

Low-Complexity SDP-ADMM for Physical-Layer Multicasting in Massive MIMO Systems

Mahmoud Zaher, and Emil Björnson
{mahmoudz, emilbjo}@kth.se

Department of Computer Science, KTH Royal Institute of Technology, Sweden

Abstract—There is a demand for the same data content from several user equipments (UEs) in many wireless communication applications. Physical-layer multicasting combines the beamforming capability of massive MIMO (multiple-input multiple-output) and the broadcast nature of the wireless channel to efficiently deliver the same data to a group of UEs using a single transmission. This paper tackles the max-min fair (MMF) multicast beamforming optimization, which is an NP-hard problem. We develop an efficient semidefinite program-alternating direction method of multipliers (SDP-ADMM) algorithm to find the near-global optimal rank-1 solution to the MMF multicast problem in a massive MIMO system. Numerical results show that the proposed SDP-ADMM algorithm exhibits similar spectral efficiency performance to state-of-the-art algorithms running on standard SDP solvers at a vastly reduced computational complexity. We highlight that the proposed ADMM elimination procedure can be employed as an effective low-complexity rank reduction method for other problems utilizing semidefinite relaxation.

Index Terms—Multicast, downlink beamforming, convex optimization, semidefinite relaxation, ADMM, massive MIMO.

I. INTRODUCTION

Massive MIMO (multiple-input multiple-output) technology has emerged as a key enabler for wireless communication systems, offering significant gains in spectral efficiency (SE) [1], [2]. By deploying many antennas at the base station (BS), massive MIMO facilitates effective beamforming and spatial multiplexing, which can greatly boost the desired signal quality of multiple user equipments (UEs). Through beamforming, physical-layer multicasting can simultaneously deliver the same information to multiple UEs within a coverage area using a single transmission. This approach mitigates the problem of co-channel interference and is particularly beneficial to efficiently utilize network resources [3].

Physical-layer multicasting can support a wide range of applications in modern wireless communication networks. For instance, mobile network operators can leverage it for video streaming of live events to multiple subscribers, providing a cost-effective alternative to traditional unicast-based content delivery. Other applications of multicasting include videoconferencing as well as broadcasting of machine learning models in federated learning applications, where multiple UEs need to receive the same data stream. Another use case is emergency broadcast systems where critical information needs to be disseminated rapidly to a large audience, ensuring efficient and timely communication.

This work was supported by the FFL18-0277 grant from the Swedish Foundation for Strategic Research.

A foundational framework for physical-layer multicasting is presented in [4], introducing quality of service (QoS) and max-min fair (MMF) designs for a single-cell single-group multicast scenario, where the QoS level ensures a minimum received SNR for all UEs, and MMF aims to maximize the lowest received SNR. This concept was later expanded in [5] to include multiple co-channel multicast groups. The problem was proven to be NP-hard, and approximate solutions were derived using semidefinite relaxation (SDR). In [4], [5], different randomization strategies were proposed to extract near-optimal rank-1 beamforming solutions from higher-rank semidefinite program (SDP) output. However, as the number of transmit antennas and UEs grow, the approximation accuracy deteriorates significantly [4]–[6]. Note that a rank-1 beamforming solution, where the BS transmits a single multicast signal using spatial multiplexing, represents a practical choice due to its ease of implementation compared to a higher-rank transmission.

Recent studies have introduced different iterative optimization techniques to address the QoS and MMF multicast problems. In [6], the authors formulate a coordinated MMF multicast problem in a multicell network, where each BS serves a single UE group, and solve it using parametric manifold optimization. The study in [3] proposes three different algorithms for the MMF problem under per-cell power constraints with joint beamforming across cooperating BSs. Among these, the difference-of-convex approximation (DCA) algorithm demonstrates superior SE while achieving lower computational complexity compared to traditional SDR methods with randomization. The DCA algorithm is built upon the successive convex approximation (SCA) framework, which represents the state-of-the-art in multicast beamforming optimization. A two-layer beamforming scheme is introduced in [7], where the first layer suppresses multicast inter-group interference, and the second layer utilizes SCA to maximize the SNR of the decoupled multicast groups. In [8], we propose a successive elimination algorithm (SEA) that relies on SDR followed by iterative elimination of higher-rank solutions to extract a near-optimal rank-1 solution to the MMF multicast problem. Additionally, the authors in [9] present alternative formulations for the QoS and MMF problems, leveraging the optimal multicast beamforming structure, which provides a reduction in computational complexity when the number of BS antennas is much greater than the number of UEs.

A common approach in the prior literature [10]–[12] is

to utilize the SCA technique together with the alternating direction method of multipliers (ADMM) to achieve a low-complexity local optimum solution to the QoS and MMF multicast problems. It is worth noting that, when general-purpose solvers were used to solve the optimization problems, our previously proposed SEA has shown superior performance in terms of SE and computational complexity compared to state-of-the-art algorithms that rely on SCA [8]. A detailed performance evaluation and comparison of the proposed algorithm to the SCA-ADMM based algorithms will be provided in the extended journal article of this paper.

