Fast First-Order Methods for the Massive Robust Multicast Beamforming Problem with Interference Temperature Constraints

Huikang Liu, Peng Wang, Anthony Man-Cho So
Dept. of System Erg. & Erg. Mgt., CUHK, Hong Kong
{hkliu, wangpeng, manchoso}@se.cuhk.edu.hk

香港中文大學 The Chinese University of Hong Kong
香港中文大學 The Chinese University of Hong Kong

Introduction

We consider the following single-group multicast beamforming problem:

$$\max_{\mathbf{w}\in\mathbb{C}^N} \min_{k\in\{1,\dots,K\}} |\mathbf{h}_k^H w|^2/\sigma_k^2 \quad \text{s.t.} \quad \|\mathbf{w}\|_2 = 1;$$

- $\mathbf{h}_k \in \mathbb{C}^N$ is the channel between the Tx and the kth Rx;
- $\mathbf{w} \in \mathbb{C}^N$ is the unit-norm beamforming vector;
- σ_k^2 is the variance of centered Gaussian noise.

We convert the above maximin problem into an equivalent minimax problem with real variables as follows:

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

Motivations

One classic approach is the projected subgradient descent method:

$$x^{s+1} = Proj_{\mathcal{S}^{N-1}}(x^s + \alpha_s d_s)$$

where $d_s \in \operatorname{conv}\bigcup_{k\in I(\mathbf{x})}\{\nabla f_k(\mathbf{x})\}$ (conv denotes the convex hull), and $I(x)$ denote the index set of active functions at x.

- $I(x)$ changes frequently when x is near the optimal point, which makes the method very ***unstable***;
- d_s is not easy to choose because there is a projection operator.

How to solve them?

Ideas of Algorithm

To improve the numerical stability and accelerate the sub-gradient method, we propose several ideas:

- We introduce the δ-active set

$$I_\delta = \{1 \le k \le K : |F(\mathbf{x}) - f_k(\mathbf{x})| \le \delta\}$$

for some $\delta > 0$ and decrease δ adaptively to improve numerical stability.

- use the projection of $\nabla f_k(\mathbf{x})$ onto the tangent space of $\mathbb{S}$ at $\mathbf{x}$:

$$g_k(\mathbf{x}) = \operatorname{grad} f_k(\mathbf{x}) = (\mathbf{I} - \mathbf{x}\mathbf{x}^T)\nabla f_k(\mathbf{x});$$

- solving the following LP to find the "best" descent direction $\mathbf{d}$ at $\mathbf{x}$:

