High Performance Adaptive Algorithms for Single-Group Multicast Beamforming

Balasubramanian Gopalakrishnan and Nicholas D. Sidiropoulos, *Fellow, IEEE*

***Abstract*—The single-group multicast beamforming problem is NP-hard, and the available approximations do not always achieve favorable performance-complexity tradeoffs. This paper introduces a new class of adaptive multicast beamforming algorithms that features guaranteed convergence and state-of-the-art performance at low complexity. Each update takes a step in the direction of an inverse signal-to-noise ratio (SNR) weighted linear combination of the SNR-gradient vectors of all the users. Convergence of this update to a Karush–Kuhn–Tucker (KKT) point of proportionally fair beamforming is established. Simulations show that the proposed approach can enable better performance than the prior state-of-art in terms of multicast rate, at considerably lower complexity. This reveals an interesting link between max-min-fair and proportionally fair multicast beamforming formulations. For cases where there is no initial channel state information at the transmitter, an online algorithm is developed that simultaneously learns the user channel correlation matrices and adapts the beamforming vector to maximize the minimum (long-term average) SNR among the users, using only periodic binary SNR feedback from each receiver. The online algorithm uses the analytic center cutting plane method to quickly learn the user correlation matrices with limited signaling overhead.**

***Index Terms*—Multicasting, beamforming, max-min-fair, proportionally fair, adaptive, online, learning.**

I. Introduction

MULTICAST beamforming is a part of the Evolved Multimedia Broadcast Multicast Service (eMBMS) in the Long-Term Evolution (LTE) standard for efficient audio and video streaming. Multicast beamforming utilizes multiple transmit antennas and channel state information at the transmitter (CSIT) to steer transmitted power towards a group of subscribers while limiting interference to other users and systems [1]. In single group multicasting, where all users are interested in the same information stream from the transmitter (Tx), the maximum common data rate is determined by the minimum received signal to noise ratio (SNR). Hence the objective is to maximize the minimum received SNR subject to transmit power constraints (max-min-fair multicast beamforming). An alternative is to minimize the transmit power subject to appropriate quality-of-service (QoS) guarantees formulated in terms of the minimum SNR for each user (QoS multicast beamforming). The two formulations are essentially equivalent from an optimization point of view [1].

All work to date on multicast beamforming has assumed that some grade of CSIT (instantaneous or statistical, perfect or inexact) is available. In practice CSI has to be acquired, and that can be a serious burden—especially when the number of users and/or antennas is large. CSIT can be acquired before beamforming optimization, but in reality channels change over time and users may drop in or out of the multicast, so it is appealing to consider joint online CSIT acquisition and beamformer adaptation. This is a challenging problem, especially when channel reciprocity cannot be assumed, and the receiver (Rx) equipment is limited in terms of computation and communication capabilities.

A. Related Work

Sidiropoulos *et al.* [1] considered max-min-fair and QoS multicast beamforming for a multi-antenna Tx serving multiple users, each with a single antenna Rx. It was shown in [1] that these two formulations are essentially equivalent NP-hard optimization problems, which can be expressed as a non-convex quadratically constrained quadratic program (QCQP). Semi-definite relaxation (SDR) followed by Gaussian randomization (SDR-G) was proposed in [1] to obtain an upper bound on the attainable minimum SNR and a good sub-optimal solution, respectively.

For a large number of antennas or users, the quality of the approximation obtained using SDR-G deteriorates considerably. This prompted a search for better approximations of the multicast beamforming problem over the past decade. The best approach so far is the successive linear approximation (SLA) algorithm proposed by Tran *et al.* for the QoS version of the problem [2]. The SLA algorithm starts with a feasible vector, say $\mathbf{w}_0$. The non-convex constraints are linearized about $\mathbf{w}_0$ using first-order Taylor series expansion, and the resulting convex problem is solved to obtain the next iterate $\mathbf{w}_1$, which is subsequently used for linearization in the next iteration. It was shown in [2] that the SLA algorithm converges to a KKT point of the QoS problem formulation, and performs better than SDR-G in simulations. SLA has lower worst-case complexity per iteration than SDR-G ($\mathcal{O}(N+K)^{3.5}$ for one SLA iteration vs. $\mathcal{O}(N^2+K)^{3.5}$ overall for SDR-G, where N is the number of antennas at the Tx and K is the number of users). However, the overall complexity of SLA can be greater than that of SDR-G, because of the outer linearization iterations required for convergence of the SLA algorithm.

A different approximation of the max-min-fair formulation was recently proposed by Demir *et al.* [3]. In [3], the non-convex

Manuscript received March 09, 2015; revised May 26, 2015; accepted May 28, 2015. Date of publication June 03, 2015; date of current version July 09, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Adel Belouchrani. Supported by NSF EARS/AST-1247885. A conference version of part of this work was presented at IEEE ICASSP, Brisbane, Australia, April 2015 [14].

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: gopa0022@umn.edu; nikos@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2441044

part of the problem is isolated to a rank-one constraint, which is replaced with an equivalent non-convex bilinear trace constraint. The interesting aspect of Demir's approach is that the resulting reformulation is naturally amenable to alternating maximization (AM). Simulations in [3] showed that the AM attained a higher minimum SNR compared to SDR-G. However, the computational complexity of the AM algorithm is higher than SDR-G, since AM involves solving an SDP (of the same size as SDR-G) at each AM step. Furthermore, our simulations indicate that SLA outperforms AM, and SLA has lower complexity than AM.

SDR-G, SLA and AM involve solving one or more convex optimization problems to obtain a good transmit beamforming vector that attains a high minimum SNR. For large N and K, the computational cost of solving these convex optimization problems becomes prohibitive, and low-complexity algorithms are needed. The first low-complexity adaptive algorithm for (max-min-fair) multicast beamforming was proposed by Lozano [4]. In every iteration of Lozano's algorithm, the new iterate is obtained by updating the previous iterate with a fixed step along the SNR gradient direction of the user with the least SNR in the previous iteration. This is followed by a scaling step to satisfy the transmit power constraint. Simulations showed that Lozano's algorithm can achieve a higher minimum SNR than the SDR-G approach when $K \gg N$. The computational complexity of Lozano's algorithm is $\mathcal{O}(KN)$ for instantaneous rank-one CSIT and $\mathcal{O}(KN^2)$ for long-term higher-rank CSIT, which is much lower than SDR-G and SLA. Matskani *et al.* [5] observed that Lozano's algorithm can exhibit limit cycle behavior, and proposed a variation called (damped) LLI (Lozano with Lopez Initialization). This employs a diminishing step size and a more sophisticated initialization using the weight vector that maximizes average SNR [6]. Simulations showed that the LLI algorithm obtains a higher minimum SNR than Lozano's algorithm at the same complexity.

Abdelkader *et al.* [7] proposed a low-complexity algorithm based on channel orthogonalization using QR decomposition, to approximate the QoS problem when $K \geq N$. For every run of the QR algorithm, a set of N out of K channel vectors are chosen randomly and stacked into a $N \times N$ matrix $\mathbf{H}$, and the QR decomposition $\mathbf{H} = \mathbf{QR}$ is obtained. The beamforming vector is modelled as a weighted linear combination of the columns of $\mathbf{Q}$ and the corresponding weights are obtained in closed form [7], followed by a scaling step to satisfy the QoS constraints. The final beamforming vector is the best obtained after a number of random draws as above. Simulations showed that when $K \gg N$, the QR algorithm performs better than the SDR-G approach, at $\mathcal{O}(N^2)$ complexity—which is much lower than SDR-G.

Multicast beamforming in the case of only $K = 2$ users was specially considered in [8], which derived the optimal solution for this case (note that the NP-hardness proof in [1] does not apply when $K < N$). Motivated by the optimal solution for $K = 2$, [8] proposed an orthogonalization-based successive beamforming (SB) algorithm for general K. As we will see in the simulations section, the performance of SB quickly becomes inferior to SDR-G as K increases, so it is not competitive.

When only imperfect CSIT is available, a robust multicast beamforming formulation (which further includes interference constraints) has been considered in [9]. The problem is formulated as a non-convex QCQP and two randomization algorithms are proposed to obtain sub-optimal solutions. Furthermore, a specific case of the problem was identified for which the optimal solution can be obtained in polynomial time via SDR [9].

QR, Lozano, and LLI feature low complexity, but also a relatively large gap to the SDR performance bound. Perhaps more importantly (since the SDR bound is generally not attainable), our simulations show that the minimum SNR attained by these algorithms is still significantly lower than that of SLA. Another drawback is that QR, Lozano, and LLI require tuning of parameters through trial and error.

Summarizing, no algorithm offers state-of-the-art performance (SLA) at low-enough complexity (QR/Lozano/LLI). One of our original goals was to fill this gap; as we will see, our new algorithms come close to SLA in terms of performance, at QR/Lozano/LLI complexity. Even better, the proposed algorithms can be used to warm-start a single iteration of SLA, and this turns out to outperform (iterative) SLA, as we will see.

Our second goal was to come up with a multicast beamforming algorithm that gradually learns the required CSI as it adapts the beamformer weights. Online algorithms for designing transmit beamforming weights for unicast transmission without initial CSIT have been developed in [10]–[12], using binary feedback from the Rx. Reference [10] proposed a variation of the Cyclic Jacobi subspace estimation algorithm to learn the instantaneous channel/channel correlation matrix $(\mathbf{H}/\mathbf{R})$ of a Multiple-Input Multiple-Output (MIMO) link. The 1-bit feedback was assumed to be based on a monotonic function of the instantaneous/average received signal power. It was shown that this algorithm asymptotically converges to the eigen-decomposition of $\mathbf{H}/\mathbf{R}$.

An adaptive thresholding algorithm was proposed in [11] to simultaneously transmit data and learn the optimal long-term beamforming vector (i.e., the principal eigenvector of $\mathbf{R}$), using binary feedback from the Rx of a Multiple-Input Single-Output (MISO) unicast channel link. For every new transmit beamforming vector, a '1' or a '0' is fed back by the Rx, based on whether the measured average SNR is $\geq$ or $<$ a pre-determined threshold. From every feedback bit, a new linear inequality involving $\mathbf{R}$ is inferred at the Tx, and $\hat{\mathbf{R}}$ is updated as the analytic center of the region formed by the positive semi-definite (p.s.d.) cone and all the linear inequalities inferred until that point. The new beamforming vector is designed to create a balance between gathering new information about $\mathbf{R}$ and attaining a high average SNR using the knowledge acquired from all the feedback bits; while the new threshold is designed in order to reduce the existing uncertainty regarding $\mathbf{R}$. Asymptotic convergence to the maximum SNR attained with perfect CSIT was established in [11].

A similar algorithm using essentially the same analytic center cutting plane method (ACCPM) as [11] was independently and simultaneously proposed in [12] for maximizing the instantaneous energy harvested at the Rx of a MIMO link. In [12], the 1-bit Rx feedback at time slot t is based on whether the energy harvested at time t is $\geq$ or $<$ that at time $t - 1$. Simulations in [11] and [12] showed the faster convergence rate of the respective algorithms to the optimal value (obtained with perfect CSIT) in comparison with [10]. The algorithms in [10]–[12] can be used to learn the user channels in a multicast setup, by considering every user of the multicast individually.

