ADAPTIVE MULTICAST BEAMFORMING: GUARANTEED CONVERGENCE AND STATE-OF-ART PERFORMANCE AT LOW COMPLEXITY

*Balasubramanian Gopalakrishnan**, *and Nicholas D. Sidiropoulos*†

Department of Electrical and Computer Engineering
University of Minnesota, MN 55455, USA

ABSTRACT

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. The associated beamformer design problem has drawn considerable attention over the last decade, but existing solutions are not quite satisfactory. The core problem is NP-hard, and the available approximations leave much to be desired in terms of achieving favorable performance-complexity trade-offs, especially for online implementation. This paper introduces a new class of adaptive multicast beamforming algorithms that simultaneously cover all bases - featuring guaranteed convergence and state-of-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 users. Convergence is established by recourse to proportional fairness. Simulation results show that the proposed algorithms outperform Semi-Definite Relaxation (SDR) and Successive Linear Approximation (SLA - the prior state-of-art) at an order of magnitude lower complexity.

Keywords: Multicast beamforming, max-min, proportional fairness, eMBMS, LTE

1. INTRODUCTION

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 the interference to other users and systems [1]. Multicasting can be broadly classified into a) single-group multicasting - where all the subscribers request a common data stream from the transmitter; and b) multiple-group multicasting - where different groups of subscribers request different data streams from the transmitter. In this paper, we consider the transmit beamforming problem for the single-group multicasting scenario. When transmitting common data to all the users, the downlink rate is restricted by the minimum received Signal to Noise Ratio (SNR) among all the users. Hence, one of the objectives is to maximize the minimum received SNR subject to a transmit power constraint, which is commonly referred to as the max-min beamforming (max-minBF) problem. An alternative problem formulation is to minimize the transmit power subject to Quality-of-Service (QoS) guarantees at the receivers of all the users (minP problem).

1.1. Related Work and Contributions

The minP and max-minBF problems were considered in [1] for the case where a multi-antenna transmitter (Tx) serves multiple users, each with a single antenna receiver (Rx). It was shown that the two formulations boil down to the same non-convex Quadratically Constrained Quadratic Programming (QCQP) problem, which is NP-hard in general; and Semi-Definite Relaxation (SDR) followed by Gaussian randomization was proposed to compute approximate solutions. When the number of antennas is large, SDR tends to be inefficient because it lifts the problem in higher-dimensional space, so several alternatives have been developed over the years. Recently, Tran *et al.* [2] proposed a Successive Linear Approximation (SLA) algorithm for approximately solving the minP problem. The SLA algorithm starts with a vector, say $\mathbf{w}_0$ which belongs to the feasible set. The non-convex constraints of the minP problem are linearized about the point $\mathbf{w}_0$ using first-order Taylor series expansion. The resulting convex problem is solved to obtain the next iterate $\mathbf{w}_1$ which is used for linearization in the next iteration. This procedure is repeated until the iterates converge to a fixed point. Simulations show that the SLA algorithm not only performs better than SDR with Gaussian randomization, but also has lower worst-case complexity - $\mathcal{O}(N+K)^{3.5}$ per iteration for SLA vs. $\mathcal{O}(N^2+K)^{3.5}$ overall for SDR, where N is the number of antennas at the Tx, K is the number of users, and the number of SLA iterations is usually small.

SDR or SLA require solving one large or many smaller (but still demanding) convex optimization problems, respectively. For large N and K, the computational burden of SDR / SLA becomes prohibitive for practical implementation, and low-complexity alternatives are needed. An iterative low complexity algorithm for approximating the max-minBF problem was first proposed by Lozano [3]. In each iteration, Lozano's algorithm takes a fixed step along the SNR gradient direction of the user that had the least SNR in the previous iteration. This is followed by scaling to satisfy the transmit power constraint. Simulations showed that Lozano's algorithm can outperform the SDR 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 - much lower than SDR and SLA. Matskani *et al.* [4] 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 more sophisticated initialization using the weight vector that maximizes average SNR [5].

