Fast Approximation Algorithms for a Class of Non-convex QCQP Problems Using First-Order Methods

Aritra Konar, *Student Member, IEEE*, and Nicholas D. Sidiropoulos, *Fellow, IEEE*

***Abstract*—A number of important problems in engineering can be formulated as non-convex quadratically constrained quadratic programming (QCQP). The general QCQP problem is NP-Hard. In this paper, we consider a class of non-convex QCQP problems that are expressible as the maximization of the point-wise minimum of homogeneous convex quadratics over a "simple" convex set. Existing approximation strategies for such problems are generally incapable of achieving favorable performance-complexity tradeoffs. They are either characterized by good performance but high complexity and lack of scalability, or low complexity but relatively inferior performance. This paper focuses on bridging this gap by developing high performance, low complexity successive *non-smooth* convex approximation algorithms for problems in this class. Exploiting the structure inherent in each subproblem, specialized first-order methods are used to efficiently compute solutions. Multicast beamforming is considered as an application example to showcase the effectiveness of the proposed algorithms, which achieve a very favorable performance-complexity tradeoff relative to the existing state of the art.**

***Index Terms*—Non-convex optimization, non-smooth optimization, quadratically constrained quadratic programming (QCQP), first-order methods, convergence, Nesterov smoothing, Nemirovski saddle point reformulation, massive multiple-input multiple-output (MIMO) communications, multicasting, per-antenna power constraints.**

I. Introduction

QUADRATICALLY constrained quadratic programming (QCQP) problems are an important class of optimization problems which involve minimizing a quadratic cost function subject to quadratic inequality constraints, and find widespread application in various engineering fields (see [2] and references therein). However, the general case of the QCQP problem is known to be NP–Hard [3]. In this paper, we consider a special class of non-convex QCQP problems which can be expressed as

$$\max_{\mathbf{x}\in\mathbb{C}^N} \quad \min_{m\in\mathcal{M}} \mathbf{x}^H \mathbf{A}_m \mathbf{x} \tag{1a}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{F} \tag{1b}$$

where $\mathcal{M} = \{1, 2, \cdots, M\}$, $\mathbf{A}_m \in \mathbb{C}^{N\times N} \succeq \mathbf{0}$, $\forall\, m \in \mathcal{M}$, and $\mathcal{F} \subseteq \mathbb{C}^N$ is a *simple*[1], compact, convex set. Note that (1) is a non-smooth, non-convex optimization problem since the point-wise minimum of convex quadratics is non-differentiable and non-concave. As a motivating example, such a formulation naturally arises in single-group multicast beamforming [4].

A. Prior Art and Motivation

Although (1) is a challenging non-convex optimization problem, there exist tractable approximation strategies for computing high quality, suboptimal solutions. A general approximation framework for solving such non-convex problems is *successive convex approximation* (SCA) [5]–[9], which is based on iteratively solving a series of convex problems obtained by constructing a convex surrogate of the non-convex objective function at each iteration. Under appropriate technical conditions, convergence of the iterates generated by the algorithm to a stationary point can be established (see [5]–[9] for details). Furthermore, if the feasible set $\mathcal{F}$ is defined by quadratic inequalities (as is the case in multicast beamforming), then the technique of *semidefinite relaxation* (SDR) [10] can be used to obtain an approximate solution to (1). In this approach, (1) is first converted into an equivalent rank-1 constrained semidefinite programming (SDP) problem, following which the rank constraint is dropped to obtain a relaxed convex SDP problem. A post-processing step is then used to convert the high rank solution of the relaxed problem into a rank-1 solution according to a randomized procedure (see [4] for details). One can also use a combined two-step strategy where SDR followed by randomization is used to yield an initial approximate solution, which is then further refined via SCA.

Regardless of how successful these approaches prove to be in yielding high quality, approximate solutions of (1), they all

Manuscript received March 6, 2016; revised September 16, 2016, December 16, 2016, and January 19, 2017; accepted February 9, 2017. Date of publication April 3, 2017; date of current version May 5, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gesualdo Scutari. This work was supported in part by the National Science Foundation under Grants AST-1247885, ECCS 1231504, and CIF-1525194. This paper was presented in part at the 17th IEEE International Workshop on Signal Processing Advances for Wireless Communications, Edinburgh, U.K., July 2016. *(Corresponding author: Nicholas D. Sidiropoulos.)*

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: konar006@umn.edu; nikos@umn.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the authors, which includes an alternative proof of Proposition 1. The material is 147 KB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2690386

[1]By *simple*, we mean that the Euclidean projection operation onto the set $\mathcal{F}$ can be computed in closed form.

suffer from the downside of having high computational complexity. SDR requires lifting the problem to higher-dimensional space, effectively squaring the number of variables, while SCA requires one to solve a sequence of general convex programming problems, each of which can be computationally expensive. In this manuscript, we focus on addressing this shortcoming by developing high performance, low complexity SCA algorithms for QCQP problems of the form (1).

B. Contributions

Our approach for developing fast SCA algorithms can be summarized as follows: at each iteration we construct a non-smooth, convex surrogate function of the non-convex objective of (1) by locally linearizing each quadratic component of (1a) about the current iterate. On replacing (1a) with its convex surrogate function, at each iteration, we obtain a non-smooth, convex optimization subproblem. The solution of each subproblem is then used as the point about which we construct a convex surrogate function in the next iteration. We establish the global convergence of the iterates generated by the resulting SCA algorithm to the set of d-stationary solutions of (1) via two different techniques. The main difference between these proof techniques is that the first reformulates (1) into a smooth problem and then passes through an argument involving the KKT conditions of said smooth problem (valid under some constraint qualification) to arrive at the desired result, while the second verifies the result directly by establishing a link between the first-order properties of the objective of (1) and its non-smooth, convex surrogate about the current iterate. The latter approach does not depend on any constraint qualifications and also establishes that all requisite regularity assumptions are automatically satisfied by (1), which makes it a stronger result compared to the former. To the best of our knowledge, this marks the first time that such a direct proof of convergence has been rigorously established for this class of problems. In fact, the proof applies to any problem which involves the maximization of the point-wise minimum of a finite number of smooth, concave functions[2] over a convex set, and hence may be of broader interest to the signal processing community.

The overall complexity of this algorithm depends on the cost incurred in solving each non-smooth, convex subproblem. The prevailing approach for solving each subproblem is to first convert it into an equivalent smooth representation and then use general purpose convex optimization solvers, which utilize interior-point methods, for obtaining solutions. However, such a smooth reformulation has undesirable consequences from a computational standpoint, since it introduces additional constraints. In addition, when the number of variables is large, then using interior-point methods to solve each subproblem can become very computationally expensive.