The advancements in multicast beamforming research highlight the increasing importance of efficient transmission techniques for emerging wireless applications. In this paper, we propose a novel SDP-ADMM algorithm to solve the MMF multicast problem in massive MIMO systems. Notably, this represents the first application of SDP-ADMM in multicast beamforming optimization, demonstrating its strong potential for improving efficiency in this field. We develop a new problem reformulation and ADMM-based implementation, extending our previously proposed SEA framework in [8], to obtain a near-global optimal rank-1 beamforming solution. A key advantage of our approach is its adaptability to various multicast optimization objectives and network configurations without requiring ADMM parameter tuning. Throughout the paper, we focus on multicast beamforming design, assuming perfect CSI at the transmitting BS. Our results confirm that the proposed SDP-ADMM algorithm achieves comparable SE performance to the SEA that relies on conventional SDP solvers, while significantly reducing computational complexity.

II. SYSTEM MODEL

This paper considers a massive MIMO BS equipped with N antennas. The BS serves K single-antenna UEs, that are arbitrarily distributed in a large service area, with a single multicast transmission. We consider a narrow-band channel, such that each channel realization is frequency-flat and quasi-static in time. The channel realizations are assumed to be available at the BS. The channel between UE k and the BS is denoted as $\mathbf{h}_k \in \mathbb{C}^N$. The system model is depicted in Fig. 1.

The received signal at UE k is computed as

$$y_k^{\text{dl}} = \mathbf{h}_k^H \mathbf{w} s + n_k, \quad (1)$$

where s denotes the zero-mean unit-variance multicast signal intended for all UEs, $\mathbf{w}$ represents the common multicast precoding vector, and $n_k \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_k^2)$ is the noise at UE k . The multicast precoding vector satisfies a short-term power constraint, which means that the power constraint must be satisfied for each channel realization. Accordingly, $\|\mathbf{w}\|^2 \leq P_{\max}$, where $P_{\max}$ represents the maximum transmit power of the BS. As a result, an achievable SE of the k^{th} UE under the perfect CSI assumption can be evaluated as

$$\text{SE}_k^{\text{dl}} = \log_2 \left(1 + \text{SNR}_k^{\text{dl}} \right). \quad (2)$$

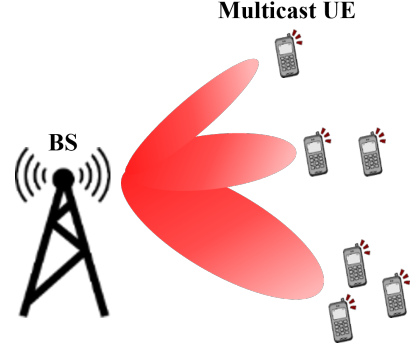


Fig. 1: Massive MIMO multicasting with irregular beamforming to a UE group.

We define the notation $\mathbf{H}_k = \mathbf{h}_k \mathbf{h}_k^H / \sigma_k^2$ and $\mathbf{W} = \mathbf{w} \mathbf{w}^H$. Utilizing the fact that $|\mathbf{h}_k^H \mathbf{w}|^2 = \text{tr}(\mathbf{h}_k \mathbf{h}_k^H \mathbf{w} \mathbf{w}^H)$, the SNR of UE k in (2) can be written as

$$\text{SNR}_k^{\text{dl}} = \frac{|\mathbf{h}_k^H \mathbf{w}|^2}{\sigma_k^2} = \text{tr}(\mathbf{H}_k \mathbf{W}). \quad (3)$$

This SNR formulation is suitable to construct the SDP of the MMF multicast problem.

A. Max-Min Fair Multicast Problem

In this paper, the primary focus is on the MMF multicast problem. The goal is to find the multicast transmit precoding vector $\mathbf{w}$ that maximizes the minimum achievable SNR in (3) among all UEs in the system under the power constraint at the BS. Since all UEs in a multicast transmission must be able to decode the same signal, it is important to note that the minimum SNR among the UEs determines what data rate can be utilized. Hence, an appropriate design criterion for the multicast problem is to maximize the minimum achievable SNR.

We utilize the relation between the QoS and MMF multicast problems to solve the MMF problem. The logic is that the target SNR of the QoS problem represents the MMF SNR that is achievable given a power budget at the BS equal to the optimum objective value achieved for the QoS problem. Since the optimum objective values of the MMF and QoS problems are monotonically non-decreasing in the BS power budget and the SNR target, respectively, a bisection search can be performed over the SNR target to identify the max-min SNR that results in an optimum objective value of the QoS problem equal to the power budget available at the BS. This relation is well-established in the previous literature, e.g., [4], [8], [10].

Let $\mathbb{H}^N$ denote the set of $N \times N$ Hermitian matrices. The linear map $\mathcal{H}(\cdot) : \mathbb{H}^N \rightarrow \mathbb{R}^K$ is defined as

$$\mathcal{H}(\mathbf{W}) = \left(\langle \mathbf{H}_1^H, \mathbf{W} \rangle, \dots, \langle \mathbf{H}_K^H, \mathbf{W} \rangle \right), \quad (4)$$

where the inner product between two matrices is given by $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^H \mathbf{B})$. The QoS problem can then be written compactly as

$$\min_{\mathbf{W}} \quad \text{tr}(\mathbf{W}) \quad (5a)$$

$$\text{s.t.} \quad \mathcal{H}(\mathbf{W}) \geq \gamma, \quad (5b)$$

$$\mathbf{W} \succeq \mathbf{0}, \quad \text{rank}(\mathbf{W}) = 1, \quad (5c)$$

where $\gamma \in \mathbb{R}^K$ represents the SNR targets of the UEs. The above problem is non-convex due to the non-convex rank-1 constraint in (5c). A relaxation of the problem is devised by dropping the rank-1 constraint to obtain the relaxed convex QoS problem. In general, such relaxation results in the solution matrix $\mathbf{W}$ having a high rank that does not satisfy the rank constraint of the original problem, requiring post-processing of the output matrix to extract a feasible rank-1 beamforming solution. Finding the optimal post-processing is challenging.

