LPA-SD: An Efficient First-Order Method for Single-Group Multicast Beamforming

Rujun Jiang
School of Data Science,
Fudan University, China
Email: rjjiang@fudan.edu.cn

Huikang Liu
Dept. of Sys. Engg. & Engg. Mgmt.,
The Chinese Univ. of Hong Kong,
Hong Kong SAR, China
Email: hkliu@se.cuhk.edu.hk

Anthony Man-Cho So
Dept. of Sys. Engg. & Engg. Mgmt.,
The Chinese Univ. of Hong Kong,
Hong Kong SAR, China
Email: manchoso@se.cuhk.edu.hk

***Abstract*—In this work, we develop a new first-order method called linear programming-assisted subgradient descent (LPA-SD) for solving the single-group multicast beamforming (SGMB) problem. As the SGMB problem is NP-hard, most existing methods focus on finding a good sub-optimal solution. Our objective is to maximize the minimum signal-to-noise ratio (SNR) subject to a given transmit power. We then propose a first-order descent algorithm on the unit sphere to solve the SGMB problem efficiently. We prove that our algorithm converges to a critical point. Our numerical results further demonstrate our algorithm outperforms the state-of-the-art method for the SGMB problem with a much faster computational speed and a better SNR, especially when the number of users or antennas is large.**

I. Introduction

In this paper, we consider the single-group multicast beamforming (SGMB) problem, where all users receive the same information stream from the transmitter (Tx), and the maximum common date rate is determined by the minimum received signal-to-noise ratio (SNR). Hence, the objective is to maximize the minimum received SNR subject to the limited transmit power (max-min-fair multicast beamforming). Another equivalent form, from an optimization point of view, is to minimize the transmit power subject to appropriate quality-of-service (QoS) guarantees formulated in terms of the minimum SNR for each user (QoS multicast beamforming) [1].

In general, the SGMB problem is a non-convex quadratically constrained quadratic programming (QCQP) problem and is known to be NP-hard [1]. A prevailing method to tackle this problem is to approximate it using the semidefinite relaxation (SDR) technique [2]. The authors in [1] used a Gaussian randomization process to obtain a good sub-optimal solution. However, the SDR-based algorithms yield solutions whose performance do not scale well with the number of users and are not computational efficient as they require lifting the problem to a higher-dimensional space. To improve the performance of the SDR method, Wu et al. [3] developed a rank-2 transmit beamformed Alamouti space-time code scheme. However, their method still needs to solve an SDP problem, and the high computation cost prevents its applications in solving large scale problems.

Tran et al. [4] proposed a more efficient method to tackle the SGMB problem using successive linear approximation (SLA). The SLA algorithm iteratively solves a series of convex problems obtained by linearly approximating each non-convex constraint. Numerical results in [4] demonstrates that SLA methods enjoy better performance than the SDR methods. However, the SLA still has a relative large CPU time and thus is not suitable for solving high-dimensional problems that arises in, e.g., future-generation wireless broadband systems [5], [6], [7]. Recently, Konar and Sidiropoulos have proposed in [8], [9] several first-order algorithms to solve the SLA sub-problem, which yields a speed up in CPU time.

To further reduce the computation cost, an adaptive algorithm focusing on the max-min fair formulation was recently developed by Gopalakrishnan and Sidiropoulos [10], where each update takes a step in the direction of a weighted linear combination of the SNR-gradient vectors of all the users. They further proposed a method by combining their algorithm and the SLA method to achieve lower CPU time than and similar SNR values as SLA.

In this paper, we design a new subgradient method, termed the *linear programming-assisted subgradient descent* (LPA-SD), based on formulating the SGMB problem as a non-smooth minimization problem over a smooth manifold; cf. [9], [11]. Our method is motivated by the first order method for an unconstrained minimax problem in [12]. Our algorithm can also be seen as an improvement on the algorithm of [10], in that the weighted SNR-gradient is set to guarantee an increment of the minimum SNR at each step and each step size is chosen by an Armijo-type rule. An advantage of our algorithm is that it achieves a slightly better SNR than the SLA using much less CPU time. Unlike the traditional subgradient method, we define a relative active set in each iteration and use a linear program (LP) to choose a suitable convex combination of the SNR-gradients of the functions defined by the relative active set as the descent direction. Combined with an Armjio-type step size rule, we can ensure that the function value decreases in each iteration. Moreover, we demonstrate that our algorithm converges to a critical point. Our extensive numerical results show that the proposed method outperforms the SLA method with a large improvement in CPU time and some improvement in the SNR values.