In order to reduce the complexity of solving each subproblem, we explore the idea of using fast, first-order methods (FOMs) for efficiently solving each subproblem in its non-smooth representation. It is shown that the cost function of each subproblem possesses special structure in the form of being expressible as the maximization of a bilinear function over a convex set. This is the key step, since this form of structure can be exploited by the *Nesterov smoothing* technique [12] and Nemirovski's *saddle point reformulation* approach [13] to allow the development of specialized FOMs for efficiently solving each SCA subproblem. These methods possess the desirable property of low per-iteration complexity and an improved iteration complexity over standard methods for non-smooth, convex-optimization (e.g., projected subgradient methods). In addition to these two methods, we also propose using an inexact version of the classical Alternating Direction Method of Multipliers (ADMM) algorithm [14]–[16] for solving each SCA subproblem, which drops expensive, exact variable updates in favor of computationally cheaper, albeit inexact updates.

In order to test the performance of the proposed FOM-based SCA techniques, we apply them to the problem of designing a max-min fair multicast beamformer. We consider the following two complimentary scenarios within the multicast context.

- A traditional multicasting scenario, where the number of users is much larger than the number of transmit antennas, under a transmit *sum power constraint*.
- A Massive MIMO multicasting scenario, where the number of transmit antennas is larger than the number of users, under *per-antenna power constraints* (PAPCs).

A comprehensive set of simulations is carried out to evaluate the performance of the proposed methods relative to the existing state-of-the-art in both scenarios.

Relative to [1], which describes the inexact-ADMM based approach for solving each SCA subproblem, the journal version adds the FOMs based on Nesterov smoothing and Nemirovski's saddle point reformulation approach, the convergence proofs for the SCA algorithm (the stronger result is included in the appendix while the other is contained in the supplementary material), and comprehensive numerical results and comparisons.

C. Organization

The rest of the paper is organized as follows. We begin with some technical preliminaries in Section II, followed by a description of the single-group multicasting problem in Section III. Our problem formulation is presented in Section IV, while Section V describes the overall SCA approach and the proposed methods for solving each subproblem. Simulation results are provided in Section VI. Conclusions are drawn in Section VII.

D. Notation

Superscript ^{H} is used to denote the Hermitian (conjugate) transpose of a vector/matrix, while ^{T} denotes plain transposition. Capital boldface is reserved for matrices, while vectors are denoted by small boldface. Scalars are represented in the normal face, while calligraphic font is used to denote sets. The set of natural numbers is represented by $\mathbb{N}$. We denote the $N-$dimensional real and complex Euclidean spaces by $\mathbb{R}^N$ and $\mathbb{C}^N$ respectively. The convex hull of a finite number of points $\{\mathbf{x}_i\}_{i=1}^N$ is denoted as $\mathrm{conv}(\mathbf{x}_i|\forall\, i=\{1,\cdots,N\})$. The domain of a function $f:\mathbb{R}^N\rightarrow\mathbb{R}$ is defined as $\mathrm{dom} f=\{\mathbf{x}\in\mathbb{R}^N|f(\mathbf{x})<+\infty\}$. If $f(.)$ is differentiable, its gradient operator is denoted

[2]or, stated equivalently, the minimization of the point-wise maximum of a finite number of smooth, convex functions.

by $\nabla(f(.))$; otherwise, $\partial f(.)$ denotes the subdifferential operator of f. For general f (differentiable or non-differentiable), the directional derivative at a point $\mathbf{x}$ in the direction $\mathbf{d}$ is given by $f'(\mathbf{x};\mathbf{d})$. For a set $\mathcal{X} \subset \mathbb{R}^N$, we use $\bar{\mathcal{X}}$ to denote its closure and $\partial\mathcal{X}$ to represent its boundary. Finally, the empty set is denoted by $\emptyset$.

II. Preliminaries

We will use the following standard definitions from convex analysis [11].

- Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ denote a single-valued, convex, lower semi-continuous function such that dom$f \neq \emptyset$. Then, the *Fenchel conjugate* of f is defined as
$$f^*(\mathbf{y}) = \sup_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x}^T\mathbf{y} - f(\mathbf{x}) \tag{2}$$
Note that $f^*(\mathbf{y})$ is also convex since it corresponds to the point-wise supremum of linear functions in $\mathbf{y}$.
- Consider a multi-valued function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. F is said to be *monotone* if
$$(F(\mathbf{x}) - F(\mathbf{y}))^T(\mathbf{x}-\mathbf{y}) \geq 0, \forall\, \mathbf{x},\mathbf{y} \in \text{dom}F \tag{3}$$
For the special case of $n = 1$, (3) implies that F is a monotonically increasing function. Thus (3) can be interpreted as an extension of the concept of monotonicity to the general case of $n \geq 1$. As an example, the gradient of a convex, differentiable function is monotone.
- A multi-valued function F is said to be *strongly monotone* with parameter $\sigma \in \mathbb{R} > 0$ if
$$(F(\mathbf{x}) - F(\mathbf{y}))^T(\mathbf{x}-\mathbf{y}) \geq \sigma\|\mathbf{x}-\mathbf{y}\|_2^2, \forall\, \mathbf{x},\mathbf{y} \in \text{dom}F \tag{4}$$
As an example, the gradient of a strongly convex, differentiable function is strongly monotone.
- Let $\mathbf{y} \in \mathbb{R}^n$ and $\mathcal{X} \subseteq \mathbb{R}^n$ be a set. Then the distance of the point $\mathbf{y}$ from the set $\mathcal{X}$ is defined as
$$d(\mathbf{y},\mathcal{X}) = \inf_{\mathbf{x}\in\mathcal{X}} \|\mathbf{y}-\mathbf{x}\|_2 \tag{5}$$

III. Single-Group Multicast Beamforming

Consider a downlink transmission scenario where a single base station (BS) equipped with N antennas wishes to transmit common information to a group of M single-antenna users over the same frequency band. Assuming perfect channel state information (CSI) is available at the BS, the goal of multicast beamforming is to exploit this knowledge and the spatial diversity offered by the multiple transmit antennas to steer transmitted power towards the group of desired users while limiting leakage to nearby co-channel users and systems [4]. Let $\mathbf{w} \in \mathbb{C}^N$ denote the desired beamforming vector and $\mathbf{h}_m \in \mathbb{C}^N$ denote the downlink channel between the BS and the m^{th} user, which is modeled as complex, random vector that is flat in frequency and quasi-static in time. Using a unit-norm beamformer, the BS transmits a zero-mean , unit-variance, common information bearing signal x to all M users. The corresponding received signal at the mth user is given by

$$y_m = \mathbf{w}^H\mathbf{h}_m x + z_m,\ \forall\, m \in \mathcal{M} \tag{6}$$

where z_m is zero-mean, wide-sense stationary additive noise at the mth receiver with variance σ_m^2, and is assumed to be independent of x and $\mathbf{h}_m$. The received signal-to-noise ratio (SNR) at the mth user is then given by $\frac{|\mathbf{w}^H\mathbf{h}_m|^2}{\sigma_m^2} = \mathbf{w}^H\mathbf{R}_m\mathbf{w}$, where $\mathbf{R}_m := \frac{\mathbf{h}_m\mathbf{h}_m^H}{\sigma_m^2} \succeq \mathbf{0}, \forall\, m \in \mathcal{M}$. Since all users are required to decode the same information bearing signal, the maximum common data rate attainable is determined by the minimum SNR. Hence, the objective is to maximize the minimum received SNR subject to unit-norm transmit sum power constraints, which leads to the following max-min fair formulation

$$\max_{\mathbf{w}\in\mathbb{C}^N} \ \min_{m\in\mathcal{M}} \mathbf{w}^H\mathbf{R}_m\mathbf{w} \tag{7a}$$
$$\text{s.t.} \quad \|\mathbf{w}\|_2 \leq 1 \tag{7b}$$

which is a non-convex QCQP problem of the form of (1). When $M \geq N$, (7) is NP–Hard [4]. Another variant of (7) seeks to minimize the transmitted power subject to user-specific quality of service (QoS) guarantees which are formulated in terms of minimum received SNR per user. From an optimization point of view, the two formulations are essentially equivalent [4].

