Adaptive Hybrid Beamforming with Massive Phased Arrays in Macro-Cellular Networks

Shahram Shahsavari, S. Amir Hosseini, Chris Ng, and Elza Erkip

***Abstract*—Hybrid beamforming via large antenna arrays has a great potential for increasing data rate in cellular networks by delivering multiple data streams simultaneously. In this paper, several beamforming design algorithms are proposed based on long-term channel information in macro-cellular environments where the base station is equipped with a massive phased array under per-antenna power constraint. Using an adaptive scheme, beamforming vectors are updated whenever the long-term channel information changes. First, the problem is studied when the base station has a single RF chain (single-beam scenario). Semi-definite relaxation (SDR) with randomization is used to solve the problem. As a second approach, a low-complexity heuristic beam composition algorithm is proposed which performs very close to the upper-bound obtained by SDR. Next, the problem is studied for a generic number of RF chains (multi-beam scenario) where the Gradient Projection method is used to obtain local solutions. Numerical results reveal that using massive antenna arrays with optimized beamforming vectors can lead to five-fold network throughput improvement over systems with conventional antennas.**

I. Introduction

In light of the rapid development of fifth generation cellular networks (5G), multi-user Multiple-Input and Multiple-Output (MIMO) systems with large number of antennas at the base station (known as Massive MIMO systems) have proven to improve the network performance significantly [1]. These systems comprise of an array of many antenna elements. The user data is precoded in the digital domain first and then, each of the digital streams is converted to a Radio Frequency (RF) signal through a circuit referred to as RF chain. Each signal is then transmitted by the antenna element connected to that RF chain. This process is best suited to a rich scattering propagation environment that provides a large number of degrees of freedom. In a macro-cellular environment, however, these conditions often do not hold. A more cost and energy efficient alternative is the use of hybrid Massive MIMO systems in such scenarios [2].

In hybrid Massive MIMO systems, there are fewer RF chains than antenna elements. This helps the overall system to operate at a lower power level and be more cost effective, since each RF chain consists of power hungry and expensive elements such as Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC) which do not follow Moore's law. However, these systems rely on accurate channel estimation and typically are applied to networks with Time Division Duplex (TDD) operation to alleviate the channel estimation overhead [2]. On the other hand, common deployment of LTE in North America is Frequency Division Duplex (FDD) based. In this paper, we focus on a class of hybrid Massive MIMO systems where all antenna elements maintain RF coherency [3]. This means that all antenna elements are closely spaced and have matching phase and magnitude characteristics at the operating frequency [4]. Using this technique, the antenna system can be used as a phased array and macro-cellular transmission is achieved through hybrid beamforming (BF) [5]. This makes it appealing for FDD systems.

In hybrid BF, each RF chain carries a stream of data and is connected to each antenna element through a separate pair of variable gain amplifier and phase shifter. By setting the values of the amplifier and phase shifts (equivalently designing BF vectors), multiple beams are generated, each carrying one data stream over the air. Generating beams using phased arrays generally requires channel information of all users. By keeping the beam pattern constant over an extended period of time, small scale channel variations can be averaged out. Hence, the BF direction corresponds to a dominant multipath component [6] which mainly depends on the user location in macro-cellular environment due to the primarily Line of Sight (LOS) channels. Whenever user location information is updated, the system can adaptively switch to a different beam pattern to constantly provide enhanced service to the users. We refer to this technique as *long-term adaptive BF*.

The radiated power from an antenna array is constrained and power constraints are chosen to limit the non-linear effects of the amplifiers [7]. Generally, two types of power constraints are considered in research problems: *i*) sum power constraint (SPC) in which an upper-bound is considered for the total power consumption of the array, and *ii*) per-antenna power constraint (PAPC) in which an upper-bound is considered for the power consumption of each antenna in the array [8]. Although it is more convenient to consider SPC for research problems [9], it is not applicable to practical implementations, where each antenna element is equipped with a separate power amplifier.

Generating adaptive beams that maximize the overall network throughput plays a significant role in exploiting the benefits of hybrid BF in a cellular system. Any method that is proposed should have a manageable complexity and operate within the power constraints of the array. The goal of this paper is to propose methods for long-term adaptive BF under PAPC to maximize the average network rate using hybrid phased

S. Shahsavari and E. Erkip are with the ECE Department of New York University, Brooklyn, NY (email: {shahram.shahsavari, elza}@nyu.edu).

S. A. Hosseini and C. Ng are with Blue Danube Systems, Warren, NJ, USA. (email: {amir.hosseini, chris.ng}@bluedanube.com).

This work was supported in part by National Science Foundation under Grants No. 1527750 and 1547332.