II. Problem Formulation

We consider a single-group multicast cell consisting of a Tx with N antennas and K single-antenna receivers (Rx's). The Tx transmits the common information-bearing signal $x \in \mathbb{C}$ to all K Rx's using a unit-norm beamforming vector $\mathbf{w} \in \mathbb{C}^N$. The corresponding received signal at the kth Rx is

$$\mathbf{y}_k = \mathbf{w}^H \mathbf{h}_k x + \mathbf{z}_k, \quad \forall k = 1, \ldots, K,$$

where $\mathbf{h}_k \in \mathbb{C}^N$ is the channel between the Tx and the kth Rx; $\mathbf{z}_k \in \mathbb{C}$ is a complex circularly symmetric Gaussian noise with mean zero and variance σ_k^2, assumed to be independent of x and $\mathbf{h}_k$. The received SNR at the kth Rx is then given by $|\mathbf{h}_k^H \mathbf{w}|^2/\sigma_k^2$. The objective of the Tx is to design the beamforming vector $\mathbf{w}$ so that the minimum SNR among the users is maximized; i.e.,

$$\max_{\mathbf{w}\in\mathbb{C}^N} \min_{k\in\{1,\ldots,K\}} \mathbf{w}^H \mathbf{R}_k \mathbf{w} \quad \text{s.t.} \quad \|\mathbf{w}\|_2 = 1, \qquad \text{(P)}$$

where $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H/\sigma_k^2$ [1], [10], [8], [9]. To facilitate our algorithmic development and theoretical analysis, we convert the above maximin problem into an equivalent minimax problem with real variables as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^{2N}} \max_{k\in\{1,\ldots,K\}} f_k(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{x}\|_2 = 1, \qquad (\overline{\text{P}})$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \in \mathbb{R}^{2N}$ with $\mathbf{w} = \mathbf{y} + i\mathbf{z}$; $f_k(\mathbf{x}) = \mathbf{x}^T \mathbf{A}_k \mathbf{x}$ with $\mathbf{A}_k = -\begin{bmatrix} \mathbf{R}_k^1 & -\mathbf{R}_k^2 \\ (\mathbf{R}_k^2)^T & \mathbf{R}_k^1 \end{bmatrix}$ and $\mathbf{R}_k = \mathbf{R}_k^1 + i\mathbf{R}_k^2$; $i = \sqrt{-1}$ is the imaginary unit. Upon defining $F(\mathbf{x}) = \max_{k\in\{1,\ldots,K\}} f_k(\mathbf{x})$, the KKT necessary condition of Problem ($\overline{\text{P}}$) can be expressed as [13, Theorem 3.5]

$$0 \in \partial F(\mathbf{x}^*) + \nu \mathbf{x}^*, \quad \|\mathbf{x}^*\|_2 = 1, \quad \nu \in \mathbb{R}. \qquad (1)$$

We call any point $\mathbf{x}^* \in \mathbb{R}^{2N}$ that satisfies the KKT condition (1) for some $\nu \in \mathbb{R}$ a *critical point* of Problem ($\overline{\text{P}}$). Since Problem ($\overline{\text{P}}$) is NP-hard in general [1], we shall focus on finding one of its critical points.

III. LP-Assisted Subgradient Descent (LPA-SD)

Note that Problem ($\overline{\text{P}}$) can be regarded as a manifold optimization problem, as it involves minimizing the non-smooth function $F : \mathbb{R}^{2N} \to \mathbb{R}$ over the smooth manifold $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^{2N} : \|\mathbf{x}\|_2 = 1\}$. This motivates us to utilize manifold optimization techniques [11] to design an algorithm for finding an (approximate) critical point of Problem ($\overline{\text{P}}$). Towards that end, let us first give an alternative characterization of the critical points of Problem ($\overline{\text{P}}$). Given $\mathbf{x} \in \mathbb{R}^{2N}$, let $I = I(\mathbf{x}) = \{k \in \{1, \ldots, K\} : f_k(\mathbf{x}) = F(\mathbf{x})\}$ be the index set of active functions and $\text{grad} f_k(\mathbf{x}) = (\mathbf{I} - \mathbf{x}\mathbf{x}^T)\nabla f_k(\mathbf{x})$ be the projection of $\nabla f_k(\mathbf{x})$ onto the tangent space of $\mathbb{S}$ at $\mathbf{x}$. Furthermore, let $\Delta_I = \{\mathbf{x} \in \mathbb{R}^{|I|} : \mathbf{e}^T\mathbf{x} = 1,\ \mathbf{x} \geq \mathbf{0}\}$ be an $|I|$-dimensional simplex, where $|I|$ is the cardinality of I and $\mathbf{e}$ is the all-one vector. We have the following theorem:

Theorem 1. *A point $\mathbf{x}^* \in \mathbb{R}^{2N}$ is a critical point of Problem ($\overline{\text{P}}$) if and only if there exists a $\lambda \in \Delta_I$ such that $\sum_{k\in I} \lambda_k \,\text{grad} f_k(\mathbf{x}^*) = \mathbf{0}$, where $I = I(\mathbf{x}^*)$.*

