Distributed Interference Alignment for MIMO Cellular Network Via Consensus ADMM

Sandeep Kumar†, and Ketan Rajawat*
Department of Electrical Engineering
Indian Institute of Technology, Kanpur, India;
emails: \{sandkr†, ketan*\}@iitk.ac.in

Abstract—This paper considers the design of beamforming matrices for multi-cell interference alignment (IA). Different from the existing centralized algorithms for coordinated IA, the focus here is on low-complexity distributed algorithms that are easy to implement on large networks. Towards this end, the rank-minimization form of the coordinated IA problem is formulated as a general-form consensus problem, that is solvable via a distributed alternating directions method of multipliers algorithm. The non-convexity of the original problem is overcome by utilizing a proximal update step. The proposed algorithm is flexible enough to allow delays and losses in the update process, while still being provably convergent to a stationary point. Simulations are carried out on a six-cell system, demonstrating the effectiveness of the proposed algorithm.

Index Terms—multi-cell coordinated beamforming, distributed optimization, interference alignment, non-convex, ADMM

I. INTRODUCTION

Coordinated multipoint (CoMP) has emerged as a potential technology for the next-generation high-density base station (BS) deployments. Of particular importance are the coordinated beamforming techniques that take advantage of the low-latency backhaul between BSs to exchange channel state information, in order to coordinate and mitigate inter-cell interference. Interference alignment (IA) via linear beamforming has been widely advocated as a low-complexity approach to CoMP [1]–[4]. Within the coordinated IA framework, the goal is to design the beamforming matrices so as to restrict the interference subspace at the receivers to a small number of dimensions.

In the general case, the design of linear beamforming matrices that achieve a certain number of degrees of freedom (DoF), is a difficult non-convex problem. Indeed, even ascertaining the feasibility of IA for a given network is non-trivial, and requires techniques from algebraic geometry [1], [5]–[7]. While simple IA designs are known in certain special cases [2], [8], iterative optimization algorithms are widely used for generic designs. The underlying idea behind these approaches is to formulate IA as a non-convex optimization problem and utilize techniques such as relaxation or block-coordinate descent to obtain approximate solutions. Within this context, the two main classes of formulations include minimization of the total interference power subject to various constraints [9], [10], and the more general rank-constrained rank-minimization formulation [2], [3], [11], [12]. Specifically, good performance has been reported within the latter framework when the a priori knowledge of the rank is included within the formulation; see [2] and references therein.

This paper considers the problem of designing linear beamforming matrices for IA in a distributed setting. Different from the centralized or fusion center-based implementations described thus far, the focus here is on fully distributed algorithms that can tolerate missing or delayed message exchanges. While several distributed algorithms exist for the more general CoMP problem, the same does not hold within the more practical multi-cell coordinated IA framework [13]–[17]. Distributed algorithms are however well-motivated for next-generation cellular networks due to their reduced backhaul signaling loads, lower computational complexity, and simpler network architecture. The last notion is particularly important for future wireless systems with a flat Internet Protocol architecture, where the BSs are directly connected to the network core [13]–[17]. Finally, the decomposition of the original network-wide optimization problem into local subproblem obviates the need for sharing the channel state information globally.

A distributed IA algorithm was proposed in [1], but it utilizes only local channel information, thus achieving limited alignment. In contrast, the goal here is to achieve network-wide alignment using a flexible inter-node message passing scheme. The main contribution of the present paper is to formulate the corresponding consensus problem and solve it via the distributed asynchronous proximal alternating direction method of multipliers (ADMM) method [18]. Towards this end, the proposed algorithm utilizes a matrix factorization approach that renders the objective function non-convex but differentiable. The non-convexity is handled via a low-complexity proximal update step, that enables asynchronous algorithm operation. As long as the worst-case delay is bounded, the proposed algorithm is known to converge to a local optimum. Only the uplink signal model is considered here, since the downlink beamformers can be obtained by using the uplink-downlink duality. Further, the proposed algorithm is also applicable to unconventional cellular networks such as next-generation device-to-device communication [19] and femtocell networks [20].