Abdelkader *et al.* [6] proposed a low-complexity algorithm based on select channel orthogonalization using QR decomposition to approximate the minP problem when $K \geq N$. For every run of this QR algorithm, a set of N out of K channels is randomly chosen and stacked into a matrix $\mathbf{H}$. The QR decomposition of $\mathbf{H}$ is obtained, and the beamforming vector is selected as a linear combination of the columns of the $\mathbf{Q}$ factor matrix in the QR decomposition, with weights obtained in closed form [6]. This is 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

*E-mail: gopa0022@umn.edu. Supported by NSF EARS / AST-1247885.
†E-mail: nikos@umn.edu. Supported by NSF EARS / AST-1247885.

better than SDR with Gaussian randomization, at $\mathcal{O}(N^2)$ complexity - which is much lower than SDR.

A new and conceptually interesting approach to multicast beamforming was recently proposed by Demir *et al*. [7]. Similar to SDR, the approach in [7] isolates the nonconvex part of the problem in a rank-one constraint, but instead of dropping it (as SDR does), it replaces it with an equivalent non-convex bilinear trace constraint. The resulting problem is still NP-hard, but amenable to alternating optimization, which is nice. The drawback is that each alternating step requires solving an SDP, and one needs to alternate till convergence, so complexity is high; and the total number of variables is doubled. Our preliminary experiments with [7] indicate that, in the case of full-rank covariance matrices (long-term CSIT) it performs close to SDR with randomization; whereas for rank-one covariances (instantaneous CSIT) it performs poorly. In any case, the complexity of [7] is much higher than all other algorithms.

SLA is the state-of-art from the performance point of view - it attains higher minimum SNR / multicast rate than other methods, but at relatively high complexity, because it entails solving a sequence of convex optimization problems. This is not appealing for implementation at a base station, particularly for high N and K. LLI and QR are of sufficiently low complexity and can outperform SDR in certain cases, but leave much to be desired in terms of minimum SNR / multicast rate performance relative to SLA. In short, no algorithm offers state-of-art performance at low-enough complexity. The more lightweight algorithms (Lozano, LLI, QR) work reasonably well, yet remain *ad-hoc* and the tuning of parameters is an art that requires trial and error.

This paper introduces a new class of adaptive multicast beamforming algorithms that simultaneously cover all bases - featuring guaranteed convergence with no parameter tuning, and state-of-art performance at low complexity. Each iteration of the *Additive Update* (AU) algorithm takes a step in the direction of an inverse-SNR weighted linear combination of the SNR-gradient vectors of all users, computed using the beamforming vector obtained in the previous iteration. This is followed by a scaling step to satisfy the transmit power constraint, and the whole procedure is repeated until the iterates converge. Convergence is established by recourse to proportional fairness - showing that the AU can be interpreted as successive convex approximation of proportionally fair beamforming. This alludes to an interesting link between max-min fairness and proportional fairness. We also propose a *Multiplicative Update* which can be viewed as a limiting case of the AU algorithm. The MU eliminates the need of choosing a step-size and converges faster than the AU, although we currently have proof of convergence only for the AU - the analysis does not carry over verbatim to the MU for technical reasons. Finally, we propose the *Multiplicative Update - Successive Linear Approximation* (MU-SLA) algorithm where the solution provided by the MU algorithm is used as initialization for a *single SLA iteration*. Simulation results show that MU-SLA outperforms SLA, while AU and MU operate close to SLA and outperform all other algorithms, at an order of magnitude lower complexity.

2. PROBLEM DESCRIPTION

We consider a single group multicast cell consisting of a Tx with N antennas serving 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 k^{th} Rx is given by

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

where $\mathbf{h}_k$ is the channel between the Tx and the k^{th} Rx which is modelled as a complex $N \times 1$ random vector that is independent of x. z_k is the additive noise at the k^{th} Rx, which has zero-mean, variance σ_k^2, and is independent of x and $\mathbf{h}_k$. The SNR at the k^{th} Rx is given by $\frac{|\mathbf{w}^H \mathbf{h}_k|^2}{\sigma_k^2}$. We can absorb σ_k into $\mathbf{h}_k$, and thereafter work with the scaled channels $\tilde{\mathbf{h}}_k = \frac{\mathbf{h}_k}{\sigma_k}$. We will assume that this has already been done, and drop the $\tilde{}$ for brevity. The problem of interest can then be written as follows.

