forall (b,b') \in \mathcal{E}_c \quad (34)$$

and (29) becomes

$$\begin{aligned} \mathbf{W}_b^{(k+1)} = \arg \min_{\mathbf{W}_b} & \left[ \frac{\lambda}{N_B} \sum_{(c,c'):c \neq c'} s_b(\mathcal{B}_c, \mathcal{B}_{c'}) \right. \\ & + \frac{\rho}{2} \|\mathbf{W}_b - \mathbf{U}_b^{(k)} + \mathbf{V}_b^{(k+1)}\|_F^2 \\ & + \frac{\rho}{2} \sum_{b':(b,b') \in \mathcal{E}_c} \|\mathbf{W}_b - \frac{1}{2} \left( \mathbf{W}_b^{(k)} + \mathbf{W}_{b'}^{(k)} \right) + \mathbf{V}_{(b,b')}^{(k+1)}\|_F^2 \\ & \left. + \frac{\rho}{2} \sum_{b':(b',b) \in \mathcal{E}_c} \|\mathbf{W}_b - \frac{1}{2} \left( \mathbf{W}_b^{(k)} + \mathbf{W}_{b'}^{(k)} \right) - \mathbf{V}_{(b',b)}^{(k+1)}\|_F^2 \right], \\ & \quad \forall b \in \mathcal{B}. \end{aligned} \quad (35)$$

Thus, the updates are necessary only for  $\{\mathbf{V}_b\}_{b \in \mathcal{B}}$ ,  $\{\mathbf{V}_e\}_{e \in \mathcal{E}_c}$ ,  $\{\mathbf{W}_b\}_{b \in \mathcal{B}}$ , and  $\{\mathbf{U}_b\}_{b \in \mathcal{B}}$ , via (26), (34), (35), and (31), respectively.

The overall algorithm is summarized in Table II. In the beginning of each iteration, all BSs transmit their latest copies of  $\mathbf{W}_b^{(k)}$  to their one-hop neighbors. Then, each BS can locally carry out the updates in (26) and (34) to obtain  $\mathbf{V}_b^{(k+1)}$  and  $\mathbf{V}_{(b,b')}^{(k+1)}$ , respectively. Note that  $\mathbf{V}_{(b,b')}^{(k+1)}$  may be stored at both BSs  $b$  and  $b'$ , which facilitates the subsequent update via (35). After that, each BS can again locally update  $\mathbf{U}_b^{(k+1)}$  using (31). In summary, all updates can be performed locally at individual BSs through exchanging  $\mathbf{W}_b^{(k)}$  only with the one-hop neighbors in  $\mathcal{G}_c$ . It can be shown that with  $\rho > 0$ , the iterates  $\{\mathbf{W}_b^{(k)}\}_{b \in \mathcal{B}}$  all converge to the solution to (14) [44].

### B. Decentralized Implementation for Dynamically Clustered MCP

In case of dynamic clustering, solving (16) in a decentralized fashion is of interest. As in the centralized implementation in Table I, alternating minimization is employed, where  $\mathbf{W}$  and  $\{\mathcal{B}_c\}$  are updated sequentially until no improvement in the cost can be made. The update for  $\mathbf{W}$  while keeping clusters  $\{\mathcal{B}_c\}$  fixed is essentially an instantiation of static clustered MCP, which can be solved in a decentralized way via the algorithm in Table II, with the only modification that the graph cut penalty  $\lambda N_B^{-1} \sum_{(c,c'):c \neq c'} s_b(\mathcal{B}_c, \mathcal{B}_{c'})$  in (35) is replaced by that of the ratio cut as  $\lambda N_B^{-1} \sum_{c=1}^{N_C} s_b(\mathcal{B}_c, \bar{\mathcal{B}}_c) / |\mathcal{B}_c|$ .

After the execution of the algorithm in Table II, all BSs have the same local copies of  $\mathbf{W}$ . Therefore, the BSs can simply use this  $\mathbf{W}$  to perform spectral clustering to obtain the clusters  $\{\mathcal{B}_c\}_{c=1}^{N_C}$ . However, this wastes computational resources since the same clustering problem is solved at all BSs. An alternative

Input: $\mathbf{A}$ and $N_C$
Output: $\Phi := [\mathbf{q}_1, \dots, \mathbf{q}_{N_B}]^T$
1: For each BS $b \in \mathcal{B}$ , randomly initialize $\mathbf{q}_b^{(0)T} \in \mathbb{R}^{N_C}$ , set $k = 0$ , and run:
2: Repeat
3: Update $\psi_b^{(k+1)T} = \sum_{b' \in \mathcal{N}_b} a_{bb'} \mathbf{q}_{b'}^{(k)T}$
4: Form $\mathbf{K}_b := \psi_b^{(k+1)} \psi_b^{(k+1)T}$ and perform consensus averaging to obtain $\mathbf{K} := \sum_{b=1}^{N_B} \mathbf{K}_b$ .
5: Compute Cholesky factorization on $\mathbf{K} = \mathbf{R}^T \mathbf{R}$ .
6: Compute $\mathbf{q}_b^{(k+1)T} = \psi_b^{(k+1)T} \mathbf{R}^{-1}$ .
7: $k \leftarrow k + 1$
8: Until convergence
9: Set $\mathbf{q}_b = \mathbf{q}_b^{(k)}$ for all $b \in \mathcal{B}$ .
10: For $N_C$ largest absolute values of eigenvalues, set $ \lambda_i  = \sqrt{K_{ii}}$ , $i = 1, 2, \dots, N_C$ .