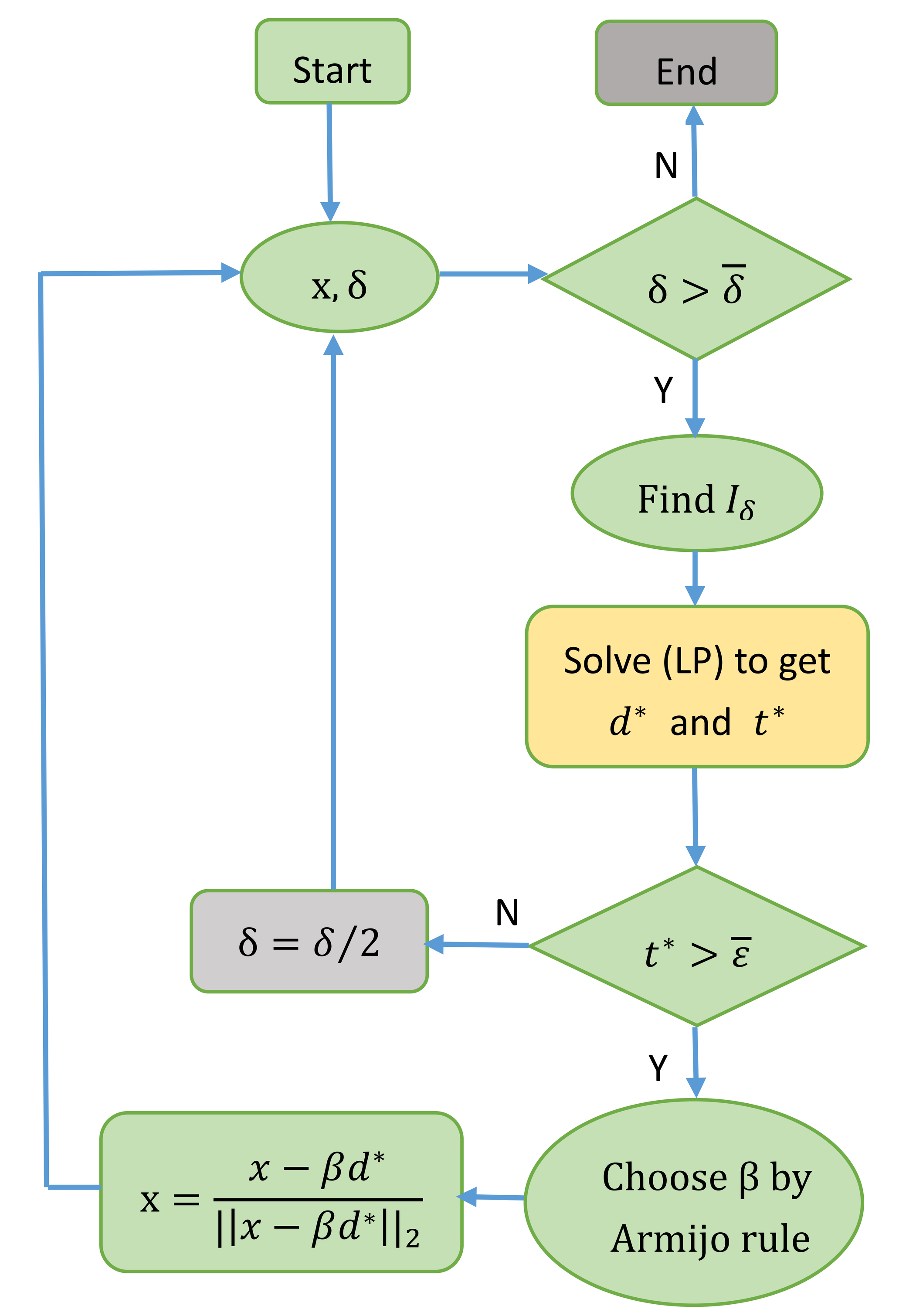
$$\begin{aligned} \max \quad & t & \text{(LP)}\\ \text{s.t} \quad & g_k(\mathbf{x})^T\mathbf{d} \ge t, \quad k \in I_\delta(\mathbf{x}), & \\ & \mathbf{d} = \sum_{k\in I(\mathbf{x})} \lambda_k g_k(\mathbf{x}), \quad \lambda \in \Delta_{I(\mathbf{x})}. & \end{aligned}$$

We then apply an Armijo-type rule to perform a line search. Specifically, we find the smallest integer $l \ge 0$ such that

$$F(\mathbf{x}^s - \gamma\theta^l\mathbf{d}^s) \le F(\mathbf{x}^s) - \tau\gamma\theta^l t^*, \qquad (1)$$

where $0 < \gamma \le 1$, $0 < \theta < 1$, and $0 < \tau \le 0.5$.

LPA-SD Method



Analysis

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

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

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

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

We call the point $\mathbf{x}$ a (δ,ϵ)*-critical point* of Problem (P) if $|\partial F_{I_\delta}| \le \epsilon$ for the δ-active set I_δ. The following theorem shows that our algorithm converges to some $(\bar{\delta}, \sqrt{\bar{\epsilon}})$ critical point of Problem (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 $0 < \tau \le 0.5$, $\bar{\delta} > 0$ and $\bar{\epsilon} > 0$, Algorithm 1 returns a $(\bar{\delta}, \sqrt{\bar{\epsilon}})$-critical point of Problem* (P) *in $O(1/\min\{\tau\theta\bar{\epsilon}^2/8L^3, \tau\theta\bar{\epsilon}\bar{\delta}/16L^2\})$ iterations.*

Numerical Results

In numerical simulations,

- we use a same random initialization for all the four algorithms;
- 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 $F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) \le 10^{-5}$. The parameters in the Armijo rule is set as $\tau = 0, \gamma = 1$ and $\theta = 0.5$.
- We also set the maximum iteration number to be 20 for each of the three SLA algorithms, and the inner iteration number of SLA-MP and SLA-LADMM is set as 1000 and 600
- in Figure 1, fixed number of Tx antennas $N = 25$ and increased number of users K from 100 to 500;
- in Figure 2, fixed number of users $K = 50$ and increased number of antennas N from 100 to 500;
- our numerical results in Figures 1 and 2 are averaged over 200 channel realizations.