Several algorithms have been proposed for obtaining approximate solutions to (7), ranging from SDR followed by randomization [4], to alternating maximization [17], and SCA [18], [19] (applied to the QoS version), which exhibit the best overall performance. Several low complexity algorithms also exist [20]–[22], although they are unable to match the performance of SCA. In [23], it was proposed to approximate (7) by a proportionally fair formulation and a first-order based *Multiplicative Update* (MU) algorithm was introduced, which was demonstrated to attain performance comparable to that of SCA at much lower complexity. This algorithm is the current state-of-the-art for solving (7) in the traditional multicasting scenario (i.e., when $M \geq N$).

Recently, Massive MIMO [24], [25], which refers to the concept of equipping cellular base stations with a very large number of transmit antennas, has emerged as a very promising paradigm for possible implementation in a future 5G wireless broadband standard [26], [27]. When considering the multicasting problem in the context of such a scenario (i.e., when $M < N$), it is more practical to replace the transmit sum power constraint (7b) by per antenna power constraints (PAPCs), since every antenna in a large scale array will be equipped with its own power amplifier, whose linearity is the performance limiting factor. Hence the single-group multicasting problem in the Massive MIMO setting can be expressed as [30]

$$\max_{\mathbf{w}\in\mathbb{C}^N} \ \min_{m\in\mathcal{M}} \mathbf{w}^H\mathbf{R}_m\mathbf{w} \tag{8a}$$
$$\text{s.t.} \quad |w(i)|^2 \leq P_i,\ \forall\, i = 1,2,\cdots,N \tag{8b}$$

where P_i denotes the power constraint for the ith transmit antenna, $\forall\, i = 1,2,\cdots,N$. Although (8) is a non-convex optimization problem, we are currently not aware of any result which establishes it as being NP–Hard. Note that (8) is a QCQP problem of the form (1), since computing the projection onto the set of constraints (8b) decouples into computing the projection of each element of $\mathbf{w}$ onto its corresponding element-wise constraint, which admits a closed form solution. When N is large

(i.e., of the order of hundreds), then solving (8) via SDR or SCA (where each convex subproblem is solved using standard interior point methods) is very computationally demanding.

IV. Problem Formulation

As a first step, we convert (1) from the complex domain to the real domain. Define the variables $\tilde{\mathbf{x}} := [\mathbf{x}_r^T, \mathbf{x}_i^T]^T \in \mathbb{R}^{2N}$, $\mathbf{x}_r = \mathrm{Re}\{\mathbf{x}\}, \mathbf{x}_i = \mathrm{Im}\{\mathbf{x}\}$ and the matrices $\tilde{\mathbf{A}}_m \in \mathbb{R}^{2N\times 2N}$ as

$$\tilde{\mathbf{A}}_m = \begin{bmatrix} \mathrm{Re}\{\mathbf{A}_m\} & -\mathrm{Im}\{\mathbf{A}_m\} \\ \mathrm{Im}\{\mathbf{A}_m\} & \mathrm{Re}\{\mathbf{A}_m\} \end{bmatrix}, \ \forall\, m \in \mathcal{M} \tag{9}$$

Note that $\mathbf{A}_m \succeq \mathbf{0}$ if and only if $\tilde{\mathbf{A}}_m \succeq \mathbf{0}$, $\forall\, m \in \mathcal{M}$. Hence, (1) can be equivalently expressed in terms of real variables as

$$\max_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \ \min_{m\in\mathcal{M}} \tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} \tag{10a}$$

$$\text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \tag{10b}$$

where $\tilde{\mathcal{F}}$ is the representation of the feasible set $\mathcal{F}$ in terms of real variables.

We now reformulate the problem into the following equivalent form.

$$\begin{aligned} \max_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \ \min_{m\in\mathcal{M}} \tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} \quad & \Leftrightarrow \min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \ -\min_{m\in\mathcal{M}} \tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} \\ \text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \quad & \quad \text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \\ & \Leftrightarrow \min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \ \max_{m\in\mathcal{M}} \tilde{\mathbf{x}}^T(-\tilde{\mathbf{A}}_m)\tilde{\mathbf{x}} \\ & \quad \text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \end{aligned}$$

Defining $\bar{\mathbf{A}}_m := -\tilde{\mathbf{A}}_m$, $\forall\, m \in \mathcal{M}$, we finally obtain

$$\min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \ \max_{m\in\mathcal{M}} \tilde{\mathbf{x}}^T \bar{\mathbf{A}}_m \tilde{\mathbf{x}} \tag{11a}$$

$$\text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \tag{11b}$$

Henceforth, we will work with this formulation. Note that the problem still remains non-convex since the point-wise maximum of concave functions is not convex. We now present the following approximation strategies for obtaining sub-optimal solutions of (11) via low-complexity algorithms.

V. Successive Convex Approximation

SCA is an iterative procedure where the original non-convex problem is approximated by a sequence of convex problems. Starting from a feasible point $\tilde{\mathbf{x}}^{(0)} \in \tilde{\mathcal{F}}$, the algorithm proceeds as follows. At every iteration $n \geq 0$, we construct a convex majorization function of the non-convex cost function (11a) about the current iterate $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(n)}$ in the following manner.

Define $f(\tilde{\mathbf{x}}) := \max_{m\in\mathcal{M}} u_m(\tilde{\mathbf{x}})$, where we have $u_m(\tilde{\mathbf{x}}) := \tilde{\mathbf{x}}^T \bar{\mathbf{A}}_m \tilde{\mathbf{x}}$, $\forall\, m \in \mathcal{M}$. Since $u_m(\tilde{\mathbf{x}})$ is concave $\forall\, m \in \mathcal{M}$, on locally linearizing $u_m(\tilde{\mathbf{x}})$ about the point $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(n)}$, we obtain the following upper bound.

$$\begin{aligned} u_m(\tilde{\mathbf{x}}) \leq \ & u_m(\tilde{\mathbf{x}}^{(n)}) + \nabla u_m(\tilde{\mathbf{x}}^{(n)})^T(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}^{(n)}) \\ = \ & (2\bar{\mathbf{A}}_m \tilde{\mathbf{x}}^{(n)})^T \tilde{\mathbf{x}} - \tilde{\mathbf{x}}^{(n)T} \bar{\mathbf{A}}_m \bar{\mathbf{x}}^{(n)} \\ = \ & \mathbf{c}_m^{(n)T} \tilde{\mathbf{x}} + d_m^{(n)}, \forall\, m \in \mathcal{M} \end{aligned} \tag{12}$$

where $\mathbf{c}_m^{(n)} := 2\bar{\mathbf{A}}_m \tilde{\mathbf{x}}^{(n)} \in \mathbb{R}^{2N}$, $d_m^{(n)} := -\tilde{\mathbf{x}}^{(n)T} \bar{\mathbf{A}}_m \tilde{\mathbf{x}}^{(n)} \in \mathbb{R}$, $\forall\, m \in \mathcal{M}$.