TABLE III

DECENTRALIZED COMPUTATION OF EIGENVECTORS OF  $\mathbf{A}$  [45].

is to perform decentralized spectral clustering, as explained next.

Spectral clustering involves two basic steps. First, one needs to compute the eigenvectors corresponding to  $N_C$  smallest eigenvalues of  $\tilde{\mathbf{L}}$ . Subsequently, the k-means algorithm must be executed on the rows of the matrix containing the  $N_C$  eigenvectors as columns. Therefore, these two steps should be implemented in a decentralized fashion, as outlined in the next subsections.

1) *Decentralized Eigenvector Computation:* The algorithm in [45] is an instance of the orthogonal iteration (OI), which is a generalization of the power method. The centralized OI would start with a random initial matrix  $\Phi \in \mathbb{R}^{N_B \times N_C}$  and repeat the steps  $\Psi \leftarrow \mathbf{A}\Phi$  and  $\Phi \leftarrow \text{orthonormalize}(\Psi)$  until convergence. The resulting  $\Phi$  matrix is the eigenvector matrix for  $\mathbf{A}$ .

In the decentralized OI, assuming that the  $b$ -th BS is equipped with the  $b$ -th row of  $\mathbf{A}$ , the  $b$ -th BS calculates only the  $b$ -th row of  $\Phi$  through localized message passing. Let  $\mathcal{N}_b := \{b' : a_{bb'} \neq 0, b' \in \mathcal{B}\}$  be the set of neighbors of BS  $b$  (including  $b$  itself) based on affinity matrix  $\mathbf{A}$ . Then, the  $b$ -th BS starts with a randomly initialized nonzero vector  $\mathbf{q}_b \in \mathbb{R}^{N_C}$ , and forms  $\psi_b := \sum_{b' \in \mathcal{N}_b} a_{bb'} \mathbf{q}_{b'}$ , where  $\{\mathbf{q}_{b'}\}_{b' \in \mathcal{N}_b, b' \neq b}$  are collected from the one-hop neighbors. To orthonormalize  $\Psi := [\psi_1, \psi_2, \dots, \psi_{N_B}]^T$  in a decentralized fashion, a QR decomposition is performed on  $\Psi = \mathbf{Q}\mathbf{R}$ . For this, the individual BSs form  $\mathbf{K}_b := \psi_b \psi_b^T$  and run a consensus averaging iteration to obtain  $\mathbf{K} := \sum_{b=1}^{N_B} \mathbf{K}_b = \Psi^T \Psi = \mathbf{R}^T \mathbf{R} \in \mathbb{R}^{N_C \times N_C}$ . The consensus averaging can be performed by sharing information only among the one-hop neighbors in the connected communication graph. Then,  $\mathbf{R}$  can be obtained at each BS via Cholesky factorization of  $\mathbf{K}$ . Subsequently, the  $b$ -th row of the orthonormalized version of  $\Psi$  can be obtained by  $\mathbf{q}_b^T = \psi_b^T \mathbf{R}^{-1}$ . Upon convergence,  $\Psi = \mathbf{A}\Phi = \Phi\Lambda$ , where  $\Lambda$  is the diagonal matrix with the eigenvalues on the diagonal. Thus,  $\mathbf{K} = \Psi^T \Psi = \Lambda^2$ . Therefore, the eigenvalue magnitudes are also obtained as the square roots of the diagonal entries in  $\mathbf{K}$ . The final algorithm is listed in Table III.

Since our clustering algorithm requires the eigenvectors  $\tilde{\mathbf{L}}$ , corresponding to the  $N_C$  smallest eigenvalues, a two-step procedure is used. First, the largest eigenvalue magnitude  $|\lambda_1|$  is obtained via the algorithm in Table III with  $\mathbf{A} := \tilde{\mathbf{L}}$  and

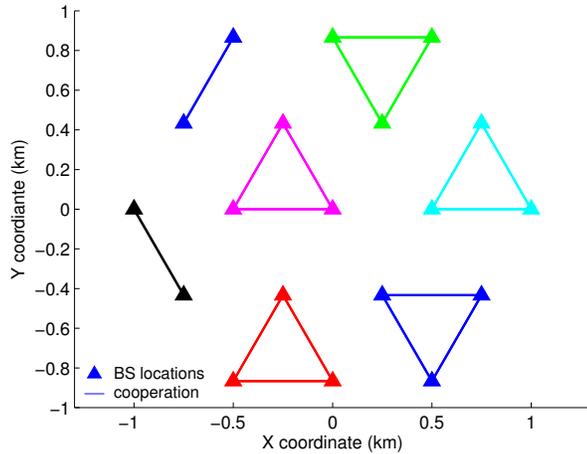


Fig. 2. A network of 19 BSs used for numerical experiments. Locations of BSs are indicated by solid triangles. The BSs belonging to a cluster are depicted in the same color and connected by solid lines. The clusters shown here were used for experiments of static clustered MCP.

$N_C = 1$ . Then, the same algorithm is executed again with  $\mathbf{A} := (|\lambda_1| + \epsilon)\mathbf{I} - \tilde{\mathbf{L}}$ , where  $\epsilon$  is a small positive number, and  $N_C$  set as the desired number of clusters. The last choice of  $\mathbf{A}$  ensures that it is positive definite with the eigenvalue order is reversed from  $\tilde{\mathbf{L}}$  as desired.

