Mirror Prox Algorithm for Large-Scale Cell-Free Massive MIMO Uplink Power Control

Muhammad Farooq⬤, *Member, IEEE*, Hien Quoc Ngo⬤, *Senior Member, IEEE*, and Le Nam Tran⬤, *Senior Member, IEEE*

Abstract—We consider the problem of max-min fairness for uplink cell-free massive multiple-input multiple-output (MIMO) subject to per-user power constraints. The standard framework for solving the considered problem is to separately solve two subproblems: the receiver filter coefficient design and the power control problem. While the former has a closed-form solution, the latter has been solved using either second-order methods of high computational complexity or a first-order method that provides an approximate solution. To deal with these drawbacks of the existing methods, we propose a mirror prox based method for the power control problem by equivalently reformulating it as a convex-concave problem and applying the mirror prox algorithm to find a saddle point. The simulation results establish the optimality of the proposed solution and demonstrate that it is more efficient than the known methods. We also conclude that for large-scale cell-free massive MIMO, joint optimization of linear receive combining and power control provides significantly better user fairness than the power control only scheme in which receiver coefficients are fixed to unity.

Index Terms—Cell-free massive MIMO, max-min fairness, power-control, mirror prox method.

I. Introduction

THE recently evolved form of the massive multiple-input multiple-output (MIMO) for the beyond fifth generation (5G) networks is cell-free massive MIMO [1], in which one or several central processing units (CPUs) control a large number of low-cost and low-power access points (APs) to serve a large number of users. The APs and users are distributed in a large coverage area. Owing to the high array gain, multiplexing gain, and macro-diversity gain, cell-free massive MIMO offers many advantages such as high energy efficiency, uniform quality of service, and flexible and cost-effective deployment [2].

To achieve the above benefits, suitable power allocation algorithms need to be employed to control the near/far effect. Since the numbers of APs and users are large, the resulting power control problems consist of many variables. Thus, simple and low-complexity solutions to power control optimization problems for cell-free massive MIMO are of particular interest. In [1], the problem of maximization of the minimum uplink rate (a.k.a. *max-min fairness*) for single-antenna APs and single-antenna users was solved using a

bisection algorithm in combination with linear programming (LP), where maximum-ratio combining (MRC) was considered at each AP for signal detection. In [3], Bashar *et al.* solved this max-min fairness problem by alternatively solving two subproblems: (i) the receiver coefficient design which is in fact a generalized eigenvalue problem (EVP), and (ii) the power control problem which was solved using geometric programming (GP). A similar alternating optimization (AO)-based scheme was used in [4] to solve a mixed quality of service (QoS) problem including the max-min fairness for a set of users and a fixed QoS for the remaining users. Again, the power control subproblem was solved using GP. In [5], Mai *et al.* employed the zero-forcing (ZF) technique to detect symbols from multi-antenna users where the power control problem was solved using the bisection algorithm. A common feature of all aforementioned studies is the use of LP or (more complex) GP by means of off-the-shelf convex solvers, and thus are suitable for small-scale scenarios. In particular, a low complexity first-order accelerated projected gradient (APG) algorithm was applied to the max-min fairness problem following Nesterov's smoothing technique [6] in both down-link [7], [8] and uplink [9] channels. However, this method can only produce an approximate solution whose accuracy is inversely proportional to the number of users, and hence, is not suitable for very large numbers of users. This clearly calls for novel solutions to the power control problem, which are more computationally efficient than LP or GP methods and more accurate than the APG method.

In this letter, we consider the max-min fairness problem subject to the power constraint at each user. We utilize the MRC technique to detect users' signals using the channel estimates. Similar to the known approaches, we also split the considered problem into two subproblems which are alternately solved. In particular, for the power control subproblem, we propose a mirror prox (MP)-based algorithm. To achieve this, we first reformulate the *nonconvex* power control problem into an equivalent convex-concave program and then propose an MP method [10] to find a saddle point. The proposed method only requires the first order information of the objective, and thus, is shown to be much faster than the LP-based and GP-based methods. Moreover, it outperforms other existing first-order methods in terms of achievable rate performance.

Notations: Bold lower and upper case letters represent vectors and matrices. $\mathcal{CN}(\mathbf{0}, \mathbf{K})$ denotes the multivariate circularly symmetric complex Gaussian random distribution with zero mean and co-variance matrix $\mathbf{K}$. $\mathbf{X}^{\mathrm{T}}$ and $\mathbf{X}^{\mathrm{H}}$ stand for the transpose and Hermitian of $\mathbf{X}$, respectively. Notation $\mathbf{e}_i$ denotes the i-th column of the identity matrix. $\nabla f(\mathbf{x})$ represents the gradient of $f(\mathbf{x})$. $\|\cdot\|$ denotes the Euclidean or ℓ_2-norm and $|\cdot|$ is the absolute value of the argument.