Define $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) := \max_{m\in\mathcal{M}} \mathbf{c}_m^{(n)T} \tilde{\mathbf{x}} + d_m^{(n)}$. It then follows that $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ possesses the following properties.

(A1) $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ is a convex function in $\tilde{\mathbf{x}}$ (being piecewise linear).

(A2) $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ is continuous in $(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$, but non-differentiable in $\tilde{\mathbf{x}}$.

(A3) $v(\tilde{\mathbf{x}}^{(n)}, \tilde{\mathbf{x}}^{(n)}) = f(\tilde{\mathbf{x}}^{(n)})$, $\forall\, \tilde{\mathbf{x}}^{(n)} \in \tilde{\mathcal{F}}$

(A4) $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) \geq f(\tilde{\mathbf{x}})$, $\forall\, \tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)} \in \tilde{\mathcal{F}}$

Properties (A3) and (A4) together imply that the piecewise linear function $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ is an upper bound of the original function $f(\tilde{\mathbf{x}})$ which is tight at $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(n)}$. In every iteration n, we replace $f(\tilde{\mathbf{x}})$ by its piecewise linear approximation about $\tilde{\mathbf{x}}^{(n)}$ to obtain the non-smooth, convex subproblem

$$\min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \ \max_{m\in\mathcal{M}} \mathbf{c}_m^{(n)T} \tilde{\mathbf{x}} + d_m^{(n)} \tag{13a}$$

$$\text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \tag{13b}$$

The overall algorithm is given by

Algorithm 1: SCA.

Initialization: Randomly generate a starting point $\tilde{\mathbf{x}}^{(0)} \in \tilde{\mathcal{F}}$ and set $n := 0$.

Repeat

- Compute $\tilde{\mathbf{x}}^{(n+1)} \in \arg\min_{\tilde{\mathbf{x}}\in\tilde{\mathcal{F}}} v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$
- Set $n := n + 1$.

Until termination criterion is met

Note that a feasible starting point can always be determined trivially since, by our assumption, $\tilde{\mathcal{F}}$ is a simple, convex set. Furthermore, we have the following chain of inequalities

$$f(\tilde{\mathbf{x}}^{(n+1)}) \leq v(\tilde{\mathbf{x}}^{(n+1)}, \tilde{\mathbf{x}}^{(n)}) \leq v(\tilde{\mathbf{x}}^{(n)}, \tilde{\mathbf{x}}^{(n)}) = f(\tilde{\mathbf{x}}^{(n)}), \quad \forall\, n = 0, 1, \cdots, \tag{14}$$

The first inequality stems from (A4), whereas the second inequality follows since $\tilde{\mathbf{x}}^{n+1}$ is an optimal solution of $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$, and the last equality is due to (A3). As a result, it can easily be seen that the algorithm produces a sequence of iterates with monotonically non-increasing cost. In addition, we have the following result, which establishes that the algorithm is also provably convergent.

Proposition 1: The iterates generated by Algorithm 1 globally converge to the set of d-stationary solutions of (11); i.e., we have

$$\lim_{n\to\infty} d(\tilde{\mathbf{x}}^{(n)}, \tilde{\mathcal{F}}^*) = 0$$

where $\tilde{\mathcal{F}}^*$ is the set of d-stationary solutions of (11).

Proof: The proof is deferred to Appendix A. ■

The main computational cost incurred is in solving a subproblem of the form (13) at each iteration. Since (13) does not have a closed form solution, we must resort to iterative methods for solving each SCA subproblem.

The standard procedure for solving (13) is to first transform it into its epigraph representation, which yields the following smooth optimization problem

$$\min_{t\in\mathbb{R},\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \quad t \tag{15a}$$

$$\text{s.t.} \quad \mathbf{c}_m^{(n)T}\tilde{\mathbf{x}} + d_m^{(n)} \leq t, \forall\, m \in \mathcal{M} \tag{15b}$$

$$\tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \tag{15c}$$

Problem (15) can be cast as a Second-order Cone Programming (SoCP) problem, and solved using interior point methods at a worst case computational complexity of $O(N^3\sqrt{M+|\tilde{\mathcal{F}}|})$ [31], where $|\tilde{\mathcal{F}}|$ is the smallest number of constraints required to define $\tilde{\mathcal{F}}$ (e.g., $|\tilde{\mathcal{F}}| = 1$ in the traditional multicasting scenario while $|\tilde{\mathcal{F}}| = N$ in the Massive MIMO setting). Coupled with the fact that each such problem has to be solved multiple times, it is clear that the overall cost incurred is expensive and can become a serious computational burden for large M or (especially) N. In hindsight, the high computational cost stems from equivalently reformulating (13) as a smooth optimization problem (15). This forces us to consider developing low-complexity alternatives for efficiently solving each non-smooth subproblem (13).

One may consider using projected sub-gradient methods [32], [33] for solving (13), which are well suited for minimizing non-differentiable convex functions subject to simple convex constraints. These methods possess the desirable property of having low per-iteration complexity in contrast to interior-point methods. The main drawback of using such methods is their slow convergence rate, which ultimately limits the attainable accuracy. Indeed, if the constraint set is compact, then for appropriately chosen step-sizes, the number of iterations required to obtain an ϵ optimal solution [3] is upper bounded by $O(\frac{1}{\epsilon^2})$. Furthermore, from results in Information-Based Complexity Theory [34], it is known that the number of iterations required to construct an ϵ optimal solution by a FOM, with knowledge of the value and subgradients of the cost function only, is no less than $O(\frac{1}{\epsilon^2})$.

Since the iteration outer bound of projected subgradient methods matches this lower bound, they are "order" optimal, which would seem to indicate that one cannot devise a FOM with a faster convergence rate for solving problems of the form (13). However, it is important to remember that projected subgradient does not utilize any specific structure the problem may possess. This implies that it may indeed be possible to devise FOMs which explicitly exploit problem structure to achieve ϵ-optimality in fewer iterations. Note that this observation is not a contradiction of the results of [34], since such FOMs are not oblivious to problem-specific structure.