B. Contributions

In this paper we consider a single-group multicast cell with N antennas at the Tx serving K single antenna users. We consider two scenarios: a) the Tx has perfect CSI for all K users, and b) the Tx has no initial CSI for any user. When perfect CSIT is available, we propose a new class of adaptive multicast beamforming algorithms comprising *Additive Update* (AU), *Multiplicative Update* (MU), and *Multiplicative Update—Successive Linear Approximation* (MU-SLA) algorithms, with guaranteed convergence and state-of-the-art performance at low complexity. In every iteration of the AU algorithm, the beamforming vector is updated by taking a step along the inverse-SNR weighted SNR-gradient direction of all the users, as computed using the previous iterate. This is followed by a scaling step to satisfy a transmit power constraint, and the whole procedure is repeated until the iterates converge. The fixed point equation of this algorithm is analyzed for a simple but insightful example, and convergence is established by interpreting the AU as successive concave approximation of (or projected gradient update for) proportionally fair beamforming. The MU algorithm, which is a limiting case of the AU algorithm, attains the same minimum SNR as AU, but has faster convergence and it also eliminates the need for step-size selection. We currently have proof of convergence only for the AU—the analysis does not carry over verbatim to the MU for technical reasons. The MU-SLA algorithm uses the solution provided by the MU algorithm as an initialization for a *single SLA iteration*. Simulations show that MU-SLA outperforms SLA, while the AU and MU operate close to SLA and outperform all the other algorithms, at an order of magnitude lower complexity. The performance-complexity tradeoff is analyzed for the proposed algorithms and the previous state-of-art using relevant simulations.

In the absence of initial CSIT, if the receivers do not have sufficient computational and energy resources to estimate, quantize and feed back accurate CSI to the Tx, we propose an online *cognitive multiplicative update* (CMU) algorithm for designing long-term beamforming vectors using binary channel quality user feedback. In the CMU algorithm, every user only feeds back a '1' or a '0' in each time slot, depending on whether its average received SNR is $\geq$ or $<$ a pre-determined threshold. Using the feedback bits from every user, the Tx learns new linear inequalities about the channel matrices of the users and updates its estimate using the ACCPM. The new beamforming vector is designed to gather useful information about the channel and also use the accumulated knowledge to attain a high minimum SNR among the users. Two threshold selection techniques at the Tx, namely i) *multiple threshold selection* and ii) *common threshold selection*, are proposed for effectively reducing the uncertainty in the channel correlation matrices of the users in each slot. It is shown that the former reduces the uncertainty faster and converges to the true channel correlation matrices at a faster rate than the latter, at the cost of higher communication overhead. A simple modification is also proposed to completely eliminate the communication overhead in ii) by varying the transmit power. Simulations show that the CMU algorithm using the aforementioned threshold selection methods converges to the performance achieved with perfect CSIT.

Relative to the conference version [14], this paper includes fixed point analysis for the AU and the MU algorithm, and a detailed convergence proof of the AU algorithm. Comprehensive simulation results are provided to assess the performance of the proposed algorithms in comparison to the prior state-of-art. The other novel contribution of this journal version relative to [14] is the CMU algorithm, which starts from no CSI and gradually acquires it on the fly, as it also adapts the beampattern, through a judicious combination of MU, ACCPM, and SNR threshold selection strategies.

II. Problem Formulation

We consider a single-group multicast cell consisting of a Tx with N antennas and K single antenna receivers. The Tx transmits the common data x which has zero-mean and unit-variance, to all the K receivers using a unit-norm beamforming vector $\mathbf{w}$. The corresponding received signal at the kth Rx is given by

$$y_k = \mathbf{w}^H \mathbf{h}_k x + z_k, \quad \forall k \in \{1, 2, \ldots, K\} \tag{1}$$

where $\mathbf{h}_k$ is the channel between the Tx and the kth Rx which is modelled as a $N \times 1$ zero-mean complex random vector. z_k is wide-sense stationary additive noise at the kth Rx, assumed independent of x and $\mathbf{h}_k$, with zero-mean and variance σ_k^2. The received SNR at the kth Rx is given by $\frac{|\mathbf{w}^H \mathbf{h}_k|^2}{\sigma_k^2} = \frac{\mathbf{w}^H \mathbf{R}_k \mathbf{w}}{\sigma_k^2}$, where $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H \succeq \mathbf{0}, \forall k \in \{1, 2, \ldots, K\}$. We can absorb σ_k into $\mathbf{h}_k$, and thereafter work with the scaled channels $\tilde{\mathbf{h}}_k = \mathbf{h}_k / \sigma_k$ (assuming that the kth Rx can estimate σ_k beforehand and inform the Tx, or scale $\mathbf{h}_k$ before sending it to the Tx). We will assume that this has already been done, and drop the $\tilde{}$ for brevity. The objective of the Tx is to design unit norm transmit beamforming vectors that maximize the minimum SNR among the users. This can be formulated as follows.

$$\mathbf{\Pi_1} \qquad \arg \max_{\mathbf{w} \in \mathbb{C}^N} \min_{k \in \{1,2,\ldots,K\}} \mathbf{w}^H \mathbf{R}_k \mathbf{w} \qquad s.t. \quad \|\mathbf{w}\|^2 = 1$$

where $\forall k \in \{1, 2, \ldots, K\}$. $\mathbf{\Pi_1}$ is NP-hard, as shown in [1].

III. Tx has Perfect CSI About $\{\mathbf{R}_k\}_{k=1}^K$

When the Tx has perfect CSI, the SLA algorithm [2] is the state-of-the-art in terms of attaining the highest possible minimum SNR/multicast rate. However, it has a relatively high worst-case complexity. The first low-complexity adaptive algorithm for multicast beamforming was Lozano's [4]. Lozano's algorithm focuses only on the weakest user in each iteration and ignores all other users. In certain cases [5], this strategy results in fluctuations in the minimum SNR due to limit cycles, as improving the SNR of one user may reduce the SNR of another and vice-versa. When there are multiple users experiencing low SNR, it makes intuitive sense that we should take all the user-channels into account while taking the next step. Furthermore, users experiencing different SNR 'grades' should be appropriately weighted in the computation of the new direction. This intuition naturally suggests the following *Additive*

Update (AU) algorithm, which we first introduce below in the context of a simplified, two-user scenario.

Example: Consider a scenario with $K = 2$ users. The initial unit norm beamforming vector is chosen randomly and is denoted by $\mathbf{w}_1$. At every iteration $n \geq 1$, the new beamforming vector iterate is obtained as follows.

$$\begin{aligned}\tilde{\mathbf{w}}_{n+1} &= \mathbf{w}_n + \alpha\left[\frac{\mathbf{h}_1\mathbf{h}_1^H\mathbf{w}_n}{|\mathbf{h}_1^H\mathbf{w}_n|^2} + \frac{\mathbf{h}_2\mathbf{h}_2^H\mathbf{w}_n}{|\mathbf{h}_2^H\mathbf{w}_n|^2}\right] \\ &= \mathbf{w}_n + \alpha\left[\frac{\mathbf{h}_1}{\mathbf{w}_n^H\mathbf{h}_1} + \frac{\mathbf{h}_2}{\mathbf{w}_n^H\mathbf{h}_2}\right] \quad (2)\end{aligned}$$

$$\mathbf{w}_{n+1} = \frac{\tilde{\mathbf{w}}_{n+1}}{||\tilde{\mathbf{w}}_{n+1}||} \quad (3)$$

where α is the fixed positive step-size for every iteration. From (2), $\tilde{\mathbf{w}}_{n+1}$ is a sum of $\mathbf{w}_n$ and a fixed step times a direction vector, which is an inverse SNR weighted sum of the SNR gradients, evaluated at $\mathbf{w}_n$. Therefore, in the $(n+1)$th iteration, $\mathbf{w}_{n+1}$ is updated along a direction that favors the user with the least SNR in the nth iteration, but also takes into account, the other user. More generally, for $K > 2$, using an inverse SNR weighted sum of the SNR gradients of all users will favor those experiencing lower SNRs in the nth iteration. This should be contrasted with [4], [5], which only focus on the weakest user. From (2), it is easy to see that the fixed point equation[1] is given by (4).

$$\mathbf{w}_{fp} = \frac{1}{c}\left[\frac{\mathbf{h}_1}{\mathbf{w}_{fp}^H\mathbf{h}_1} + \frac{\mathbf{h}_2}{\mathbf{w}_{fp}^H\mathbf{h}_2}\right] \quad (4)$$

where $c \in \mathbb{R}^1$ is a constant introduced to scale the magnitude of $\mathbf{w}_{fp}$ to unity.

Proposition 1: For $K = 2$, the multicast rate attained by a fixed point of the proposed AU algorithm is $r_{\min} = \log_2(1 + \mathrm{SNR}_{\min,fp})$, where

$$\mathrm{SNR}_{\min,fp} = \frac{1}{2}\left[\min(||\mathbf{h}_1||^2, ||\mathbf{h}_2||^2) + \min\left(\frac{||\mathbf{h}_1||}{||\mathbf{h}_2||}, \frac{||\mathbf{h}_2||}{||\mathbf{h}_1||}\right)|\mathbf{h}_1^H\mathbf{h}_2|\right] \quad (5)$$

Proof: See Appendix A. ∎

It is instructive to compare this result to the max-min SNR in two special cases. The best situation for multicast beamforming is when $\mathbf{h}_2 = s\mathbf{h}_1$, for some $s \in \mathbb{C}$, i.e., the two user channel vectors are collinear. Then $\mathrm{SNR}_{\min,fp} = \min(||\mathbf{h}_1||^2, ||\mathbf{h}_2||^2)$, which is equal to the optimum max-min SNR. The worst situation is when the two channel vectors are orthogonal, $\mathbf{h}_1^H\mathbf{h}_2 = 0$, in which case $\mathrm{SNR}_{\min,fp} = \frac{1}{2}\min(||\mathbf{h}_1||^2, ||\mathbf{h}_2||^2)$, while the optimum max-min SNR can be easily shown to be $\frac{||\mathbf{h}_1||^2||\mathbf{h}_2||^2}{||\mathbf{h}_1||^2+||\mathbf{h}_2||^2}$. Without loss of generality, assume $||\mathbf{h}_1||^2 \leq ||\mathbf{h}_2||^2$. Then $\mathrm{SNR}_{\min,fp} = \frac{1}{2}||\mathbf{h}_1||^2$, while the optimum max-min SNR is $\frac{||\mathbf{h}_1||^2}{\frac{||\mathbf{h}_1||^2}{||\mathbf{h}_2||^2}+1}$ [8], and satisfies $\frac{||\mathbf{h}_1||^2}{2} \leq \frac{||\mathbf{h}_1||^2}{\frac{||\mathbf{h}_1||^2}{||\mathbf{h}_2||^2}+1} \leq ||\mathbf{h}_1||^2$. We see that the AU fixed point is optimum in the case of balanced channel norms, and no worse than 3 dB off the optimum even in the worst (near-far) case. While this is clearly a toy example (e.g., the NP-hardness proof in [1] does not apply when $K = 2$), it is still satisfying to see that the simple AU iteration is so close to optimum in these two extreme cases. Motivated by these preliminary observations, we next consider the AU algorithm for general K.

[1]A fixed point of a mapping $\mathbf{f}(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$ is any $\mathbf{x} \in \mathbb{R}^N$ satisfying $\mathbf{f}(\mathbf{x}) = \mathbf{x}$.