2) *Decentralized k-Means*: To finally obtain the desired clusters  $\{\mathcal{B}_c\}_{c=1}^{N_C}$ , the k-means algorithm must be performed on  $\{\mathbf{q}_b\}$ . A decentralized k-means algorithm developed in [46] employs an ADMM procedure, and does not require the exchange of raw data among the agents, but only the local estimates of the centroids in the one-hop neighborhood. In our setup, each BS possesses data vector  $\mathbf{q}_b$  from the decentralized eigendecomposition, and can readily execute this algorithm to obtain the cluster assignments. The decentralized k-means converges to a locally optimal solution [46].

The overall algorithm is the same as Table I, with line 6 substituted by the procedure explained in Sec. V-B1, line 7 by Sec. V-B2, and line 8 by the algorithm in Table II (with the ratio cut penalty).

*Remark 3*: The communication overhead of the proposed algorithms is commensurate to how often the algorithms are run, which, in turn, should depend on the coherence time of the channels. Focusing on the MSs with slow channel variation, and also initializing the iterative algorithms with the iterates from the most recent round, the impact of the feedback overhead can be minimized.

## VI. NUMERICAL TESTS

To test the proposed algorithms, a simple network comprising 19 BSs with 500 m cell radius is considered. The locations of the BSs are shown in Fig. 2 as triangles. The single antenna MSs were dropped uniformly in the coverage area. The channels between the MSs and the BSs are composite of the path-loss, log-normal shadowing with standard deviation 8 dB, and Rayleigh small-scale fading. The path-loss followed the 3GPP urban model given as  $PL(\text{dB}) = 148.1 + 37.6 \log_{10} d$ , where  $d$  is the distance between the BS and the MS in km. A path-loss exponent of  $\alpha = 3.76$  was used. The MSs were

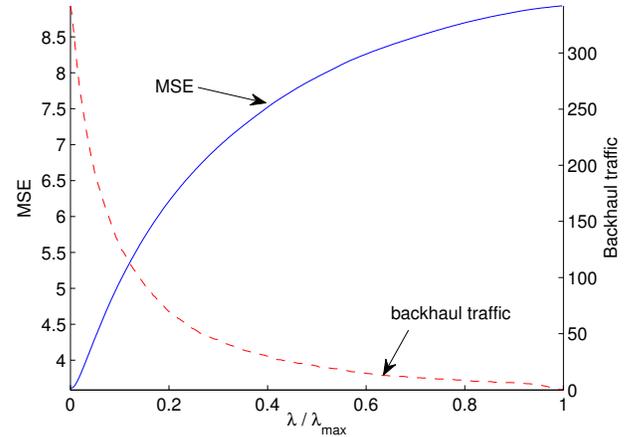


Fig. 3. MSE incurred by distributed cooperation in the network shown in Fig. 2. The number of antennas  $A = 1$  was used.

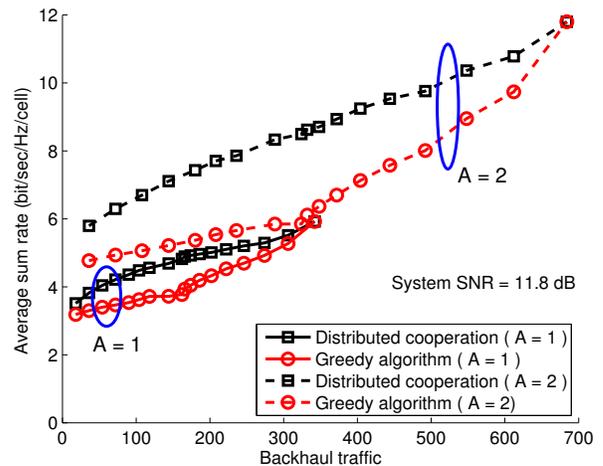


Fig. 4. Average sum rates in distributed cooperation at various backhaul traffic levels. The curves with solid lines represent the case of single-antenna BSs ( $A = 1$ ) and the ones with dashed lines correspond to  $A = 2$ . The total number of MSs is the same as the total number of antennas. The proposed distributed cooperation outperforms the greedy algorithm in both cases.

assigned to the BSs with highest long-term channel gain (that is, the channel gain with the small-scale fading averaged out). When not otherwise noted, single-antenna BSs are assumed. In all the experiments, the number MSs served by each BS is equal to the number of antennas at the BS. Thus, the total number of MSs is equal to  $N_A$ .

### A. Distributed Cooperation

The plots for MSE vs. backhaul traffic in distributed cooperation for 19 cell network of single antenna ( $A = 1$ ) BSs and 19 MSs is shown in Fig. 3. The system signal-to-noise-power-ratio (SNR), defined as the average SNR at which the serving BS receives the signals from the MS located at the cell edge without accounting for inter-cell interference, was 6.2 dB. Parameter  $\lambda_{\max}$  denotes the value of  $\lambda$  that yields sparse equalizer matrix with all off-diagonal entries as zero.

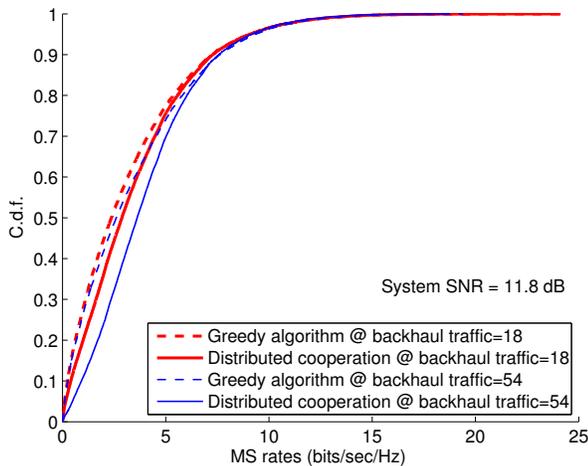


Fig. 5. C.d.f.'s of MS rates at two values of backhaul traffic. Distributed cooperation increases the system fairness.