This paper is organized as follows. Sec. II introduces the system model considered here, and formulates the required non-convex optimization problem. The proposed distributed asynchronous consensus-based ADMM algorithm is detailed in Sec. III. Sec. IV provides the simulations results that
demonstrate the efficacy of the proposed algorithm.

II. SYSTEM MODEL

Consider a cellular network of $G$ BSs, each equipped with $N$ antennas. The backhaul network only allows communication between adjacent BSs, so that the neighborhood $N_g$ of the $g$-th BS includes all the adjacent BSs connected to $g$. The BS serves $K$ mobile users, denoted by the set $U_g$, and equipped with $M$ antennas each. There are a total of $GK$ users, with $K$ users per cell. Following the notation in [2], such a network is referred to as a $(G, K, N, M)$-network.

We consider the uplink case, where user $j \in U_g$ transmits a precoded signal vector over an $M \times N$ channel, and is received by the set of BSs $N_g^d := N_g \cup \{g\}$. For any user $1 \leq j \leq GK$, denoting the signal vector by $s_j \in \mathbb{C}^d$ and the transmit precoder by $V_j \in \mathbb{C}^{M \times d}$, then the received signal at the BS $g$ is given by

$$y_g = \sum_{i \in N_g^d \cup \{g\}} H_{ij} V_j s_j + n_g$$

(1)

where $H_{ij} \in \mathbb{C}^{M \times N}$ is the channel gain matrix between user $j$ and BS $g$, and $n_g \in \mathbb{C}^N$ is the circularly symmetric, additive white Gaussian noise vector. Subsequently, the received signal is decoded by the receive beamforming matrix $U_k$ for $k \in U_g$, resulting in

$$U_k^H y_g = \sum_{i \in N_g^d \cup \{g\}} U_k^H H_{ij} V_j s_j + U_k^H n_g$$

(2)

From (2), it can be observed that the inter-cell interference space for user $k \in U_g$ is the span of $M \times d$ matrices $\{H_{ij} V_j \forall i \in N_g^d \cup \{g\}\}$. Henceforth, the set $\{i \in N_g^d \cup \{g\}\}$ is compactly written as $\{j \notin U_g\}$.

When symbol extensions over time and frequency are not allowed, interference alignment entails obtaining matrices $\{V_j , U_j\} _j$ that satisfy [2, 5]

$$\sum_{j \notin U_g \cap \{i \}} U_k^H H_{ij} V_j = 0 \quad j \notin U_g , k \in U_g$$

(3)

$$\text{rank}(U_k^H H_{ij} V_k) = 3$$

(4)

for all $1 \leq g \leq G$. Alternatively, following the analysis in [2], it is possible to eliminate $\{U_k\}$, and impose the constraints in (3) by defining the interference matrix $R_g \in \mathbb{R}^{N \times (G-1)Kd}$, formed using the column vectors from the set $\{H_{ij} V_j \forall j \notin U_g\}$. For a $(3, 2, 4, 3)$ system, with 1 DoF/user, $R_1$ is a $4 \times 4$ matrix given by

$$R_1 = [H_{(3,1)} V_3 H_{(4,1)} V_4 H_{(5,1)} V_5 H_{(6,1)} V_6]$$

(5)

Further, decomposing $R_g = C_g D_g$. Where, $C_g \in \mathbb{C}^{N \times (N-Kd)}$, and the matrix $D_g \in \mathbb{R}^{(N-Kd) \times (G-1)Kd}$ is formed by concatenating the $(N-Kd) \times d$ matrices $\{D_{ij}\}_{i \neq g}$. This decomposition yields the following equivalent optimization problem

$$\min_{\{V_j\} \{C_g\} \{D_g\}} \sum_{g=1}^G \sum_{i \in N_g^d \cup \{g\}} \|H_{ij} V_j - C_g D_{ij}\|^2_F$$

s.t. $V_k (1 : d , 1 : d ) = I \quad \forall \ g , k \in U_g$ (6)