$$\mathbf{\Pi_1} \qquad \arg \max_{\|\mathbf{w}\|^2=1} \min_{k \in \{1,2,..,K\}} |\mathbf{w}^H \mathbf{h}_k|^2$$

3. ADDITIVE UPDATE ALGORITHM

The first adaptive algorithm for multicast beamforming was Lozano's [3]. In each iteration, Lozano's algorithm takes a step in a direction that improves the SNR of the weakest user - the one attaining the lowest SNR in the previous iteration. In other words, Lozano's algorithm focuses only on a single (the currently weakest) user in each iteration, temporarily ignoring all other users. This seems reasonable, yet it is a culprit behind 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 into account when 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.

$$\tilde{\mathbf{w}}_{n+1} = \mathbf{w}_n + \alpha \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right), \; \mathbf{w}_{n+1} = \frac{\tilde{\mathbf{w}}_{n+1}}{\|\tilde{\mathbf{w}}_{n+1}\|} \tag{2}$$

where $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H, \forall k \in \{1, 2, ..., K\}$, $\alpha > 0$ is a constant step size, and $\varepsilon > 0$ is introduced for numerical stability. The initial $\mathbf{w}_1$ can be randomly drawn (and normalized to unit norm), or designed using another low-complexity algorithm. At the $(n+1)^{st}$ iteration, the update direction is a linear combination of the gradients of the SNR of all the users at the point $\mathbf{w}_n$. For the $(n+1)^{st}$ iteration, the weight of the SNR gradient term of each user in the update direction is inversely proportional to the SNR of that user attained using the previous iterate $\mathbf{w}_n$. Therefore, in the $(n+1)^{st}$ iteration, $\mathbf{w}_{n+1}$ is updated along a direction that not only favors the user with the least SNR in the n^{th} iteration, but also takes into account all users - emphasizing those that experienced low SNR in the n^{th} iteration. This is to be contrasted with [3], [4], which only focus on the weakest link.

Inverse SNR-weighting of the gradient vectors intuitively aims to balance the SNR of all users. But can this intuition be rigorously justified? On a more basic level, does this procedure converge? If it does, then it must converge to a vector that satisfies the fixed point equation

$$\mathbf{w}_{FP} = \frac{1}{c} \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_{FP}^H \mathbf{R}_k \mathbf{w}_{FP} + \varepsilon} \right) \mathbf{w}_{FP} \tag{3}$$

for some constant $c \in \mathbb{R}$.

Proposition 1: The beamforming vector obtained at the $(n+1)^{st}$ iteration of the AU algorithm can be interpreted as the solution of a strongly convex approximation (cf. (5) and (4)) of the proportional fairness [8] multicast beamforming problem $\mathbf{\Pi_2}$ maximizing the geometric mean of the SNR of the users at the point $\mathbf{w} = \mathbf{w}_n$.

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

It can be shown that the Hessian of $f(\mathbf{w})$ (objective function in $\mathbf{\Pi_2}$) is indefinite. Therefore, $\mathbf{\Pi_2}$ is a non-concave maximization problem which is difficult to solve in general. Consider a strongly concave approximation of $f(\mathbf{w})$.

$$f(\mathbf{w}) \approx f(\mathbf{w}_n) + \overbrace{\left(\sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right)^H}^{(\nabla f(\mathbf{w}_n))^H} (\mathbf{w} - \mathbf{w}_n) - \frac{\|\mathbf{w} - \mathbf{w}_n\|^2}{2\alpha} \tag{4}$$

where α is the same as in (2). Denote the right hand side of (4) 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. Instead of solving $\mathbf{\Pi_2}$, suppose that we iteratively solve $\mathbf{\Pi_{2r}}$ to obtain $\mathbf{w}_{n+1}$ from $\mathbf{w}_n$.