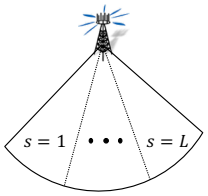


Figure 1: Single cell scenario with L sections.

arrays with arbitrary number of beams. First, we focus the optimization on an individual cell where the interference from other cells is treated as noise. We use well-known theoretical and numerical techniques for finding the optimal beam pattern as well as a theoretical upper bound for the solution. Then, we propose a low complexity heuristic algorithm that performs close to the obtained upper bound.

Notation: $(.)^T$, $(.)^H$, $\mathrm{Tr}\{.\}$, and $||.||_{\mathrm{F}}$ are transpose, hermitian, trace and Frobenius norm operations, respectively. $[N]$ denotes the set of integers from 1 to N.

II. Problem Statement

We consider the downlink of a single-cell scenario consisting of a base station (BS) with M antennas and $L \ll M$ radio frequency (RF) transceiver chains [2]. Since each RF chain can carry a single data stream, the BS can serve L User Equipments (UEs) simultaneously. As a result, the cell site is partitioned into L sections (Fig. 1) and one UE per section is activated at each time slot as will be explained later. We assume that user equipments (UEs) are clustered in hotspots within the cell. Let H_{si}, $(s,i) \in [L] \times [K_s]$ denote hotspot i from section s which consists of a group of N_{si} nearby UEs. We let K_s be the number of hotspots in section s and $K = \sum_s K_s$ denotes the total number of hotspots in the cell. The fraction of UEs located at H_{si} among the UEs in section s is defined by $\alpha_{si} = N_{si}/N_s$, where $N_s = \sum_i N_{si}$ is the total number of UEs in section s. Let U^n_{si}, $n \in [N_{si}]$, denote the n^{th} UE of hotspot H_{si}.

We consider a macro-cellular environment in which the channels are primarily LOS with the possibility of having local scatterers around the UEs. We assume that only the long-term channel state information of UEs is available at the BS and can be used to perform long-term BF. Furthermore, we assume that the long-term channel vectors between the BS and the users belonging to a hotspot are the same due to their proximity. Let $\mathbf{g}_{si} = \sqrt{\beta_{si}}\mathbf{h}_{si}$, denote the long-term channel vector between the BS and the UEs located at H_{si}, where $\beta_{si} \in \mathbb{R}^+$ and $\mathbf{h}_{si} \in \mathbb{C}^M$ denote the pathloss and spatial signature between BS and H_{si}, respectively. We consider the Vandermonde model where $\mathbf{h}_{si} = [e^{j\theta_{si}}, e^{j2\theta_{si}}, \ldots, e^{jM\theta_{si}}]^T$. A use case of this channel model is when the users are located in the far-field of a uniform linear array with M antennas in a primarily line-of-sight environment [9]. In such cases we have $\theta_{si} = 2\pi d \sin(\psi_{si})/\lambda$, where d denotes the spacing between successive elements, λ is the wavelength, and ψ_{si} is direction of H_{si} relative to the BS. In order to model other types of antenna arrays such as rectangular and circular arrays, $\mathbf{h}_{si}$ can be changed accordingly. We note that this model relates the long-term channel information to the location of the hotspots. In [10], the validity of this model is demonstrated using a variety of test-bed experiments.

Each RF chain is connected to each antenna element through a separate pair of variable gain amplifier and phase shifter. We model the corresponding gain and phase shift by a complex coefficient. As a result, there are M complex coefficients corresponding to each RF chain creating a BF vector. The radiation pattern (or equivalently beam pattern) of the antenna array corresponding to each RF chain can be modified by controlling the corresponding BF vector [5]. We assume that the BS uses BF vector $\mathbf{w}_s \in \mathbb{C}^M$, $s \in [L]$ to generate a beam pattern (or a beam in short) for serving UEs located in section s. To reduce the complexity, these BF vectors are designed based on the long-term channel information and are adaptively modified when the long-term channel information changes, i.e., when there is substantial change in the geographical distribution of the hotspots. Moreover, we assume that the UEs are scheduled within each section based on a round robin scheduler. The same approach is valid when random scheduling is utilized within each section. Let $q^n_{si} \in \mathbb{C}$ denote the signal to be sent to U^n_{si}, where $\mathbb{E}(s^n_{si}) = 0$ and $\mathbb{E}(|q^n_{si}|^2) = 1$. Also, let $U^{n^*_{si}}_{si}$ be the scheduled UE in section s at a generic time slot. Hence, the BS transmit vector is $\mathbf{x} = \sqrt{P}\sum_{s\in[L]} q^{n^*_{si}}_{si}\mathbf{w}_s$, where P denotes the average transmit power of the BS. Subsequently, $U^{n^*_{si}}_{si}$ receives signal

$$y^{n^*_{si}}_{si} = \sqrt{P\beta_{si}}q^{n^*_{si}}_{si}\mathbf{h}^H_{si}\mathbf{w}_s + \sum_{s'\neq s}\sqrt{P\beta_{s'i}}q^{n^*_{si}}_{s'i}\mathbf{h}^H_{si}\mathbf{w}_{s'} + v,$$