$\mathcal{P}_\mathcal{C}(\mathbf{x})$ denotes the projection of $\mathbf{x}$ onto the convex set $\mathcal{C}$, i.e. $\mathcal{P}_\mathcal{C}(\mathbf{x}) = \arg\min_{\mathbf{z} \in \mathcal{C}} \lVert \mathbf{x} - \mathbf{z} \rVert$.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a cell-free massive MIMO uplink scenario where L single-antenna users are served by M APs with K antennas per AP. All APs and users are randomly distributed in a coverage area. The APs are connected to a CPU via high-capacity dedicated links. We denote by ζ_{ml} the large-scale fading coefficient between the l-th user and the m-th AP, and $\mathbf{h}_{ml} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_K)$ is the vector of the small-scale fading coefficient for all antennas at the m-th AP. The channel vector between the l-th user and the m-th AP is modeled as

$$\mathbf{h}_{ml} = \zeta_{ml}^{1/2} \tilde{\mathbf{h}}_{ml}. \tag{1}$$

The length (in symbols) of the uplink training is denoted by τ_p. Let η_p be the transmit power of each pilot symbol and ϕ_l be the pilot sequence of unit norm transmitted from user l to the APs. Let $\hat{\mathbf{h}}_{ml}$ be the minimum mean square error (MMSE) channel estimate of $\mathbf{h}_{ml}$. Then $\hat{\mathbf{h}}_{ml} \sim \mathcal{CN}(\mathbf{0}, g_{ml}\mathbf{I}_K)$, where [1]

$$g_{ml} \triangleq \frac{\eta_p \tau_p \zeta_{ml}^2}{\eta_p \tau_p \sum_{i=1}^{L} \zeta_{mi} \lvert \phi_i^\mathsf{H} \phi_l \rvert^2 + 1}. \tag{2}$$

To formulate the problem of interest, we define $\mathbf{p} = [p_1, \ldots, p_L]$ as the vector of power control coefficients, $\mathbf{q} = [\mathbf{q}_1; \mathbf{q}_2; \ldots; \mathbf{q}_L]$ as the receiver coefficient where $\mathbf{q}_l = [q_{1l}; q_{2l}; \ldots; q_{Ml}]$ associates with the l-th user, and $\bar{p}$ as the maximum transmit power at each individual user. With this setup, an achievable signal-to-interference plus noise ratio (SINR) is given by [1]

$$\gamma_l(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{q}_l^\mathsf{H}\big(\mathbf{g}_{ll}\mathbf{g}_{ll}^\mathsf{H} p_l\big)\mathbf{q}_l}{\mathbf{q}_l^\mathsf{H}\big(\sum_{i \neq l}^{L} \mathbf{g}_{li}\mathbf{g}_{li}^\mathsf{H} p_i + \frac{1}{K}\sum_{i=1}^{L} \bar{\mathbf{G}}_{li}p_i + \frac{1}{K}\tilde{\mathbf{G}}_l\big)\mathbf{q}_l}. \tag{3}$$

In (3), $\bar{\mathbf{G}}_{li} \in \mathbb{R}_+^{M \times M}$ and $\tilde{\mathbf{G}}_l \in \mathbb{R}_+^{M \times M}$ are diagonal matrices with $[\bar{\mathbf{G}}_{li}]_{m,m} = g_{ml}\zeta_{mi}$ and $[\tilde{\mathbf{G}}_l]_{m,m} = g_{ml}$, respectively, and

$$\mathbf{g}_{li} \triangleq \lvert \phi_l^\mathsf{H} \phi_i \rvert \left[g_{1l}\frac{\zeta_{1i}}{\zeta_{1l}}; g_{2l}\frac{\zeta_{2i}}{\zeta_{2l}}; \ldots; g_{Ml}\frac{\zeta_{Mi}}{\zeta_{Ml}} \right]. \tag{4}$$

Note that the achievable rate of the l-th user is given by

$$\mathcal{R}_l(\mathbf{p}, \mathbf{q}) = \log_2\left(1 + \gamma_l(\mathbf{p}, \mathbf{q})\right). \tag{5}$$

Similar to [3], we consider the problem of maximization of the minimum rate, which is mathematically stated as

$$\begin{aligned}
\underset{\mathbf{q}, \mathbf{p}}{\text{maximize}} \quad & \min_{1 \leq l \leq L} \gamma_l(\mathbf{p}, \mathbf{q}) \\
\text{subject to} \quad & \lVert \mathbf{q}_l \rVert = 1, \ l = 1, \ldots, L, \\
& 0 \leq \mathbf{p} \leq \bar{p}. \tag{$\mathcal{P}_1$}
\end{aligned}$$

First, we note that $(\mathcal{P}_1)$ is nonconvex, and thus, finding a globally optimal solution is difficult and not practically useful. The standard framework for solving $(\mathcal{P}_1)$ which is adopted in previous studies such as [1], [3], and [4] is based on AO. In this way, two subproblems arise: the receiver coefficient design and the power control problem. Specifically, the receiver coefficient design is obtained by fixing the power allocation $\mathbf{p}$, which admits a closed-form solution given by [3] and [4]

$$\mathbf{q}_l^* = \frac{\sqrt{p_l}\mathbf{W}_l^{-1}\mathbf{g}_{ll}}{\lVert \sqrt{p_l}\mathbf{W}_l^{-1}\mathbf{g}_{ll} \rVert}, \ l = 1, 2, \ldots, L, \tag{6}$$

where $\mathbf{W}_l = \sum_{i \neq l}^{L} \mathbf{g}_{li}\mathbf{g}_{li}^\mathsf{H} p_i + \frac{1}{K}\sum_{i=1}^{L} \bar{\mathbf{G}}_{li}p_i + \frac{1}{K}\tilde{\mathbf{G}}_l$. After updating the receiver coefficients for all users, we need to find the power coefficients, leading to the following power control problem:

$$\boxed{\underset{0 \leq \mathbf{p} \leq \bar{p}}{\text{maximize}} \quad \min_{1 \leq l \leq L} \gamma_l(\mathbf{p})} \tag{$\mathcal{P}_2$}$$