where $I$ denotes the $d \times d$ identity matrix. Here, $V_k (1 : d , 1 : d )$ is the column reduced echelon form of $V_k$, and the constraint $V_k (1 : d , 1 : d ) = I$ takes care of the rank constraint on $V_k$. The problem in (6) is non-convex and has been solved via the block-coordinate descent method in [2]. However, the performance of the BCD method was only studied via simulations, and no results on the asymptotic behavior of the variables were provided. Further, the BCD approach requires the channel state information to be collected at a central location, where the optimization is carried out. The present paper advocates a distributed consensus-based algorithm where communication is required only between the neighboring BSs. By exploiting the partially separable structure of the objective function in (6), the proposed algorithm not only eliminates the need for a central controller, but also allows asynchronous updates at the BSs. Finally, the asynchronous ADMM algorithm is guaranteed to converge to a stationary point of (6).

III. DISTRIBUTED ADMM FOR IA

In order to formulate (6) as a consensus problem, collect the local variables at the $g$-th BS as $X_g := \{V_k \}_{k \in U_g}, C_g, \{D_{ij}\}_{j \neq g}$. Since the $g$-th summand in (6) depends on the variables $\{X_i\} _i \in N_g^d$, introduce copies $X_g := \{\tilde{V}_j \}_{j \notin U_g} , \tilde{C}_g, \{\tilde{D}_{ij}\}_{j \neq g}$ corresponding to each $X_g$. It is now possible to formulate (6) as the following general-form consensus problem [18, 21]

$$\min_{X, \tilde{X}} \sum_{g=1}^G \sum_{i \in N_g^d \cup \{g\}} \|H_{ij} V_j - C_g D_{ij}\|^2_F$$

s.t. $V_j = \tilde{V}_j , C_i = \tilde{C}_i , D_{i\ell} = \tilde{D}_{i\ell}$, $1 \leq g \leq G, i \in N_g^d , j \in U_i , \ell \neq U_i$ (9)

$$\tilde{V}_j (1 : d , 1 : d ) = I \quad j \in \tilde{U}$$

(10)

where the set of all users $U := \cup_{g=1}^G U_g$, and $X$ and $\tilde{X}$ collect the variables $\{X_g\}$ and $\{\tilde{X}_g\}$, respectively. It is now possible to write down the augmented Lagrangian by associating dual variables $\{\gamma_{ij}\} , \{\Gamma_{ig}\}$, and $\{\Delta_{i\ell}\}$ with the three sets of constraints in (9). Since the constraint in (10) is local to each BS, it is left implicit at this stage. Collecting all the dual variables in $\Xi$, the augmented Lagrangian becomes

$$L(X, \tilde{X}, \Xi) = \sum_{g=1}^G \sum_{i \in N_g^d \cup \{g\}} \|H_{ij} V_j - C_g D_{ij}\|^2_F +$$

$$\sum_{g=1}^G \sum_{i \in N_g^d \cup \{g\}} \langle \gamma_{ij} , V_j - \tilde{V}_j \rangle + \frac{\rho_g}{2} \|V_j - \tilde{V}_j\|^2_F$$

$$+ \sum_{g=1}^G \sum_{i \in N_g^d \cup \{g\}} \langle \Gamma_{ig} , C_i - \tilde{C}_i \rangle + \frac{\rho_g}{2} \|C_i - \tilde{C}_i\|^2_F$$

$$+ \sum_{g=1}^G \sum_{i \in N_g^d \cup \{g\}} \langle \Delta_{i\ell} , D_{i\ell} - \tilde{D}_{i\ell} \rangle + \frac{\rho_g}{2} \|D_{i\ell} - \tilde{D}_{i\ell}\|^2_F$$

(11)

Here, $\rho_g \in \mathbb{R}^+$ is a penalty parameter that controls the rate of convergence. Next, the proximal ADMM algorithm for solving (8) is detailed.
A. Distributed Proximal ADMM