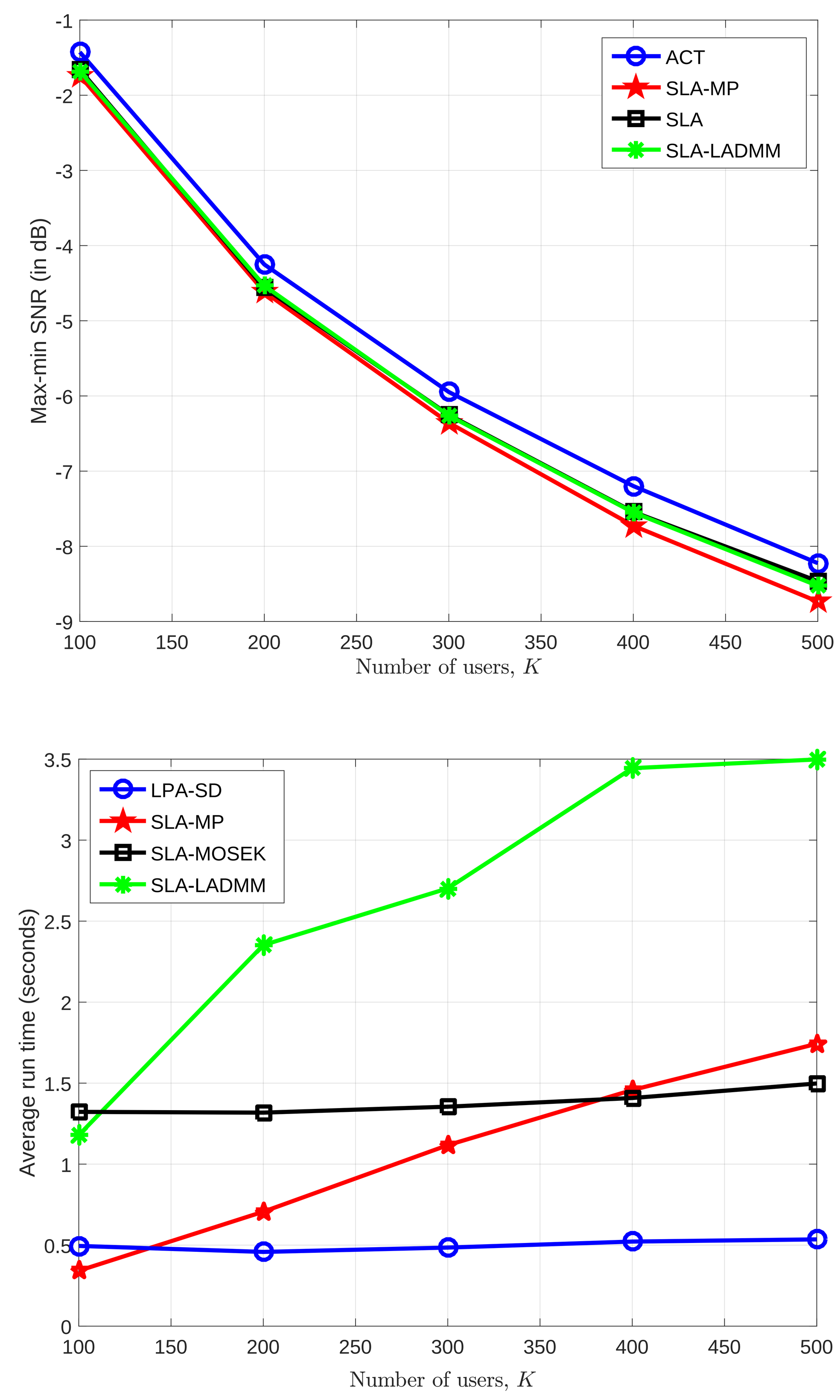


Figure 1: Evolution of minimum SNR and CPU time versus users number K from 100 to 500 for antennas number $N = 25$.