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

It can be seen that the solution of $\mathbf{\Pi_{2r}}$ can be obtained in closed form and is given as follows:

$$\mathbf{w}_{n+1} = \frac{\mathbf{w}_n + \alpha \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right) \mathbf{w}_n}{\left\| \mathbf{w}_n + \alpha \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right) \mathbf{w}_n \right\|} \tag{5}$$

It can be seen from (2) and (5) that the $(n+1)^{st}$ 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 the SNR of all the users served by the Tx.

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 $L_{\nabla f} = \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2} \right) = \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\|\mathbf{h}_k\|^2}{\varepsilon^2} \right)$ and $\lambda_{\max}^2(\mathbf{R}_k) = \|\mathbf{h}_k\|^2$ is the maximum eigenvalue of $\mathbf{R}_k$.

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

$$\nabla_{\mathbf{w}} f(\mathbf{w}_n) = \sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \tag{6}$$

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 (2)). It can be seen that $\mathbf{w}_{n+1}$ in (5) 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 $L_{\nabla f}$.
- $\|\nabla^2 f(\mathbf{w})\|_F \leq \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2} \right) = \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\|\mathbf{h}_k\|^2}{\varepsilon^2} \right) =: L_{\nabla f}, \forall \|\mathbf{w}\| \leq 1$.

In simplifying the upper bound for $\|\nabla^2 f(\mathbf{w})\|_F$, we have used that $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H$, $\forall k = 1, 2, ..., K$. Using the convergence results for the projected gradient method in [9, Chapter 2, p. 240], it can be shown that iterates of the AU algorithm in (2) converge to a Karush-Kuhn-Tucker(KKT) point of $\mathbf{\Pi_2}$ if $0 < \alpha \leq \frac{2}{L_{\nabla f}}$.

3.1. Multiplicative Update algorithm

Here, we consider a limiting case of the AU algorithm which we will call the *Multiplicative Update* (MU) algorithm. The update step of the beamforming vector in the $(n+1)^{st}$ iteration is given below.

$$\tilde{\mathbf{w}}_{n+1} = \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right); \quad \mathbf{w}_{n+1} = \frac{\tilde{\mathbf{w}}_{n+1}}{\|\tilde{\mathbf{w}}_{n+1}\|} \tag{7}$$

The new iterate is the unit vector along a linear combination of the SNR gradient direction of all the K users (i.e., only the direction vector of AU algorithm). From (3) and (7) it can be seen that the MU algorithm has the same fixed point condition as the AU algorithm. The main motivation behind proposing the MU algorithm is two-fold. First and foremost, simulations show that the MU algorithm always converges to the same fixed point as the AU algorithm, and generally does so much faster than the AU algorithm. Second, the MU algorithm does not require choosing a step-size α. Unlike the AU algorithm, however, we do not have theoretical proof of convergence of the MU algorithm at this point.

To gain more insight about the MU algorithm, consider again the proportional fairness multicast beamforming problem $\mathbf{\Pi_2}$. Since the objective function is not concave, consider its first order Taylor series about $\mathbf{w} = \mathbf{w}_n$ (i.e., the objective function of $\mathbf{\Pi_{2m}}$)

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

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 and is equal to the update in (7). Therefore the $(n+1)^{st}$ iterate of the MU algorithm is the solution of successive linear approximation of $\mathbf{\Pi_2}$ at $\mathbf{w} = \mathbf{w}_n$. It can be seen from (2) that the AU update approaches the MU update as α increases. 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.

3.2. MU-SLA algorithm

An iterative successive linear approximation (SLA) algorithm was proposed by Tran *et al*. [2] to approximately solve the following NP-hard problem.