Starting with arbitrary $\Xi(1)$ and $X(1)$, the ADMM algorithm consists of the following two minimization operations at iteration $t$

$$\hat{X}(t+1) = \arg \min_X L(X(t), \hat{X}, \Xi(t)) \quad \text{s. t. (10)}$$ (12)

$$X(t+1) = \arg \min_X L(X, \hat{X}(t+1), \Xi(t)),$$ (13)

while the dual update that takes the form

$$\Upsilon_{gj}(t+1) = \Upsilon_{gj}(t) + \rho_g (V_{gj}(t+1) - \tilde{V}_{gj}(t+1))$$ (14a)

$$\Gamma_{ig}(t+1) = \Gamma_{ig}(t) + \rho_g (C_{ig}(t+1) - \tilde{C}_{i}(t+1))$$ (14b)

$$\Delta_{i\ell g}(t+1) = \Delta_{i\ell g}(t) + \rho_g (D_{i\ell g}(t+1) - \tilde{D}_{i\ell}(t+1))$$ (14c)

Of these, the minimization in (12) is straightforward, since it involves minimization of a convex quadratic function subject to an linear equality constraint, and can be carried out separately at each base station. Specifically, the updates at node $i$ take the form:

$$\tilde{V}_{g}(t+1) = \mathcal{P}_d \left( \frac{\sum_{g \in N_i} (\rho_g V_{gj}(t) + \Upsilon_{gj}(t))}{\sum_{j \in U_i} \rho_g} \right) , j \in U_i$$ (15a)

$$\tilde{C}_{i}(t+1) = \frac{\sum_{g \in N_i} (\rho_g C_{ig}(t) + \Gamma_{ig}(t))}{\sum_{g \in N_i} \rho_g}$$ (15b)

$$\tilde{D}_{i\ell}(t+1) = \frac{\sum_{g \in N_i \ell \notin U_i} (\rho_g D_{i\ell g}(t) + \Xi_{i\ell g}(t))}{\sum_{g \in N_i} \rho_g} , \ell \notin U_i$$ (15c)

where the projection operation in (15a) is defined such that the $(i,j)$-th entry of $\mathcal{P}_d(A)$, for a given matrix $A$ is given by:

$$\mathcal{P}_d(A) = \begin{cases} 1 & 1 \leq i = j \leq d \\ 0 & 1 \leq i \neq j \leq d \\ |A|_{ij} & i, j > d \end{cases}$$ (16)

The minimization in (13) is also separable, but is difficult since $L(X, \hat{X}(t+1), \Xi(t))$ is non-convex in $X$. To this end, the non-convex term in $L(X, \hat{X}(t+1), \Xi(t))$ is replaced with its first order approximation, calculated at $X(t+1)$. This yields the proximal variant of the ADMM, with the updates given by

$$V_{gj}(t+1) = \tilde{V}_{gj}(t+1) - \frac{2}{\rho_g} \tilde{H}_{gj}^H \tilde{H}_{gj} \tilde{V}_{gj}(t+1)$$

$$\quad + \frac{2}{\rho_g} \tilde{H}_{gj}^H \tilde{C}_{g}(t+1) \tilde{D}_{gj}(t+1) - \frac{1}{\rho_g} \Upsilon_{gj}(t)$$ (17a)

$$\tilde{C}_{g}(t+1) = \tilde{C}_{g}(t+1) - \frac{1}{\rho_g} \Gamma_{gg}(t)$$

$$\quad - \frac{2}{\rho_g} \sum_{j \notin U_i} \tilde{C}_{g}(t+1) \tilde{D}_{gj}(t+1) \tilde{D}_{gj}(t+1)^H$$

$$\quad - \sum_{j \notin U_i} \tilde{H}_{gj} \tilde{V}_{gj}(t+1) \tilde{D}_{gj}(t+1)$$ (17b)

$$\tilde{D}_{gj}(t+1) = \tilde{D}_{gj}(t+1) - \frac{1}{\rho_g} \Delta_{gj}(t)$$

$$\quad + \frac{2}{\rho_g} \tilde{C}_{g}^H(t+1) \tilde{C}_{g}(t+1) \tilde{D}_{gj}(t+1)$$

$$\quad + \frac{2}{\rho_g} \tilde{C}_{g}^H(t+1) \tilde{H}_{gj} \tilde{V}_{gj}(t+1)$$ (17c)