A. Additive Update Algorithm

In this section, we consider the case when there are $K \geq 2$ users and the matrices $\{\mathbf{R}_k\}_{k=1}^K \succeq \mathbf{0}$ have rank ≥ 1. An example of higher-rank scenario is when the objective is to maximize the minimum average SNR (instead of instantaneous SNR) among the users. In this case, $\{\mathbf{R}_k\}_{k=1}^K$ are the channel correlation matrices, which are full rank with probability one if the channel vectors are drawn from a continuous distribution. The motivation to consider average SNR is that instantaneous channel estimation and feedback requires much higher computation and communication overhead relative to infrequent channel correlation feedback.

For general K, the AU weight vector update is

$$\begin{aligned}\tilde{\mathbf{w}}_{n+1} &= \mathbf{w}_n + \alpha\left(\sum_{k=1}^K \frac{\mathbf{R}_k}{\mathbf{w}_n^H\mathbf{R}_k\mathbf{w}_n + \varepsilon}\right)\mathbf{w}_n \\ \mathbf{w}_{n+1} &= \frac{\tilde{\mathbf{w}}_{n+1}}{||\tilde{\mathbf{w}}_{n+1}||} \quad (6)\end{aligned}$$

where α is a positive constant step size, and ε is a positive constant that is introduced for numerical stability. The AU update takes all the user channels into consideration, favoring weaker users more than stronger ones (i.e., those with lower SNR over those with higher SNR attained by $\mathbf{w}_n$ in the previous iteration). The fixed point equation of the AU algorithm is:

$$\mathbf{w}_{fp} = \frac{1}{c}\left(\sum_{k=1}^K \frac{\mathbf{R}_k}{\mathbf{w}_{fp}^H\mathbf{R}_k\mathbf{w}_{fp} + \varepsilon}\right)\mathbf{w}_{fp} \quad (7)$$

Whereas the AU update has been intuitively developed and motivated up to this point, the following proposition reveals that it can be viewed as an approximation of a problem that is related to (but different from) the max-min-fair formulation $\mathbf{\Pi_1}$.

Proposition 2: The beamforming vector obtained at the $(n+1)$th iteration of the AU algorithm is the solution of a strongly concave approximation (cf. (8) and (9)) of the proportionally fair [13] multicast beamforming problem $\mathbf{\Pi_2}$ at $\mathbf{w} = \mathbf{w}_n$.

$$\mathbf{\Pi_2} \quad \mathbf{w}^* = \arg\max_{||\mathbf{w}||^2=1} \frac{1}{2}\sum_{k=1}^K \log(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon)$$

Proof: It can be shown that $\mathbf{\Pi_2}$ is a non-convex optimization problem [13] which is difficult to solve in general. Consider a strongly concave approximation of $f(\mathbf{w})$ at the point $\mathbf{w} = \mathbf{w}_n$.

$$f(\mathbf{w}) \approx f(\mathbf{w}_n) + \overbrace{\left(\sum_{k=1}^K \frac{(\mathbf{R}_k\mathbf{w}_n)^H(\mathbf{w}-\mathbf{w}_n)}{\mathbf{w}_n^H\mathbf{R}_k\mathbf{w}_n + \varepsilon}\right)}^{\nabla f(\mathbf{w}_n)^H(\mathbf{w}-\mathbf{w}_n)} - \frac{||\mathbf{w}-\mathbf{w}_n||^2}{2\alpha} \quad (8)$$

Denote the right hand side of (8) as $u(\mathbf{w}, \mathbf{w}_n)$. The sum of the first two terms in $u(\mathbf{w}, \mathbf{w}_n)$ is the first order Taylor series approximation of $f(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}_n$. The last term in $u(\mathbf{w}, \mathbf{w}_n)$ is a proximal regularizer which is included to make $u(\mathbf{w}, \mathbf{w}_n)$

strongly concave ($\alpha > 0$). Instead of solving $\mathbf{\Pi_2}$, suppose that we iteratively solve $\mathbf{\Pi_{2r}}$ to obtain $\mathbf{w}_{n+1}$ from $\mathbf{w}_n$.

$$\mathbf{\Pi_{2r}} \quad \mathbf{w}_{n+1} = \arg \max_{\|\mathbf{w}\|^2=1} u(\mathbf{w}, \mathbf{w}_n)$$

From the definition of $u(\mathbf{w}, \mathbf{w}_n)$, it can be seen that the solution of $\mathbf{\Pi_{2r}}$ can be obtained in closed form as shown below.

$$\mathbf{w}_{n+1} = \frac{\mathbf{w}_n + \alpha\left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon}\right)\mathbf{w}_n}{\left\|\mathbf{w}_n + \alpha\left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon}\right)\mathbf{w}_n\right\|} \tag{9}$$

It can be seen from (6) and (9) that the $(n+1)$th iterate of the AU algorithm is the solution of $\mathbf{\Pi_{2r}}$. Hence the AU algorithm obtains a beamforming vector that promotes proportional fairness in terms of SNR among the users. ∎

The natural next question is whether the AU algorithm converges. The following result shows that it does.

Theorem 1: The iterates obtained from the AU algorithm converge to a KKT point of $\mathbf{\Pi_2}$, provided $0 < \alpha \leq \frac{2}{L_{\nabla_f}}$, where $L_{\nabla_f} = \sum_{k=1}^{K}(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2})$ and $\lambda_{\max}(\mathbf{R}_k)$ is the maximum eigenvalue of $\mathbf{R}_k$ (when $\mathbf{R}_k = \mathbf{h}_k\mathbf{h}_k^H$, $\lambda_{\max}^2(\mathbf{R}_k) = \|\mathbf{h}_k\|^2$).

Proof: See Appendix B. ∎

B. Multiplicative Update Algorithm

The proof of Theorem 1 in Appendix B requires the technical condition $0 < \alpha \leq \frac{2}{L_{\nabla_f}}$, but our experiments indicate that AU converges even for $\alpha > \frac{2}{L_{\nabla_f}}$. This motivates the following limiting version of the AU algorithm, which we will call the *Multiplicative Update* (MU) algorithm. The update step in the $(n+1)$th iteration is given by:

$$\tilde{\mathbf{w}}_{n+1} = \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon}\right),$$
$$\mathbf{w}_{n+1} = \frac{\tilde{\mathbf{w}}_{n+1}}{\|\tilde{\mathbf{w}}_{n+1}\|} \tag{10}$$

The new iterate is the unit vector along the inverse SNR weighted SNR gradient direction of all the K users (i.e., only the direction vector of AU). It can be seen from (6) that the AU update approaches the MU update as α increases. From (7) and (10) it can also be seen that the MU algorithm has the same fixed point condition as the AU algorithm. Simulations indicate that the MU algorithm always converges faster than and to the same fixed point as AU, without requiring any parameter tuning. The technical difficulty of using Theorem 1 for proving convergence of the MU algorithm at this point is that the proof in Theorem 1 places an upper bound on the step-size value of the gradient update, for the iterates to converge.

To gain more insight about the MU algorithm, consider the proportionally fair multicast beamforming problem $\mathbf{\Pi_2}$. Since the objective is non-concave, consider the maximization of its first order Taylor series ($\mathbf{\Pi_{2m}}$) about $\mathbf{w} = \mathbf{w}_n$.

$$\mathbf{\Pi_{2m}} \quad \arg \max_{\|\mathbf{w}\|^2=1} f(\mathbf{w}_n) + \left(\sum_{k=1}^{K} \frac{(\mathbf{R}_k\mathbf{w}_n)^H(\mathbf{w} - \mathbf{w}_n)}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon}\right)$$

where $f(\mathbf{w})$ is the objective function in $\mathbf{\Pi_2}$. It is straightforward to see that the solution of $\mathbf{\Pi_{2m}}$ can be obtained in closed form

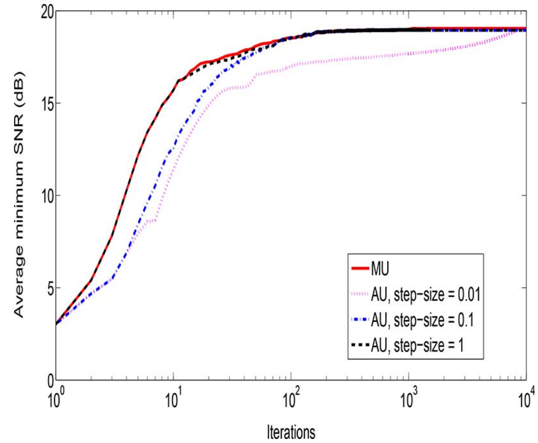

Fig. 1. Comparison of convergence rate of MU and AU algorithm for $N = 25$, $K = 500$.

and is equal to the update in (10). Therefore the $(n+1)$th iterate of the MU algorithm is the solution of the linear approximation $\mathbf{\Pi_{2m}}$ of $\mathbf{\Pi_2}$ at $\mathbf{w} = \mathbf{w}_n$. The difference with the AU is that in $\mathbf{\Pi_{2m}}$ we do not have the proximal regularization term that we had in $\mathbf{\Pi_{2r}}$.

Fig. 1 compares the minimum SNR obtained from the MU with that of the AU algorithm for various step-sizes α when $\mathbf{h}_k \sim \mathcal{C}N(\mathbf{0}, \mathbf{I}), \forall k$ and $\mathbf{R}_k = \mathbf{h}_k\mathbf{h}_k^H$. The plot has been obtained after averaging over 100 Monte-Carlo simulations. It can be seen that MU and the AU algorithms converge to the same minimum SNR and the convergence rate of the AU approaches the MU algorithm as α increases from 0.01 to 1.

C. MU-SLA Algorithm

An iterative successive linear approximation (SLA) algorithm has been proposed by Tran *et al.* [2] to approximately solve the following NP-hard problem $\mathbf{\Pi_3}$ [1].

$$\mathbf{\Pi_3} \quad \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2$$
$$\text{s.t.} \quad |\mathbf{w}^H\mathbf{h}_k|^2 \geq 1, \quad \forall k \in \{1, 2, \ldots K\}$$

The SLA algorithm starts with a feasible initialization $\mathbf{w}_0$. The non-convex SNR constraints for all the K users are linearized around $\mathbf{w}_0$ using Taylor series expansion and the resulting quadratic program is solved to obtain $\mathbf{w}_1$, which is subsequently used as the linearization point for the next iteration. The quadratic program solved in the $(n+1)$th SLA iteration is:

$$\mathbf{\Pi_{3r}} \quad \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2$$
$$\text{s.t.} \quad \|\mathbf{p}_{n,k}\|^2 + 2\mathbf{p}_{n,k}^T(\mathbf{v}_k - \mathbf{p}_{n,k}) \geq 1$$
$$\mathbf{v}_k = [\Re(\mathbf{w}^H\mathbf{h}_k), \Im(\mathbf{w}^H\mathbf{h}_k)]^T, \ \forall k \in \{1, 2, \ldots K\}$$

where $\Re(\cdot)$ ($\Im(\cdot)$) takes the real (imaginary) part of its argument, and $\mathbf{p}_{n,k} = [\Re(\mathbf{w}_n^H\mathbf{h}_k), \Im(\mathbf{w}_n^H\mathbf{h}_k)]^T$. Note that SLA was developed for the QoS rather than the max-min-fair formulation, but the two are equivalent up to normalization [1].