Proof. Since $\partial F(\mathbf{x}) = \text{co} \bigcup_{k\in I(\mathbf{x})} \{\nabla f_k(\mathbf{x})\}$ (co denotes the convex hull) for any $\mathbf{x}$, we have $\mathbf{g} = \sum_{k\in I} \lambda_k \nabla f_k(\mathbf{x}^*) \in \partial F(\mathbf{x}^*)$ for any $\lambda \in \Delta_I$. It follows that $\mathbf{0} = \mathbf{g} + \nu\mathbf{x}^*$ if and only if $(\mathbf{I} - \mathbf{x}^*(\mathbf{x}^*)^T)\mathbf{g} = \mathbf{0}$, or equivalently, $\sum_{k\in I} \lambda_k \,\text{grad} f_k(\mathbf{x}^*) = \mathbf{0}$. □

Note that when $K = 1$ (i.e., $F(\mathbf{x}) = f_1(\mathbf{x})$), Theorem 1 reduces to the definition of critical points for smooth functions on $\mathbb{S}$ in [11].

Now, for any $\mathbf{x} \neq \mathbf{0}$, let $R(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$ be the retraction of $\mathbf{x}$ back to $\mathbb{S}$. The Jacobian of R is given by $J_R(\mathbf{x}) = \mathbf{I}/\|\mathbf{x}\|_2 - \mathbf{x}\mathbf{x}^T/\|\mathbf{x}\|_2^3$. When $\|\mathbf{x}\|_2 = 1$, the Chain Rule yields

$$\begin{aligned} \nabla(f_k \circ R)(\mathbf{x}) &= J_R(\mathbf{x})^T \nabla f_k(R(\mathbf{x})) \\ &= \nabla f_k(\mathbf{x}) - \mathbf{x}^T \nabla f_k(\mathbf{x})\mathbf{x} \\ &= \text{grad} f_k(\mathbf{x}). \end{aligned} \qquad (2)$$

For simplicity, let $g_k(\mathbf{x}) = \text{grad} f_k(\mathbf{x})$ for $k = 1, \ldots, K$. Note that any $\mathbf{d} \in \partial F(\mathbf{x})$ that satisfies $g_k(\mathbf{x})^T \mathbf{d} > 0$ for $k \in I(\mathbf{x})$ gives a descent direction $-\mathbf{d}$ with respect to F at $\mathbf{x}$. This motivates us to formulate the following LP to find the "best" descent direction at $\mathbf{x}$:

$$\begin{aligned} \max \quad & t \\ \text{s.t} \quad & g_k(\mathbf{x})^T \left(\textstyle\sum_{k\in I(\mathbf{x})} \lambda_k g_k(\mathbf{x}) \right) \geq t, \\ & k \in I(\mathbf{x}), \lambda \in \Delta_{I(\mathbf{x})}. \end{aligned}$$

Upon defining the $|I| \times |I|$ matrix $\mathbf{B}$ by $B_{pq} = g_{i_p}(\mathbf{x})^T g_{i_q}(\mathbf{x})$ for $i_p, i_q \in I$, we can write the above LP more compactly as

$$\max\ t \quad \text{s.t} \quad \mathbf{B}\lambda \geq t\mathbf{e},\ \lambda \in \Delta_I(\mathbf{x}). \qquad \text{(LP)}$$

Since $\nabla f_k(\mathbf{x}) = 2A_k\mathbf{x}$ and $\nabla^2 f_k(\mathbf{x}) = \mathbf{A}_k$, for any $\mathbf{x} \in \mathbb{S}$, we have $\nabla^2 f_k(\mathbf{x}) \preceq L\mathbf{I}$ and $\|\nabla f_k(\mathbf{x})\|_2 \leq 2L$ for $k = 1, \ldots, K$, where $L \geq \lambda_{\max}(\mathbf{A}_k)$ and $\lambda_{\max}(\mathbf{A}_k)$ is the largest singular value of $\mathbf{A}_k$. We then have the following result:

Proposition 1. *Let t^* be the optimal value of* (LP)*. Then, we have $t^* \geq 0$ and*

$$\frac{t^*}{2L} \leq |\partial F_{I(\mathbf{x})}| \leq \sqrt{t^*},$$

where $|\partial F_{I(\mathbf{x})}| = \min_{\lambda\in\Delta_{I(\mathbf{x})}} \left\| \sum_{k\in I(\mathbf{x})} \lambda_k g_k(\mathbf{x}) \right\|_2$.