We now demonstrate that the cost function of (13a) possesses specific structure which can be exploited. First, we define the matrix $\mathbf{C}^{(n)} := [\mathbf{c}_1^{(n)}, \cdots, \mathbf{c}_M^{(n)}]^T \in \mathbb{R}^{M\times 2N}$ and the vector $\mathbf{d} = [d_1^{(n)}, \cdots, d_M^{(n)}]^T \in \mathbb{R}^M$. Then, $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ can be equivalently expressed as

$$v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) = \max_{m\in\mathcal{M}} \mathbf{c}_m^{(n)T}\tilde{\mathbf{x}} + d_m^{(n)} \tag{16a}$$

$$= \max_{\substack{\tilde{\mathbf{y}}\geq\mathbf{0}, \mathbf{1}^T\tilde{\mathbf{y}}=1,\\ \tilde{\mathbf{y}}\in\mathbb{R}^M}} (\mathbf{C}^{(n)}\tilde{\mathbf{x}} + \mathbf{d}^{(n)})^T\tilde{\mathbf{y}} \tag{16b}$$

To see that this holds, note that (16b) corresponds to maximizing a linear function over the $M-$dimensional probability simplex. Hence, the maximum is attained at one of the vertices of the simplex, which are given by the canonical basis vectors of $\mathbb{R}^M$. From the definition of $\mathbf{C}^{(n)}$ and $\mathbf{d}^{(n)}$, it is then evident that the equivalence holds. Defining $\phi^{(n)}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) := (\mathbf{C}^{(n)}\tilde{\mathbf{x}} + \mathbf{d}^{(n)})^T\tilde{\mathbf{y}}$ and Δ_M as the $M-$dimensional probability simplex, (13) can be equivalently reformulated as

$$\min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \quad \max_{\tilde{\mathbf{y}}\in\Delta_M} \phi^{(n)}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \tag{17a}$$

$$\text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \tag{17b}$$

This special problem structure is the cornerstone of our algorithmic approach for solving (17), as it is well suited for the application of several specialized FOMs. We now discuss methods available in the existing optimization literature which are capable of solving problems of the form (17) efficiently.

A. Smoothing via Conjugation

We first point out that one can consider using the *log-sum-exp* function as a smooth surrogate for the cost function of (13), since the log-sum-exp can be interpreted as a differentiable approximation of the point-wise maximum function [35, p. 72]. We now show that it is possible to rigorously derive a more general result.

In his seminal work [12], Nesterov considered the following problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad q(\mathbf{x}) \tag{18a}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{C} \tag{18b}$$

where $\mathcal{C} \subseteq \mathbb{R}^n$ is a simple, compact, convex set and $q : \mathbb{R}^n \to \mathbb{R}$ is a non-differentiable, Lipschitz continuous convex function which admits the following representation

$$q(\mathbf{x}) = \sup_{\mathbf{y}\in\operatorname{dom} p} ((\mathbf{C}\mathbf{x} + \mathbf{d})^T\mathbf{y} - p(\mathbf{y}))$$

$$= p^*(\mathbf{C}\mathbf{x} + \mathbf{d}) \tag{19}$$

where $\mathbf{C} \in \mathbb{R}^{m\times n}, \mathbf{d} \in \mathbb{R}^m$, $p : \mathbb{R}^m \to \mathbb{R}$ is a closed, convex function with bounded domain and $p^*(\mathbf{x})$ denotes the Fenchel conjugate of $p(\mathbf{y})$. Hence, $q(\mathbf{x})$ belongs to the class of non-differentiable convex functions which can be expressed as Fenchel conjugates of other convex functions. Nesterov's technique for solving (18) is based on *smoothing*-constructing a smooth approximation of $q(\mathbf{x})$ which possesses a Lipschitz continuous gradient, followed by minimizing the approximation by an accelerated FOM [36]. We now succinctly summarize the details of this technique as presented in [37].

Let $r(\mathbf{y})$ be a continuous, strongly convex function defined over the closure of the domain of p such that $\inf_{\mathbf{y}\in\operatorname{dom} p} r(\mathbf{y}) = 0$.

[3] Here, we define ϵ optimality in terms of the cost function.

Then, consider the function

$$q_\mu(\mathbf{x}) = \sup_{\mathbf{y}\in\mathrm{dom}\, p} ((\mathbf{C}\mathbf{x}+\mathbf{d})^T\mathbf{y} - p(\mathbf{y}) - \mu r(\mathbf{y})) = (p+\mu r)^*(\mathbf{C}\mathbf{x}+\mathbf{d}) \quad (20)$$

where $(p+\mu r)^*(\mathbf{x})$ is the Fenchel conjugate of the strongly convex function $(p+\mu r)(\mathbf{y})$ and $\mu \in \mathbb{R} > 0$. Nesterov established that the function $q_\mu(\mathbf{x})$ possesses the following properties.

(B1) $q_\mu(\mathbf{x})$ is well defined and differentiable at all $\mathbf{x}$, and $\nabla q_\mu(\mathbf{x})$ is Lipschitz continuous with Lipschitz constant $L_\mu \propto \frac{1}{\mu}$.

(B2) $q_\mu(\mathbf{x}) \leq q(\mathbf{x}) \leq q_\mu(\mathbf{x}) + \mu D$, where $D = \sup_{\mathbf{y}\in\mathrm{dom}\, p} r(\mathbf{y})$.

Hence, $q_\mu(\mathbf{x})$ can be interpreted as a smooth approximation of $q(\mathbf{x})$, where μ is a parameter which controls the level of smoothing. Replacing $q(\mathbf{x})$ by $q_\mu(\mathbf{x})$ in (18), we obtain the smooth optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad q_\mu(\mathbf{x}) \quad (21a)$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{C} \quad (21b)$$

which can be solved to a numerical accuracy of ϵ_μ in $O(\sqrt{\frac{L_\mu}{\epsilon_\mu}})$ iterations using an accelerated FOM [12], [36]. It can also be shown that we have

$$q(\mathbf{x}) - q^*(\mathbf{x}) \leq q_\mu(\mathbf{x}) - q_\mu^*(\mathbf{x}) + \mu D \quad (22)$$

where $q^*(\mathbf{x})$ and $q_\mu^*(\mathbf{x})$ denote the optimal values of (18) and (21) respectively. If we define $\epsilon_\mu = q_\mu(\mathbf{x}) - q_\mu^*(\mathbf{x})$, then from (22) it is evident that an ϵ-optimal solution of (18) can be obtained by solving (21) to a numerical accuracy of $\epsilon_\mu = \epsilon - \mu D$; i.e., the smooth approximation (21) is solved to a higher degree of accuracy than the original non-smooth problem (18).