SLA solves a relatively complex quadratic program in each iteration, and the final result depends on the initialization. MU iterations, on the other hand, are very fast; but MU is geared towards proportional fairness, not max-min fairness, so the final result of MU can be refined to improve its max-min fairness. This naturally motivates a two-step *MU-SLA* algorithm which can take advantage of the high-quality solutions obtained quickly via the MU algorithm, and the ability of the SLA algorithm to perform accurate 'last mile' minimum SNR refinement.

In more detail, the MU-SLA algorithm takes the solution obtained from the MU algorithm, denoted by $\mathbf{w}_{\rm MU}$, scales it by the inverse square root of the minimum SNR attained using $\mathbf{w}_{\rm MU}$ (to maintain feasibility for $\mathbf{\Pi_3}$) and then uses this vector to initialize and solve a *single* SLA iteration. The resulting vector determines the transmit beamforming vector direction, which is then scaled to the desired transmit power. Our experiments (presented in the simulation results section) indicate that, in terms of minimum SNR and hence multicast rate, MU-SLA is the new state-of-art, as it outperforms all other multicast beamforming algorithms available as of this writing.

IV. Tx Has No Initial CSI About $\{\mathbf{R}_k\}_{k=1}^K$

Here, we assume that the matrices $\mathbf{R}_k,\ k \in \{1,2,\ldots,K\}$ are channel correlation matrices and have rank ≥ 1. It is also assumed that all the receivers have limited computational/energy resources. As a result, the conventional training method of channel correlation matrix estimation at every Rx followed by quantization and feedback to the Tx cannot be used. In this section, we propose an online transmit beamforming algorithm for a single group multicast network where the Tx uses binary feedback from every Rx to simultaneously learn $\{\mathbf{R}_k\}_{k=1}^K$ and design beamforming vectors that attain a high minimum average SNR.

Time is divided into slots of length T seconds, with the duration of each slot long enough for every Rx to perform accurate power estimation. At time $tT+\tau$, where $t \in \mathbb{Z}$ is an integer slot index and $\tau \in [0,T)$ is 'fast time', the channel from the Tx to the kth Rx is modeled as a zero-mean $N \times 1$ complex random vector $\mathbf{h}_k(tT+\tau)$, with a correlation matrix $\mathbf{R}_k = E(\mathbf{h}_k(tT+\tau)\mathbf{h}_k(tT+\tau)^H) \succeq \mathbf{0},\ \forall k \in \{1,2,\ldots,K\}$ and $\forall t \in \mathbb{Z},\ \forall \tau \in [0,T)$. At every time slot t, the Tx sends a zero-mean unit-variance common message $x(tT+\tau)$ times a unit-norm complex beamforming vector $\mathbf{w}_t$ to all the receivers in the downlink. The received signal at the kth Rx is

$$y_k(tT+\tau) = \mathbf{w}_t^H \mathbf{h}_k(tT+\tau)x(tT+\tau) + z_k(tT+\tau) \quad (11)$$

$\forall k \in \{1,2,\ldots,K\}$, where $z_k(tT+\tau)$ is the additive noise at Rx k (assumed to be wide-sense stationary) with zero-mean, variance σ_k^2, and independent of $x(tT+\tau)$ and $\mathbf{h}_k(tT+\tau)\ \forall t$ and $\forall \tau$. In the sequel, we assume that the received signal has been multiplied by $\frac{1}{\sigma_k}$ and absorb this factor into $\mathbf{h}_k(tT+\tau)$, for convenience. This makes the noise power equal to 1, and SNR equal to signal power.

In order to decode $x(tT+\tau)$, every Rx should estimate the complex scalar $\mathbf{w}_t^H \mathbf{h}_k(tT+\tau)$. One way to accomplish this task is to transmit pilot symbols at the start of every time slot to aid every Rx in this estimation of $\mathbf{w}_t^H \mathbf{h}_k(tT+\tau)$ in the presence of AWGN [15]. An alternative is to use differential modulation. During every time slot t, the kth Rx measures the average SNR $\mathbf{w}_t^H \mathbf{R}_k \mathbf{w}_t$ and compares it with a pre-determined threshold $\gamma_k(t)$. A '1' or a '0' is fed back by the kth Rx when its average SNR is $\geq$ or $<$ $\gamma_k(t)$. It is assumed that there are no significant measurement errors (inequality flips) at the Rx or in the communication of the 1-bit feedback to the Tx. Based on the 1-bit feedback from the kth Rx at time slot t, the Tx learns that

$$\begin{cases} \mathbf{w}_t^H \mathbf{R}_k \mathbf{w}_t \geq \gamma_k(t), & \text{when } s_k(t) = 1; \text{ or} \\ \mathbf{w}_t^H \mathbf{R}_k \mathbf{w}_t < \gamma_k(t), & \text{when } s_k(t) = 0, \end{cases} \quad (12)$$

where $k \in \{1,2,\ldots,K\}$ and $s_k(t)$ is the 1-bit feedback from the kth Rx at time slot t. Here, we extend the ACCPM-based adaptive beamforming algorithm in [11] from the case of a single-user MISO link to a multi-user multicast scenario, and propose an online Cognitive Multiplicative Update (CMU) algorithm. The CMU algorithm appropriately designs a sequence of $\{\mathbf{w}_t, \{\gamma_k(t)\}_{k=1}^K\}_t$ that enables the Tx to learn $\{\mathbf{R}_k\}_{k=1}^K$ using binary feedback from all the receivers and attain a high value for the minimum average SNR among the users.

A. Cognitive Multiplicative Update Algorithm

Exploration-Exploitation Tradeoff: At every time slot, the Tx has to design the beamforming vector in such a way that it can not only infer new information about $\{\mathbf{R}_k\}_{k=1}^K$ (exploration), but also use the knowledge accumulated from the feedback bits in previous time slots, to attain a high value of minimum average SNR (exploitation) among all the receivers. Since the Tx does not have any initial CSI, it is desirable to focus on exploration initially. As time progresses (number of feedback bits from each Rx increases), and the Tx is progressively able to accurately estimate the $\{\mathbf{R}_k\}_{k=1}^K$ matrices, preference can be shifted to exploitation. At the end of time slot t, the Tx has learned the following inequalities about $\{\mathbf{R}_k\}_{k=1}^K$ using the feedback bits from every Rx.

$$\begin{aligned} &{\mathbf{w}_i}^H \mathbf{R}_k \mathbf{w}_i \geq \gamma_k(i), \quad \forall i \in \mathcal{G}_{k1}; \\ &{\mathbf{w}_j}^H \mathbf{R}_k \mathbf{w}_j < \gamma_k(j), \quad \forall j \in \mathcal{G}_{k2} \end{aligned} \quad (13)$$

where $\mathcal{G}_{k1} = \{i : 1 \leq i \leq t,\ s_k(i) = 1\}$, $\mathcal{G}_{k2} = \{j : 1 \leq j \leq t,\ s_k(j) = 0\}$, $\mathcal{G}_{k1} \bigcup \mathcal{G}_{k2} = \{1,2,\ldots,t\}$ and $k \in \{1,2,\ldots,K\}$.

Channel Correlation Matrix Estimation: We propose to update $\hat{\mathbf{R}}_k(t),\ \forall k \in \{1,2,\ldots,K\}$ (the Tx-side estimate of $\mathbf{R}_k$ at time t) as follows.

$$\mathbf{\Pi_4} \quad \hat{\mathbf{R}}_k(t) = \arg\max_{\mathbf{R}} \sum_{i \in \mathcal{G}_{k1}} \log(\mathrm{Tr}\,(\mathbf{W}_i \mathbf{R}) - \gamma_k(i)) + \sum_{j \in \mathcal{G}_{k2}} \log(\gamma_k(j) - \mathrm{Tr}\,(\mathbf{W}_j \mathbf{R})) + \log\det \mathbf{R}$$

where $\mathbf{W}_i = \mathbf{w}_i \mathbf{w}_i^H$ and the term ${\mathbf{w}_i}^H \mathbf{R} \mathbf{w}_i$ has been rewritten as $\mathrm{Tr}(\mathbf{W}_i \mathbf{R})$. $\mathbf{\Pi_4}$ is a convex optimization problem which obtains the analytic center of the region formed by the linear inequalities till time slot t (13) for a particular k and the positive semi-definite cone [16], [17]. It can be solved efficiently using interior point methods.

Design of Beamforming Vector $\mathbf{w}_{t+1}$: Once the Tx updates $\{\hat{\mathbf{R}}_k(t)\}_{k=1}^K$, we formulate $\mathbf{\Pi_5}$ to design $\mathbf{w}_{t+1}$.

$\mathbf{\Pi_5}$

$$\mathbf{w}_{t+1} = \arg\max_{\|\mathbf{w}\|^2=1} \sum_{k=1}^K \frac{\log(\mathbf{w}^H \hat{\mathbf{R}}_k(t)\mathbf{w} + \varepsilon) - \lambda_t \mathbf{w}^H \mathbf{V}_t \mathbf{w}}{2}$$

where $\mathbf{V}_t = \mathbf{V}_{w,t}\mathbf{V}_{w,t}^H$, $\mathbf{V}_{w,t} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t]$, $\varepsilon > 0$ and λ_t is a non-increasing function of t, e.g., $\lambda_t = \frac{\lambda}{\lceil 0.1t \rceil}$, with $\lambda_1 \gg 1$. The objective function of $\mathbf{\Pi_5}$ comprises two terms. The first term promotes proportional fairness of SNR among users (if $\hat{\mathbf{R}}_k(t)$ is close to $\mathbf{R}_k, \forall k$) and the second term promotes the choice of a vector that is least similar to the beamforming vectors in all the previous time slots (since it minimizes the norm of the vector $\mathbf{V}_{w,t}^H\mathbf{w}$ whose ith entry is $\mathbf{w}^H\mathbf{w}_i, \forall i \leq t$). It has been shown that the proportional fairness function is non-concave. We replace the non-concave objective function of $\mathbf{\Pi_5}$ with the first-order Taylor series approximation, resulting in an optimization problem $\mathbf{\Pi_{5r}}$ which is obtained after removing the constant terms that are irrelevant to the optimization.

$\mathbf{\Pi_{5r}}$

$$\mathbf{w}_{t+1} = \arg \max_{\|\mathbf{w}\|^2=1} \left(\sum_{k=1}^{K} \frac{\hat{\mathbf{R}}_k(t)\mathbf{w}_t}{\mathbf{w}_t^H\hat{\mathbf{R}}_k(t)\mathbf{w}_t + \varepsilon} - \lambda_t \mathbf{V}_t \mathbf{w}_t \right)^H \mathbf{w}$$

The closed form solution of $\mathbf{\Pi_{5r}}$ is given by:

$$\tilde{\mathbf{w}}_{t+1} = \left(\sum_{k=1}^{K} \frac{\hat{\mathbf{R}}_k(t)}{\mathbf{w}_t^H\hat{\mathbf{R}}_k(t)\mathbf{w}_t + \varepsilon} \right) \mathbf{w}_t - \lambda_t \mathbf{V}_t \mathbf{w}_t,$$
$$\mathbf{w}_{t+1} = \frac{\tilde{\mathbf{w}}_{t+1}}{\|\tilde{\mathbf{w}}_{t+1}\|}. \tag{14}$$

The possibility of obtaining a beamforming vector update in closed form is the main motivation behind approximating the objective function of $\mathbf{\Pi_5}$ with the first-order Taylor series of the whole function as opposed to a concave approximation of the proportional fairness term alone (non-concave part), which would result in solving an optimization problem, thereby increasing the overall complexity significantly in every step.