Observe here that updates are not specified for $\{C_{ig}\}_{i \in N_g}$ and $\{D_{i\ell g}\}_{i \in N_g}$. It is assumed that the dual variables are initialized so that $\Gamma_{ig} = 0$ for all $i \in N_g$ and $\Delta_{i\ell g} = 0$ for all $i \in N_g$, which yields the updates

$$C_{ig}(t+1) = \tilde{C}_{g}(t+1) , i \in N_g$$ (18)

$$\tilde{D}_{i\ell g}(t+1) = \tilde{D}_{i\ell}(t+1) , \ell \notin U_i , i \in N_g$$ (19)

since $\Gamma_{ig}(t+1) = \Gamma_{ig}(t) = 0$ and $\Delta_{i\ell g}(t+1) = \Delta_{i\ell}(t) = 0$ for all $t \geq 1$. In other words, the updates in (14b) and (14c) for $i \neq g$ are also not required.

The proximal ADMM based algorithm presented here is synchronous, and requires each node to allocate a fixed amount of computational and bandwidth resources per iteration. For instance, the computationally intensive steps, (15) and (17), are required to be carried out at each iteration. Similarly, each node must exchange the updates at every iteration, which may be difficult for some nodes in times of duress. Indeed, such an algorithm forces the network to wait for the slowest node, without allowing them to temporarily lag behind others.

B. Asynchronous Distributed Proximal ADMM

This section proposes an asynchronous ADMM algorithm that addresses the challenges outlined Sec. III. This is accomplished via two modifications: (a) first, nodes are allowed to skip updates in (15), and (b) second, the nodes are allowed to utilize old consensus variables $\hat{X}(\tau(t+1))$ in (17), where $\tau(t+1) \leq t+1$ for all $t \geq 1$. Denoting the nodes that do carry out the update in (15) by $S(t)$, the “update” at nodes $i \notin S(t)$ simply involves carrying forward the values of the consensus variables from the previous iteration, that is, $\tilde{V}_{gj}(t+1) = V_{gj}(t)$, $\tilde{C}_{i}(t+1) = C_{i}(t)$, and $\tilde{D}_{i\ell}(t+1) = D_{i\ell}(t)$, for all $j \in U_i$ and $\ell \notin U_i$. Such nodes also need not transmit anything since the neighboring nodes, after waiting for a fixed amount of time, may simply carry-forward upon not hearing anything.

Algorithm 1 describes an implementation of the asynchronous algorithm for interference alignment for a cellular system. Before concluding this section, the following remark on the convergence of the proposed algorithm is due.

**Remark** Observe that the algorithm allows two modes of asynchrony, namely skipped updates and bounded delays. Let the worst-case frequency at which the updates in (15) occur be denoted by $f$ and let the delay be such that $\tau(t+1) \leq \tau < \infty$. Then the convergence of the algorithm can be established similarly as in [18], provided that $\rho_g$ is chosen in accordance with $f$ and $\tau$.

IV. Simulation Results

This section provides simulation results for the proposed interference alignment algorithm on a $(6, 1, 3, 3)$ cellular MIMO network shown in Fig. 1. We set the degrees of freedom $d = 1$, so that the interference at every receiver must occupy two or fewer dimensions. Assuming that the inter-cell interference is caused only by users within the neighboring BSSs, the feasibility condition for node $g$ is $M + N \geq \sum_{i \in N_g} (K_i + 1)$. This condition results in one or two redundant antennas at corner BSSs with at most three neighbors each, but is only met with equality at the two nodes in the center.
Algorithm 1 Distributed Asynchronous ADMM with Optional Updates