B. Successive Elimination Algorithm

Low-rank matrix solutions to SDP problems belong to the boundary of the feasible set and not the interior. This means that the solution to the relaxed SDP problem will most likely have a high rank, even if a rank-1 solution exists. In fact, standard SDP solvers will always favor a higher-rank solution over the lower-rank counterpart, even if both can achieve the same objective value [13]. Previous works relying on SDR have generally used a randomization procedure [5] to extract a rank-1 beamforming solution. However, this method does not scale well and a large number of candidate sets must be tested when increasing the number of BS antennas and UEs for satisfactory SE performance. For that purpose, we utilize the state-of-the-art SEA proposed in our primal work [8]. The idea is to iteratively eliminate the output higher-rank solutions to the SDP that result in an optimum objective value greater than or equal to that achieved with the optimum rank-1 solution, until a near-optimal rank-1 solution is achieved. The elimination is done through penalizing the directions of the eigenvectors corresponding to the second-largest eigenvalues of the higher-rank solutions of the previous iterations. The penalty is in the form of a quadratic matrix product $\zeta \sum_r \mathbf{u}_r^H \mathbf{W} \mathbf{u}_r$, where $\zeta \geq 1$ represents the penalty factor and the $\mathbf{u}_r, \forall r$, represent the eigenvectors corresponding to the second-largest eigenvalues of the previous higher-rank solutions.

In this paper, we propose a new formulation that is suitable for developing a computationally fast and effective ADMM algorithm to solve the QoS problem. The penalized relaxed QoS problem is formulated as

$$\min_{\mathbf{W}} \quad \langle \mathbf{\Lambda}, \mathbf{W} \rangle \quad (6a)$$

$$\text{s.t.} \quad \mathcal{H}(\mathbf{W}) \geq \gamma, \quad (6b)$$

$$\mathbf{W} \succeq \mathbf{0}, \quad (6c)$$

where $\mathbf{\Lambda} \in \mathbb{C}^{N \times N}$ is a design parameter that is responsible for the elimination of the higher-rank solutions and improves the convergence speed of the proposed algorithm, and will be detailed later.

III. SDP-ADMM: EFFICIENT MULTICAST BEAMFORMING

In this section, we develop an efficient SDP-ADMM algorithm to solve the MMF multicast beamforming problem. The proposed algorithm relies on ADMM and utilizes the relation between the MMF and QoS problems as well as the SEA to find the near-optimal rank-1 MMF beamforming vectors.

A. A Fast SDP-ADMM Algorithm For The QoS Problem

Since the penalized relaxed QoS problem in (6) is convex and satisfies Slater's condition, strong duality holds and it is easier to solve the dual SDP problem. We begin by constructing the Lagrangian of problem (6) as

$$L_q(\mathbf{W}, \mathbf{y}, \mathbf{S}) = \langle \mathbf{\Lambda}, \mathbf{W} \rangle - \langle \mathbf{y}, \mathcal{H}(\mathbf{W}) - \gamma \rangle - \langle \mathbf{S}, \mathbf{W} \rangle, \quad (7)$$

where $\mathbf{y} \in \mathbb{R}_+^K$ and $\mathbf{S} \in \mathbb{H}_+^N$ represent the nonnegative and positive semidefinite Lagrange multipliers of the constraints in (6b) and (6c), respectively. The adjoint operator of $\mathcal{H}$ is $\mathcal{H}^H(\cdot) : \mathbb{R}^K \rightarrow \mathbb{H}^N$, and is defined as $\mathcal{H}^H(\mathbf{y}) = \sum_{k=1}^K y_k \mathbf{H}_k$. Taking the derivative of the Lagrangian with respect to $\mathbf{W}$ and equating it to $\mathbf{0}$, we get

$$\mathcal{H}^H(\mathbf{y}) + \mathbf{S} = \mathbf{\Lambda}. \quad (8)$$

Plugging this condition back in (7), the dual function becomes $g(\mathbf{W}, \mathbf{y}, \mathbf{S}) = \mathbf{y}^T \gamma$ when the condition in (8) is satisfied. The dual SDP problem can then be formulated as

$$\min_{\mathbf{y}, \mathbf{S}} \quad -\mathbf{y}^T \gamma \quad (9a)$$

$$\text{s.t.} \quad \mathcal{H}^H(\mathbf{y}) + \mathbf{S} = \mathbf{\Lambda}, \quad (9b)$$

$$\mathbf{S} \succeq \mathbf{0}, \quad \mathbf{y} \geq \mathbf{0}. \quad (9c)$$

We define the indicator function $\mathbb{I}_{\mathbf{y} \geq \mathbf{0}}(\mathbf{y})$ as

$$\mathbb{I}_{\mathbf{y} \geq \mathbf{0}}(\mathbf{y}) = \begin{cases} 0 & \text{for } \mathbf{y} \geq \mathbf{0}, \\ +\infty & \text{otherwise.} \end{cases} \quad (10)$$