The weight λ_t in $\mathbf{\Pi_5}$ decides the extent of preference given to the proportional fairness term (exploitation) in comparison with the diversity promoting term $\mathbf{w}^H\mathbf{V}_t\mathbf{w}$ (exploration). We propose to choose λ_t as a non-increasing function of t with $\lambda_1 \gg 1$ (for e.g., $\lambda_t = \frac{\lambda_1}{\lceil 0.1t \rceil}$). For small t, since $\lambda_t \gg 1$, the choice of weight vector is dictated by $\mathbf{w}^H\mathbf{V}_t\mathbf{w}$, thereby yielding diverse weight vectors that explore different directions for gathering information about $\{\mathbf{R}_k\}_{k=1}^K$. For large t, $\lambda_t \ll 1$, the Tx would have obtained sufficient information to accurately estimate $\{\mathbf{R}_k\}_{k=1}^K$ and preference shifts to the proportional fairness term, resulting in weight vectors that attempt to achieve a high minimum average SNR value among all the receivers. As $t \to \infty$, $\lambda_t \to 0$, the performance of the CMU algorithm asymptotically approaches that of the MU algorithm (where Tx has perfect CSI) if $\hat{\mathbf{R}}_k(t) \to \mathbf{R}_k, \forall k \in \{1, 2, \ldots, K\}$ (which is accomplished by appropriately designing the thresholds).

Design of Thresholds $\{\boldsymbol{\gamma}_k(t+1)\}_{k=1}^K$*:* Once $\mathbf{w}_{t+1}$ is designed, the Tx has to choose the thresholds $\{\boldsymbol{\gamma}_k(t+1)\}_{k=1}^K$ for the K receivers. $\{\boldsymbol{\gamma}_k(t+1)\}_{k=1}^K$ has to be chosen in such a way that the subsequent inequality inferred by the Tx from $\{s_k(t+1)\}_{k=1}^K$ significantly reduces the uncertainty about $\{\mathbf{R}_k\}_{k=1}^K$ at time slot t denoted by the region $\mathcal{P}_k(t) = \{\mathbf{R} : \mathbf{R} \succeq 0, {\mathbf{w}_i}^H\mathbf{R}\mathbf{w}_i \geq \gamma_k(i), \forall i \in \mathcal{G}_{k1}, {\mathbf{w}_i}^H\mathbf{R}\mathbf{w}_i < \gamma_k(i), \forall i \in \mathcal{G}_{k2}, \mathcal{G}_{k1} \bigcup \mathcal{G}_{k2} = \{1, 2, \ldots, t\}\}, \forall k \in \{1, 2, \ldots, K\}$. In this regard, we propose two threshold selection techniques.

Multiple Threshold Selection: Here, the Tx selects a unique threshold for every Rx at time slot $(t+1)$ which is given by

$$\gamma_k(t+1) = \mathbf{w}_{t+1}^H\hat{\mathbf{R}}_k(t)\mathbf{w}_{t+1}, \quad \forall k \in \{1, 2, \ldots, K\} \tag{15}$$

This selection (inspired by the ACCPM in convex optimization) ensures that the hyperplane corresponding to the inequality inferred about $\mathbf{R}_k$ at the Tx from $s_k(t+1)$, i.e., $\mathbf{w}_{t+1}^H\mathbf{R}\mathbf{w}_{t+1} = \gamma_k(t+1)$, $\mathbf{R} \in \mathcal{C}^{N\times N}$, $\mathbf{R} \succeq \mathbf{0}$ passes through the analytic center of $\mathcal{P}_k(t)$ i.e., $\hat{\mathbf{R}}_k(t)$, $\forall k \in \{1, 2, \ldots, K\}$. The analytic center $\hat{\mathbf{R}}_k(t)$ maximizes the product of distances to the defining hyperplanes and the p.s.d. cone in $\mathcal{P}_k(t)$ and gives the deepest interior point of $\mathcal{P}_k(t)$. Hence for a given $\mathbf{w}_{t+1}$, this choice of $\{\gamma_k(t+1)\}_{k=1}^K$ ensure that each of the K inequalities inferred by the Tx from $\{s_k(t+1)\}_{k=1}^K$ significantly reduces the uncertainty about $\mathbf{R}_k, \forall k \in \{1, 2, \ldots, K\}$. Using the convergence analysis of ACCPM [18], it can be shown that $\hat{\mathbf{R}}_k(t)$ is confined to a ball of radius r around $\mathbf{R}_k$, $\forall k \in \{1, 2, \ldots, K\}$ within $\mathcal{O}(\frac{N^2}{r^2})$ iterations. As $t \to \infty, \lambda_t \to 0$, $\hat{\mathbf{R}}_k(t) \to \mathbf{R}_k, \forall k$ and the performance of the CMU algorithm asymptotically approaches that of the MU algorithm with perfect knowledge of $\{\mathbf{R}_k\}_{k=1}^K$ at the Tx, even though CMU starts with no CSIT. However, the downlink signaling overhead is very high, because the Tx has to communicate $\gamma_k(t+1)$ to the kth Rx, $\forall k \in \{1, 2, \ldots, K\}$ for every time slot. This overhead increases linearly with K.

Common Threshold Selection: Here, the Tx selects a common threshold $\gamma(t+1)$ for all the users at time slot $(t+1)$.

$$\gamma(t+1) = \mathbf{w}_{t+1}^H\hat{\mathbf{R}}_{k_t}(t)\mathbf{w}_{t+1}, \text{ with } k_t := \text{mod}(t, K) + 1 \tag{16}$$

From (16), it can be seen that the common threshold at each time slot t is selected in a round-robin fashion. From the linear inequalities inferred by the Tx at time $t+1$, this selection ensures guaranteed reduction in the uncertainty region of user k_t only. Therefore, the uncertainty region of the channel correlation matrix of every user is certainly reduced at least once every K time slots. The convergence proof of ACCPM [18] can be used to show that $\hat{\mathbf{R}}_k(t)$ is confined to a ball of radius r around $\mathbf{R}_k(t)$, $\forall k \in \{1, 2, \ldots, K\}$ within $\mathcal{O}(\frac{KN^2}{r^2})$ iterations. In the worst-case, the convergence rate of common threshold selection will be K times slower than that of multiple threshold selection; but in practice, inequalities designed to reduce the uncertainty for one user will also reduce the uncertainty for other users. On the other hand, the per-slot communication overhead for the common threshold selection technique remains fixed even as K increases, since a single threshold is communicated to all K users. It should also be noted that the threshold communication can be avoided completely in this case, by keeping the thresholds at the users-side fixed for all t and scaling the transmit power instead. The set of linear inequalities inferred by the Tx at time $t+1$ can be modified as shown below.

$$\tilde{\mathbf{w}}_{t+1}^H\mathbf{R}_k\tilde{\mathbf{w}}_{t+1} \gtreqless 1, \quad \forall k \in \{1, 2, \ldots, K\} \tag{17}$$

where $\tilde{\mathbf{w}}_{t+1} = \sqrt{\frac{1}{\gamma_{t+1}}}\mathbf{w}_{t+1}$ and $\gtreqless$ means that the Tx will choose the inequality as $\geq$ or $<$ based on whether the 1-bit feedback is a '1' or a '0' respectively. In order to account for the variation of transmit power, the power amplifiers should have

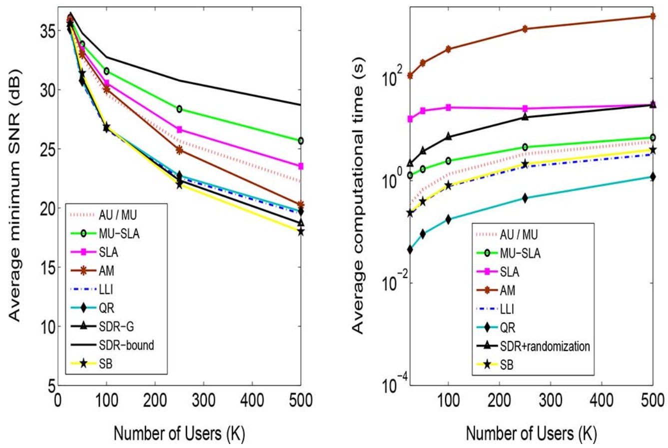

Fig. 2. Comparison of average minimum SNR and computation time versus K for N = 20 antennas when the user channels are drawn from an i.i.d. Gaussian distribution.

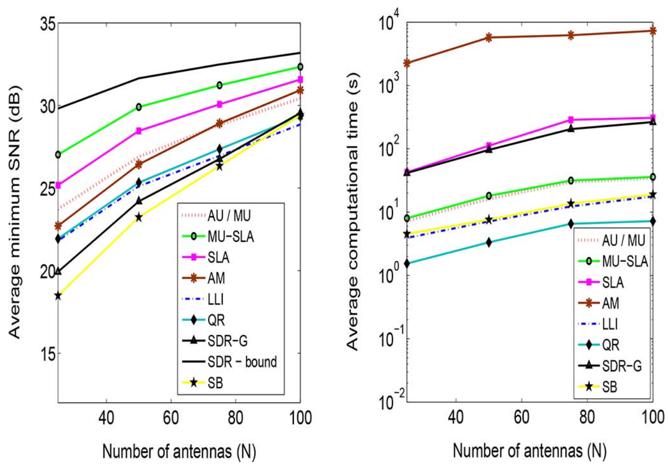

Fig. 3. Comparison of average minimum SNR and computation time versus N for K = 450 users when the user channels are drawn from an i.i.d. Gaussian distribution.

a much wider linear operating region to avoid non-linearities in the measurement of the received signal power and average SNR.

V. Simulation Results

A. Tx Has Perfect CSI

The average minimum SNR of the AU/MU and the MU-SLA algorithms are compared with the SDR upper bound and other state-of-the-art algorithms, namely SDR-G [1] (10^4 randomizations)[2], SLA [2], AM [3], LLI [5], QR algorithm [7] and the SB algorithm [8]. For the AU algorithm, the step-size is selected to satisfy the Lipschitz continuity condition. For Figs. 2–3 and 6–7, the channel vectors $\mathbf{h}_k$ are drawn from an i.i.d. $\mathcal{CN}(\mathbf{0}, \mathbf{I})$ distribution. The codes are executed using CVX [19] as the modeling language. The plots are obtained after averaging over 100 Monte-Carlo (MC) runs. For each run, the AU and the MU algorithms are executed until $\|\mathbf{w}_{n+1} - \mathbf{w}_n\| \leq 10^{-4}$ or a maximum of 1000 iterations, whichever comes first.