Proof. The dual of (LP) is given by

$$\min\ v \quad \text{s.t} \quad \mathbf{B}^T\mathbf{u} \leq v\mathbf{e},\ \mathbf{e}^T\mathbf{u} = 1,\ \mathbf{u} \geq \mathbf{0}. \qquad \text{(DLP)}$$

Let (t^*, λ^*) and $(v^*, \mathbf{u}^*)$ be optimal solutions to (LP) and (DLP), respectively. Then, by strong duality and boundedness of (DLP) (which is obvious since $\mathbf{u}$ belongs to a simplex), we have $t^* = v^*$. Moreover, the dual feasibility of $\mathbf{u}^*$ implies that $(\mathbf{u}^*)^T\mathbf{B}^T\mathbf{u}^* \leq v^*(\mathbf{u}^*)^T\mathbf{e} = v^*$. Since $\mathbf{B}^T = \mathbf{B} \succeq \mathbf{0}$, we obtain $(\mathbf{u}^*)^T\mathbf{B}^T\mathbf{u}^* \geq 0$ and hence $t^* = v^* \geq 0$.

Next, by the optimality of t^*, we know that $t^* \leq \mathbf{g}^T\left(\sum_{k\in I(\mathbf{x})}\lambda_k^* g_k(\mathbf{x})\right)$, where $\mathbf{g}$ is any convex combination of $\{g_k(\mathbf{x}) : k \in I(\mathbf{x})\}$. Hence, we have $t^* \leq \|\mathbf{g}\|_2 \cdot \left\|\sum_{k\in I(\mathbf{x})}\lambda_k^* g_k(\mathbf{x})\right\|_2 \leq 2L\|\mathbf{g}\|_2$ for all $\mathbf{g}$. This establishes the first inequality. Furthermore, since $t^* = v^* \geq (\mathbf{u}^*)^T\mathbf{B}^T\mathbf{u}^* = \left\|\sum_{k\in I(\mathbf{x})} u_k^* g_k(\mathbf{x})\right\|_2^2$, we have the second inequality. □

In practice, to improve numerical stability, we use the δ-active set $I_\delta = I_\delta(\mathbf{x}) = \{k \in \{1,\ldots,K\} : |F(\mathbf{x}) - f_k(\mathbf{x})| \leq \delta\}$ for some $\delta > 0$ and decrease δ adaptively to accelerate our algorithm. We call the point $\mathbf{x}$ a (δ,ϵ)-*critical point* of Problem $(\overline{\mathrm{P}})$ if $|\partial F_{I_\delta}| \leq \epsilon$ for the δ-active set I_δ. Given $(\bar{\delta},\bar{\epsilon})$, our proposed algorithm for finding an $(\bar{\delta},\sqrt{\bar{\epsilon}})$-critical point of Problem $(\overline{\mathrm{P}})$ is shown in Algorithm 1. In the sth iteration, we first choose a δ_s-active set and solve (LP) to find a descent direction. We consider three cases:

1) If $\delta_s \leq \bar{\delta}$ and the optimal value of (LP) satisfies that $t^* \leq \bar{\epsilon}$, then we are already at a $(\bar{\delta},\sqrt{\bar{\epsilon}})$-critical point and thus terminate our algorithm.
2) If $t^* > \bar{\epsilon}$, then $-\mathbf{d}^s = -\sum_{k\in I_{\delta_s}(\mathbf{x}^s)}\lambda_k^* g_k(\mathbf{x}^s)$ obtained from (LP) is a descent direction. We then apply an Armijo-type rule to perform a line search. Specifically, we find the smallest integer $l \geq 0$ such that
$$F(\mathbf{x}^s - \gamma\theta^l\mathbf{d}^s) \leq F(\mathbf{x}^s) - \tau\gamma\theta^l t^*, \tag{3}$$
where $0 < \gamma \leq 1$, $0 < \theta < 1$, and $0 < \tau \leq 0.5$.
3) Otherwise, we have $\delta_s > \bar{\delta}$ and $t^* \leq \bar{\epsilon}$. In this case, the δ_s-active set may be too loose and we reduce the value of δ_s by half and re-solve (LP) to see which case we are in. Note that since $\bar{\delta} > 0$, this step will be repeated for at most $\lceil\log(\delta_{s-1}/\bar{\delta})\rceil$ times, which is finite.

In our implementation, we start with some relatively large δ_s to achieve a large decrease in the first few iterations and then reduce δ_s if Case 3 occurs. The following theorem shows that our algorithm converges to some $(\bar{\delta},\sqrt{\bar{\epsilon}})$ critical point of Problem $(\overline{\mathrm{P}})$.