Then, an equivalent reformulation of the dual SDP problem is

$$\min_{\mathbf{y}, \mathbf{S}} \quad -\mathbf{y}^T \gamma + \mathbb{I}_{\mathbf{y} \geq \mathbf{0}}(\mathbf{y}) \quad (11a)$$

$$\text{s.t.} \quad \mathcal{H}^H(\mathbf{y}) + \mathbf{S} = \mathbf{\Lambda}, \quad (11b)$$

$$\mathbf{S} \succeq \mathbf{0}. \quad (11c)$$

The augmented Lagrangian, in scaled form, for the dual SDP corresponding to the linear constraints can thus be written as

$$L_\rho(\mathbf{W}, \mathbf{y}, \mathbf{S}) = -\mathbf{y}^T \gamma + \mathbb{I}_{\mathbf{y} \geq \mathbf{0}}(\mathbf{y}) + \frac{\rho}{2} \left\| \mathcal{H}^H(\mathbf{y}) + \mathbf{S} - \mathbf{\Lambda} + \mathbf{W} \right\|_F^2. \quad (12)$$

Utilizing the ADMM algorithm [14], the minimization is done with respect to each block of variables separately while the other variables are kept fixed. At each iteration, the following updates are computed sequentially:

$$\mathbf{y}^{i+1} = \arg \min_{\mathbf{y}} L_\rho(\mathbf{W}^i, \mathbf{y}, \mathbf{S}^i), \quad (13)$$

$$\mathbf{S}^{i+1} = \arg \min_{\mathbf{S}} L_\rho(\mathbf{W}^i, \mathbf{y}^{i+1}, \mathbf{S}), \quad \text{s.t. } \mathbf{S} \succeq \mathbf{0}, \quad (14)$$

$$\mathbf{W}^{i+1} = \mathbf{W}^i + \mathcal{H}^H(\mathbf{y}^{i+1}) + \mathbf{S}^{i+1} - \mathbf{\Lambda}. \quad (15)$$

1) *y-Update*: Introducing an auxiliary variable $\mathbf{z}$ and the indicator function $\mathbb{I}_{\mathbf{z} \geq \mathbf{0}}(\mathbf{z})$, the update of $\mathbf{y}$ in problem (13) is equivalent to solving the following problem:

$$\min_{\mathbf{y}, \mathbf{z}} \quad -\mathbf{y}^T \boldsymbol{\gamma} + \frac{\rho}{2} \left\| \mathcal{H}^H(\mathbf{y}) + \mathbf{S}^i - \boldsymbol{\Lambda} + \mathbf{W}^i \right\|_F^2 + \mathbb{I}_{\mathbf{z} \geq \mathbf{0}}(\mathbf{z}) \quad (16a)$$

$$\text{s.t.} \quad \mathbf{y} = \mathbf{z}. \quad (16b)$$

This problem can be efficiently solved using an (inner) ADMM algorithm to find the optimal $\mathbf{y}$ -update. The augmented Lagrangian of problem (16) can be formulated with the scaled dual variable $\mathbf{g}$ as

$$L_{\rho, \mu}(\mathbf{W}^i, \mathbf{y}, \mathbf{S}^i, \mathbf{z}, \mathbf{g}) = -\mathbf{y}^T \boldsymbol{\gamma} + \frac{\mu}{2} \|\mathbf{y} - \mathbf{z} + \mathbf{g}\|_2^2 + \mathbb{I}_{\mathbf{z} \geq \mathbf{0}}(\mathbf{z}) + \frac{\rho}{2} \left\| \mathcal{H}^H(\mathbf{y}) + \mathbf{S}^i - \boldsymbol{\Lambda} + \mathbf{W}^i \right\|_F^2. \quad (17)$$

The update of the variable blocks is done successively using the ADMM algorithm, while other variables are kept fixed. Accordingly, the inner ADMM iterations require computing the following updates:

$$\mathbf{y}^{j+1} = \arg \min_{\mathbf{y}} L_{\rho, \mu}(\mathbf{W}^i, \mathbf{y}, \mathbf{S}^i, \mathbf{z}^j, \mathbf{g}^j), \quad (18)$$

$$\mathbf{z}^{j+1} = \arg \min_{\mathbf{z}} L_{\rho, \mu}(\mathbf{W}^i, \mathbf{y}^{j+1}, \mathbf{S}^i, \mathbf{z}, \mathbf{g}^j), \quad (19)$$

$$\mathbf{g}^{j+1} = \mathbf{g}^j + \mathbf{y}^{j+1} - \mathbf{z}^{j+1}. \quad (20)$$

Note that the iteration index for the outer ADMM algorithm is omitted for the variable $\mathbf{y}$ to simplify the notation. For the inner $\mathbf{y}$ -update, the minimization of (17) with respect to $\mathbf{y}$ can be done by equating its first-order derivative to zero, that is

$$\begin{aligned} & -\boldsymbol{\gamma} + \mu (\mathbf{y}^{j+1} - \mathbf{z}^j + \mathbf{g}^j) \\ & + \rho \mathcal{H} \left(\mathcal{H}^H(\mathbf{y}^{j+1}) + \mathbf{S}^i - \boldsymbol{\Lambda} + \mathbf{W}^i \right) = \mathbf{0}. \end{aligned} \quad (21)$$

Reordering the terms and using the distributive property of linear maps lead to

$$\begin{aligned} & \rho \mathcal{H} \left(\mathcal{H}^H(\mathbf{y}^{j+1}) \right) + \mu \mathbf{y}^{j+1} \\ & = \boldsymbol{\gamma} - \rho \mathcal{H}(\mathbf{S}^i - \boldsymbol{\Lambda} + \mathbf{W}^i) + \mu (\mathbf{z}^j - \mathbf{g}^j). \end{aligned} \quad (22)$$