where $v \sim \mathcal{CN}(0,\sigma^2)$ is the noise. The first term corresponds to the desired signal received from beam s and the second term is the interference received from other $L-1$ beams. Therefore, whenever U^n_{si} is scheduled, the corresponding SINR is

$$\mathrm{SINR}_{si}(\mathbf{W}) = \frac{\mathbf{w}^H_s\mathbf{Q}_{si}\mathbf{w}_s}{1+\sum_{s'\neq s}\mathbf{w}^H_{s'}\mathbf{Q}_{si}\mathbf{w}_{s'}}, \qquad (1)$$

where $\mathbf{Q}_{si} = \gamma_{si}\mathbf{h}_{si}\mathbf{h}^H_{si}$ with $\gamma_{si} = P\beta_{si}/\sigma^2$. The BF matrix is defined as $\mathbf{W} \triangleq [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L] \in \mathbb{C}^{M\times L}$ and has columns corresponding to BF vectors of different sections. We note that PAPC corresponds to $\sum_{s\in[L]}|W_{ms}|^2 \leq 1/M, \forall m$. Hence we define the feasible set as $\mathcal{A} = \{\mathbf{W} \in \mathbb{C}^{M\times L} |\ \forall m : \sum_{s\in[L]}|W_{ms}|^2 \leq 1/M\}$. The goal is to find BF matrix $\mathbf{W}$ which maximizes a network utility function, denoted by $R(\mathbf{W})$ over the feasible set $\mathcal{A}$. In this paper, we consider average network rate as the network utility, i.e.,

$$R(\mathbf{W}) = \sum_{s\in[L]}\sum_{i\in[K_s]}\alpha_{si}\log(1+\mathrm{SINR}_{si}). \qquad (2)$$

Hence, the problem can be formulated as follows

$$\Pi_L:\ \mathbf{W}^{opt} = \underset{\mathbf{W}\in\mathcal{A}}{\mathrm{argmax}}\ R(\mathbf{W}).$$

We note that the sub-index L in Π_L corresponds to the number of the beams (equivalently number of RF chains). Although there is no minimum utility constraint defined for individual UEs in problem Π_L, sum-log maximization induces a type of

proportional fairness. It can be shown that problem Π_L is not in a convex form [11, chapters 3, 4]. Therefore, finding the globally optimal solution of this problem is difficult. In section III, we study the single-beam ($L = 1$) problem Π_1 to find local solutions and an upper-bound to evaluate the performance. In section IV, we provide an iterative algorithm to find a sub-optimal solution of problem Π_L for arbitrary L.

In Sections III and IV, we will need to find the *projection* of a general beamforming matrix $\mathbf{W} \in \mathbb{C}^{M\times L}$ on set $\mathcal{A}$ which is defined as

$$\mathbb{P}_{\mathcal{A}}(\mathbf{W}) = \underset{\mathbf{X}\in\mathcal{A}}{\operatorname{argmin}} \ ||\mathbf{X} - \mathbf{W}||_{\mathrm{F}}^2. \tag{3}$$

We note that $\mathcal{A}$ is a closed convex set which leads to a unique $\mathbb{P}_{\mathcal{A}}(\mathbf{W})$ for every $\mathbf{W} \in \mathbb{C}^{M\times L}$, introduced by Lemma 1. The proof is provided in [12, Appendix A].

Lemma 1. *We have* $\hat{\mathbf{W}} = \mathbb{P}_{\mathcal{A}}(\mathbf{W})$ *if and only if for every* $m \in \{1, 2, \ldots, M\}$

$$\begin{cases} \hat{W}_{ms} = W_{ms} & \text{if } \sum_s |W_{ms}|^2 \leq 1/M, \\ \hat{W}_{ms} = W_{ms}/(\sqrt{M\sum_s |W_{ms}|^2}) & \text{if } \sum_s |W_{ms}|^2 > 1/M. \end{cases}$$

III. Single-Beam Scenario

In this section, we study problem Π_1 where every UE is served by a single-beam generated by BF vector $\mathbf{w}$, i.e., there is only one section in the cell ($s = 1$) and PAPC corresponds to $|w_m|^2 \leq 1/M, \forall m$. In this scenario, there is no interference since one UE is scheduled per time slot. Hence we have $R(\mathbf{w}) = \sum_{i\in[K]} \alpha_i \log(1 + \mathbf{w}^H \mathbf{Q}_i \mathbf{w})$. Please note that we drop index s in the single-beam scenario because $s = 1$. Next, we derive an upper-bound for the optimal value of Π_1 and provide two different methods to obtain local solutions of this problem. We will use the upper-bound as a benchmark in the simulations to evaluate the effectiveness of the local solutions.