The role of the smoothing parameter μ is now discussed. Clearly, a smaller μ results in less smoothing which corresponds to a more accurate approximation of $q(\mathbf{x})$, but results in more iterations required to solve (21) via an accelerated FOM since L_μ is large. On the other hand, a larger μ produces a more smooth approximation (since L_μ is small), but results in having to solve (21) to a higher degree of accuracy in order to obtain an ϵ-optimal solution of (18). Overall, for a given ϵ, if we choose $\mu = \frac{\epsilon}{2D}$, it can be shown that using an accelerated FOM to solve (21), one can obtain an ϵ-optimal solution of (18) in no more than $O(\frac{1}{\epsilon})$ iterations, which represents an order of magnitude improvement over the $O(\frac{1}{\epsilon^2})$ iterations required by subgradient methods.

From (16), it is clear that Nesterov's smoothing technique can be applied to solve (17). Define the function $r(\tilde{\mathbf{y}}) := \sum_{m=1}^M \tilde{y}_m \log \tilde{y}_m + \log M$, which is continuous and strongly convex everywhere over Δ_M. Then, the smooth approximation of $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ is given by

$$v_\mu(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) = \sup_{\tilde{\mathbf{y}}\in\Delta_M} (\mathbf{C}^{(n)}\tilde{\mathbf{x}} + \mathbf{d}^{(n)})^T\tilde{\mathbf{y}} - \mu r(\tilde{\mathbf{y}}) = \mu \log\left(\sum_{m=1}^M \exp\left(\frac{\mathbf{c}_m^{(n)T}\tilde{\mathbf{x}} + d_m^{(n)}}{\mu}\right)\right) - \mu \log M \quad (23)$$

where $\mu \in \mathbb{R} > 0$ is the smoothing parameter. Note that if we set $\mu = 1$ and neglect the last term, we re-obtain the log-sum-exp function. Replacing $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ by $v_\mu(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ in (17), our optimization problem becomes

$$\min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \quad \log\left(\sum_{m=1}^M \exp\left(\frac{\mathbf{c}_m^{(n)T}\tilde{\mathbf{x}} + d_m^{(n)}}{\mu}\right)\right) \quad (24a)$$

$$\text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \quad (24b)$$

which is a smooth optimization problem, and can be solved using an accelerated FOM. Utilizing Nesterov's smoothing technique to solve each SCA subproblem, the overall SCA algorithm is given by

Algorithm 2: Nesterov SCA.

Initialization: Randomly generate a feasible starting point $\tilde{\mathbf{x}}^{(0)} \in \tilde{\mathcal{F}}$. Set $n := 0$.

Repeat

- Compute $\tilde{\mathbf{x}}^{(n+1)} \in \arg\min_{\tilde{\mathbf{x}}\in\tilde{\mathcal{F}}} v_\mu(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$ using an accelerated FOM.
- Set $n := n+1$.

Until termination criterion is met.

The per-iteration cost of an accelerated FOM is dominated by the cost of forming the gradient (since all projections can be computed in closed form), which incurs a modest cost of $O(MN)$ flops in our case. In order to obtain further savings in computation, at each SCA iteration n, one can warm-start the accelerated FOM with the current iterate $\tilde{\mathbf{x}}^{(n)}$.

B. Convex-Concave Saddle Point Reformulation

The following technique is attributed to Nemirovski [13]. Let $\mathcal{X} \subset \mathbb{R}^n, \mathcal{Y} \subset \mathbb{R}^m$ be convex, compact sets and let $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be a continuous function which is convex in $\mathbf{x} \in \mathbb{R}^n$ and concave in $\mathbf{y} \in \mathbb{R}^m$. Define the function $g(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x}, \mathbf{y})$ and consider the following problem

$$\min_{\mathbf{x}\in\mathcal{X}} g(\mathbf{x}) = \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x}, \mathbf{y}) \quad (25)$$

From Sion's minimax equality theorem [38], we have that

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{y}\in\mathcal{Y}} \min_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x}, \mathbf{y}) \quad (26)$$

which implies that the optimal solution $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*) \in \mathcal{X} \times \mathcal{Y} := \mathcal{Z}$ of (25) corresponds to a saddle point of $\phi(\mathbf{x}, \mathbf{y})$, i.e., we have

$$\phi(\mathbf{x}^*, \mathbf{y}) \leq \phi(\mathbf{x}^*, \mathbf{y}^*) \leq \phi(\mathbf{x}, \mathbf{y}^*) \,\forall\, (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y} \quad (27)$$

Given a candidate solution $\bar{\mathbf{z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \mathcal{Z}$, its degree of suboptimality can be evaluated via the following primal-dual gap

$$\epsilon_{\mathrm{sad}}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \max_{\mathbf{y}\in\mathcal{Y}} \phi(\bar{\mathbf{x}}, \mathbf{y}) - \min_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x}, \bar{\mathbf{y}}) \quad (28)$$

Using (28), we can obtain the following inequality

$$\begin{aligned} g(\mathbf{x}) - \min_{\mathbf{x}\in\mathcal{X}} g(\mathbf{x}) &= \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x},\mathbf{y}) - \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x},\mathbf{y}) \\ &\leq \max_{\mathbf{y}\in\mathcal{Y}} \phi(\mathbf{x},\mathbf{y}) - \min_{\mathbf{x}\in\mathcal{X}} \phi(\mathbf{x},\mathbf{y}) \\ &= \epsilon_{\mathrm{sad}}(\mathbf{x},\mathbf{y}) \end{aligned} \tag{29}$$

Hence, the $\mathbf{x}$ component of a point $(\mathbf{x},\mathbf{y}) \in \mathcal{Z}$ for which $\epsilon_{\mathrm{sad}}(\mathbf{x},\mathbf{y}) \leq \epsilon$ also corresponds to an ϵ-optimal solution of (25). The overall problem of determining a saddle point $\mathbf{z}^* = (\mathbf{x}^*,\mathbf{y}^*) \in \mathcal{Z}$ of $\phi(\mathbf{x},\mathbf{y})$ can be cast as solving the associated variational inequality

$$\Psi(\mathbf{z}^*)^T(\mathbf{z}-\mathbf{z}^*) \geq 0,\ \forall\, \mathbf{z} \in \mathcal{Z} \tag{30}$$

where we define the *saddle-point operator* Ψ as

$$\Psi(\mathbf{z}) := \Psi(\mathbf{x},\mathbf{y}) = \begin{bmatrix} \nabla_{\mathbf{x}}\phi(\mathbf{x},\mathbf{y}) \\ -\nabla_{\mathbf{y}}\phi(\mathbf{x},\mathbf{y}) \end{bmatrix} \tag{31}$$

In addition, it can be shown that Ψ is monotone and Lipschitz continuous on $\mathcal{Z}$ (see [13]). The field of variational inequalities is a well studied subject which finds application in diverse areas (see [39] and references therein) and can be analyzed using tools from fixed-point theory. Hence, one can attempt to solve (30) by a fixed-point iteration, such as the *generalized* projected gradient method [40]. However, the convergence of this method is not guaranteed (see [40] for counter-examples), unless Ψ is *strongly* monotone. This restriction can be overcome by using a modified version of the method, known as the *extragradient* algorithm [41], which only requires the assumption of monotonicity and Lipschitz continuity of Ψ to guarantee convergence to the solution of (30).