The operator $\mathcal{H}(\mathcal{H}^H(\mathbf{y}))$ can be equivalently formulated as $\mathcal{H}(\mathcal{H}^H(\mathbf{y})) = \mathbf{H}\mathbf{H}^H\mathbf{y}$, where $\mathbf{H} \in \mathbb{C}^{K \times N^2}$ is defined as $\mathbf{H} = [\text{vec}(\mathbf{H}_1)^H; \dots; \text{vec}(\mathbf{H}_K)^H]$ and $\text{vec}(\mathbf{H}_k)$ is the column-wise vectorization of $\mathbf{H}_k$. This reformulation allows for a closed-form update of $\mathbf{y}$ as

$$\mathbf{y}^{j+1} = \left(\rho \mathbf{H}\mathbf{H}^H + \mu \mathbf{I}_K \right)^{-1} \left(\boldsymbol{\gamma} - \rho \mathcal{H}(\mathbf{S}^i - \boldsymbol{\Lambda} + \mathbf{W}^i) + \mu (\mathbf{z}^j - \mathbf{g}^j) \right). \quad (23)$$

Note that the matrix inverse in (23) needs to be computed only once before the start of the main (outer) ADMM algorithm.

Moreover, the term $\boldsymbol{\gamma} - \rho \mathcal{H}(\mathbf{S}^i - \boldsymbol{\Lambda} + \mathbf{W}^i)$ is independent of the inner ADMM iterations so is computed once for every $\mathbf{y}$ -update of the outer ADMM iterations. The remaining term is relatively cheap to compute, which results in a fast inner ADMM update.

As for the update of the auxiliary variable $\mathbf{z}$, problem (19) is equivalent to solving the following problem:

$$\min_{\mathbf{z}} \quad \left\| \mathbf{y}^{j+1} + \mathbf{g}^j - \mathbf{z} \right\|_2^2 \quad (24a)$$

$$\text{s.t.} \quad \mathbf{z} \geq \mathbf{0}. \quad (24b)$$

It is clear from problem (24) that the optimum $\mathbf{z}$ -update is given by

$$\mathbf{z}^{j+1} = \max \left(\mathbf{y}^{j+1} + \mathbf{g}^j, \mathbf{0} \right). \quad (25)$$

2) *S-Update*: We define the variable $\mathbf{X}^{i+1} = \boldsymbol{\Lambda} - \mathcal{H}^H(\mathbf{y}^{i+1}) - \mathbf{W}^i$. Problem (14) can be rewritten as

$$\min_{\mathbf{S}} \quad \left\| \mathbf{S} - \mathbf{X}^{i+1} \right\|_F^2 \quad (26a)$$

$$\text{s.t.} \quad \mathbf{S} \succeq \mathbf{0}. \quad (26b)$$

The $\mathbf{S}$ -update is then given by

$$\mathbf{S}^{i+1} = \mathbf{X}^{i+1} \triangleq \mathbf{Q}_+^{i+1} \boldsymbol{\Sigma}_+^{i+1} (\mathbf{Q}_+^{i+1})^H, \quad (27)$$

where $\boldsymbol{\Sigma}_+$ is a diagonal matrix with the non-negative eigenvalues of $\mathbf{X}^{i+1}$ and $\mathbf{Q}_+^{i+1}$ denotes a matrix with the corresponding eigenvectors as columns.

3) *Convergence Criteria*: The stopping criteria for the outer ADMM updates are chosen to verify the convergence of the primal and dual variables. To determine termination, the following two conditions must be satisfied at the i^{th} iteration:

$$\frac{|\text{tr}(\mathbf{W}^i - \mathbf{W}^{i-1})|}{\text{tr}(\mathbf{W}^i)} < \epsilon_{\text{dual}}, \quad (28)$$

$$\frac{\left\| \mathbf{S}^i - \mathbf{S}^{i-1} \right\|_F}{\left\| \mathbf{S}^i \right\|_F} < \epsilon_{\text{prim}}, \quad (29)$$

where $\epsilon_{\text{dual}} > 0$ and $\epsilon_{\text{prim}} > 0$ are predefined stopping conditions. We highlight that the stopping criterion for the dual variable $\mathbf{W}$ directly translates into the relative change in the original optimization objective. To avoid early termination due to stagnation, both conditions are deemed necessary to achieve the best possible outcome. As for the inner ADMM algorithm with cheap updates, we avoid computing stopping criteria to reduce the complexity per inner iteration. Instead, the algorithm is run for a fixed number of iterations T .

B. Designing $\boldsymbol{\Lambda}$

The SEA is a method to extract a near-optimal rank-1 solution through the elimination of the higher-rank solutions from the SDP. This method was introduced in our primal work [8]. The idea is to iteratively modify the optimization objective by introducing a penalty on the directions of the eigenvectors corresponding to the second largest eigenvalues of the higher-rank solutions outputted by the SDP in the previous iterations. The purpose of the design parameter $\boldsymbol{\Lambda}$ is to incorporate

this penalty and improve the convergence speed of our SDP-ADMM implementation.