A. Semi-definite relaxation with randomization

Since $\mathbf{w}^H \mathbf{Q}_i \mathbf{w}$ is a complex scalar, we have $\mathbf{w}^H \mathbf{Q}_i \mathbf{w} = (\mathbf{w}^H \mathbf{Q}_i \mathbf{w})^T = \gamma_i \mathbf{h}_i^H \mathbf{X} \mathbf{h}_i$, where $\mathbf{X} = \mathbf{w}\mathbf{w}^H \in \mathbb{C}^{M\times M}$ is a rank-one positive semi-definite matrix. Using this transformation, semi-definite relaxation of problem Π_1 is as follows.

$$\begin{aligned} \mathbf{\Pi}_{1r}: \ \mathbf{X}_r^{opt} = & \underset{\mathbf{X}\in\mathbb{C}^{M\times M}}{\operatorname{argmax}} \ \sum_{i=1}^{K} \alpha_i \log(1 + \gamma_i \mathbf{h}_i^H \mathbf{X} \mathbf{h}_i) \\ & \text{subject to: } \ \mathbf{X} \geq 0, \ \forall m : X_{mm} \leq 1/M. \end{aligned}$$

We remark that Π_1 is equivalent to Π_{1r} plus a non-convex constraint $Rank(\mathbf{X}) = 1$. Removing the rank-one constraint enlarges the feasible set and makes it possible to find solutions with higher objective value. Hence, the optimal objective value of Π_{1r} is an upper-bound for the optimal objective value of Π_1. This is a well-known technique called 'Semi-definite Relaxation (SDR)' [13]. Note that Π_{1r} can be solved using convex programming techniques [11]. After solving the convex problem Π_{1r} there are two possibilities:

1) $\mathbf{Rank(X_r^{opt}) = 1}$: in this case the upper-bound is tight and we have $\mathbf{X}_r^{opt} = \mathbf{w}^{opt}\mathbf{w}^{opt\,H}$, where $\mathbf{w}^{opt}$ is the solution of Π_1.
2) $\mathbf{Rank(X_r^{opt}) > 1}$: in this case, the upper-bound is not tight and finding the global solution of Π_1 is difficult. However, there are a number of methods developed to generate a reasonable BF vector $\mathbf{w}$ for problem Π_1 by processing $\mathbf{X}_r^{opt}$ [13]. For example, using eigenvalue decomposition, we have $\mathbf{X}_r^{opt} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$. Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M$ be the eigenvectors in descending order of eigenvalues. One simple approach is to use the eigenvector corresponding to the maximum eigenvalue and form BF vector as $\mathbf{w}_{mev} = \frac{\mathbf{v}_1}{\sqrt{M}||\mathbf{v}_1||_2}$. It should be noted that normalization is necessary for feasibility. Although this simple method is optimal when $Rank(\mathbf{X}_r^{opt}) = 1$, it is not the best strategy when $Rank(\mathbf{X}_r^{opt}) > 1$. Using different '*randomization*' techniques can lead to better solutions [13]. Let us define $\mathbf{w}_{sdr} = \frac{\mathbf{b}}{\sqrt{M}||\mathbf{b}||_2}$, where $\mathbf{b} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{e}$ with random vector $\mathbf{e} \in \mathbb{C}^M$. The elements of $\mathbf{e}$ are *i.i.d.* random variables uniformly distributed on the unit circle in the complex plane. Alternative distributions such as Gaussian distribution can also be adopted for $\mathbf{e}$. The randomization method is to generate a number of BF vectors $\{\mathbf{w}_{sdr}\}$ and pick the one resulting in the highest objective value of Π_1. Note that using $\mathbf{e} = [1, 0, 0, \ldots, 0]^T$ would lead to $\mathbf{w}_{sdr} = \mathbf{w}_{mev}$. The number of random instances denoted by N_{trial} depends on the number of the hotspots, which is discussed more in the numerical examples.

B. Single-beam sub-beam composition

In this section, we introduce a heuristic algorithm to find a BF vector $\mathbf{w}$ for Π_1 with a relatively good performance compared to the upper-bound obtained by SDR. Suppose there is only one hotspot in the network, say H_i. Using Cauchy-Schwarz inequality, we can show that the solution of Π_1 is $\mathbf{w} \triangleq \mathbf{h}_i/\sqrt{M}$. In this technique, which is referred to as *conjugate beamforming*, the BS creates a narrow beam towards the location of H_i [5, chapter 19]. We can generalize this method to generate a beam pattern serving all the hotspots, by summing up the individually optimal BF vectors, and normalizing the result to satisfy PAPC. Hence, the resulting BF vector is $\mathbf{w}_{sbc} \triangleq \mathbb{P}_{\mathcal{A}}(\sum_{i=1}^{K} \mathbf{w}_i)$, where $\mathbb{P}_{\mathcal{A}}(.)$ is given by Lemma 1. We call this method *single-beam sub-beam composition (SB-SBC)* due to the fact that we form a beam pattern by adding up multiple sub-beams.