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

The SLA algorithm should be started with a feasible initialization $\mathbf{w}_0$. The non-convex constraints for all the K users are linearized around $\mathbf{w}_0$ using their first order Taylor series expansion and the resulting quadratic programming problem is solved to obtain $\mathbf{w}_1$, which is subsequently used for linearization in the next iteration. Motivated by the high-quality solutions obtained via AU / MU, and the potential of SLA for "last mile" refinement, we propose combining the two for cases where the computational complexity of *one* (as opposed to many) SLA iteration(s) is acceptable. The idea is to run MU until convergence, scale the resulting vector $\mathbf{w}_{MU}$ by the inverse square root of the minimum SNR attained using $\mathbf{w}_{MU}$ (to maintain feasibility for $\mathbf{\Pi_3}$) and then use the result to initialize a single SLA iteration. The resulting vector determines the transmit beamforming vector direction, which is then scaled to the desired transmit power. This is the MU-SLA algorithm. As it turns out, MU-SLA consistently outperforms all other methods in terms of attained minimum SNR / multicast rate, as illustrated in the simulations. This is because one iteration of SLA refines the solution of MU, but also MU provides a very good initialization to SLA.

4. SIMULATION RESULTS

In this section, the minimum SNR performance of the proposed algorithms i.e., the AU, MU, and MU-SLA are compared with the SDR upper bound and state-of-the-art algorithms, namely SLA [2], SDR with Gaussian randomization [1], Lozano's algorithm with Lopez initialization and damping [4] and the QR algorithm [6]. For the AU algorithm, the step-size is selected to satisfy the condition in Theorem 1.

For the simulations, the channel vectors $\mathbf{h}_k$ were drawn from an i.i.d. $\mathcal{CN}(\mathbf{0}, \mathbf{I})$ distribution. The codes were executed using CVX [10] as the modelling language. The plots were obtained after averaging over 100 Monte-Carlo (MC) runs. For each run, the AU and the MU algorithms were executed until $\|\mathbf{w}_{n+1} - \mathbf{w}_n\| \leq 10^{-4}$ or until reaching 1000, whichever occurs first. Fig. 1 compares the average (taken over MC runs) minimum SNR performance and the average (again taken over MC runs) computational time of all the algorithms versus K for $N = 25$ transmit antennas. Similarly, Fig. 2 compares the average minimum SNR performance and the computational time of various algorithms versus N for $K = 500$ users.

It can be seen that the MU-SLA algorithm attains the highest average minimum SNR among all the algorithms; whereas the average minimum SNR attained by the MU / AU algorithm is very close to the SLA algorithm (which performs the best among all the state of the art methods used for comparison) and significantly better than SDR. Furthermore, the average computation time of the MU-SLA algorithm is very close to the MU / AU algorithm, both of which are significantly less than the SLA and the SDR with randomization algorithms. Also, it can be seen from Fig. 1 and Fig. 2 that the gap between the SDR upper bound and the average minimum SNR achieved by the algorithms increases as $\frac{K}{N}$ increases ($\approx$ 0.4 dB for $K = 25, N = 25$ to $\approx$ 3.5 dB for $K = 500, N = 25$ for the MU-SLA algorithm in Fig. 1; and $\approx$ 0.5 dB for $K = 500, N = 150$ to $\approx$ 3.2 dB for $K = 500, N = 25$ for MU-SLA algorithm in Fig. 2). This behavior is in concurrence with the results on multicast capacity in [11]: it is difficult to attain a high SNR for all the users as K increases relative to N when the corresponding channels are drawn from an i.i.d. zero-mean complex Gaussian distribution. Also it can be seen that the minimum SNR increases as $\frac{K}{N}$ decreases because the Tx has more degrees of freedom at its disposal using which it is able to select better transmit beamforming vectors that attain higher minimum SNR.

5. CONCLUSION

In this paper, we considered the transmit beamforming problem for a single group multicast cell and proposed novel low-complexity adaptive algorithms, namely the AU algorithm, the MU algorithm, and the MU-SLA algorithm. These new algorithms attain very favorable performance - complexity trade-offs. MU-SLA outperforms all other available algorithms for multicast beamforming, including SDR and SLA; while MU / AU are close to SLA, which was the previous state-of-art method in terms of attaining the highest minimum SNR / multicast rate. This is quite remarkable given the low complexity of MU / AU, and even MU-SLA, as compared to SLA and SDR, and the fact that multicast beamforming is NP-hard. We proved that AU is guaranteed to converge via an interesting link to proportional fairness, also exploiting very recent results on the convergence of successive convex approximation for certain types of non-convex problems.