Since the trace and summation are linear operators, the objective of the original QoS problem (5) and the modified penalty of the SEA can be reformulated as

$$\begin{aligned} \text{tr}(\mathbf{W}) + \sum_r \zeta_r \mathbf{u}_r^H \mathbf{W} \mathbf{u}_r &= \text{tr}(\mathbf{W}) + \text{tr}\left(\sum_r \zeta_r \mathbf{U}_r \mathbf{W}\right) \\ &= \text{tr}\left(\left(\mathbf{I}_N + \sum_r \zeta_r \mathbf{U}_r\right) \mathbf{W}\right) \\ &= \left\langle \mathbf{I}_N + \sum_r \zeta_r \mathbf{U}_r, \mathbf{W} \right\rangle, \end{aligned} \quad (30)$$

where $\mathbf{U}_r = \mathbf{u}_r \mathbf{u}_r^H$. In this work, we make use of the linearity of the derivatives in the ADMM update equations by modifying the penalty proposed in [8]. The design parameter Λ is then set to

$$\Lambda = c\mathbf{I}_N + \sum_r \zeta_r \mathbf{U}_r, \quad (31)$$

where $c \geq 1$ is a tunable weight set to improve the convergence speed towards the original optimization objective. Further, the penalty factors ζ_r , $\forall r$ are modified and set equal to the corresponding eigenvalues of $\mathbf{U}_r$. We stress that this penalty is particularly beneficial for the ADMM algorithm updates. In this way, after a penalty is applied to a higher-rank output solution, the second strongest component of this higher-rank precoding matrix $\mathbf{W}$ is completely eliminated from the subsequent iterates $\{\mathbf{S}^i\}$ and $\{\mathbf{W}^i\}$ of our SDP-ADMM algorithm. We have observed that this modification improves the convergence speed and the numerical stability of the proposed SDP-ADMM algorithm compared to the previously proposed penalty term, however, it is not well-suited for higher-order methods.

C. MMF Multicast Beamforming

Since every UE in a multicast group must be able to decode the data, we assume a common SNR target for all the UEs, that is $\gamma = \gamma_c$. However, we stress that the proposed algorithm is applicable to different SNR targets. The complete procedure is detailed in Algorithm 1. Note that P_T represents the power budget at the BS and κ determines the elimination interval width, i.e., the bisection search interval for subsequent higher-rank solution elimination.

Two fundamental advantages of the proposed algorithm are worth noting. The first is that it does not require initialization with a feasible solution to the original QoS problem, which is necessary for algorithms that rely on the SCA technique (the state-of-the-art in multicast beamforming optimization). Note that it might be hard to find an effective low-complexity solution to use as initialization, specially when the number of UEs grows large. Another advantage is that the elimination procedure can perform the bisection search over the SNR target within a relatively small interval. This allows

Algorithm 1 MMF Multicast Beamforming via SDP-ADMM

Input: The interval $[\gamma_{\text{lo}}, \gamma_{\text{up}}]$ that contains the optimum objective value of the MMF problem, $\gamma_{\text{lo}} = 0$ and $\gamma_{\text{up}} = \min_k \frac{P_T \|\mathbf{h}_k\|^2}{\sigma_k^2}$. Solution tolerance $\epsilon > 0$. ADMM penalty parameters ρ and μ , and stopping conditions $\epsilon_{\text{dual}} > 0$ and $\epsilon_{\text{prim}} > 0$. Number of inner ADMM iterations T . Initialize $\mathbf{y}^0 = \mathbf{z}^0 = \mathbf{g}^0 = \mathbf{0}$, $\mathbf{S}^0 = \mathbf{W}^0 = \frac{P_T}{N} \mathbf{I}_N$.

```

1: repeat
2:   Set  $\gamma_c \leftarrow (\gamma_{\text{lo}} + \gamma_{\text{up}}) / 2$ .
3:   Initialize  $\mathbf{W}^0$  as a scaled version of the solution to the
   previous bisection iteration with the factor  $\gamma_c / \gamma_{\text{prev}}$ .
4:   Solve the QoS problem (6) using the SDP-ADMM
   algorithm in Section III-A for the current  $\gamma_c$ .
5:   if  $\text{tr}(\mathbf{W}) > P_T$  then
6:     Set  $\gamma_{\text{up}} \leftarrow \gamma_c$ .
7:   else
8:     Set  $\gamma_{\text{lo}} \leftarrow \gamma_c$ .
9:   end if
10:  Set  $\gamma_{\text{prev}} \leftarrow \gamma_c$ .
11: until  $\gamma_{\text{up}} - \gamma_{\text{lo}} < \epsilon$ 
12: Set  $r \leftarrow 1$ .
13: Set  $\Lambda \leftarrow c\mathbf{I}_N$ .
14: while  $\text{rank}(\mathbf{W}) \neq 1$  do
15:   Set  $\zeta_r \leftarrow$  Second strongest eigenvalue of  $\mathbf{W}$ .
16:   Set  $\mathbf{U}_r \leftarrow \mathbf{u}_r \mathbf{u}_r^H$ , where  $\mathbf{u}_r$  is the corresponding
   eigenvector to  $\zeta_r$ .
17:   Update  $\Lambda \leftarrow \Lambda + \zeta_r \mathbf{U}_r$ .
18:   Set  $\gamma_{\text{lo}} \leftarrow \kappa \gamma_c$ .
19:   Repeat steps 1-11.
20:   Set  $r \leftarrow r + 1$ .
21: end while

```

Output: The near-optimal rank-1 solution.

for a reduction in the computational time of the elimination procedure iterations as compared to the first solution (the conventional higher-rank SDP solution). The reason is that the upper bound of the bisection search interval will be the optimum γ_c achieved in the previous bisection iteration, whereas the lower bound is chosen with a small separation away from it (Step 18).