Thus,  $\lambda = \lambda_{\max}$  represents no cooperation, whereas  $\lambda = 0$ , full cooperation among all 19 BSs. It can be seen that even with a small amount of backhaul traffic, significant reduction in MSE can be achieved. For instance, it can be seen that at  $\lambda \approx 0.3\lambda_{\max}$ , only about 12% of the backhaul traffic necessary for full cooperation (or about 40 units) is incurred, but this achieves an MSE of about 7.0. Compared to the MSE achievable by full cooperation at 3.6, and that with no cooperation at 8.9, this already captures about 36% of the entire feasible reduction.

The performance of the proposed algorithm is compared to that of a greedy clustering algorithm in [20]. The greedy clustering algorithm can be summarized as follows. Given a cluster size (the number of BSs per cluster), the greedy scheme picks the BS that yields the largest increase in the sum-rate as a member of the cluster sequentially. Once a cluster is formed, the algorithm moves on to the next cluster until all BSs belong to clusters. No inter-cluster backhauling is considered in the greedy scheme in [20]. However, it is assumed that full cooperation is performed inside clusters. Fig. 4 depicts the achieved average per-cell rates of the proposed distributed cooperation and the greedy scheme as functions of the total backhaul traffic incurred in the network, at the system SNR 11.8 dB. Again the backhaul traffic in both the cases was equal to the  $\ell_0$  norm of corresponding  $\tilde{\mathbf{W}}$  matrices. The curves with solid lines represent the case of single-antenna BSs ( $A = 1$ ) and the ones with dashed lines correspond to  $A = 2$ . The total number of MSs is equal to the total number of antennas. In both cases, it is clearly seen that the proposed scheme achieves the trade-offs better than the greedy algorithm, and the gap between the two increases with the larger number of antenna elements.

Fig. 5 shows the cumulative distribution function (c.d.f.) of the rates of the individual MSs achieved through distributed cooperation of single-antenna BSs. Two sets of curves are plotted. One set was obtained at the backhaul traffic amount of 18 units, and the other set of 54 units. In each set, the solid curve represents the c.d.f. due to the proposed algorithm,

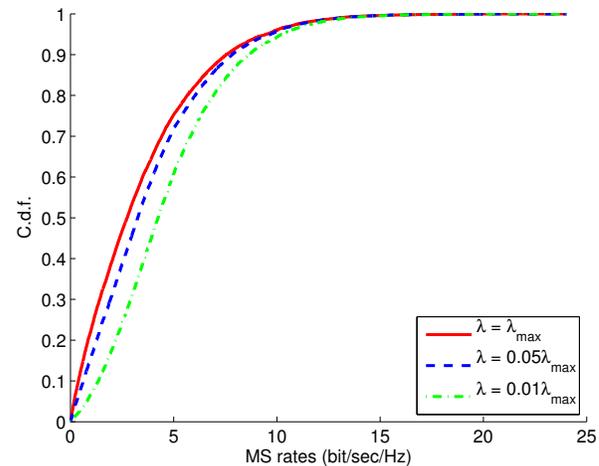


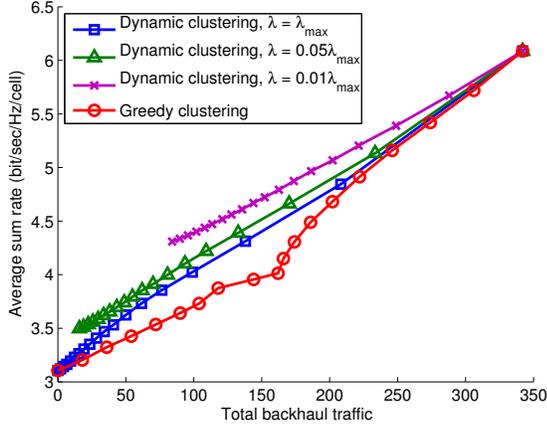
Fig. 6. C.d.f.'s of MS rates at different backhaul traffic levels. Static clustered cooperation with the clusters shown in Fig. 2 was used. The system fairness is improved as more inter-cluster backhaul traffic is allowed.

and the dashed one corresponds to that of the greedy scheme presented for comparison. For greedy clustering, the cluster sizes of 2 and 4 yield the backhaul traffic of 18 and 54, respectively. For distributed cooperation, the value of  $\lambda$  was adjusted to yield the same amounts of backhaul traffic. It can be observed that the cell-edge users enjoy far better rates under the proposed distributed cooperation at the same backhaul traffic level.