Adding up individually optimal BF vectors and projecting the result on the feasible set $\mathcal{A}$ will perturb each of them. Therefore, $\mathbf{w}_{sbc}$ would not exactly point towards all the hotspots. To compensate for this disturbance, we use another approach called single-beam phase optimized sub-beam composition (SB-POSBC). In SB-POSBC, we add a separate phase shift for each BF vector $\mathbf{w}_i$ in the summation, i.e., we define $\mathbf{w}_{posbc} \triangleq \mathbb{P}_{\mathcal{A}}(\sum_{i=1}^{K} e^{j\phi_i}\mathbf{w}_i)$. By choosing a set of appropriate phase shifts, $\mathbf{w}_{posbc}$ leads to a beam pattern which points to all the hotspots, hence, it leads to a better network utility. Since it is not easy to find optimal phase shifts analytically, one approach is to try a number of randomly chosen sets of phase shifts and pick the one which leads to the highest objective value in Π_1. One can think of this random

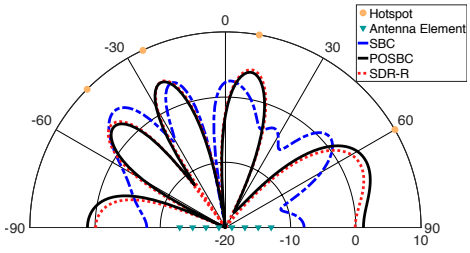


Figure 2: Comparison between the beam patterns (in dB) generated by SB-SBC and SB-POSBC for a uniform linear antenna array with eight antennas (8-ULA) and four hotspots.

trials as the counterpart of randomization technique described in section III-A. Note that if $\forall i : \phi_i = 0$ then $\mathbf{w}_{posbc} = \mathbf{w}_{sbc}$, hence, if the case of zero phase shifts is included in the set of random phase shifts, we can ensure that SB-POSBC will perform at least as good as SB-SBC. One important parameter in SB-POSBC is the number of random trials of phase shift sets denoted by N_{trial} which will be studied in section V-A.

Figure 2 depicts a network with four hotspots. This figure also depicts the beam patterns corresponding to BF vectors $\mathbf{w}_{sbc}$, $\mathbf{w}_{posbc}$, and $\mathbf{w}_{sdr}$ with $N_{trial} = 1000$. We can observe how phase shifts in SB-POSBC compensate for the perturbation caused by SB-SBC. Furthermore, we can also see that SB-POSBC creates a similar beam to SDR with randomization while its complexity is much lower.

IV. Multi-Beam Scenario

In this section, we study problem Π_L for generic L. First we present a heuristic similar to SB-SBC and SB-POSBC, described in Section III-B, and then we introduce an iterative algorithm to find a local solution of problem Π_L.

A. Multi-beam sub-beam composition

Similar to what is described in Section III-B, one can obtain L BF vectors each of which generates a beam to cover a section. To this end, we can consider each section and its associated hotspots and use SB-SBC (or SB-POSBC) to find a BF vector for that section. Furthermore, we assume that the power is equally divided among the BF vectors. Hence, after applying SB-SBC (or SB-POSBC) to find a BF vector for each section separately, we divide all the vectors by $1/\sqrt{L}$. We call this method MB-SBC (or MB-POSBC). We note that this method does not consider inter-beam interference, because each BF vector is obtained independently from the others.

B. Gradient projection

Numerical optimization methods can be used to find a local solution of Π_L for arbitrary L. These methods are more valuable when it is difficult to find a closed-form solution, such as non-convex non-linear optimization. Although there is no guarantee that these methods find the global optimum, they converge to a local optimum of if some conditions hold; we refer the reader to [14] for details. To find a local solution for problem Π_L, we use an iterative numerical method called 'Gradient Projection (GP)'. Although there are different types of GP, we use one that includes two steps at each iteration: *i*) taking a step in the gradient direction of the objective function with a step-size satisfying a condition called Armijo Rule (AR), and *ii*) projecting the new point on the feasible set. Let $\mathbf{W}^{[k]}$ be the BF matrix at iteration k. We define

$$\mathbf{W}^{[k+1]} = \mathbb{P}_{\mathcal{A}}(\mathbf{W}^{[k]} + r^{[k]}\mathbf{G}^{[k]}), \tag{4}$$

where $\mathbf{G}^{[k]} = [\mathbf{g}_1^{[k]}, \mathbf{g}_2^{[k]}, \ldots, \mathbf{g}_L^{[k]}]$ with $\mathbf{g}_s^{[k]} \triangleq \nabla_{\mathbf{w}_s} R(\mathbf{W}^{[k]})$, $r^{[k]} > 0$. $r^{[k]}$ denotes the step-size at iteration k and $\mathbb{P}_{\mathcal{A}}(\mathbf{W})$ is the projection of BF matrix $\mathbf{W}$ on the feasible set $\mathcal{A}$ which is given by Lemma 1. We observe that the projection rule is relatively simple and does not impose high implementation complexity to the problem.