To solve $(\mathcal{P}_2)$, existing methods [1], [3], [4] involve second order methods, i.e., GP or LP, which require very high complexity, and thus, is not suitable for large-scale cell-free massive MIMO. To overcome the complexity issue, a first-order algorithm was introduced in [9] but it cannot yield a solution of high accuracy for a large number of users. In the next section, we propose a more efficient method for solving (7) based on the MP framework.

III. PROPOSED SOLUTION FOR POWER CONTROL PROBLEM

A. Equivalent Convex-Concave Reformulation of (7)

The proposed method is developed based on an equivalent convex-concave reformulation of $(\mathcal{P}_2)$. To this end, it is easy to see that we can equivalently rewrite $(\mathcal{P}_2)$ as

$$\underset{\mathbf{p}}{\text{minimize}} \quad \max_{1 \leq l \leq L} \gamma_l^{-1}(\mathbf{p}), \tag{7}$$

where

$$\gamma_l^{-1}(\mathbf{p}) = p_l^{-1}\left(\sum_{i \neq l}^{L} a_{li}p_i + \sum_{i=1}^{L} b_{li}p_i + c_l\right), \tag{8}$$

where $a_{li} = \frac{\mathbf{q}_l^\mathsf{H}\mathbf{g}_{li}\mathbf{g}_{li}^\mathsf{H}\mathbf{q}_l}{\mathbf{q}_l^\mathsf{H}\mathbf{g}_{ll}\mathbf{g}_{ll}^\mathsf{H}\mathbf{q}_l}$, $b_{ki} = \frac{(1/K)\mathbf{q}_l^\mathsf{H}\bar{\mathbf{G}}_{li}\mathbf{q}_l}{\mathbf{q}_l^\mathsf{H}\mathbf{g}_{ll}\mathbf{g}_{ll}^\mathsf{H}\mathbf{q}_l}$, and $c_k = \frac{(1/K)\mathbf{q}_l^\mathsf{H}\tilde{\mathbf{G}}_l\mathbf{q}_l}{\mathbf{q}_l^\mathsf{H}\mathbf{g}_{ll}\mathbf{g}_{ll}^\mathsf{H}\mathbf{q}_l}$. We remark that $\gamma_l^{-1}(\mathbf{p})$ is non-convex but can be converted to a single convex form as for geometric programming by making a change of variables $\theta_i = \log p_i$ [3], [4]. In this letter, *we propose the change of variables $\theta_i = \log \omega p_i$ or $p_i = \frac{1}{\omega}e^{\theta_i}$* and reformulate (8) as

$$\begin{aligned}
f_l(\boldsymbol{\theta}) &\triangleq \gamma_l^{-1} = \omega e^{-\theta_l}\left(\sum_{i \neq l}^{L} \frac{a_{li}}{\omega}e^{\theta_i} + \sum_{i=1}^{L} \frac{b_{li}}{\omega}e^{\theta_i} + c_l\right) \\
&= \sum_{i \neq l}^{L} a_{li}e^{(\mathbf{e}_i - \mathbf{e}_l)^\mathsf{T}\boldsymbol{\theta}} + \sum_{i=1}^{L} b_{li}e^{(\mathbf{e}_i - \mathbf{e}_l)^\mathsf{T}\boldsymbol{\theta}} \\
&\quad + \bar{c}_l e^{-\mathbf{e}_l^\mathsf{T}\boldsymbol{\theta}}, \tag{9}
\end{aligned}$$

where $\bar{c}_l = c_l\omega$ and $\boldsymbol{\theta} = [\theta_1; \theta_2; \ldots; \theta_L]$. Now (7) is equivalent to

$$\underset{\boldsymbol{\theta} \in \Theta}{\text{minimize}} \quad \left[f(\boldsymbol{\theta}) \triangleq \max_{1 \leq l \leq L} f_l(\boldsymbol{\theta})\right] \tag{$\mathcal{P}_3$},$$

where $\Theta \triangleq \{\boldsymbol{\theta} | \boldsymbol{\theta} \leq \bar{\theta}\}$ and $\bar{\theta} = \log(\omega\bar{p})$. Note that $f_l(\boldsymbol{\theta})$ is convex and so is $(\mathcal{P}_3)$. In the context of projected gradient methods, the introduction of ω effectively scales the gradient of $f_l(\boldsymbol{\theta})$. We shall numerically demonstrate that a proper value of ω can speed up the convergence of the proposed algorithm as shown in the next section.