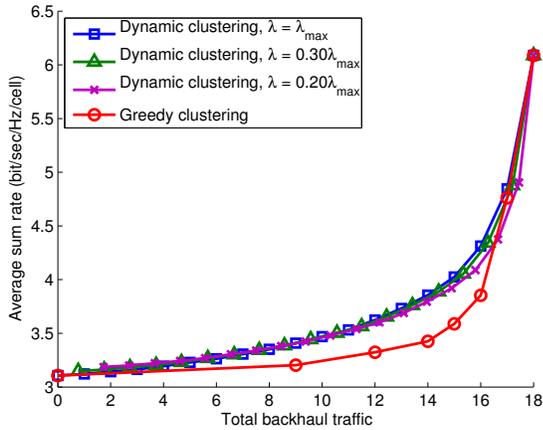
### B. Clustered Cooperation

First, the performance of the static clustered cooperation is verified. A seven-cluster partitioning as shown in Fig. 2 was employed. Single-antenna BSs were employed and full cooperation was used within clusters. The number of MSs in the system is 19. Due to the fixed clustering, the MSs at the cluster edges may experience excessive interference. Fig. 6 depicts the c.d.f. of the MS rates with varying degrees of inter-cluster cooperation. The curve for  $\lambda = \lambda_{\max}$  represents the performance without any inter-cluster cooperation. The average inter-cluster backhaul amounts due to  $\lambda = 0.05\lambda_{\max}$  and  $\lambda = 0.01\lambda_{\max}$  turned out to be 28.5 and 124, respectively. It can be seen that the rates of the cluster-edge mobiles are improved significantly through inter-cluster cooperation.

To assess performance of the dynamic clustered cooperation, the average per-cell rates for different backhaul traffic amounts are plotted in Fig. 7 at the system SNR equal to 11.8 dB. The algorithm in Table I was used with the initial equalizer  $\mathbf{W}_0$  set to the LMMSE equalizer  $\mathbf{W}_{\text{lmmse}}$ . Although we did not formally prove the convergence of the algorithm, it always converged within 2-3 iterations in our experiments. For comparison, the curve from the greedy clustering case is shown again. The markers in the greedy clustering curve correspond to different cluster sizes, which range from 1 to 19. Similarly, the markers in the dynamic clustering curves represent a variable number of clusters  $N_C$ , which also ranges from 1 to 19. The effect of inter-cluster cooperation was examined by varying the value of  $\lambda$ , where  $\lambda = \lambda_{\max}$  again



(a) Full distributed cooperation within clusters

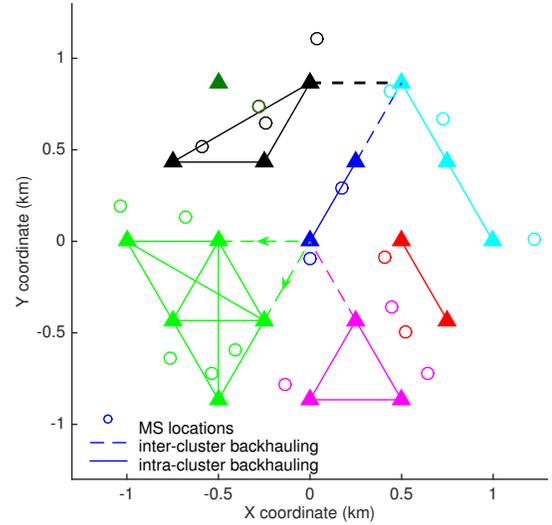


(b) Centralized cooperation within clusters

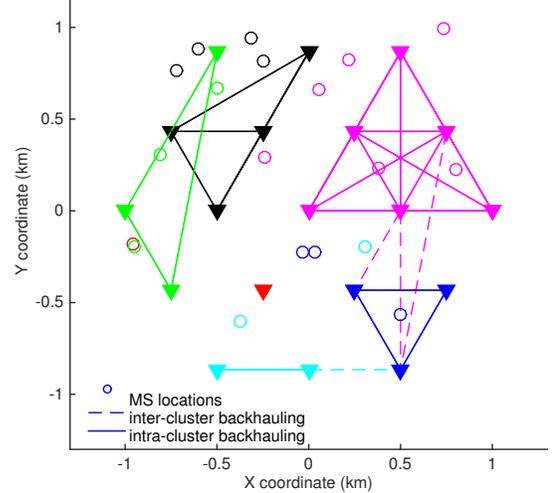
Fig. 7. Average sum rates in dynamic clustered cooperation at various backhaul traffic levels. Panel (a) is for the case where full distributed cooperation takes place within the clusters. Panel (b) corresponds to the case where centralized cooperation takes place at the cluster heads. Dynamic clustered cooperation outperforms greedy clustering in both cases.

signifies no inter-cluster cooperation. The backhaul traffic includes both the inter- and the intra-cluster backhauling. Fig. 7(a) was plotted by modeling the intra-cluster traffic for a cluster  $\mathcal{B}_c$  as  $|\mathcal{B}_c|(|\mathcal{B}_c| - 1)$  units, which represents the case where distributed cooperation occurs inside the clusters. On the other hand, the intra-cluster traffic per cluster  $\mathcal{B}_c$  was assumed equal to  $|\mathcal{B}_c| - 1$  units in Fig. 7(b), which corresponds to the situation where a BS inside each cluster is elected as the cluster head for centralized cooperation inside the cluster, with all the other cluster members feeding their samples to the cluster head. It can be seen that even without inter-cluster cooperation, the dynamic clustering algorithm outperforms the greedy scheme under both models of intra-cluster backhaul traffic. The performance is seen to be further improved by having inter-cluster cooperation, especially at low backhaul traffic, which is the regime of practical interest. However, this additional gain seems to be limited, in particular when central processing per cluster is assumed, as the spectral clustering-based dynamic BS partitioning itself has already absorbed the major portion of the MCP gain.