It is worth mentioning that at a given bisection iteration over the SNR target, the initial point for $\mathbf{W}$ is chosen as a scaled version of the solution to the previous bisection iteration. This provides a warm start to the ADMM algorithm and can greatly speed up the convergence to the optimal solution. In fact, if the QoS problem is solved exactly, i.e., a rank-1 solution is obtained directly from the SDP, then the scaled $\mathbf{W}$ is actually the optimal solution for the new SNR target. Otherwise, this scaled solution is fine-tuned with the ADMM algorithm for each step of the bisection search over the SNR target.

IV. NUMERICAL EVALUATION

In this section, we use Monte Carlo simulations to verify the effectiveness of the proposed SDP-ADMM optimization

TABLE I: Network simulation parameters.

Area of interest	750 m × 750 m
Bandwidth	20 MHz
Number of BS antennas	$N = 36$
Number of UEs	$K = \{15, 30\}$
BS transmit power	$P_T = 40$ W
DL noise power	-94 dBm

procedure to solve the MMF multicast problem. We consider a massive MIMO system with a half-wavelength-spaced uniform linear array of $N = 36$ antennas deployed at the BS, serving an area of 750 m × 750 m. We assume the BS serves $K = \{15, 30\}$ UEs, that are randomly and uniformly distributed within the area of interest, using a single multicast transmission. The simulation parameters are summarized in Table I. The channel between the BS and an arbitrary UE k is modeled by correlated Rayleigh fading as $\mathbf{h}_k \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{R}_k)$, where $\mathbf{R}_k \in \mathbb{C}^{N \times N}$ represents the spatial correlation matrix, generated using the local scattering model in [1]. The average channel gain, $\beta_k = \frac{1}{N} \text{tr}(\mathbf{R}_k)$, is calculated using the 3GPP Urban Microcell model with correlated shadowing among the UEs. More precisely, the large-scale fading coefficients are given by

$$\beta_k = -30.5 - 36.7 \log_{10} \left(\frac{d_k}{1 \text{ m}} \right) + F_k \text{ dB}, \quad (32)$$

where d_k represents the distance between the BS and UE k , and $F_k \sim \mathcal{N}(0, 4^2)$ denotes the shadow fading. The shadowing is correlated between the BS and different UEs as

$$\mathbb{E}\{F_k F_i\} = 4^{2-\delta_{ki}/9 \text{ m}}, \quad (33)$$

where δ_{ki} is the distance between UE k and UE i .

The SDP-ADMM penalty parameters are set to $\rho = 1$ and $\mu = 10^5$. The stopping conditions for the outer ADMM algorithm are set to $\epsilon_{\text{dual}} = 10^{-5}$ and $\epsilon_{\text{prim}} = 10^{-4}$, whereas the maximum number of iterations is set to 1000. For the inner ADMM updates, the number of iterations is fixed to $T = 50$. The tunable weight defined for improving the convergence speed of the algorithm is set to $c = 5$. The stopping criterion for the bisection search is chosen to be $\epsilon = 0.1$ and the elimination interval uses $\kappa = 0.9$. We use the same platform for performing the simulations, a 4 core Intel(R) Core i5-10310U CPU with 1.7 GHz base frequency and 4.4 GHz turbo frequency. All programs are written in Matlab. The CDF curves are generated using 2000 simulation samples.

Figs. 2 and 3 plot the cumulative distribution function (CDF) of the max-min SE for the proposed SDP-ADMM algorithm for $K = 15$ and $K = 30$ UEs, respectively. The state-of-the-art SEA algorithm [8], which utilizes CVX [15], and the SDR upper bound are shown for comparison. It is clear that the proposed low-complexity SDP-ADMM algorithm exhibits similar SE performance to that of the SEA. Later, we will show the tremendous saving in computational time for the proposed SDP-ADMM algorithm. For $K = 15$ UEs, both the SDP-ADMM and SEA algorithms almost achieve the SDR upper bound. Remarkably, the proposed SDP-ADMM algorithm is

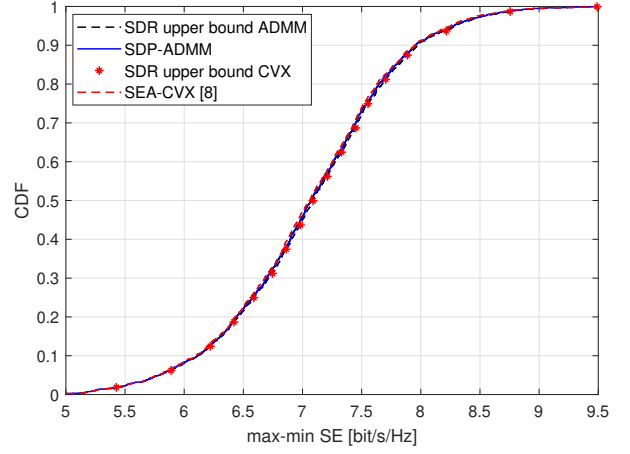
Fig. 2: CDF of the max-min SE for $K = 15$ UEs.

TABLE II: Average computational time in seconds.