Although $(\mathcal{P}_3)$ is convex, solving it efficiently is still challenging since its objective is nonsmooth due to the max operator. A solution to deal with the nonsmoothness of $(\mathcal{P}_3)$ is to adopt a smoothing technique as done in [9] but such a method can only produce an approximate solution. To derive an efficient solution, we recall the following equality

$$\max_{1 \leq l \leq L} f_l(\boldsymbol{\theta}) = \max_{\boldsymbol{\lambda} \in \Delta} \sum_{l=1}^{L} \lambda_l f_l(\boldsymbol{\theta}), \tag{10}$$

where $\boldsymbol{\lambda} = [\lambda_1; \lambda_2; \ldots; \lambda_L] \in \mathbb{R}^L$ and $\Delta \triangleq \{\boldsymbol{\lambda} \mid \mathbf{1}^\mathsf{T}\boldsymbol{\lambda} = 1; \boldsymbol{\lambda} \geq 0\}$ is the standard simplex. Thus, $(\mathcal{P}_3)$ can

be equivalently rewritten as

$$\boxed{\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}}\ \underset{\boldsymbol{\lambda}\in\Delta}{\text{maximize}}\ \phi(\boldsymbol{\theta},\boldsymbol{\lambda}),} \qquad (\mathcal{P}_4)$$

where $\phi(\boldsymbol{\theta},\boldsymbol{\lambda}) = \sum_{l=1}^{L}\lambda_l f_l(\boldsymbol{\theta})$. Note that $\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ is convex w.r.t $\boldsymbol{\theta}$ for a given $\boldsymbol{\lambda}$ and concave (in fact linear) w.r.t $\boldsymbol{\lambda}$ for a given $\boldsymbol{\theta}$.

B. Proposed Mirror Prox Algorithm for Solving $(\mathcal{P}_4)$

It is now clear that we can find a saddle point of $(\mathcal{P}_4)$ to solve the power control problem, which indeed motivates the application of the MP algorithm in this letter. To explain the idea of the MP algorithm, let $\mathbf{z} = (\boldsymbol{\theta},\boldsymbol{\lambda})$ and $\mathbf{F}(\mathbf{z}) = [\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda}), -\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})]$ which is a monotone operator associated with $(\mathcal{P}_4)$, i.e., $\langle\mathbf{F}(\mathbf{z}) - \mathbf{F}(\mathbf{z}'),\mathbf{z} - \mathbf{z}'\rangle \geq 0$. The idea of the MP algorithm is to find a point $\mathbf{z}^*$ such that $\langle\mathbf{F}(\mathbf{z}),\mathbf{z} - \mathbf{z}^*\rangle \geq 0$, which is indeed a saddle point of $(\mathcal{P}_4)$. The proposed method is a special case of the MP algorithm with the Euclidean setup. The motivation is that the Euclidean projections onto Θ and Δ can be done in closed-form. Under the Euclidean setup, the distance between $\mathbf{z} = (\boldsymbol{\theta},\boldsymbol{\lambda})$ and $\mathbf{z}' = (\boldsymbol{\theta}',\boldsymbol{\lambda}')$, denoted by $D(\mathbf{z},\mathbf{z}')$, is defined as

$$D(\mathbf{z},\mathbf{z}') = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2 + \frac{1}{2}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|^2. \qquad (11)$$

Note that $D(\mathbf{z},\mathbf{z}')$ is the sum of individual Euclidean distances.

The step-by-step description of the proposed method for solving $(\mathcal{P}_4)$ is as follows. Let $\mathbf{z}^n = (\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n)$ be the n-th iterate. To obtain the next iterate, we perform the following two *proximal mappings*

$$\hat{\mathbf{z}}^n = \arg\min_{\mathbf{z}\in\Theta\times\Delta}\langle\mu^n\mathbf{F}(\mathbf{z}^n),\mathbf{z}\rangle + D(\mathbf{z}^n,\mathbf{z}), \qquad (12a)$$

$$\mathbf{z}^{n+1} = \arg\min_{\mathbf{z}\in\Theta\times\Delta}\langle\mu^n\mathbf{F}(\hat{\mathbf{z}}^n),\mathbf{z}\rangle + D(\mathbf{z}^n,\mathbf{z}), \qquad (12b)$$

where $\mu^n > 0$ is a step size which needs to be chosen properly to guarantee the convergence. We remark that the two steps above are similar to a classical gradient-type method since the vector field $\mathbf{F}(\mathbf{z})$ acts as a descent direction. From the current iterate $\mathbf{z}^n$, we first move along the gradient of individual variables to obtain an intermediate point $\hat{\mathbf{z}}^n = (\hat{\boldsymbol{\theta}}^n,\hat{\boldsymbol{\lambda}}^n)$ as shown in (12a). The main difference and in fact the novel idea of the MP algorithm are as follows. To obtain the next iterate from $\mathbf{z}^n$, we *do not* use the gradients at $\mathbf{z}^n$. Instead, $\mathbf{z}^{n+1}$ is obtained using the gradient at the intermediate point $\hat{\mathbf{z}}^n$ as given in (12b).