Instances of cooperating clusters formed by the dynamic



(a) no shadowing, system SNR = 11.2 dB



(b) with shadowing, system SNR = 16.2 dB

Fig. 8. Clusters obtained by dynamic clustering algorithm for the network of 19 single-antenna BSs/MSs shown in Fig. 2. Each MS is served by the BS with highest channel gain. The clusters shown are for  $\lambda = 0.1\lambda_{\max}$ .

clustering algorithm are depicted in Fig. 8. The solid lines represent the (bi-directional) intra-cluster cooperation links, and the dashed arrows denote the inter-cluster cooperation, obtained with  $\lambda = 0.1\lambda_{\max}$ . To check that the formed clusters make intuitive sense, the shadowing and the small-scale fading were suppressed when generating Fig. 8(a). Thus, each MS is served by the closest BS, and the cluster formation is solely dictated by the geometry of the BS and the MS locations. The MS locations are marked by the circles in Fig. 8. It can be observed that the clusters are mostly formed in such a way to protect the cell-edge users. Fig. 8(b) presents the case with shadow fading and the system SNR equal to 16.2 dB. Each MS is associated with the BS with highest channel gain, which might not be the closest BS necessarily. The color of each MS is matched to that of the cluster by which the MS is served.

The c.d.f. curve of the MS rates for dynamic clustered cooperation with  $N_C = 7$  and  $\lambda = \lambda_{\max}$  is shown in Fig. 9. For comparison, the curves corresponding to the greedy scheme

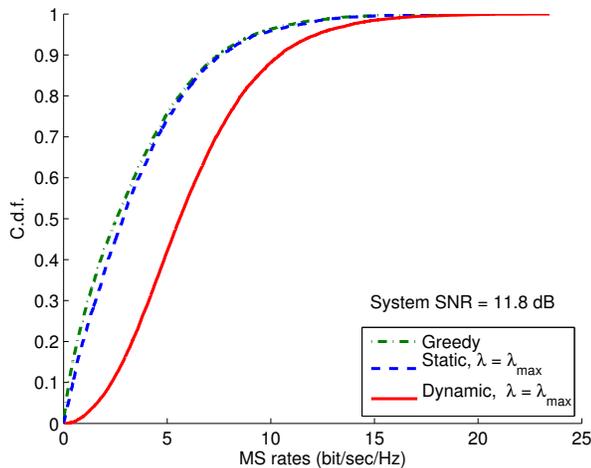


Fig. 9. C.d.f.'s of MS rates with dynamic clustered cooperation. The network in Fig. 2 with  $A = 1$  was used. Full cooperation was used within the clusters and no inter-cluster cooperation was allowed. Dynamic clustered cooperation performs the best, followed by the static clustered and the greedy scheme.

and the static clustered cooperation with the clusters in Fig. 2 are also provided. The figure underlines the clear improvement in fairness using the proposed dynamic clustering even without inter-cluster cooperation.

### C. Convergence of Decentralized Implementation

To illustrate the convergence of the decentralized algorithm for static clustered MCP in Table II, the left panel in Fig. 10 depicts the evolution of the difference between the objective in (21) for the centralized optimization and the objective (22) for the decentralized optimization is plotted. The clusters in Fig. 2 is again used with  $\lambda = 0.4\lambda_{\max}$  and  $\rho = 0.1$ . The communication graph for the decentralized algorithm was chosen so that each BS can communicate with its nearest neighbors. The plot shows that the decentralized objective quickly converges to that of the centralized algorithm. In the right panel of Fig. 10, the convergence of the iterates of

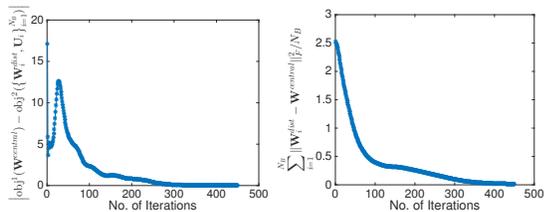


Fig. 10. Convergence of the decentralized algorithm for static clustered MCP.

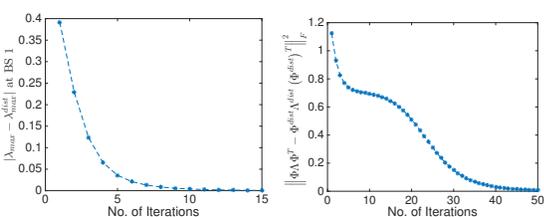


Fig. 11. Convergence of decentralized eigenvector computation.

the equalizer matrix is shown. It can be observed that the decentralized solution reaches global consensus toward the centralized solution.

The convergence of the decentralized eigenvector computation via the algorithm in Table III is illustrated in Fig. 11. The sparse equalizer obtained in Fig. 10 was used to obtain  $\tilde{\mathbf{L}}$ , and the largest eigenvalue was first computed using algorithm in Table III. The left panel of Fig. 11 shows the evolution of the difference between the centrally computed eigenvalue and the local copy at BS 1. It can be seen that the local copy converges quickly to the central solution. After this,  $N_C = 9$  top eigenvectors of  $\mathbf{A} = (\lambda_{\max}^{\text{dist}} + 0.1)\mathbf{I} - \tilde{\mathbf{L}}$  are computed. The right panel of Fig. 11 depicts the convergence of  $\Phi^{\text{dist}} \mathbf{\Lambda}^{\text{dist}} (\Phi^{\text{dist}})^T$ , where  $\Phi^{\text{dist}}$  contains the  $N_C$  top eigenvectors from the distributed algorithm, and  $\mathbf{\Lambda}^{\text{dist}}$  is the diagonal matrix with  $N_C$  top eigenvalues of  $\mathbf{A}$ . Again, the convergence is seen to be quite fast.