Theorem 2. *Let $\{\mathbf{x}^s\}$ be the sequence generated by Algorithm 1 and $\{\beta_s\}$ be the step sizes chosen by the Armijo-type rule. Then, given parameters $0 < \tau \leq 0.5$, $\bar{\delta} > 0$ and $\bar{\epsilon} > 0$, Algorithm 1 returns a $(\bar{\delta},\sqrt{\bar{\epsilon}})$-critical point of Problem $(\overline{\mathrm{P}})$ in $O(1/\min\{\tau\theta\bar{\epsilon}^2/8L^3, \tau\theta\bar{\epsilon}\bar{\delta}/16L^2\})$ iterations.*

Proof. From Proposition 1, we know that if $t^* \leq \bar{\epsilon}$, then $|\partial F_{I_{\delta_s}}| \leq \sqrt{t^*} \leq \sqrt{\bar{\epsilon}}$. Note that every iteration before termination must have the property that $t^* \geq \bar{\epsilon}$ or $\delta_s \geq \bar{\delta}$. Now, let us analyze the decrease in each iteration.

Let $\lambda^* = \mathrm{argmin}_\lambda \left\|\sum_{k\in I_{\delta_s}}\lambda_k g_k(\mathbf{x}^s)\right\|_2$ and define $\mathbf{d}^s = \sum_{k\in I_{\delta_s}}\lambda_k^* g_k(\mathbf{x}^s)$. Then, for the *true* active set I_s in iteration s, we have $I_s \subset I_{\delta_s}$ and
$$|\partial F_{I_s}| \geq |\partial F_{I_{\delta_s}}| \geq \frac{t^*}{2L} \geq \frac{\bar{\epsilon}}{2L}. \tag{4}$$

Algorithm 1 LPA-SD: First-order method for Problem $(\overline{\mathrm{P}})$

Input: parameters $\bar{\delta}$, $\bar{\epsilon}$, τ, γ, θ, initial value of δ_s and maximum iteration number M.

initialize $\mathbf{x}^0$.
for $s = 0, 1, \ldots$ **do**
  compute $F = \max_k f_k(\mathbf{x}^s)$
  compute $I_{\delta_s} = \{i : |F - f_k(\mathbf{x}^s)| \leq \delta_s\}$
  while $\delta_s \geq \bar{\delta}$ **do**
    solve (LP) with active set I_{δ_s} to get the direction $\mathbf{d}^s$ and optimal value t^*
    if $t^* \leq \bar{\epsilon}$ and $\delta_s \leq \bar{\delta}$ **then**
      return
    else if $t^* \leq \bar{\epsilon}$ **then**
      set $J = I_{\delta_s}$, $\delta_s = \delta_s/2$, $I_{\delta_s} = \{i \in J : |F - f_k(\mathbf{x}^s)| \leq \delta_s\}$
    else
      break
    end if
  end while
  choose step size $\beta_s > 0$ via (3) and update
$$\mathbf{x}^{s+1} = \frac{\mathbf{x}^s - \beta_s\mathbf{d}^s}{\|\mathbf{x}^s - \beta_s\mathbf{d}^s\|_2}$$
end for