1) Intuition: To gain further insights into the proposed MP algorithm, we note that (12a) implies

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\Theta}\langle\mu\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n),\boldsymbol{\theta}\rangle + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^n\|^2$$
$$= \mathcal{P}_\Theta(\boldsymbol{\theta}^n - \mu^n\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n)) \qquad (13)$$

and, similarly,

$$\hat{\boldsymbol{\lambda}}^n = \mathcal{P}_\Delta(\boldsymbol{\lambda}^n + \mu^n\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n)). \qquad (14)$$

An important remark is in order. Since $\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ is convex in $\boldsymbol{\theta}$ and concave in $\boldsymbol{\lambda}$, $-\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ and $\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ are the descent and ascent directions, respectively. In this regard, the MP algorithm is similar to the gradient descent and ascent algorithm. That is, the MP algorithm simultaneously minimizes and maximizes $\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ to reach a saddle point.

However, the gradients at the intermediate point $(\hat{\boldsymbol{\theta}}^n,\hat{\boldsymbol{\lambda}}^n)$ are used to obtain $(\boldsymbol{\theta}^{n+1},\boldsymbol{\lambda}^{n+1})$ from $(\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n)$.

To obtain a convergent algorithm, the step size in each iteration n needs to satisfy the following condition

$$\delta^n \triangleq \mu^n\nabla_{\boldsymbol{\theta}}\phi(\hat{\boldsymbol{\theta}}^n,\hat{\boldsymbol{\lambda}}^n)(\hat{\boldsymbol{\theta}}^n - \boldsymbol{\theta}^{n+1})$$
$$- \mu^n\nabla_{\boldsymbol{\lambda}}\phi(\hat{\boldsymbol{\theta}}^n,\hat{\boldsymbol{\lambda}}^n)(\hat{\boldsymbol{\lambda}}^n - \boldsymbol{\lambda}^{n+1})$$
$$- \frac{1}{2}\|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^n\|^2 - \frac{1}{2}\|\boldsymbol{\lambda}^{n+1} - \boldsymbol{\lambda}^n\|^2 \leq 0. \qquad (15)$$

Note that the above inequality is true if $\mu^n \leq \mathcal{L}^{-1}$, where $\mathcal{L}$ is the Lipschitz constant of $\mathbf{F}(\mathbf{z})$. In practice, we find μ^n by a back tracking line search. In summary, the proposed MP algorithm for solving $(\mathcal{P}_4)$ is outlined in Algorithm 1.

Algorithm 1 MP Algorithm for Solving $(\mathcal{P}_4)$

1: Initialization: $\mu^0 > 0$, $\rho \in (0,1)$
2: $\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^{\text{initial}}$, $\boldsymbol{\lambda}^1 \leftarrow \boldsymbol{\lambda}^{\text{initial}}$
3: **for** $n = 1, 2, \ldots$ **do**
4:    $\mu^n = \mu^{n-1}/\rho$
5:    **repeat**
6:      $\mu^n \leftarrow \mu^{n-1} \times \rho$
7:      $\hat{\boldsymbol{\theta}}^n \leftarrow \mathcal{P}_\Theta(\boldsymbol{\theta}^n - \mu^n\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n))$
8:      $\hat{\boldsymbol{\lambda}}^n \leftarrow \mathcal{P}_\Delta(\boldsymbol{\lambda}^n + \mu^n\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta}^n,\boldsymbol{\lambda}^n))$
9:      $\boldsymbol{\theta}^{n+1} \leftarrow \mathcal{P}_\Theta(\boldsymbol{\theta}^n - \mu^n\nabla_{\boldsymbol{\theta}}\phi(\hat{\boldsymbol{\theta}}^n,\hat{\boldsymbol{\lambda}}^n))$
10:      $\boldsymbol{\lambda}^{n+1} \leftarrow \mathcal{P}_\Delta(\boldsymbol{\lambda}^n + \mu^n\nabla_{\boldsymbol{\lambda}}\phi(\hat{\boldsymbol{\theta}}^n,\hat{\boldsymbol{\lambda}}^n))$
11:    **until** $\delta^n \leq 0$
12: **end for**
13: $\boldsymbol{\theta}^* \leftarrow \frac{\sum_{i=1}^{n}\mu^i\boldsymbol{\theta}^i}{\sum_{i=1}^{n}\mu^i}$, $\boldsymbol{\lambda}^* \leftarrow \frac{\sum_{i=1}^{n}\mu^i\boldsymbol{\lambda}^i}{\sum_{i=1}^{n}\mu^i}$

---

2) Computation of Gradients: To implement Algorithm 1, we need to calculate $\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ and $\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ which are given in closed form as

$$\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda}) = \sum_{l=1}^{L}\lambda_l\nabla_{\boldsymbol{\theta}}f_l(\boldsymbol{\theta})$$
$$\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda}) = [f_1(\boldsymbol{\theta}); f_2(\boldsymbol{\theta}); \ldots; f_L(\boldsymbol{\theta})], \qquad (16)$$

where, from (9), we have

$$\nabla_{\boldsymbol{\theta}}f_l(\boldsymbol{\theta}) = \sum_{i\neq l}^{L}a_{li}e^{(\mathbf{e}_i - \mathbf{e}_l)^{\mathsf{T}}\boldsymbol{\theta}}(\mathbf{e}_i - \mathbf{e}_l)$$
$$+ \sum_{i=1}^{L}b_{li}e^{(\mathbf{e}_i - \mathbf{e}_l)^{\mathsf{T}}\boldsymbol{\theta}}(\mathbf{e}_i - \mathbf{e}_l) - \bar{c}_l e^{-\mathbf{e}_l^{\mathsf{T}}\boldsymbol{\theta}}\mathbf{e}_l. \qquad (17)$$