## VII. CONCLUSIONS

Backhaul-constrained MCP was considered for cellular uplinks. Exploiting recent compressive sensing techniques, a reduced-backhaul linear equalizer was obtained. In the case of clustered cooperation, sparsity was promoted on inter-cluster feedback. For the dynamic clustering set-up, a *joint* cluster formation and equalization problem was formulated, and an iterative algorithm based on spectral clustering was developed. Decentralized implementations of the latter were introduced, and validated with simulations. Numerical tests also demonstrated that significant reduction in MSE is possible at small increase in backhaul traffic.

## REFERENCES

- [1] M. K. Karakayali, G. J. Foschini, and R. A. Valenzuela, "Network coordination for spectrally efficient communications in cellular systems," *IEEE Wireless Commun.*, vol. 13, no. 4, pp. 56–61, Aug. 2006.
- [2] P. Marsch, M. Grieger, and G. Fettweis, "Large scale field trial results on different uplink coordinated multi-point (CoMP) concepts in an urban environment," in *Proc. of the WCNC*, Cancun, Mexico, Jun. 2011, pp. 1858–1863.
- [3] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011.
- [4] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: a new look at interference," *J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.
- [5] C. Yang, S. Han, X. Hou, and A. F. Molisch, "How do we design CoMP to achieve its promised potential?" *IEEE Wireless Commun.*, vol. 20, no. 1, pp. 67–74, Feb. 2013.
- [6] S. V. Hanly and P. Whiting, "Information-theoretic capacity of multi-receiver networks," *Telecommun. Syst.*, vol. 1, no. 1, pp. 1–42, Dec. 1993.
- [7] S. Shamai and B. M. Zaidel, "Enhancing the cellular downlink capacity via co-processing at the transmitting end," in *Proc. of the 53rd IEEE Vehicular Tech. Conf.*, vol. 3, May 2001, pp. 1745–1749.
- [8] E. Aktas, J. Evans, and S. Hanly, "Distributed decoding in a cellular multiple-access channel," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 241–250, Jan. 2008.
- [9] O. Simeone, O. Somekh, H. V. Poor, and S. Shamai, "Local base station cooperation via finite-capacity links for the uplink of linear cellular networks," *IEEE Trans. Info. Theory*, vol. 55, no. 1, pp. 190–204, Jan. 2009.
- [10] J. Hoydis, M. Kobayashi, and M. Debbah, "Optimal channel training in uplink network MIMO systems," *IEEE Trans. Sig. Proc.*, vol. 59, no. 6, pp. 2824–2833, Jun. 2011.