For any $\xi \in [0,1]$, note that $\left\|(1-\xi)\mathbf{x}^{s+1} + \xi\mathbf{x}^s\right\|_2^2 = \xi^2(\mathbf{x}^s)^T\mathbf{x}^s + (1-\xi)^2(\mathbf{x}^{s+1})^T\mathbf{x}^{s+1} + 2\xi(1-\xi)(\mathbf{x}^{s+1})^T\mathbf{x}^s \geq \xi^2 + (1-\xi)^2 \geq 1/2$. This follows since $\mathbf{d}^s$ is orthogonal to $\mathbf{x}^s$ and hence the angle between $\mathbf{x}^{s+1}$ and $\mathbf{x}^s$ is less than 90 degrees. Using (2), for any $\alpha > 0$, we have
$$\begin{aligned} f_k(\mathbf{x}^{s+1}) &= (f_k \circ R)(\mathbf{x}^s - \alpha\mathbf{d}^s) \\ &= (f_k \circ R)(\mathbf{x}^s) + \nabla(f_k \circ R)(\mathbf{x}^s)^T(-\alpha\mathbf{d}^s) \\ &\quad + \frac{1}{2}(-\alpha\mathbf{d}^s)^T\nabla^2(f_k \circ R)(\tilde{\mathbf{x}}^{s+1})(-\alpha\mathbf{d}^s) \\ &\leq f_k(\mathbf{x}^s) - \alpha\mathrm{grad}f_k(\mathbf{x}^s)^T\mathbf{d}^s + L\|\alpha\mathbf{d}^s\|^2 \\ &\leq f_k(\mathbf{x}^s) - \alpha\mathrm{grad}f_k(\mathbf{x}^s)^T\mathbf{d}^s + 4\alpha^2L^3, \end{aligned}$$
where $\mathbf{x}^{s+1} = R(\mathbf{x}^s - \alpha\mathbf{d}^s)$ and $\tilde{\mathbf{x}}^{s+1} = \mathbf{x}^s + \xi(\mathbf{x}^{s+1} - \mathbf{x}^s)$ for some $\xi \in [0,1]$. The first equality uses Taylor expansion in Lagrangian form; the first inequality is because $\left\|\nabla^2(f_k \circ R)(\tilde{\mathbf{x}}^{s+1})\right\|_2 = \left\|J_R^T(\tilde{\mathbf{x}}^{s+1})\nabla^2 f_k(\tilde{\mathbf{x}}^{s+1})J_R(\tilde{\mathbf{x}}^{s+1})\right\|_2$ and $\nabla^2 f_k(\tilde{\mathbf{x}}^{s+1}) \leq L\mathbf{I}$ and $\sqrt{2}\mathbf{I} \succeq \mathbf{I}/\left\|\tilde{\mathbf{x}}^{s+1}\right\|_2 \succeq J_R(\tilde{\mathbf{x}}^{s+1}) \succeq \mathbf{0}$ as $J_R(\tilde{\mathbf{x}}^{s+1}) = \mathbf{I}/\left\|\tilde{\mathbf{x}}^{s+1}\right\|_2 - \tilde{\mathbf{x}}^{s+1}(\tilde{\mathbf{x}}^{s+1})^T/\left\|\tilde{\mathbf{x}}^{s+1}\right\|_2^3$; the second inequality is because $\|\mathbf{d}^s\|_2 \leq 2L$. Hence, for any step size $\alpha \in (0, t^*/8L^3]$, we have
$$\begin{aligned} f_k(\mathbf{x}^{s+1}) &\leq f_k(\mathbf{x}^s) - \alpha\mathrm{grad}f_k(\mathbf{x}^s)^T\mathbf{d}^s + 4\alpha^2L^3 \\ &\leq f_k(\mathbf{x}^s) - \frac{1}{2}\alpha t^* \end{aligned}$$
for $k \in I_s$ and
$$\begin{aligned} f_k(\mathbf{x}^{s+1}) &= f_k(\mathbf{x}^s) - \alpha\mathrm{grad}f_k(\mathbf{x}^s)^T\mathbf{d}^s + 4\alpha^2L^3 \\ &\leq f_k(\mathbf{x}^s) + 4\alpha L^2 + \alpha t^* \end{aligned}$$

for $k \notin I_s$. Now, define $\eta_s = F(\mathbf{x}^s) - \max_{k\notin I_s} f_k(\mathbf{x}^s) > \delta_s \geq \bar{\delta}$. For all $\alpha \leq \eta_s/2(4L^2+t^*)$, we have

$$\max_{k\notin I_s} f_k(\mathbf{x}^{s+1}) \leq F(\mathbf{x}^s) - \frac{\eta_s}{2}.$$

Define $\alpha_0 = \min\{t^*/8L^3,\ \eta_k/2(4L^2+t^*)\}$. Then, we have $\alpha_0 \geq \min\{t^*/8L^3, \bar{\delta}/16L^2\}$ because $t^* \leq 4L^2$. Hence, we have

$$F(\mathbf{x}^s) - F(\mathbf{x}^{s+1}) \geq \min\left\{\frac{\alpha t^*}{2}, \frac{\eta_s}{2}\right\} \geq \min\left\{\frac{\alpha\bar{\epsilon}}{2}, \frac{\bar{\delta}}{2}\right\}$$

for all $\alpha \leq \alpha_0$. Moreover, the Armijo step size in (3) is bounded below by $\beta_s \geq \theta\alpha_0$ because $0 < \tau \leq 0.5$. It follows that

$$\begin{aligned} F(\mathbf{x}^s) - F(\mathbf{x}^{s+1}) &\geq \tau\beta_s t^* \\ &\geq \tau\theta\bar{\epsilon}\min\left\{\frac{t^*}{8L^3}, \frac{\bar{\delta}}{16L^2}\right\} \\ &\geq \min\left\{\frac{\tau\theta\bar{\epsilon}^2}{8L^3}, \frac{\tau\theta\bar{\epsilon}\bar{\delta}}{16L^2}\right\} \end{aligned}$$

and the proof is completed. □

IV. Simulation Results

We compare our first-order method (LPA-SD) with the standard SLA algorithm (SLA-MOSEK) with codes provided by the authors in [4] and two SLA algorithms (SLA-MP and SLA-LADMM) with codes provided by the authors in [9]. Each subproblem of the standard SLA was a convex quadratic programming problem and solved by the MOSEK solver [14] using the modeling language YALMIP [15] as in [4]. The two SLA algorithms in [9] use a Mirror-Prox algorithm and a linearized ADMM to solve the subproblems in each iteration of the SLA algorithm, respectively. Each LP problem in our algorithm is solved by `linprog` in MATLAB optimization toolbox with default accuracy. The channel vectors of all users are generated with standard complex normal distribution $\mathcal{CN}(0, \mathbf{I})$ and the noise variance σ_k is set to be 1 for all users. All experiments were run on a Windows desktop with 8 Intel i7 cores (3.40GHz) and 16GB of RAM.