The step-size calculation rule directly affects the convergence of GP. Applying AR to problem Π_L, we have $r^{[k]} = \tilde{r}\beta^{l^{[k]}}$ where $\tilde{r} > 0$ is a fixed scalar and $l^{[k]}$ is the smallest non-negative integer satisfying $R(\mathbf{W}^{[k+1]}) - R(\mathbf{W}^{[k]}) \geq \sigma \operatorname{Re}\{\operatorname{Tr}\{(\mathbf{W}^{[k+1]} - \mathbf{W}^{[k]})^H \mathbf{G}^{[k]}\}\}$ and $\mathbf{W}^{[k+1]}$ is given by (4). In order to find $r^{[k]}$ at iteration k, we start from $l^{[k]} = 0$ and increase $l^{[k]}$ one unit at a time until the above condition is satisfied. $0 < \sigma < 1$ and $0 < \beta < 1$ are AR parameters. In practice, σ is usually chosen close to zero, e.g., $\sigma \in [10^{-5}, 10^{-1}]$. Also, β is usually chosen between 0.1 and 0.5 [14].

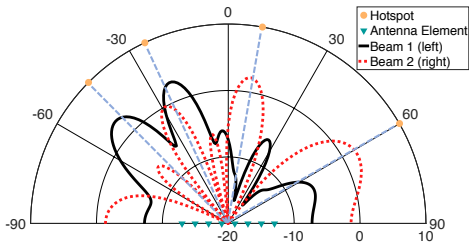
Lemma 2. *Let $\{\mathbf{W}^{[k]}\}$ be a sequence of BF matrices generated by gradient projection in* (4) *with step-size $r^{[k]}$ chosen by the Armijo rule, described above. Then, every limit point of $\{\mathbf{W}^{[k]}\}$ is stationary.*

We refer the reader to [14, chapter 2] for the proof. To implement GP, we need an initial point $\mathbf{W}^{[1]}$ and a termination condition. We use MB-SBC described in Section IV-A to generate an initial point for the numerical examples. For the termination condition, we define the error as $err^{[k+1]} \triangleq ||\mathbf{W}^{[k+1]} - \mathbf{W}^{[k]}||_{\mathrm{F}}$ and we stop after iteration k if $err^{[k+1]} \leq \epsilon$, where ϵ is a predefined error threshold. Although the numerical examples will show that GP converges fast with AR, we specify a threshold for the number of iterations denoted by N_{iter} to avoid slow convergence.

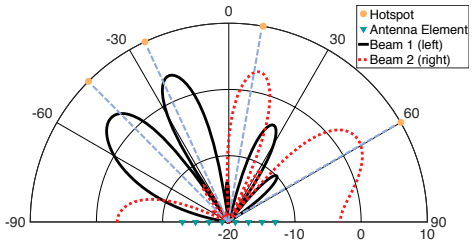
Fig. 3 illustrates a network with two beams where sections 1 and 2 are the left and right half planes, respectively, and each beam serves 2 hotspots. Fig. 3a shows the beams generated by double-beam SBC described in Section IV-A. We observe that the BS suffers from inter-beam interference in this case. GP takes the solution of double-beam SBC as initial point and iteratively updates the BF coefficients of each beam (Fig. 3b), which greatly reduces the inter-beam interference.

V. Numerical Results and Implementation

In this section, we provide extensive simulation and implementation results to evaluate and compare the performance of the proposed methods. We simulate the downlink of a three dimensional network with a BS consisting of a 4×12 uniform rectangular array serving a 120° sector of a cell. Table I lists the network parameters. Hotspots are distributed uniformly at random in a ring around the BS with inner and outer radii of



(a) Double-beam SBC



(b) Gradient projection

Figure 3: Beam patterns generated by double-beam SBC and gradient projection in a sample network with 2 beams, four hotspots, and a uniform linear array with eight antennas.

300 m and 577 m, respectively. We use CVX package [15] to solve the convex problem Π_{1r}. We also use $\epsilon = 10^{-4}$ and $N_{iter} = 10^4$ for GP.