The idea of classical projected gradient descent was extended to non-Euclidean geometries by the *Mirror Descent* algorithm [42], [43], which uses a distance generating function to exploit the specific geometry of the constraint set. In [13], Nemirovski proposed a variant of the Mirror Descent algorithm, known as the *Mirror-Prox* algorithm, for solving variational inequalities of the form (30), which can be interpreted as a generalization of the extragradient algorithm to non-Euclidean geometries. We now summarize the details of the Mirror-Prox algorithm as presented in [44, Section 5.2.3].

Let the sets $\mathcal{X}$ and $\mathcal{Y}$ be endowed with norms $\|.\|_{\mathcal{X}}$ and $\|.\|_{\mathcal{Y}}$ respectively. Assume that $\phi(\mathbf{x},\mathbf{y})$ is $(\beta_{11},\beta_{12},\beta_{21},\beta_{22})$-smooth in the following sense.

$$\|\nabla_{\mathbf{x}}\phi(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{x}}\phi(\mathbf{x}',\mathbf{y})\|_{\mathcal{X},^*} \leq \beta_{11}\|\mathbf{x}-\mathbf{x}'\|_{\mathcal{X}}, \tag{32a}$$

$$\|\nabla_{\mathbf{y}}\phi(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}}\phi(\mathbf{x},\mathbf{y}')\|_{\mathcal{Y},^*} \leq \beta_{22}\|\mathbf{y}-\mathbf{y}'\|_{\mathcal{Y}}, \tag{32b}$$

$$\|\nabla_{\mathbf{x}}\phi(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{x}}\phi(\mathbf{x},\mathbf{y}')\|_{\mathcal{X},^*} \leq \beta_{12}\|\mathbf{y}-\mathbf{y}'\|_{\mathcal{Y}}, \tag{32c}$$

$$\|\nabla_{\mathbf{y}}\phi(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}}\phi(\mathbf{x}',\mathbf{y})\|_{\mathcal{Y},^*} \leq \beta_{21}\|\mathbf{x}-\mathbf{x}'\|_{\mathcal{X}}, \tag{32d}$$

$$\forall\, \mathbf{z} = (\mathbf{x},\mathbf{y}) \in \mathcal{Z},\ \mathbf{z}' = (\mathbf{x}',\mathbf{y}') \in \mathcal{Z}$$

where $\|.\|_{\mathcal{X},^*}$ and $\|.\|_{\mathcal{Y},^*}$ denote the dual norms of $\|.\|_{\mathcal{X}}$ and $\|.\|_{\mathcal{Y}}$ respectively. Define $\Phi_{\mathcal{X}}(\mathbf{x})$ to be a *mirror map* for $\mathcal{X}$, which possesses the following properties [44, Section 4.1]

(C1) $\Phi_{\mathcal{X}} : \mathcal{D}_{\mathcal{X}} \to \mathbb{R}$, where $\mathcal{D}_{\mathcal{X}} \subset \mathbb{R}^n$ is a non-empty, convex open set which contains $\mathcal{X}$ in its closure (i.e., $\mathcal{X} \subset \overline{\mathcal{D}}_{\mathcal{X}}$) and $\mathcal{X} \cap \mathcal{D}_{\mathcal{X}} \neq \emptyset$.

(C2) $\lim_{\mathbf{x}\to\partial\mathcal{D}_{\mathcal{X}}} \|\nabla\Phi_{\mathcal{X}}(\mathbf{x})\| \to \infty$

(C3) $\Phi_{\mathcal{X}}(\mathbf{x})$ is strongly convex and continuously differentiable on $\mathcal{D}_{\mathcal{X}}$.

(C4) The *Bregman Divergence* associated with $\Phi_{\mathcal{X}}$ is defined as

$$D_{\Phi_{\mathcal{X}}}(\mathbf{x},\mathbf{x}') := \Phi_{\mathcal{X}}(\mathbf{x}) - \Phi_{\mathcal{X}}(\mathbf{x}') - \nabla\Phi_{\mathcal{X}}(\mathbf{x}')^T(\mathbf{x}-\mathbf{x}'), \quad \forall\, \mathbf{x},\mathbf{x}' \in \mathcal{D}_{\mathcal{X}} \tag{33}$$

Similarly, define $\Phi_{\mathcal{Y}}(\mathbf{y})$ to be a mirror map for $\mathcal{Y}$. We now consider the mirror map $\Phi(\mathbf{z}) = \Phi(\mathbf{x},\mathbf{y}) = \Phi_{\mathcal{X}}(\mathbf{x}) + \Phi_{\mathcal{Y}}(\mathbf{y})$ for $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$, defined on $\mathcal{D} = \mathcal{D}_{\mathcal{X}} \times \mathcal{D}_{\mathcal{Y}}$. Define $\beta := \max(\beta_{11},\beta_{12},\beta_{21},\beta_{22})$ and $\alpha := \frac{1}{2\beta}$. The Mirror-Prox algorithm is then given by the following steps

Algorithm 3: Mirror Prox.

Initialization: Define $\mathbf{z}_j = [\mathbf{x}_j^T,\mathbf{y}_j^T]^T$, $\mathbf{w}_j = [\mathbf{u}_j^T,\mathbf{v}_j^T]^T$, $\Psi(\mathbf{z}_j) = [\nabla_{\mathbf{x}}\phi(\mathbf{x}_j,\mathbf{y}_j)^T, -\nabla_{\mathbf{y}}\phi(\mathbf{x}_j,\mathbf{y}_j)^T]^T$, and $\Psi(\mathbf{w}_j) = [\nabla_{\mathbf{x}}\phi(\mathbf{u}_j,\mathbf{v}_j)^T, -\nabla_{\mathbf{y}}\phi(\mathbf{u}_j,\mathbf{v}_j)^T]^T$ for $j \geq 0$. Let $\mathbf{z}_0 = \arg\min_{\mathbf{z}\in\mathcal{Z}\cap\mathcal{D}} \Phi(\mathbf{z})$. Set $j := 0$, $\mathbf{w}_0 = \mathbf{z}_0$.

Repeat

1) $\nabla\Phi(\mathbf{w}'_{j+1}) = \nabla\Phi(\mathbf{z}_j) - \alpha\Psi(\mathbf{z}_j)$
2) $\mathbf{w}'_{j+1} = \nabla\Phi^{-1}(\nabla\Phi(\mathbf{z}_j) - \alpha\Psi(\mathbf{z}_j))$
3) $\mathbf{w}_{j+1} = \arg\min_{\mathbf{z}\in\mathcal{Z}\cap\mathcal{D}} D_{\Phi}(\mathbf{z},\mathbf{w}'_{j+1})$
4) $\nabla\Phi(\mathbf{z}'_{j+1}) = \nabla\Phi(\mathbf{z}_j) - \alpha\Psi(\mathbf{w}_{j+1})$
5) $\mathbf{z}'_{j+1} = \nabla\Phi^{-1}(\nabla\Phi(\mathbf{z}_j) - \alpha\Psi(\mathbf{w}_{j+1}))$
6) $\mathbf{z}_{j+1} = \arg\min_{\mathbf{z}\in\mathcal{Z}\cap\mathcal{D}} D_{\Phi}(\mathbf{z},\mathbf{z}'_{j+1})$
7) Set $j := j+1$.