1: Set $t = 1$, initialize Complex $\{X_{ig}, \Xi_{ig}, Z_i\}$ for all $i \in N_g^g, g = 1, \ldots, G$
2: for $t = 1, 2, \ldots$ do
3:   (Optional) Send $(\rho_g V_{jg}(t) + \Upsilon_{jg}(t))$ to neighbors $i \in N_g^g$ s.t. $j \in U_i$
4:   Send $(\rho_g C_{ig}(t) + \Gamma_{ig}(t))$ to neighbors $i \in N_g^g$
5:   (Optional) Update $\{V_k(t + 1)\}_{k \in V_t, C_g(t + 1), \Delta_{ig}(t)}$ as in (15a), (15b) and (15c) and transmit to $i \in N_g^g$
6:   end if
7:   if $\{V_j(t + 1)\}_{j \in V_t}, C_i(t + 1), \Delta_{jt}(t + 1)$ not received from some $i \in N_g^g$ then
8:      Set $\{V_j(t + 1) = V_j(t), C_i(t + 1) = C_i(t), \Delta_{jt}(t + 1) = \Delta_{jt}(t)\$
9:   end if
10: Update the primal variable $V_{jg}(t + 1), C_{0}(t + 1), \Delta_{g}(t + 1)$ as in (17a), (17b) and (17c)
11: Update the dual variable $\Upsilon_{jg}(t + 1), \Gamma_{0g}(t + 1), \Delta_{g}(t + 1)$ as in (14a), (14b) and (14c)
12: Update $C_{ig}(t + 1) = C_i(t + 1), \Gamma_{ig}(t + 1) = 0, \forall i \in N_g^g$
13: Update $\Delta_{g}(t + 1) = \Delta_{g}(t + 1), \Delta_{g}(t + 1) = 0, \forall i \in N_g^g$ and $l : l \notin U_k$
14: if $\|\tilde{X}_g(t + 1) - \tilde{X}_g(t)\| \leq \delta$ then
15:      terminate loop
16: end if
17: end for

Monte-carlo simulations are carried out over 400 independent realizations, with channel coefficients drawn from complex circularly symmetric Gaussian distribution with unit variance in each dimension. The signal-to-interference ratio (SIR) is utilized as a metric to quantify the interference suppression. Specifically, whenever the SIR exceeds a preset threshold on all streams, the interference is declared aligned. The top plot in Fig. 2 shows the number of instances of network-wide alignment for different SIR thresholds. Perfect alignment is achieved when a high SIR threshold is met at all streams, while a low SIR threshold is of practical importance, where it is sufficient to allow the interference power to stay near the noise floor. It is emphasized that similar to the block-coordinate descent method in [2], the proposed algorithm always achieves perfect alignment eventually.

The bottom plot in Fig. 2 shows the number of interference-free dimensions. Let $\sigma_n(A)$ denote the $n$-th singular value of a matrix $A$. Then, the number of interference-free dimensions is defined as $\eta_1 - \eta_2$, where $\eta_1 := |\{n | \sum_k \sigma_n(U_k^T H_k^T V_k) > -2\}|$ for all $k$ and $\eta_2 := |\{n | \sigma_n(U_k^T R_k g, H_k^T V_n) \in \{u, \ldots, n \neq k\} > -5\}|$. As observed earlier, the proposed algorithm also attains the full number of interference-free dimensions asymptotically, similar to the algorithms proposed in [2]. The computational and communication complexity is controlled by various system parameters, namely $G, |N_g|, K_g, N_g, M_g$, and $d$. In most practical systems, the number of antennas at each user $M_g$, the degrees of freedom $d$, the number of neighboring cells $|N_g|$ are small constants. Therefore, the complexity of the proposed algorithm is only studied with respect to $N := \max_g N_g$ and $K := \max_g K_g$. Further, the feasibility condition necessitates that $N \approx O(K)$. With these assumptions, it can be shown that the proposed algorithm requires $O(K^4)$ computations per iteration, and incurs a communication cost of $O(K^2)$ per iteration. In contrast, the BCD algorithm in [2] is centralized and incurs a computational complexity of $O(K^6)$ per iteration. On the other hand, if $K$ is fixed, the per-iteration computational complexity is $O(N^3)$ for the proposed algorithm and $O(N^6)$ for the BCD algorithm. It is remarked however that the number of iterations required by proposed algorithm is generally several times those required by the BCD algorithm. Indeed, for the $(6, 1, 3, 3)$ system considered here, the overall run-time for both algorithms is roughly the same, despite the proposed algorithm being distributed and asynchronous.

Fig. 1: The $(6, 1, 3, 3)$ cellular network

Fig. 2: Network-wide alignment and free dimensions
REFERENCES