Table I: Simulation parameters

Parameter	Value
Scenario	single-beam, double-beam
Cell radius	577 m
Bandwidth	20 MHz
Noise spectral density	-174 dBm/Hz
BS transmit power (P)	20 dBm
Number of hotspots (K)	$4, 8, 16$
Pathloss in dB (β^{-1})	$128.1 + 37.6 \log_{10}(d \text{ in km})$

A. *Effect of number of trials on the performance*

In this section, we consider the single-beam scenario described in Section III. We focus on SDR with randomization (SDR-R) and SB-POSBC described in Sections III-A and III-B, respectively. In both of these algorithms there are N_{trial} random trials. To evaluate the performance of these algorithms, we consider 100 random network realizations. We run both algorithms with $N_{trial} = 10^0, 10^1, 10^2, 10^3, 10^4$ and find the BF vectors $\mathbf{w}_{sdr}$ and $\mathbf{w}_{posbc}$ and the corresponding network utilities in bps/Hz. We also obtain the Upper-Bound (UB) by solving the relaxed problem Π_{1r}. Table II lists the average performance of these algorithms given two values for number of hotspots, K. We observe that the larger N_{trial} becomes, the closer the performance gets to the UB, which in turn slows down the pace of improvement. For larger K, however, the performance keeps increasing with the number of trials, which suggests that the number of trials should be proportional to the number of hotspots. While both algorithms provide performance close to the UB with large enough N_{trial}, SDR-R outperforms SB-POSBC in some cases. This improved performance comes at the cost of higher computational complexity.

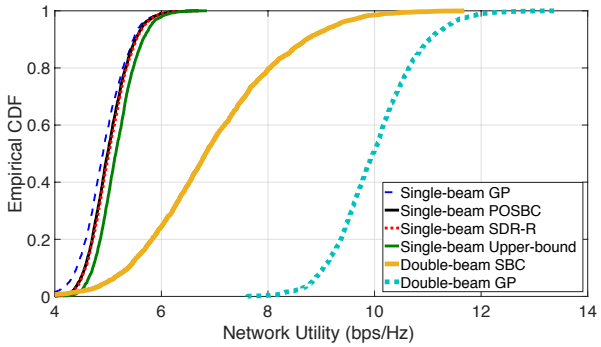


Figure 4: Empirical CDF of network utility.

Table II: Average network utility in (bps/Hz)

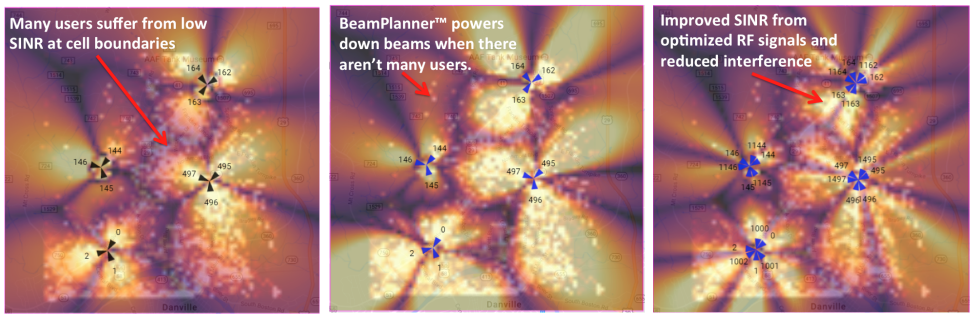
K	Method	N_{trial}				
		10^0	10^1	10^2	10^3	10^4
4	SB – POSBC	5.039	5.505	5.602	5.627	5.637
	SDR – R	5.252	5.584	5.609	5.611	5.611
	UB	5.783	5.783	5.783	5.783	5.783
16	SB – POSBC	3.617	4.220	4.403	4.498	4.556
	SDR – R	4.123	4.521	4.572	4.585	4.587
	UB	4.715	4.715	4.715	4.715	4.715

B. *Computational complexity and performance*

In this part, we compare the performance and numerical complexity of different algorithms devised for the single-beam scenario ($L = 1$). Table III lists the average throughput (in bps/Hz) and average time spent on a typical desktop computer (3.1 GHz Core i5 CPU, 16 GB RAM) to find the BF vector using GP, SB-POSBC, and SDR-R algorithms. The reported values are average values over 100 random network realizations. It is assumed that $N_{trial} = 10^3$ for SDR-R and SB-POSBC. While GP and SB-POSBC have sub-second runtime, SDR-R has a much higher complexity. This is because a complex convex optimization problem has to be solved in the first step of SDR-R, whereas GP and SB-POSBC only require simple mathematical operations. We also observe that the performance of these algorithms are very close. Overall, we can conclude that GP and SB-POSBC are superior to SDR-R since they achieve similar performance with much lower computational complexity. The results also reveal that network utility decreases for each of the algorithms when the number of hotspots increases. This is the cost of having a single beam pattern. In fact, given a fixed antenna array aperture, it is more difficult to provide good BF gain for larger number of hotspots with a single beam.