We use a same random initialization for all the four algorithms. We set the maximum iteration number to be 150 in our numerical comparisons to achieve a tradeoff between accuracy and time complexity. The initial δ is set to be 1, $\bar{\delta}$ is set to be 10^{-5} and $\bar{\epsilon} = 10^{-3}$. We also terminate our algorithm if $F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) \leq 10^{-5}$. The parameters in the Armijo rule is set as $\tau = 0$, $\gamma = 1$ and $\theta = 0.5$. (We find $\tau = 0$ is sufficient to guarantee a good piratical performance of our algorithm, although our theoretical analysis requires $\tau > 0$ being a small positive number.) We also set the maximum iteration number to be 20 for each of the three SLA algorithms as in [4] and [9]. And the inner iteration number of SLA-MP and SLA-LADMM is set as 1000 and 600 for a tradeoff between accuracy and time complexity. For SLA-MP, we set $\epsilon_b = 1e-6$ and the step-size $\alpha = 2/L$. For SLA-LADMM, we set the penalty parameter ρ to be 1. Note that we try our best to improve the tradeoff between accuracy and time complexity for the SLA algorithms in [4] and [9]. We present our numerical results in Figures 1 and 2, which are averaged over 200 channel realizations. The comparison of average minimum SNR and CPU time for fixed number of Tx antennas $N = 25$ and increased number of users from 100 to 500 is shown in Figures 1, while the comparison for fixed number of users $K = 50$ and increased number of antennas N from 100 to 500 is shown in Figure 2. Before discussing the numerical

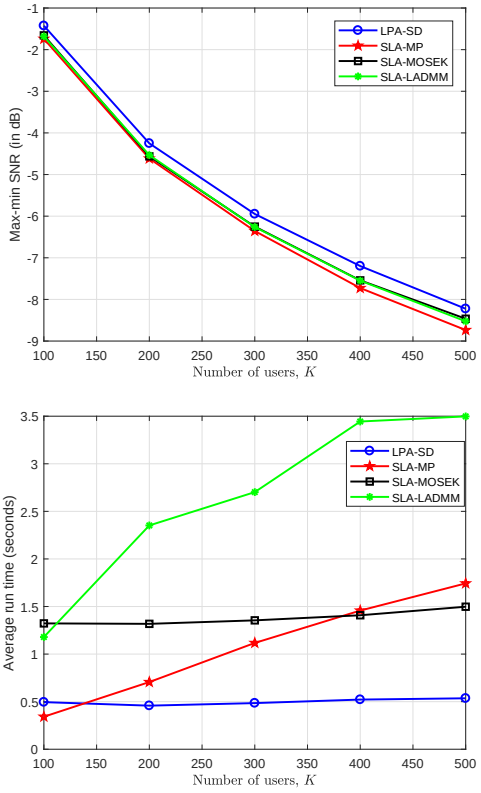

Fig. 1. Evolution of minimum SNR and CPU time versus users number K from 100 to 500 for antennas number $N = 25$.

results, let us illustrate how to efficiently implement the Armijo rule in iteration s. Note that $\mathbf{x} = \begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}$,$\mathbf{w} = \mathbf{y} + i\mathbf{z}$ and $f_k(\mathbf{x}) = -\left\|\frac{\mathbf{h}_k^H\mathbf{w}}{\sigma_k}\right\|_2^2$. For ease of notation, let $\mathbf{d} = \begin{bmatrix}\mathbf{s}\\ \mathbf{t}\end{bmatrix}$ and $\mathbf{u} = \mathbf{s} + i\mathbf{t}$. Then $f_k(\frac{\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s}{\|\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s\|_2}) = -\left\|\frac{\mathbf{h}_k^H(\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s)}{\sigma_k\|\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s\|_2}\right\|_2^2 = -\left\|\frac{\mathbf{h}_k^H\mathbf{w}^s - \gamma\theta^l\mathbf{h}_k^H\mathbf{u}^s}{\sigma_k\|\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s\|_2}\right\|_2^2$. Hence in iteration s, the complexity of our algorithm is $O(KN + l^sK)$ plus the time cost by the LP solver: the estimations of $\mathbf{h}_k^H\mathbf{w}^s$ and $\mathbf{h}_k^H\mathbf{u}^s$ for $k = 1, \ldots, K$ cost $O(KN)$; each evaluation of $f_k(\frac{\mathbf{x}^s - \gamma\theta^l\mathbf{u}^s}{\|\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s\|_2})$ only need an estimation of $\left\|\mathbf{w}^s - \gamma\theta^l\mathbf{u}^s\right\|_2$ with complexity $O(N)$ and several scalar operations with complexity $O(l^s)$ where we denote l^s the times of line search with the Armijo rule at iteration s, and thus the total cost is $O(l^sK + KN)$. From the figures we can observe that our first-order method has slightly better