K	SDP-ADMM	SEA [8]
15	0.6	8.6
30	2.1	17.8

able to achieve a small SE improvement over [8] for the denser scenario with $K = 30$ UEs, thanks to the refined penalty term that is specifically tailored for the SDP-ADMM implementation. We stress that this modified penalty term is not suitable for algorithms that rely on higher-order methods.

We highlight that we have selected our previously proposed SEA as a benchmark as it outperforms state-of-the-art SCA-based methods when standard solvers were used to solve the optimization problems, as demonstrated in [8]. The SE improvement over SCA-based methods stems from their use of a gradient descent approach, which ensures convergence only to a stationary point of the non-convex MMF problem. In contrast, the proposed algorithm begins with the optimal higher-rank solution obtained via SDR and iteratively reduces its rank by penalizing eigenvectors associated with the second-largest eigenvalues. This process continues until a near-optimal rank-1 solution is reached in the orthogonal subspace of these eigenvectors. At each iteration, the solution matrices are Hermitian positive semidefinite with distinct non-zero eigenvalues, ensuring that the penalty minimally impacts the optimality of subsequent iterations. As a result, this approach efficiently converges to a near-global optimal rank-1 solution to the NP-hard MMF multicast problem.

Table II presents the average computational time of the proposed SDP-ADMM algorithm as compared to the benchmark. For the case of $K = 15$ UEs, the proposed algorithm is able to achieve more than 10 times reduction in computational time as compared to the state-of-the-art SEA utilizing CVX and attains the same SE performance. As for the dense scenario of $K = 30$ UEs, the proposed algorithm brings an 88% reduction in computational time while achieving a small improvement in SE.

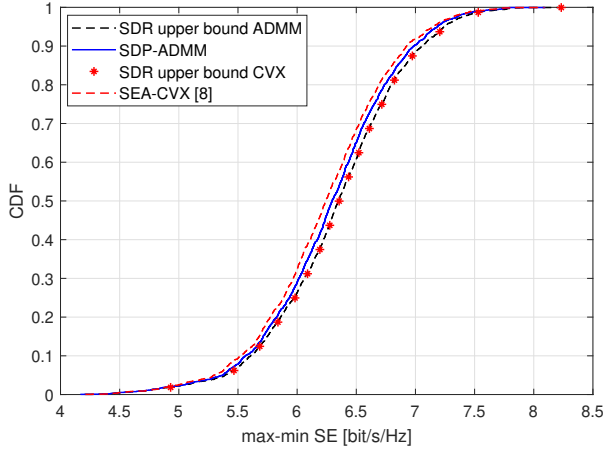


Fig. 3: CDF of the max-min SE for $K = 30$ UEs.

V. CONCLUSIONS

This paper addresses the MMF multicast optimization problem in a massive MIMO system. We develop a fast SDP-ADMM algorithm that can achieve a near-global optimal rank-1 beamforming solution to this NP-hard problem. The proposed solution relies on the SEA to extract a rank-1 solution from the output of the SDP. A novel problem reformulation and penalization scheme is proposed that is specifically tailored for our SDP-ADMM implementation. The distinction of the proposed algorithm is that it is applicable to a wide-class of optimization problems where SDR is utilized and a low-rank solution is desirable. Compared to standard SDP solvers, SDP-ADMM attains similar SE performance with about 10 times reduction in computational time. Numerical results show the robustness of the proposed SDP-ADMM algorithm where it maintains its performance for different simulation settings without the need for ADMM parameter tuning.

REFERENCES

- [1] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.
- [2] M. Zaher, Ö. T. Demir, E. Björnson, and M. Petrova, "Learning-based downlink power allocation in cell-free massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 174–188, 2023.
- [3] G.-W. Hsu, B. Liu, H.-H. Wang, and H.-J. Su, "Joint beamforming for multicell multigroup multicast with per-cell power constraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4044–4058, 2016.
- [4] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE transactions on signal processing*, vol. 54, no. 6, pp. 2239–2251, 2006.
- [5] E. Karipidis, N. D. Sidiropoulos, and Z.-Q. Luo, "Quality of service and max-min fair transmit beamforming to multiple cochannel multicast groups," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1268–1279, 2008.
- [6] L. Zhou, L. Zheng, X. Wang, W. Jiang, and W. Luo, "Coordinated multicell multicast beamforming based on manifold optimization," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1673–1676, 2017.
- [7] M. Sadeghi, L. Sanguinetti, R. Couillet, and C. Yuen, "Reducing the computational complexity of multicasting in large-scale antenna systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2963–2975, 2017.

- [8] M. Zaher, E. Björnson, and M. Petrova, "Near-optimal cell-free beamforming for physical layer multigroup multicasting," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2024.
- [9] M. Dong and Q. Wang, "Multi-group multicast beamforming: Optimal structure and efficient algorithms," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3738–3753, 2020.
- [10] E. Chen and M. Tao, "ADMM-based fast algorithm for multi-group multicast beamforming in large-scale wireless systems," *IEEE Transactions on Communications*, vol. 65, no. 6, pp. 2685–2698, 2017.
- [11] A. Konar and N. D. Sidiropoulos, "Fast approximation algorithms for a class of non-convex QCQP problems using first-order methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3494–3509, 2017.
- [12] C. Zhang, M. Dong, and B. Liang, "Ultra-low-complexity algorithms with structurally optimal multi-group multicast beamforming in large-scale systems," *IEEE Transactions on Signal Processing*, 2023.
- [13] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.
- [14] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <https://cvxr.com/cvx>, Mar. 2014.