Table III: Average run time (in seconds) and utility in (bps/Hz)

K	Method	Run Time	Network Utility
4	GP	0.024	5.541
	SB – POSBC	0.081	5.627
	SDR – R	9.268	5.611
16	GP	0.040	4.958
	SB – POSBC	0.293	4.498
	SDR – R	21.026	4.585



(a) Network with passive antennas. Average throughput is 53.0 Mbps/km^2.

(b) Network with optimized single beam phased arrays. Average throughput is 89.8 Mbps/km^2.

(c) Network with optimized double beam phased arrays. Average throughput is 237.0 Mbps/km^2.

Figure 5: Sample cellular network located in Danville, VA optimized using BeamPlanner^{TM} software.

C. Performance evaluation

In this section, we consider single-beam ($L = 1$) and double-beam ($L = 2$) scenarios. In order to compare the performance of the algorithms described in Sections III and IV, we consider 4000 random network realizations and calculate the network utility corresponding to each algorithm for each realization. For the single-beam scenario, the upper-bound of the network utility is obtained for each realization by solving problem Π_{1r}. It is assumed that $N_{trial} = 10^3$ for SDR-R and SB-POSBC. Fig. 4 illustrates the empirical CDF of network utility corresponding to each algorithm for $K = 8$ hotspots. We observe that SDR-R outperforms SB-POSBC and GP in the single-beam scenario. Moreover, SB-POSBC performs very close to SDR-R. Having two beams will double the number of transmissions compared to the single-beam scenario which can potentially lead to significant network utility improvement if the interference due to multi-user activity (i.e. inter-beam interference) is managed appropriately. Since double-beam GP considers interference, it leads to almost $2\times$ improvement in network utility compared to the single-beam algorithms. On the other hand, the performance of double-beam SBC is remarkably inferior to double-beam GP, due to lack of interference management.

D. Implementation

Based on the multi-cell generalization of the proposed algorithms, a commercial software has been developed by *Blue Danube Systems* called *BeamPlanner^{TM}*. The software is designed to optimize beam patterns in macro-cellular networks to enable effective antenna deployment. Fig. 5 illustrates the map of a sample cellular network in Danville, VA in three different deployment scenarios. Fig. 5a represents the case where all cells are equipped with conventional passive antennas, whereas the other two figures showcase the deployment of *BeamCraft^{TM} 500*, an active antenna array designed and manufactured by *Blue Danube Systems*. The white dots show the distribution of demand inside the network and the illuminated patterns illustrate the SINR at each point. Fig. 5b shows the single-beam scenario where the beams are optimized using the GP algorithm. Fig. 5c illustrates the same result for the double-beam scenario. It can be seen that double-beam active antenna arrays with optimal beam patterns can offer close to $5\times$ throughput improvement over current systems with conventional antennas.

VI. Conclusion

We have studied a hybrid BF problem in a single macro-cell scenario where the BS is equipped with a massive phased array. We have proposed several algorithms with different complexities to design BF vectors when long-term channel information is available. Extensive simulation and implementation results have revealed the effectiveness of the proposed algorithms in macro-cellular networks.

References

[1] T. L. Marzetta *et al.*, *Fundamentals of Massive MIMO*. Cambridge University Press, 2016.

[2] A. F. Molisch *et al.*, "Hybrid beamforming for massive MIMO: A survey," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 134–141, 2017.

[3] M. Banu, "HDAAS: An efficient massive MIMO technology," 4th Brooklyn 5G Summit, April 2017.

[4] Y. T. Lo and S. Lee, *Antenna Handbook: theory, applications, and design*. Springer Science & Business Media, 2013.

[5] S. J. Orfanidis, *Electromagnetic waves and antennas*. Rutgers University New Brunswick, NJ, 2002.

[6] Q. Li *et al.*, "MIMO techniques in WiMAX and LTE: a feature overview," *IEEE Communications Magazine*, vol. 48, no. 5, 2010.

[7] J. Joung *et al.*, "A survey on power-amplifier-centric techniques for spectrum-and energy-efficient wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 315–333, 2015.

[8] W. Yu and T. Lan, "Transmitter optimization for the multi-antenna downlink with per-antenna power constraints," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2646–2660, 2007.

[9] E. Karipidis, N. D. Sidiropoulos, and Z. Q. Luo, "Far-field multicast beamforming for uniform linear antenna arrays," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4916–4927, Oct 2007.

[10] C. Zhang and R. C. Qiu, "Massive MIMO testbed - implementation and initial results in system model validation," vol. abs/1501.00035, 2015.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] S. Shahsavari *et al.*, "Adaptive hybrid beamforming with massive phased arrays in macro-cellular networks," *arXiv*, vol. abs/1801.09029, 2018.

[13] Z.-Q. Luo *et al.*, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[14] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 2016.

[15] M. Grant, S. Boyd, and Y. Ye, "CVX: MATLAB software for disciplined convex programming," 2008.