Fig. 2 compares the average minimum SNR and the average computation time[3] (averaged over MC runs) of all the algorithms versus the number of users (K) for a fixed $N = 20$ transmit antennas. Fig. 3 compares the variation of the same metrics with the number of transmit antennas N for $K = 450$. It can be seen that the MU-SLA algorithm attains the highest minimum SNR among all the algorithms ($\approx$ 0.8 dB above SLA for $N = 20, K = 500$); whereas the minimum SNR attained by the MU/AU algorithm is close but inferior to the SLA algorithm ($\approx$ 1 dB below SLA for $N = 20, K = 500$). It is also interesting to note that the minimum SNR of the AU/MU algorithm is slightly lower than the AM algorithm when K is small, but it outperforms the latter when $K > 150$ ($\approx$ 2 dB above AM for $N = 20, K = 500$).

The average computation time of the MU-SLA algorithm is very close to the AU/MU algorithm ($\mathcal{O}(10^1)$s for $K = 450$), both of which are significantly less than the SLA, the SDR-G ($\mathcal{O}(10^2)$s) and the AM ($\mathcal{O}(10^3)$s) algorithms. The MU-SLA algorithm outperforms the state-of-the-art by attaining the highest minimum SNR at a computational complexity similar to or much lower than all the high-performance algorithms. The gap between the SDR upper bound and the average minimum SNR achieved by the algorithms increases with $\frac{K}{N}$ ($\approx$ 0.4 dB for $K = 25, N = 20$ to $\approx$ 3.2 dB for $K = 500, N = 20$ for the MU-SLA algorithm in Fig. 2 and $\approx$ 0.5 dB for $K = 450, N = 100$ to $\approx$ 3 dB for $K = 450, N = 10$ for the MU-SLA algorithm in Fig. 3). This is in concurrence with the results on multicast capacity in [20]: it is difficult to attain a high minimum SNR among the users as $\frac{K}{N}$ increases when the channels are drawn from an i.i.d. zero-mean complex Gaussian distribution.

Figs. 4 and 5 compare the average minimum SNR when the channels are drawn from a Rician distribution for $N = 20$. This simulation models a practical scenario, where the users are clustered into multiple spatial groups in a given area (e.g., University campus), and the channel vectors of the users in a group are correlated. For this simulation, the users are clustered into G spatial groups and the channel to every user is modelled as follows: $\mathbf{h}_k = \mathbf{h}_{LOS,k_g} + \sigma_{k_g} * (\mathrm{randn}(N,1) + \sqrt{-1} * \mathrm{randn}(N,1))$, where $k_g = \mathrm{mod}(k, G)$ and $\mathbf{h}_{LOS,k_g}, k_g \in \{0, 1, \ldots G-1\}$ are the line-of-sight channels (common to all users in a group) from the Tx to the various clusters. If $\sigma_{k_g} \ll 1$, then the common line-of-sight component dominates the channels of the users in group k_g, thereby making them highly correlated. In Fig. 4, $\sigma_{k_g} = 10^{-3}, \forall k_g \in \{0, 1, 2, \ldots, G-1\}$ and $G \in \{5, 25\}$. Since $\sigma_{k_g} \ll 1$, the correlation of user-channels in a group is very high and the whole system can be approximated as single Tx with N antennas serving G users (instead of K).

As expected, the variation in the average minimum SNR of all the algorithms with K is rather small, because G is fixed. Also, it can be seen from Fig. 4 that the gap between the average minimum SNR for all the algorithms and the SDR bound is much less than that in Fig. 2 (1.3 dB for $G = 5, K = 450$ and 1.5 dB for $G = 25, K = 450$ vs. 9 dB in Fig. 2 for SDR-G at $K = 450$). For $G = 5$, all the algorithms perform close to optimal

[2]We also tried 10^5 randomizations in several test cases, and this made SDR-G much slower without noticeably improving its performance. SDR-G does not seem to work well when K is large, something that was also noted in earlier papers, e.g., [7].

[3]The computation time depends on software, hardware, coding quality and other implementation issues. We used standard/author-supplied codes where possible and carefully coded our implementations to ensure that the results are fair and indicative of the *relative* complexity of the different algorithms.

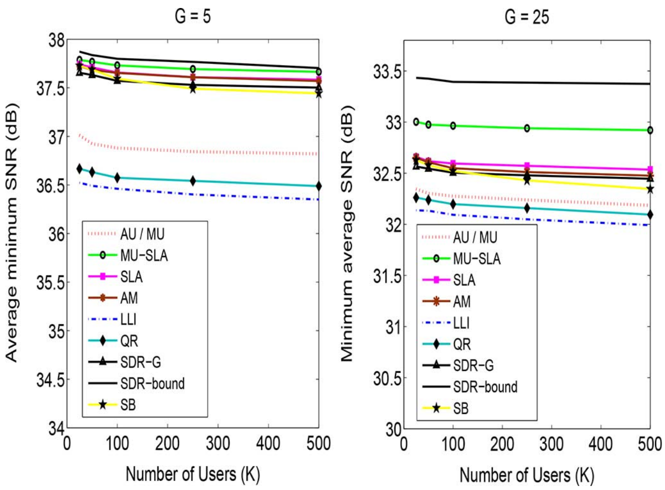

Fig. 4. Comparison of average minimum SNR versus K for N = 20 antennas when channels to all users are drawn from a mixture of $G = 5$ (left) and $G = 25$ (right) Rician distributions.

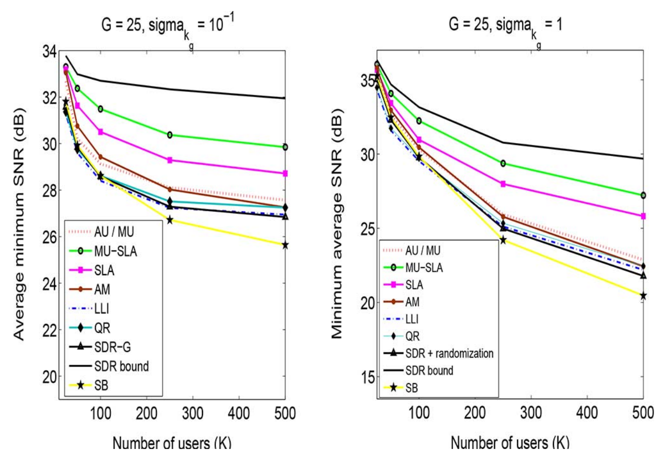

Fig. 5. Comparison of average minimum SNR versus K for N = 20 antennas when channels to all users are drawn from a mixture of $G = 25$ Rician distributions with $\sigma_{k_g} = 10^{-1}$ (left) and $\sigma_{k_g} = 1$ (right).

SDR bound because the Tx has more degrees of freedom than number of groups $(N > G)$ and the gap increases as G increases to 25 and $N < G$.

In Fig. 5, the value of G is fixed at 25 and $\sigma_{k_g} \in \{0.1, 1\}$. In Fig. 5, the variation in the minimum SNR with respect to K increases as σ_{k_g} increases. As σ_{k_g} increases, the LOS component becomes less dominant, the correlation between different channels to users in a particular group decreases, and the performance approaches the case where the channels to all the K users are drawn from an i.i.d. complex Gaussian distribution with zero mean (in Fig. 2).

Fig. 6 compares the minimum average SNR and the average computation time when $\{\mathbf{R}_k\}_{k=1}^K$ are full-rank channel correlation matrices. For each Monte-Carlo run, $\mathbf{R}_k = \mathbf{M}_k \mathbf{M}_k^H$, $\forall k \in \{1, 2, \ldots, K\}$, where the entries of $\mathbf{M}_k$ are drawn from an i.i.d. $\mathcal{CN}(0, 1)$ distribution. The QR algorithm is not used for comparison because it is not applicable when the matrices $\{\mathbf{R}_k\}_{k=1}^K$ have rank > 1. The SLA algorithm has been modified appropriately to work in this case (see Appendix D). It can be seen that the minimum SNR attained by MU-SLA algorithm is higher than all the other algorithms (0.5 dB above SLA at $K = 450$) and the minimum SNR attained by the AU/MU algorithm is lesser than the SLA algorithm (1.8 dB below SLA), but still higher than SDR-G (for $K > 150$) and AM for $K = 450$. The average computation time for the MU-SLA and the AU/MU algorithms ($\mathcal{O}(10^0)s$ at $K = 450$) are order(s) of magnitude lower than SDR-G ($\mathcal{O}(10^1)s$), SLA ($\mathcal{O}(10^2)s$) and AM ($\mathcal{O}(10^3)s$).

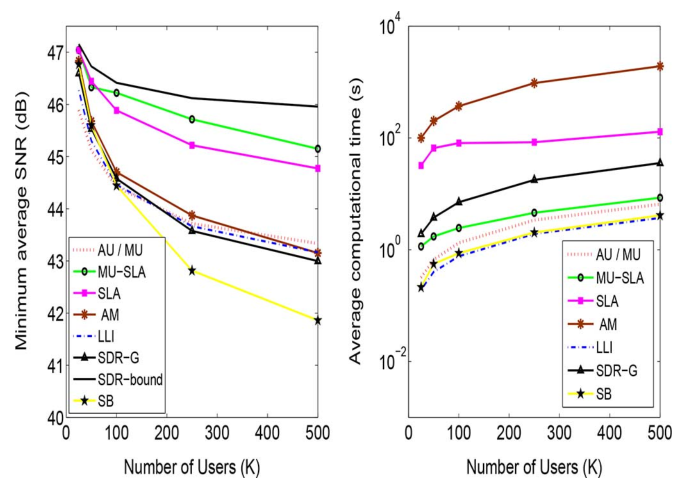

Fig. 6. Comparison of average minimum SNR and computation time versus K for $N = 20$ antennas, when $\{\mathbf{R}_k\}_{k=1}^K$ are full rank.

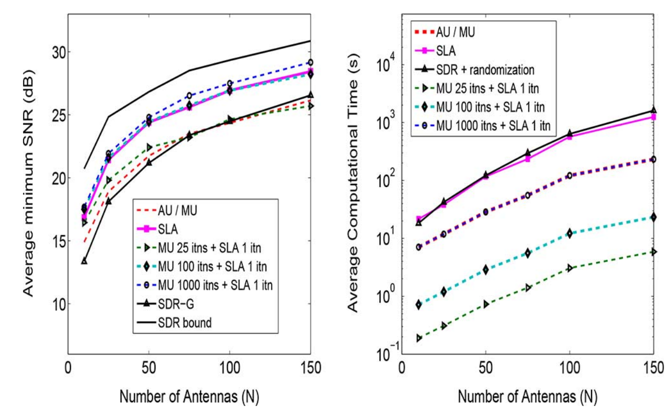

Fig. 7. Comparison of minimum average SNR and computation time versus N for $K = 500$ users, when channels to all users are drawn from an i.i.d. complex Gaussian distribution.