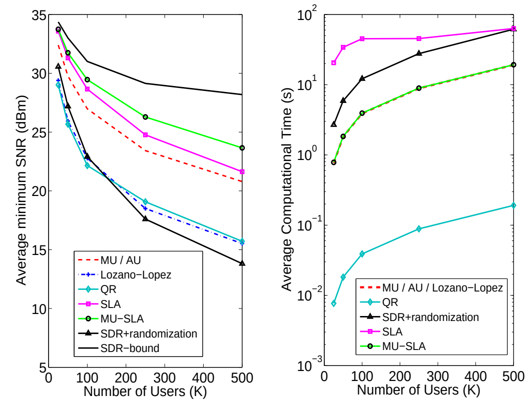

Fig. 1. Comparison of average minimum SNR and average computational time versus the number of users (K) for a) MU / AU, b) MU-SLA, c) SLA, d) Lozano with Lopez initialization, e) QR, f) SDR with 1000 Gaussian randomizations, and g) SDR upper bound for N = 25 antennas.

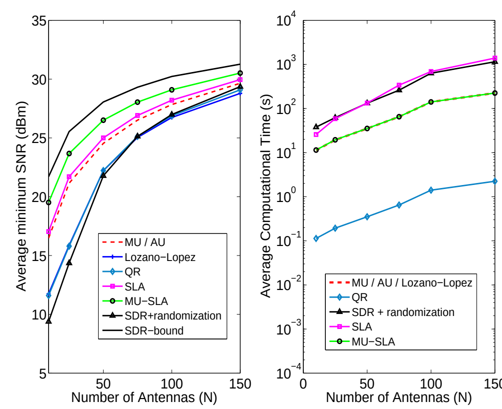

Fig. 2. Comparison of average minimum SNR and average computational time versus number of antennas (N) for a) MU / AU, b) MU-SLA, c) SLA, d) Lozano with Lopez initialization, e) QR, f) SDR with 1000 Gaussian randomizations, and g) SDR upper bound for K = 500 users.

6. REFERENCES

[1] N.D. Sidiropoulos, T. Davidson and Z.Q. Luo, "Transmit beamforming for Physical Layer Multicasting," *IEEE Trans. on Signal Processing*, vol. 54, no. 6, pp. 2239-2251, June 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 Processing Letters*, vol. 21, no. 1, pp. 114-117, Jan. 2014.

[3] A. Lozano, "Long-Term Transmit Beamforming for Wireless Multicasting," *Proc. of IEEE ICASSP*, pp. 417-420, Apr. 15-20, 2007, Honolulu, USA.

[4] 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. on Signal Processing*, vol. 57, no. 12, pp. 4882-4894, Dec. 2009.

[5] M.J. Lopez, "Multiplexing, Scheduling, and Multicasting Strategies for Antenna Arrays in Wireless Networks," *Ph.D. Dissertation*, Dept. of Elect. Eng. and Comp. Sci., MIT, Cambridge, MA, 2002.

[6] A. Abdelkader, A.B. Gershman and N.D. Sidiropoulos, "Multiple-Antenna Multicasting using Channel Orthogonalization and Local Refinement," *IEEE Trans. on Signal Processing*, vol. 58, no. 7, pp. 3922-3927, July 2010.

[7] O.T. Demir and T.E. Tuncer, "Alternating Maximization Algorithm for the Broadcast Beamforming," *Proc. of EUSIPCO*, Sep. 1-5, 2014, Lisbon, Portugal.

[8] Z.Q. Luo and S. Zhang, "Dynamic Spectrum Management: Complexity and Duality," *IEEE Trans. on Signal Processing*, vol. 2, no. 1, pp. 57-73, Feb. 2008.

[9] D.P. Bertsekas, *Nonlinear Programming*, Belmont, Massachusetts, U.S.A.: Athena Scientific, 2008.

[10] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," *http://cvxr.com/cvx*, April 2011.

[11] N. Jindal and Z.Q. Luo, "Capacity Limits of Multiple Antenna Multicast," *Proc. of IEEE ISIT*, pp. 1841-1845, July 9-14, 2006, Seattle, USA.