- [11] 3rd Generation Partnership Project (3GPP), "Further advancements for E-UTRA physical layer aspects," Tech. Rep. TR36.814, Mar. 2010.
- [12] S. Brueck, "Backhaul requirements for centralized and distributed cooperation techniques," presented at ITG Heidelberg, Jul. 2010. [Online]. Available: <http://www.ikr.uni-stuttgart.de/Content/itg/fg524/Meetings/2010-07-08-Heidelberg/index.html>
- [13] C. Hoymann, L. Falconetti, and R. Gupta, "Distributed uplink signal processing of cooperating base stations based on IQ sample exchange," in *Proc. of the IEEE Intl. Conf. Commun.*, Dresden, Germany, Jun. 2009.
- [14] S. Shamai, O. Simeone, O. Somekh, A. Sanderovich, B. M. Zaidel, and H. V. Poor, "Information-theoretic implications of constrained cooperation in simple cellular models," in *Proc. IEEE Personal, Indoor, Mobile Radio Commun. Symp. (PIMRC)*, Cannes, France, Sep. 2008, pp. 1–5.
- [15] A. Sanderovich, O. Somekh, H. V. Poor, and S. Shamai, "Uplink macro diversity of limited backhaul cellular network," *IEEE Trans. Info. Theory*, vol. 55, no. 8, pp. 3457–3478, Aug. 2009.
- [16] P. Marsch and G. Fettweis, "A framework for optimizing the uplink performance of a distributed antenna systems under a constrained backhaul," in *Proc. of the IEEE Intl. Conf. Commun.*, Glasgow, Scotland, Jun. 2007, pp. 975–979.
- [17] —, "Uplink CoMP under a constrained backhaul and imperfect channel knowledge," *IEEE Trans. Wireless Commun.*, vol. 10, no. 6, pp. 1730–1742, Jun. 2011.
- [18] L. Zhou and W. Yu, "Uplink multicell processing with limited backhaul via per-base-station successive interference cancellation," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 10, pp. 1981–1993, Oct. 2013.
- [19] S. Venkatesan, "Coordinating base stations for greater uplink spectral efficiency in a cellular network," in *Proc. of the 18th Annual IEEE Intl. Symp. on Pers., Indoor and Mobile Radio Commun. (PIMRC)*, Athens, Greece, Sep. 2007.
- [20] A. Papadogiannis, D. Gesbert, and E. Hardouin, "A dynamic clustering approach in wireless networks with multi-cell cooperative processing," in *Proc. of the IEEE Intl. Conf. Commun.*, Beijing, China, May 2008, pp. 4033–4037.
- [21] J. Zhang, R. Chen, J. G. Andrews, A. Ghosh, and R. W. Heath Jr., "Networked MIMO with clustered linear processing," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1910–1921, Apr. 2009.
- [22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
- [23] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
- [24] J. Zhao, T. Q. S. Quek, and Z. Lei, "Coordinated multipoint transmission with limited backhaul data transfer," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2762–2775, Jun. 2013.
- [25] M. Hong, R. Sun, H. Baligh, and Z.-Q. Luo, "Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 226–240, Feb. 2013.
- [26] W.-C. Liao, M. Hong, Y.-F. Liu, and Z.-Q. Luo, "Base station activation and linear transceiver design for optimal resource management in heterogeneous networks," *IEEE Trans. Sig. Proc.*, vol. 62, no. 15, pp. 3939–3952, Aug. 2013.
- [27] S. Luo, R. Zhang, and T. J. Lim, "Downlink and uplink energy minimization through user association and beamforming in C-RAN," *IEEE Trans. Sig. Proc.*, vol. 14, no. 1, pp. 494–508, Jan. 2014.
- [28] M. Hong and Z.-Q. Luo, "Distributed linear precoder optimization and base station selection for an uplink heterogeneous network," *IEEE Trans. Sig. Proc.*, vol. 61, no. 12, pp. 3214–3228, Jun. 2013.
- [29] F. Zhuang and V. K. N. Lau, "Backhaul limited asymmetric cooperation for MIMO cellular networks via semidefinite relaxation," *IEEE Trans. Sig. Proc.*, vol. 62, no. 3, pp. 684–693, Feb. 2014.
- [30] T. R. Lakshmana, C. Botella, and T. Svensson, "Partial joint processing with efficient backhauling in coordinated multipoint networks," in *Proc. of the IEEE 75th Vehicular Technology Conf. (VTC Spring)*, 2012, pp. 1–5.
- [31] Y. Li, X. Wang, S. Zhou, and S. Alshomrani, "Uplink coordinated multipoint reception with limited backhaul via cooperative group decoding," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3017–3030, Jun. 2014.
- [32] S.-J. Kim, S. Jain, and G. B. Giannakis, "Backhaul-constrained multicell cooperation using compressive sensing and spectral clustering," in *Proc. of the IEEE 13th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2012, pp. 65–69.
- [33] C. T. K. Ng and H. Huang, "Linear precoding in cooperative MIMO cellular networks with limited coordination clusters," *J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1446–1454, Dec. 2010.
- [34] G. Caire, S. A. Ramprasad, H. C. Papadopoulos, C. Pepin, and C.-E. W. Sundberg, "Multiuser MIMO downlink with limited inter-cell cooperation: Approximate interference," in *Proc. of the 46th Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Sep. 2008, pp. 730–737.
- [35] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
- [36] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. R. Statist. Soc. B*, vol. 68, no. 1, pp. 49–67, Feb. 2006.
- [37] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [38] S. Parkvall, E. Dahlman, A. Furuskär, Y. Jading, M. Olsson, S. Wänstedt, and K. Zangi, "LTE-Advanced – Evolving LTE towards IMT-Advanced," in *Proc. IEEE 68th Veh. Technol. Conf. (VTC)-Fall*, Calgary, Canada, Sep. 2008.
- [39] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Design*, vol. 11, no. 9, pp. 1074–1085, Sep. 1992.
- [40] M. Meila and W. Pentney, "Clustering by weighted cuts in directed graphs," in *Proc. of the SIAM Intl. Conf. on Data Mining*, Minneapolis, MN, Apr. 2007, pp. 135–144.
- [41] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy link—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Sig. Proc.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.
- [42] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, "Group-lasso on splines for spectrum cartography," *IEEE Trans. Sig. Proc.*, vol. 59, no. 10, pp. 4648–4663, 2011.
- [43] 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, 2011.
- [44] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [45] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," *J. Comput. and Syst. Sci.*, vol. 74, no. 1, pp. 70–83, Feb. 2008.
- [46] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE J. Selected Topics in Sig. Proc.*, vol. 5, no. 4, pp. 707–724, Aug. 2011.



**Swayambhoo Jain** completed his Bachelor of Technology degree in Electronics and Communication Engineering from National Institute of Technology, Calicut, India in 2007. He worked as Research & Development Engineer at IBM Systems and Technology Group – Banalore from 2007 - 2010. He obtained his Master of Science degree in Electrical Engineering from University of Minnesota, MN, USA in 2012. Since 2012, he has been pursuing PhD. in Electrical Engineering from University of Minnesota, MN, USA. His research interests include wireless communication, statistical signal processing and machine learning.



**Seung-Jun Kim (SM'12)** received his B.S. and M.S. degrees from Seoul National University in Seoul, Korea in 1996 and 1998, respectively, and his Ph.D. from the University of California at Santa Barbara in 2005, all in electrical engineering. From 2005 to 2008, he worked for NEC Laboratories America in Princeton, New Jersey, as a research staff member. He was with Digital Technology Center and Department of Electrical and Computer Engineering at the University of Minnesota during 2008-

2014, where his final title was Research Associate Professor. In August 2014, he joined Department of Computer Science and Electrical Engineering at the University of Maryland, Baltimore County, as an Assistant Professor. His research interests include statistical signal processing, optimization, and machine learning, with applications to wireless communication and networking, future power systems, and big data. He is serving as an Associate Editor for *IEEE Signal Processing Letters*.



**G. B. Giannakis (Fellow'97)** received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engineering, 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. Since 1999 he has been a professor with the Univ. of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing - subjects on which he has published more than 380 journal papers, 650 conference papers, 22 book chapters, two edited books and two research monographs (h-index 114). Current research focuses on learning from Big Data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the (co-) inventor of 25 patents issued, and the (co-) recipient of 8 best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.