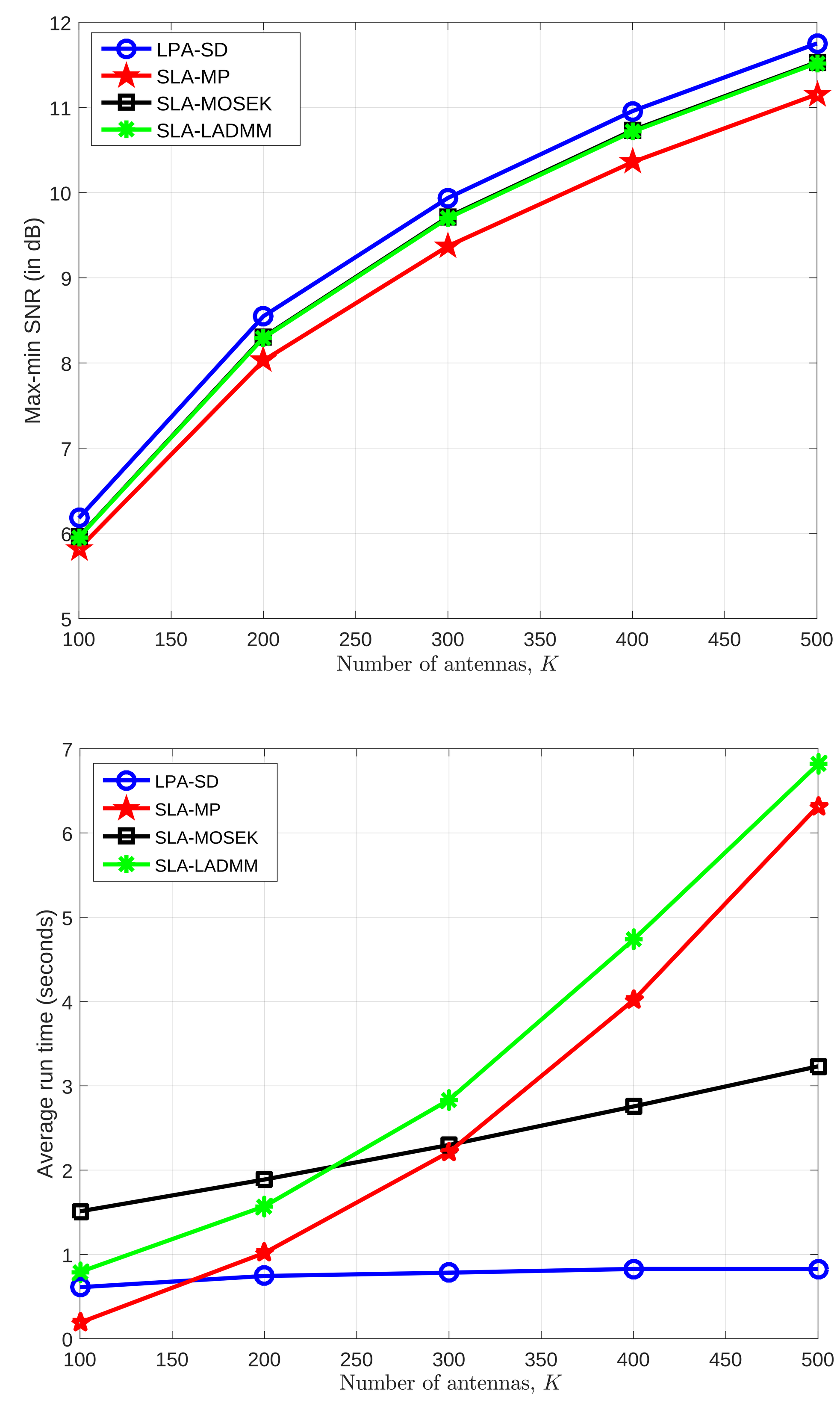


Figure 2: Evolution of minimum SNR and CPU time versus antennas number N from 100 to 500 for user number $K = 50$.