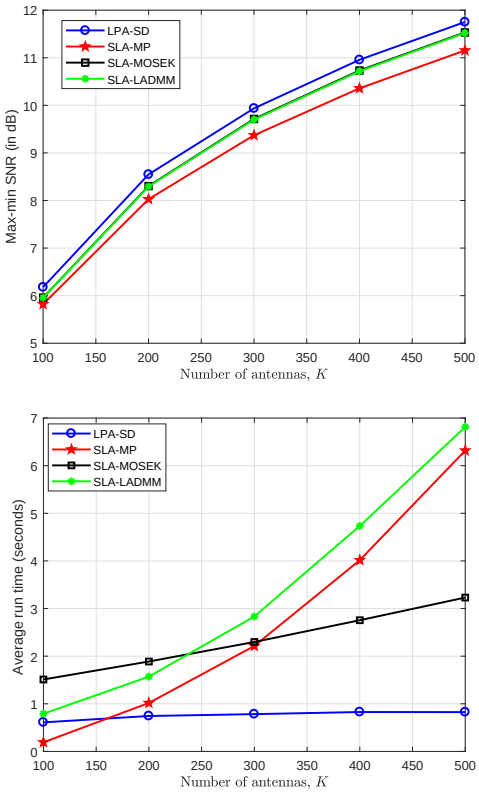

Fig. 2. Evolution of minimum SNR and CPU time versus antennas number N from 100 to 500 for user number $K = 50$.

average minimum SNR performance than the three SLA based algorithms and takes much less CPU time. The similar SNR performance may be because all these algorithms are theoretically guaranteed to converge to a critical point and the initial points are the same. The low time complexity of our algorithm may be because of the lower number of operations of our algorithm as analyzed below. Note that i) the SLA-MOSEK solves a convex quadratic programming problem for at most 20 iterations, ii) SLA-MP (SLA-LADMM, respectively) has 20 outer iterations and 1000 (600, respectively) inner iterations, and each inner iteration takes $O(KN)$ operations. As anlayzed in last paragraph, out algorithm takes $O(l^s K + KN)$ operations plus solving a LP problem in each iteration (up to 150). The figures further reveal that the speed of our method is almost independent of the number of antennas as the bulk of the cost of our method is solving the LP, whose dimension is determined by the cardinality of the relative active set. Particularly when the number of the antennas is $N = 25$, the dimension of $\mathbf{x}$ is 50 and we also found that the cardinality of relative active functions never exceeded 50 in our simulation. In the setting of fixed users $K = 50$, the cardinality of relative active functions cannot exceed the number of users, i.e., 50. Our further investigation also verified that most of the time complexity (more than 80% of total CPU time) is in solving the LP problem. Hence the low dimension of the LP problem and the low iteration number are the main reasons that makes our algorithms efficient. The evolution of figures also indicates that the outperformace of our algorithm may be even more significant for higher dimensional problem settings.

V. Conclusion

In the paper, we propose a new first-order algorithm called LPA-SD for the max-min-fair formulation for the SGMB problem. We further demonstrate convergence of our algorithm to a critical point. Numerical results show that our algorithm has slightly better minimum SNR values and much less CPU time than the state of the art algorithms in the literature for solving the SGMB problem. As future research, we would like to extend our algorithm to problems with more general objective and constraint, e.g., more complex objective and any closed convex set constraint whose projection is easy to compute (e.g., massive MIMO multicasting).

References

[1] 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, June 2006.

[2] 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.

[3] S. X. Wu, A. M.-C. So, and W.-K. Ma, "Rank-two transmit beamformed Alamouti space-time coding for physical-layer multicasting," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 2793–2796.

[4] 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, 2014.

[5] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[6] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, 2013.

[7] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, 2014.

[8] A. Konar and N. D. Sidiropoulos, "Fast feasibility pursuit for non-convex QCQPs via first-order methods," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4064–4068.

[9] ——, "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.

[10] B. Gopalakrishnan and N. D. Sidiropoulos, "High performance adaptive algorithms for single-group multicast beamforming," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4373–4384, Aug 2015.

[11] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[12] R. Jiang and D. Li, "Novel reformulations and efficient algorithm for the generalized trust region subproblem," *arXiv preprint arXiv:1707.08706*, 2017.

[13] E. J. Balder, "On generalized gradients and optimization," http://www.staff.science.uu.nl/~balde101/cao10/cursus08_3.pdf, 2008.

[14] APS Mosek, "The MOSEK optimization software", *Online at* http://www.mosek.com.

[15] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," *in Proc. CACSD*, Taipei, Sep. 4, 2004.