Remark 1: As mentioned above, the proposed change of variable $\theta_i = \log\omega p_i$ is equivalent to scaling the gradient of $f_l(\boldsymbol{\theta})$, compared to the standard change of variable $\theta_i = \log p_i$. This scaling effect is reflected by the term $\bar{c}_l$ in the above equation. The main motivation for introducing ω is to balance the gradients $\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$ and $\nabla_{\boldsymbol{\lambda}}\phi(\boldsymbol{\theta},\boldsymbol{\lambda})$, which can accelerate the convergence of Algorithm 1 as numerically illustrated in the next section.

3) Projections Onto Θ and Δ: It is easy to see that $P_\Theta(\mathbf{x})$ admits a closed form solution as follows

$$P_\Theta(\mathbf{x}) = \begin{cases} x_k, & x_k \leq \bar{\theta} \\ \bar{\theta}, & x_k > \bar{\theta} \end{cases}, k = 1, \ldots, L. \qquad (18)$$

Also, the projection onto a standard simplex is given by

$$P_\Delta(\boldsymbol{\lambda}) = [\boldsymbol{\lambda} - \beta]_+, \qquad (19)$$

where β is the solution to the equation

$$\sum_{k=1}^{L}[\lambda_k - \beta]_+ = 1, \qquad (20)$$

which can be found by bisection. In summary, to solve $(\mathcal{P}_1)$ we keep alternately computing (6) and running Algorithm 1 until convergence.

C. Convergence Analysis

We first provide the convergence analysis of Algorithm 1. Let f^* be the optimal objective of $(\mathcal{P}_3)$ and $\tilde{\boldsymbol{\theta}}^n = \left(\sum_{n=1}^{N} \mu^n\right)^{-1} \sum_{n=1}^{N} \mu^n \boldsymbol{\theta}^n$ be the obtained solution after N iterations. Note that $\tilde{\boldsymbol{\theta}}^n$ is the weighted average of the iterates $\boldsymbol{\theta}^n$ up to iteration N. Then it is shown that $f(\tilde{\boldsymbol{\theta}}^n) - f^* \leq \frac{1}{\sum_{n=1}^{N} \mu^n}\left(\Omega + \sum_{n=1}^{N} \delta^n\right)$, where Ω is a constant that depends on the distance generating function. By the line search procedure in Algorithm 1, we have that $\delta^n \leq 0$ and $\mu^n \geq \mathcal{L}^{-1}$, and thus, $f(\tilde{\boldsymbol{\theta}}^n) - f^* \leq \Omega L/N$. That is, Algorithm 1 can achieve a $\mathcal{O}(1/N)$-rate of convergence. The proof follows closely the steps in [11, Proposition 6.1], and thus, is omitted here for brevity.

D. Complexity Analysis and Comparison

We now present the per-iteration complexity analysis of Algorithm 1. It is easy to see that L multiplications are required to compute $f_l(\boldsymbol{\theta}^n)$. Therefore, the complexity of finding the objective is $\mathcal{O}(L^2)$. Similarly, we can find that the complexity of $\nabla f_l(\boldsymbol{\theta})$ is also $\mathcal{O}(L^2)$. It is clear from (18) that $P_\Theta(\mathbf{x})$ has complexity of $\mathcal{O}(L)$. For $P_\Delta(\boldsymbol{\lambda})$, by sorting the elements of $\boldsymbol{\lambda}$ in the ascending order, the complexity of the projection onto a simplex from [12, Th. 2.2] is $\mathcal{O}(L)$ complexity in the worst case. In summary, the per-iteration complexity of the proposed algorithm for solving $(\mathcal{P}_2)$ is $\mathcal{O}(L^2)$. If a bisection search is used with LP as done in [1], the *worst-case* per-iteration complexity is $\mathcal{O}(L^{3.5})$ [13, eq. (8.1.6)]. The complexity of the GP-based methods in [4] and [3] is $\mathcal{O}(L^{4.5})$ [13, Sec. 6.3.1].

IV. NUMERICAL RESULTS

We evaluate the performance of our proposed method in terms of the achievable rate and the run time. We randomly distribute APs and users over a $D \times D$ km^2. Channel coefficients are generated using (1), in which the large-scale fading coefficient between the m-th AP and the l-th user is modeled as $\zeta_{ml} = \mathrm{PL}_{ml} z_{ml}$, where PL_{ml} is the corresponding path loss, and z_{ml} represents the log-normal shadowing between the m-th AP and the l-th user with mean zero and standard deviation σ_{sh}, respectively. In this letter, we adopt the three-slope path loss model and model parameters as in [8]. Noise figure is set to 9 dB. We assume pilot sequences to be pairwisely orthogonal to avoid the effect of pilot contamination. The lengths of the coherence interval and the uplink training phase are set to $\tau_c = 200, \tau_p = 20$, respectively. If not otherwise mentioned, we set $\eta_p = 0.2$ W and $\bar{p} = 0.2$ W. The number of antennas at each AP is $K = 1$.