In Fig. 7, we explore the SNR performance-computation time tradeoff for the MU-SLA algorithm. For this simulation, the MU algorithm is run for 25, 100 and 1000 iterations. The beamforming vectors at the end of these iterations are scaled as mentioned in Section III.C and used as initialization for one SLA iteration. The resultant beamforming vector direction is scaled to the required transmit power. From Fig. 5, it can be seen that the minimum SNR of the MU-SLA algorithm improves with more iterations of the MU algorithm (e.g., at $N = 150$, 1000 MU iterations with 1 SLA iteration attains ≈ 3.5 dB higher minimum SNR than 25 MU iterations with 1 SLA iteration). This means that the SLA algorithm is quite sensitive to the quality of the initialization. On the other hand, the average computation time required for the MU-SLA algorithm increases with the number of MU iterations. From the plots, the best tradeoff seems to be the MU-SLA algorithm with 100 MU iterations and 1 SLA iteration, since its performance is ≈ 0.5 dB below the MU-SLA

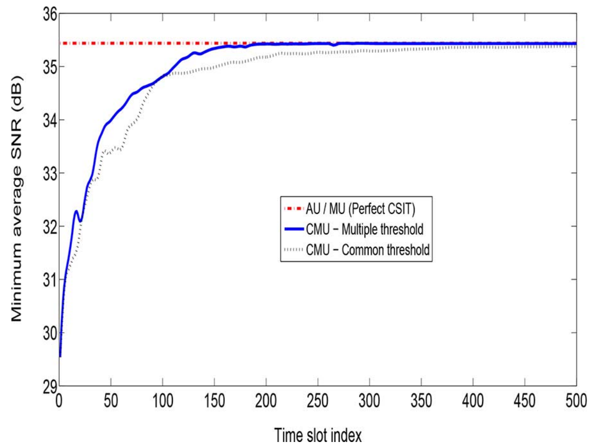

Fig. 8. Average minimum SNR of CMU algorithm for $N = 5$, $K = 20$.

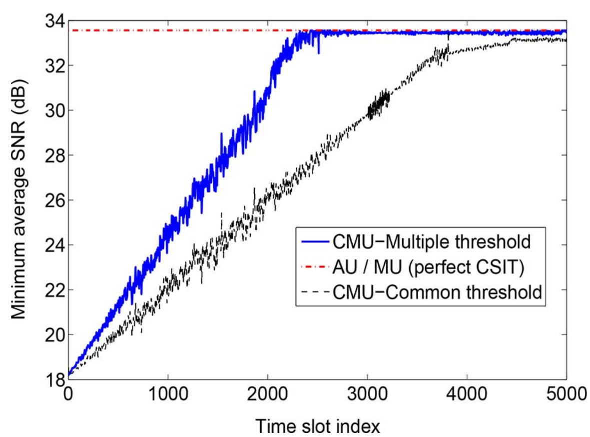

Fig. 9. Average minimum SNR of CMU algorithm for $N = 20$, $K = 50$.

algorithm with 1000 MU iterations and 1 SLA iteration, while the average computation time is $\mathcal{O}(10^0)$ seconds at $N = 150$.

B. Tx has no Initial CSI

Figs. 8 and 9 compare the minimum average SNR of the CMU algorithm with that of MU (perfect CSIT) for ($N = 5$, $K = 20$), and ($N = 20$, $K = 50$) respectively. The value of λ_t is chosen as $\lambda_t = \frac{1}{\lceil 0.1t \rceil}$, i.e., $\lambda_1 = \lambda = 1$. We tried different values of λ ranging from 0.01 to 100, and found that $\lambda = 1$ provided a good tradeoff between exploration and exploitation (i.e., by the time λ_t was very small, the Tx had just about acquired a good estimate of the user-channels) for the values of N and K used for the simulation results in Figs. 8 and 9. We note that λ should be an increasing function of N and K, because the number of unknowns grows like KN^2. For example, we tried $N = 25$ and $K = 50$ and found that $\lambda = 5$ worked best in that case. For each MC run, $\mathbf{R}_k = \mathbf{M}_k\mathbf{M}_k^H$, $\forall k \in \{1, 2, \ldots, K\}$, where the entries of $\mathbf{M}_k$ are drawn from an i.i.d. $\mathcal{CN}(0, 1)$ distribution. The dotted horizontal red line in each of the figures is the minimum average SNR attained by the MU algorithm with perfect CSIT.

It can be seen from Figs. 8 and 9 that the performance of the CMU algorithm with both threshold selection methods converges to the performance attained with perfect CSIT. It is evident that the CMU with multiple threshold selection converges faster than the common threshold selection (≈ 200 vs. 450 time slots when $N = 5, K = 20$ and ≈ 2500 vs. 4800 time slots when $N = 20, K = 50$).

It is interesting to note that the common threshold method is only ≈ 2.5 times slower for $N = 5$ and ≈ 2 times slower for $N = 20$ than the multiple threshold method (the worst-case scenario is $K = 20$ times slower in the first case and $K = 50$ times slower in the second). As alluded to earlier, this is due to the fact that the common threshold selection not only reduces the uncertainty in the channel correlation matrix of the user chosen via round-robin selection, but also decreases the uncertainty of the other users as well, although not to the extent accomplished by multiple threshold selection in every time slot.

VI. CONCLUSION

We considered the single group multicast network beamforming problem and proposed novel adaptive algorithms of low complexity, namely the AU, MU and MU-SLA, that obtain transmit beamforming vectors which attain a high minimum SNR when the Tx has perfect CSI. The fixed point equation of AU and MU was studied, and proof of convergence of the AU algorithm to a KKT point of proportionally fair beamforming was established. Extensive simulations were used to show that the MU-SLA algorithm outperforms the prior state-of-art (SLA) in terms of minimum SNR (and therefore multicast rate), whereas the AU/MU algorithm attains minimum SNR close to SLA, at far lower complexity.

When the Tx does not have any initial CSI and the receivers have limited computational resources, an online CMU algorithm based on ACCPM and appropriate threshold selection techniques were proposed to enable the Tx to learn the channel correlation matrices and design long-term transmit beamforming vectors simultaneously, using binary feedback from every Rx. Asymptotic convergence of the CMU algorithm to perfect-CSIT performance was established by invoking convergence results for the ACCPM from convex optimization, and verified in pertinent simulations. A variation of the common threshold selection technique was proposed to eliminate the threshold communication overhead at the cost of varying the transmit power (and the associated difficulties this imposes on power amplifiers). It was interesting to see that the convergence rate of the common threshold selection technique was much faster than the worst-case rate, thereby making it a strong contender for multicasting scenarios with only limited on-line feedback from the user terminals.

Last but not least, it is interesting to note that the proposed algorithms appear to be useful in the context of wireless power transfer—a concept that has gained traction recently, as a means of charging electrical devices without using power cables [21]. In this regard, the problem of maximizing the total sum-power harvested at all the receivers of a multi-user MISO system subject to a transmit power constraint was considered in [21]. The authors approximated the non-convex optimization problem using SDR and proved optimality of the SDR solution. Maximizing the sum-power harvested at all the receivers may lead to non-uniform power harvesting across different receivers. A 'fair' alternative is to maximize the minimum power harvested at every Rx subject to transmit power constraints. This latter formulation is exactly same as $\mathbf{\Pi_1}$, where $\mathbf{w}$ is the transmit beamforming vector, $\mathbf{R}_k$ is the channel correlation matrix of the kth Rx and $\mathbf{w}^H\mathbf{R}_k\mathbf{w}$ represents the average power harvested at the kth Rx. Therefore, the proposed algorithms can be used verbatim for this application as well.

Appendix A

Taking the scalar dot product with $\mathbf{w}_{fp}$ on both sides of (4) and using the fact that $\|\mathbf{w}_{fp}\|^2 = 1$, we get $c = 2$. It can be seen from (4) that $\mathbf{w}_{fp}$ is a linear combination of $\mathbf{h}_1$ and $\mathbf{h}_2$. Furthermore $\mathbf{w}_{fp}\, e^{j\phi}$ is also a fixed point of (4), $\forall \phi \in [-\pi, \pi]$ (i.e., the set of fixed points of (4) is closed under rotation). Consider one such fixed point.

$$\mathbf{w}_{fp} = a\mathbf{h}_1 + be^{j\theta}\mathbf{h}_2, \quad a, b \in \mathbb{R}^1,\ \theta \in [-\pi, \pi] \tag{18}$$

Using the closure property of the set of fixed points of (4), we can also assume that $a, b \in \mathbb{R}^1_+$ without loss of generality. Equating the right hand side of (4) and (18) after substituting for $\mathbf{w}_{fp}$ from (18) and using the fact that $c = 2$, we get

$$2a\mathbf{h}_1 + 2be^{j\theta}\mathbf{h}_2 = \frac{\mathbf{h}_1}{a\|\mathbf{h}_1\|^2 + be^{-j\theta}\mathbf{h}_2^H\mathbf{h}_1} + \frac{\mathbf{h}_2}{a\mathbf{h}_1^H\mathbf{h}_2 + be^{-j\theta}\|\mathbf{h}_2\|^2} \tag{19}$$

Assuming that $\mathbf{h}_1$ and $\mathbf{h}_2$ are linearly independent, we can equate their corresponding coefficients on both sides.

$$2a^2\|\mathbf{h}_1\|^2 + 2abe^{-j\theta}\mathbf{h}_2^H\mathbf{h}_1 = 1 \tag{20}$$
$$2abe^{j\theta}\mathbf{h}_1^H\mathbf{h}_2 + 2b^2\|\mathbf{h}_2\|^2 = 1 \tag{21}$$

In (20), since $a, b \in \mathbb{R}^1_+$, it is clear that $\theta = \angle(\mathbf{h}_2^H\mathbf{h}_1)$. Substituting this value of θ in (20) and (21) and equating the corresponding terms in the left hand side, we get

$$2a^2\|\mathbf{h}_1\|^2 + 2ab|\mathbf{h}_2^H\mathbf{h}_1| = 2b^2\|\mathbf{h}_2\|^2 + 2ab|\mathbf{h}_1^H\mathbf{h}_2| \tag{22}$$

This implies $a\|\mathbf{h}_1\| = b\|\mathbf{h}_2\|$. Using the fact that $\|\mathbf{w}_{fp}\|^2 = 1$ and $\theta = \angle(\mathbf{h}_2^H\mathbf{h}_1)$, from (20) and (22), we get

$$a = \frac{1}{\sqrt{2\left[\|\mathbf{h}_1\|^2 + \left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\right)\left|\mathbf{h}_2^H\mathbf{h}_1\right|\right]}} \tag{23}$$

Now the characterization of the fixed point is complete. We next compute the SNR at each of the receivers using this transmit beamforming vector $\mathbf{w}_{fp}$. The SNR at the kth Rx is $|\mathbf{w}_{fp}^H\mathbf{h}_k|^2,\ k \in \{1, 2\}$.

$$\begin{aligned}\left|\mathbf{w}_{fp}^H\,\mathbf{h}_1\right|^2 &= \left(a\|\mathbf{h}_1\|^2 + b\left|\mathbf{h}_2^H\mathbf{h}_1\right|\right)^2\\ &= a^2\left[\|\mathbf{h}_1\|^2 + \left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\right)\left|\mathbf{h}_2^H\mathbf{h}_1\right|\right]^2\\ &= \frac{1}{2}\left[\|\mathbf{h}_1\|^2 + \left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\right)\left|\mathbf{h}_2^H\mathbf{h}_1\right|\right]\end{aligned} \tag{24}$$