In Fig. 1, we show the convergence of Algorithm 1. For each channel realization, we run Algorithm 1 for a fixed $\mathbf{q}$ and plot the average achievable rate over 100 randomly generated channel realizations. We compare Algorithm 1 with two other iterative baseline schemes: the gradient descent accent (GDA) method [14], [15] and the APG method combined with the smoothing technique [9]. We also benchmark Algorithm 1 with
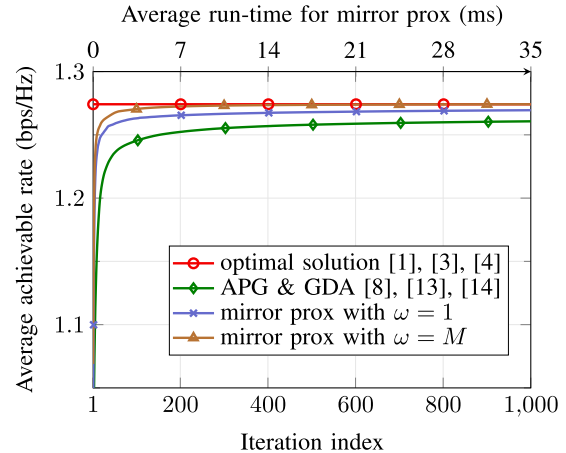

Fig. 1. Convergence performance of Algorithm 1, GDA, APG averaged over 100 channels. The simulation parameters taken are $M = 150, L = 50$, and $D = 1$.
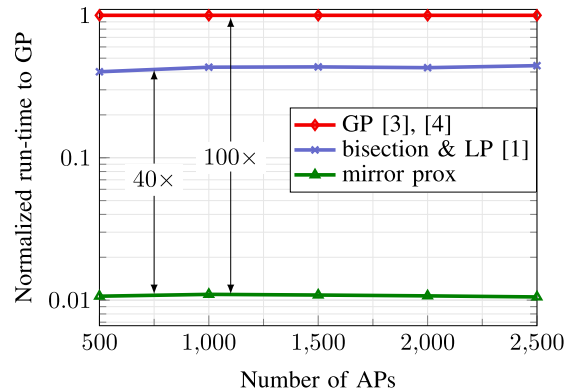

Fig. 2. Comparison of run-time (in seconds) between Algorithm 1 and the bisection-based method for $L = 40$.

the optimal solution obtained by either the bisection method with LP [1] or the GP in [3] and [4].

We can see that Algorithm 1 based on the MP method has better objective than GDA and APG. Also, Algorithm 1 takes lesser number of iterations. Thus, GDA and APG [9] are not suitable for the applications where a fast convergence rate is required. Note that APG presented in [9] is essentially an approximate solution. On the other hand, *Algorithm 1 is an exact method and thus can reach the optimal solution at convergence as clearly seen in Fig. 1*. Note that we also demonstrate the impact of introducing ω when reformulating $(\mathcal{P}_2)$ into $(\mathcal{P}_4)$. As can be observed clearly, a proper value of ω can indeed speed up the convergence of Algorithm 1 very significantly. By extensive simulation settings, we find that $\omega = M$ yields a good convergence rate for Algorithm 1 overall, which is shown in Fig. 1. Further, we remark that the proposed algorithm converges within 500 iterations. Also, the average run-time per iteration is very small, as shown in Fig. 1.

The main advantage of Algorithm 1 over second-order methods in [1], [3], and [4] is that each iteration of Algorithm 1 is very memory efficient and computationally cheap, and hence, can be executed very fast. To demonstrate this, we compare the run-time of Algorithm 1 with the bisection method with LP in [1], *both normalized to the run-time of the GP approach* in [4] and [3] for solving $(\mathcal{P}_2)$ in Fig. 2.
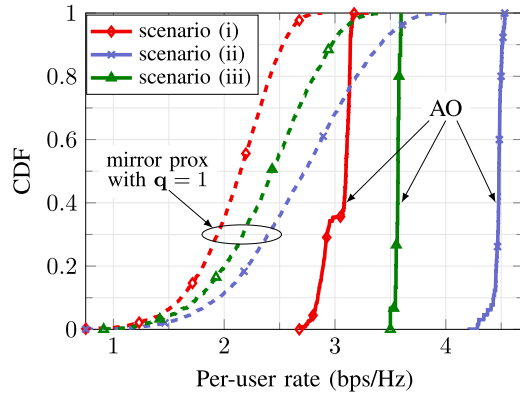
Fig. 3.   CDF of per-user achievable rate for three scenarios: (i) $M = 500$, $L = 50, D = 1$, (ii) $M = 2000$, $L = 100, D = 2$, and (iii) $M = 2000$, $L = 50, D = 2$. Solid lines indicate the CDF of the per-user rate using both power and receiver filtering control, and dashed lines represent the CDF of the per-user rate using power control only.

This shows the factor at which a particular method is faster than the GP method with normalized run-time of 1. We run the codes on a 64-bit Windows operating system with 16 GB RAM and Intel CORE i7, 3.7 GHz. All iterative methods are terminated when the difference of the objective for the latest two iterations is less than 10^{-4}. We observe that solving $(\mathcal{P}_2)$ using a bisection search with LP is faster than converting it to a single GP, which is then solved by a dedicated GP solver. Most importantly, Algorithm 1 is 40 times faster than the LP method and approximately 100 times faster than the GP approach as shown in Fig. 2.

In the next experiment, we investigate cell-free massive MIMO for large-scale scenarios. Instead of solving a generalized eigenvalue problem to find $\mathbf{q}_l^*$ as suggested in [3] and [4] which incurs a complexity of $O(M^3)$, we can reduce this complexity to $O(M^2)$ using the method in [9, Appendix]. This complexity reduction and the low complexity of Algorithm 1 indeed allow us to investigate the performance of uplink large-scale cell-free massive MIMO, which has never been reported previously. In particular, we plot the cumulative distribution function (CDF) of the per-user achievable rate for 100 channel realizations in Fig. 3. Three large-scale scenarios detailed in the caption of Fig. 3 are considered. Note that the AP-density, defined as the number of APs per km^2 of area, is the same for all three scenarios. In particular, we compare the CDF of the per-user rate where both receiver filtering and power control are considered (i.e. alternately computing (6) and running Algorithm 1 until convergence, referred to as AO in the figure) to that where the power control only scheme (i.e. Algorithm 1) is adopted in which AP-weighting coefficients are all set to one.

It can be seen from Fig. 3 that the system performance improves when the number of APs increases (even the AP-density is fixed). Let us discuss scenarios (i) and (iii) first. Although the AP density is the same for both scenarios, the user-density (defined as the number of users per km^2 of area) for scenario (iii) is much smaller than that for scenario (i), and hence, users in scenario (iii) suffer less interference than users in scenario (i) do. Thus, users in scenario (iii) have higher rates than those in scenario (i). If we increase the number of users in scenario (iii) from 50 to 100 as in scenario (ii), then the user-density and thereby inter-user interference increase,

which decreases the per-user rate accordingly. Furthermore, it is interesting to note that for scenarios (ii) and (iii), considering both power and receiver filtering control can deliver more universally good services to the users than the power control only scheme. However, this comes at the cost of extra complexity imposed by the computation of the receiver filter coefficients. Therefore, it is important to consider both the receiver coefficient design and power control problems for large-scale scenarios.

V. CONCLUSION

In this letter, we have considered the max-min fairness problem for uplink cell-free massive MIMO. As in previous studies, we have decomposed the problem into the receiver filtering design problem and the power control problem. In particular, we have proposed a novel power control scheme based on the MP method. The numerical results have demonstrated that the proposed method is superior to other existing methods such as LP-based, GDA-based, and APG-based methods. More specifically, the proposed method achieves the same objective as these conventional schemes but in much lesser time. The proposed scheme has enabled us to investigate the performance of cell-free massive MIMO in large-scale scenarios. We have also numerically shown that the proposed solution provides better fairness among the users compared to the power control only scheme where AP-weighting coefficients are all set to one.

REFERENCES

[1] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[2] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99878–99888, 2019.

[3] M. Bashar, K. Cumanan, A. G. Burr, M. Debbah, and H. Q. Ngo, "On the uplink max-min SINR of cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2021–2036, Apr. 2019.

[4] M. Bashar, K. Cumanan, A. G. Burr, H. Q. Ngo, and H. V. Poor, "Mixed quality of service in cell-free massive MIMO," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1494–1497, Jul. 2018.

[5] T. C. Mai, H. Q. Ngo, and T. Q. Duong, "Uplink spectral efficiency of cell-free massive MIMO with multi-antenna users," in *Proc. 3rd Int. Conf. Recent Adv. Signal Process., Telecommun. Comput. (SigTelCom)*, Mar. 2019, pp. 126–129.

[6] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

[7] M. Farooq, H. Q. Ngo, and L. N. Tran, "Accelerated projected gradient method for the optimization of cell-free massive MIMO downlink," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.* Aug./Sep. 2020, pp. 1–6.

[8] M. Farooq, H. Q. Ngo, E.-K. Hong, and L.-N. Tran, "Utility maximization for large-scale cell-free massive MIMO downlink," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 7050–7062, Oct. 2021.

[9] M. Farooq, H. Q. Ngo, and L. N. Tran, "A low-complexity approach for max-min fairness in uplink cell-free massive MIMO," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–6.

[10] A. Nemirovski, "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM J. Optim.*, vol. 15, no. 1, pp. 229–251, 2004.

[11] A. Juditsky and A. Nemirovski, "First order methods for nonsmooth convex large-scale optimization, II: Utilizing problems structure," *Optim. Mach. Learn.*, vol. 30, no. 9, pp. 149–183, Mar. 2011.

[12] Y. Chen and X. Ye, "Projection onto a simplex," Univ. Florida Digit. Collections, Gainesville, FL, USA, Tech. Rep., Feb. 2011.

[13] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: SIAM, 1994.

[14] G. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, no. 4, pp. 747–756, 1976.

[15] A. Nedić and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J. Optim. Theory Appl.*, vol. 142, no. 1, pp. 205–228, Mar. 2009.