Similarly

$$\begin{aligned}\left|\mathbf{w}_{fp}^H\,\mathbf{h}_2\right|^2 &= \left|a\mathbf{h}_1^H\mathbf{h}_2 + be^{-j\theta}\|\mathbf{h}_2\|^2\right|^2\\ &= a^2\left[\left|\mathbf{h}_1^H\mathbf{h}_2\right| + \|\mathbf{h}_1\|\|\mathbf{h}_2\|\right]^2\\ &= \frac{1}{2}\frac{\left[\left|\mathbf{h}_1^H\mathbf{h}_2\right| + \|\mathbf{h}_1\|\|\mathbf{h}_2\|\right]^2}{\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\left[\left|\mathbf{h}_1^H\mathbf{h}_2\right| + \|\mathbf{h}_1\|\|\mathbf{h}_2\|\right]}\\ &= \frac{1}{2}\left[\|\mathbf{h}_2\|^2 + \left(\frac{\|\mathbf{h}_2\|}{\|\mathbf{h}_1\|}\right)\left|\mathbf{h}_2^H\mathbf{h}_1\right|\right]\end{aligned} \tag{25}$$

From (24) and (25), the minimum SNR among the receivers $\text{SNR}_{\min}$ and the associated multicast rate $r_{\min}$ are given by

$$\begin{aligned}\text{SNR}_{\min} &= \min\left(\left|\mathbf{w}_{fp}^H\,\mathbf{h}_1\right|^2, \left|\mathbf{w}_{fp}^H\,\mathbf{h}_2\right|^2\right)\\ &= \frac{1}{2}\Big[\min(\|\mathbf{h}_1\|^2, \|\mathbf{h}_2\|^2)\\ &\quad + \min\left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}, \frac{\|\mathbf{h}_2\|}{\|\mathbf{h}_1\|}\right)\left|\mathbf{h}_1^H\mathbf{h}_2\right|\Big]\\ r_{\min} &= \log_2(1 + \text{SNR}_{\min})\end{aligned} \tag{26}$$

Appendix B

The gradient of $f(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}_n$ is given by

$$\nabla_{\mathbf{w}} f(\mathbf{w}_n) = \sum_{k=1}^{K} \frac{\mathbf{R}_k\mathbf{w}_n}{\mathbf{w}_n^H\mathbf{R}_k\mathbf{w}_n + \varepsilon} \tag{27}$$

Now suppose that a projected gradient update algorithm is used for finding the local maxima of the constrained non-concave maximization problem $\mathbf{\Pi_2}$, where the update step at iteration $n + 1$ is given by $\tilde{\mathbf{w}}_{n+1} = \mathbf{w}_n + \alpha\nabla_{\mathbf{w}} f(\mathbf{w}_n), \mathbf{w}_{n+1} = \mathcal{P}_{S_w}(\tilde{\mathbf{w}}_{n+1})$, $\mathcal{P}_{S_w}(.)$ is the projection of the argument onto the set $S_w = \{\mathbf{w} : \|\mathbf{w}\|^2 = 1\}$ and α is a positive step size (same as in (6)). It can be seen that $\mathbf{w}_{n+1}$ in (9) is the optimal projection of the gradient update $\tilde{\mathbf{w}}_{n+1}$ onto the unit ball S_w. Furthermore, it can be shown that

- $\nabla f(\mathbf{w})$ is Lipschitz continuous in $\mathbf{w}$ with a Lipschitz constant $L_{\nabla f}$; See Appendix C.
- $\|\nabla^2 f(\mathbf{w})\|_F \leq \sum_{k=1}^{K}\left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda^2_{\max}(\mathbf{R}_k)}{\varepsilon^2}\right) =: L_{\nabla f}$, $\forall \|\mathbf{w}\| \leq 1$; See Appendix C.

Using the convergence results for the projected gradient method in ([22], Chapter 2, p. 240), it can now be shown that iterates of the AU algorithm in (6) converge to a Karush-Kuhn-Tucker(KKT) point of $\mathbf{\Pi_2}$ if $0 < \alpha \leq \frac{2}{L_{\nabla f}}$.

Appendix C

$$\begin{aligned}\nabla^2 f(\mathbf{w}) &= \sum_{k=1}^{K} \frac{(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon)\mathbf{R}_k - 2\mathbf{R}_k\mathbf{w}\mathbf{w}^H\mathbf{R}_k}{(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon)^2}\\ \|\nabla^2 f(\mathbf{w})\| &\leq \sum_{k=1}^{K} \left\|\frac{\mathbf{R}_k}{\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon}\right\|\\ &\quad + 2\left\|\frac{\mathbf{R}_k\mathbf{W}\mathbf{R}_k}{(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon)^2}\right\|\\ \|\nabla^2 f(\mathbf{w})\| &\leq \sum_{k=1}^{K}\left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda^2_{\max}(\mathbf{R}_k)}{\varepsilon^2}\right)\end{aligned} \tag{28}$$

where $\mathbf{W} = \mathbf{w}\mathbf{w}^H$, and we have used that $\mathbf{w}^H\mathbf{R}_k\mathbf{w} \geq 0, \forall \mathbf{w}$, and $\mathbf{w}^H\mathbf{R}_k\mathbf{w} \leq \lambda_{\max}(\mathbf{R}_k), \forall \mathbf{w} : \|\mathbf{w}\| = 1$. From (28), it can be seen that $\nabla^2 f(\mathbf{w})$ can be universally bounded over the feasible region. Furthermore it can also be seen that $\nabla f(\mathbf{w})$ is continuously differentiable. Hence, $\nabla f(\mathbf{w})$ is Lipschitz continuous in $\mathbf{w}$. Finally, $\nabla u(\mathbf{w}, \mathbf{w}(n))$ is a linear combination of two Lipschitz continuous functions i.e., $\nabla f(\mathbf{w}_n)$ and $\frac{(\mathbf{w} - \mathbf{w}_n)}{\alpha}$. Therefore, $\nabla u(\mathbf{w}, \mathbf{w}(n))$ is also Lipschitz continuous.

APPENDIX D

When the matrices $\{\mathbf{R}_k\}_{k=1}^{K}, \forall k \in \{1, 2, \ldots K\}$ have rank > 1, the optimization problem in [2] is given by $\mathbf{\Pi_6}$ and the nth SLA iteration is the solution of $\mathbf{\Pi_{6r}}$.

$$\begin{aligned} \mathbf{\Pi_6} \quad & \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \mathbf{w}^H \mathbf{R}_k \mathbf{w} \geq 1, \quad \forall k \in \{1, 2, \ldots K\} \\ \mathbf{\Pi_{6r}} \quad & \mathbf{w}_{n+1} = \arg \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + 2\Re[(\mathbf{R}_k \mathbf{w}_n)^H (\mathbf{w} - \mathbf{w}_n)] \geq 1 \end{aligned}$$

ACKNOWLEDGMENT

We would like to thank Ozlem Tugfe Demir and T. Engin Tuncer for explaining the Alternating Maximization algorithm in [3], which is used as one of the benchmarks in this paper.

REFERENCES

[1] N. D. Sidiropoulos, T. Davidson, and Z. Q. Luo, "Transmit beamforming for physical layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006.
[2] L. N. Tran, M. F. Hanif, and M. Juntti, "A conic quadratic programming approach to physical layer multicasting for large-scale antenna arrays," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 114–117, Jan. 2014.
[3] O. T. Demir and T. E. Tuncer, "Alternating maximization algorithm for the broadcast beamforming," in *Proc. EUSIPCO*, Lisbon, Portugal, Sep. 1–5, 2014.
[4] A. Lozano, "Long-term transmit beamforming for wireless multicasting," in *Proc. IEEE ICASSP*, Honolulu, HI, USA, Apr. 15–20, 2007, pp. 417–420.
[5] E. Matskani, N. D. Sidiropoulos, Z. Q. Luo, and L. Tassiulas, "Efficient batch and adaptive approximation algorithms for joint multicast beamforming and admission control," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4882–4894, Dec. 2009.
[6] M. J. Lopez, "Multiplexing, scheduling, and multicasting strategies for antenna arrays in wireless networks," Ph.D. dissertation, Elect. Eng. and Comp. Sci. Dept., MIT, Cambridge, MA, USA, 2002.
[7] A. Abdelkader, A. B. Gershman, and N. D. Sidiropoulos, "Multiple-antenna multicasting using channel orthogonalization and local refinement," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3922–3927, July 2010.
[8] H. Kim, D. J. Love, and S. Park, "Optimal and successive approaches to signal design for multiple antenna physical layer multicasting," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 2316–2327, Aug. 2011.
[9] Y. Huang, Q. Li, W.-K. Ma, and S. Zhang, "Robust multicast beamforming for spectrum sharing-based cognitive radios," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 527–533, Jan. 2012.
[10] Y. Noam and A. J. Goldsmith, "The one-bit null space learning algorithm and its convergence," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6135–6149, Dec. 2013.
[11] B. Gopalakrishnan and N. D. Sidiropoulos, "Cognitive transmit beamforming from binary CSIT," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 895–906, 2015.
[12] J. Xu and R. Zhang, "Energy beamforming with one-bit feedback," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5370–5381, Oct. 2014.
[13] Z. Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Trans. Signal Process.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.
[14] B. Gopalakrishnan and N. D. Sidiropoulos, "Adaptive multicast beamforming: Guaranteed convergence and state-of-art performance at low complexity," in *Proc. ICASSP*, Brisbane, Australia, Apr. 19–24, 2015.
[15] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, ser. Prentice-Hall Signal Processing Series. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
[16] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in systems and control theory," in *SIAM Studies in Applied Mathematics*. Philadelphia, PA, USA: SIAM, 1994.
[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[18] S. Boyd and L. Vandenberghe, "Notes on localization and cutting-plane methods," in *EE392 Lecture Notes*. Stanford, CA, USA: Stanford Univ., Sep. 2003.
[19] M. Grant and S. Boyd, CVX: Matlab Software for Disciplined Convex Programming. Apr. 2011 [Online]. Available: http://cvxr.com/cvx
[20] N. Jindal and Z. Q. Luo, "Capacity limits of multiple antenna multicast," in *Proc. IEEE ISIT*, Seattle, WA, USA, Jul. 9–14, 2006, pp. 1841–1845.
[21] J. Xu, L. Liu, and R. Zhang, "Multiuser MISO beamforming for simultaneous wireless information and power transfer," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 26–31, 2013, pp. 4754–4758.
[22] D.P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 2008.

Balasubramanian Gopalakrishnan received his B.Tech. degree in electronics and telecommunication engineering from the University of Kerala, India, in 2006, his M.Tech degree in electronics and electrical communication engineering from the Indian Institute of Technology, India, in 2008, and his Ph.D. in electrical and computer engineering from the University of Minnesota in 2015. He is currently a member of technical staff at Qualcomm, San Jose. His research interests are in wireless communications, signal processing, and optimization.

Nicholas D. Sidiropoulos (F'09) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Greece, and M.S. and Ph.D. degrees in electrical engineering from the University of Maryland—College Park, in 1988, 1990 and 1992, respectively. He served as Assistant Professor at the University of Virginia (1997–1999); Associate Professor at the University of Minnesota—Minneapolis (2000–2002); Professor at the Technical University of Crete, Greece (2002–2011); and Professor at the University of Minnesota—Minneapolis (2011 to present). His current research focuses primarily on signal and tensor analytics, with applications in cognitive radio, big data, and preference measurement. He received the NSF/CAREER award (1998), the IEEE Signal Processing Society (SPS) Best Paper Award (2001, 2007, 2011), the IEEE SPS Meritorious Service Award (2010), and the Distinguished Alumni Award from the Department of ECE, University of Maryland. He has served as IEEE SPS Distinguished Lecturer (2008–2009), and Chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He is also a Fellow of EURASIP.