Until termination criterion is met.

Note that the functional $\nabla\Phi^{-1}$ exists and is well defined since the gradient of a strongly convex function is strongly monotone. In addition, the existence and uniqueness of the minimizers of steps 3) and 6) follow from properties (C2) and (C3) of mirror maps, respectively.

The overall algorithm consists of two iterations of Mirror Descent. The first three steps of the algorithm (i.e., going from $\mathbf{z}_j$ to $\mathbf{w}_{j+1}$) correspond to a single step of Mirror Descent, whereas in the subsequent three steps, a similar procedure is followed, albeit with a slight difference; the algorithm again starts from $\mathbf{z}_j$ (instead of $\mathbf{w}_{j+1}$), but uses an operator evaluation at $\mathbf{w}_{j+1}$ to obtain $\mathbf{z}_{j+1}$. If the mirror maps of $\mathcal{X}$ and $\mathcal{Y}$ are chosen to be $\frac{1}{2}\|\mathbf{x}\|_2^2$ and $\frac{1}{2}\|\mathbf{y}\|_2^2$ respectively, then it can be shown that Mirror-Prox reduces to the extragradient algorithm of [41]. In [13], Nemirovski established convergence of the ergodic average of the iterates $(\mathbf{x}_j,\mathbf{y}_j)$ generated by the algorithm. To be more specific, he proved the following sub-optimality bound in terms of the primal-dual gap.

$$\epsilon_{\mathrm{sad}}\left(\frac{1}{T}\sum_{j=0}^{T-1}\mathbf{x}_j, \frac{1}{T}\sum_{j=0}^{T-1}\mathbf{y}_j\right) \leq O\left(\frac{1}{T}\right) \tag{34}$$

Combining this result with (29) implies an iteration outer bound of $O(\frac{1}{\epsilon})$ for guaranteeing convergence to an ϵ-optimal solution of (25).

Note that problem (17) fits the framework proposed by Nemirovski, since it corresponds to a smooth, *bilinear* saddle-point reformulation of the non-smooth problem (13). Define $\Phi(\tilde{\mathbf{x}}) = \frac{1}{2}\|\tilde{\mathbf{x}}\|_2^2$ and $\Phi(\tilde{\mathbf{y}}) = \sum_{m=1}^{M} \tilde{y}_m \log \tilde{y}_m$ to be the mirror maps for the sets $\tilde{\mathcal{F}}$ and $\widetilde{\mathcal{Y}} := \{\tilde{\mathbf{y}} \in \Delta_M\}$, respectively. Then, the mirror map $\Phi(\tilde{\mathbf{z}})$ defined $\forall\, \tilde{\mathbf{z}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \in \widetilde{\mathcal{Z}} := \tilde{\mathcal{F}} \times \widetilde{\mathcal{Y}}$ is given by

$$\begin{aligned}\Phi(\tilde{\mathbf{z}}) &= \Phi(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \\ &= \Phi(\tilde{\mathbf{x}}) + \Phi(\tilde{\mathbf{y}}) \\ &= \frac{1}{2}\|\tilde{\mathbf{x}}\|_2^2 + \sum_{m=1}^{M} \tilde{y}_m \log \tilde{y}_m \end{aligned} \tag{35}$$

from which it follows that

$$\nabla\Phi(\tilde{\mathbf{z}}) = \begin{bmatrix} \tilde{\mathbf{x}} \\ \log \tilde{y}_1 + 1 \\ \vdots \\ \log \tilde{y}_M + 1 \end{bmatrix}, \nabla^{-1}\Phi(\tilde{\mathbf{z}}) = \begin{bmatrix} \tilde{\mathbf{x}} \\ \exp(\tilde{y}_1 - 1) \\ \vdots \\ \exp(\tilde{y}_M - 1) \end{bmatrix} \tag{36}$$

Furthermore, the Bregman Divergence associated with $\Phi(\tilde{\mathbf{z}})$ can be expressed as

$$\begin{aligned} D_\Phi(\tilde{\mathbf{z}}, \tilde{\mathbf{z}}') &= \Phi(\tilde{\mathbf{z}}) - \Phi(\tilde{\mathbf{z}}') - \nabla\Phi(\tilde{\mathbf{z}}')^T(\tilde{\mathbf{z}} - \tilde{\mathbf{z}}') \\ &= \frac{1}{2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}'\|_2^2 + \sum_{m=1}^{M} \tilde{y}_m \log \frac{\tilde{y}_m}{\tilde{y}'_m} - \sum_{m=1}^{M} (\tilde{y}_m - \tilde{y}'_m) \end{aligned} \tag{37}$$

where $\tilde{\mathbf{z}} = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}}), \tilde{\mathbf{z}}' = (\tilde{\mathbf{x}}', \tilde{\mathbf{y}}') \in \mathcal{D}$. Thus, the non-Euclidean projection problem

$$\begin{aligned} &\min_{\tilde{\mathbf{z}} \in \widetilde{\mathcal{Z}} \cap \mathcal{D}} D_\Phi(\tilde{\mathbf{z}}, \tilde{\mathbf{z}}') \\ &= \min_{\substack{\tilde{\mathbf{x}} \in \tilde{\mathcal{F}}, \\ \tilde{\mathbf{y}} \in \Delta_M}} \frac{1}{2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}'\|_2^2 + \sum_{m=1}^{M} \tilde{y}_m \log \frac{\tilde{y}_m}{\tilde{y}'_m} - \sum_{m=1}^{M} (\tilde{y}_m - \tilde{y}'_m) \end{aligned} \tag{38}$$

can be decomposed into the two problems

$$\min_{\tilde{\mathbf{x}} \in \tilde{\mathcal{F}}} \quad \frac{1}{2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}'\|_2^2, \tag{39a}$$

$$\min_{\tilde{\mathbf{y}} \in \Delta_M} \quad \sum_{m=1}^{M} \tilde{y}_m \log \frac{\tilde{y}_m}{\tilde{y}'_m} - \sum_{m=1}^{M} (\tilde{y}_m - \tilde{y}'_m) \tag{39b}$$

The first problem is an Euclidean projection onto $\tilde{\mathcal{F}}$, which can be solved in closed form, while the second problem involves a non-Euclidean projection onto the $M-$dimensional probability simplex, where "distances" are measured using the unnormalized Kullback-Leibler (KL) divergence. This problem admits a simple closed form solution [44, p. 302] given by

$$\tilde{\mathbf{y}} = \begin{cases} \tilde{\mathbf{y}}', & \tilde{\mathbf{y}}' \in \Delta_M \\ \dfrac{\tilde{\mathbf{y}}'}{\|\tilde{\mathbf{y}}'\|_1}, & \text{otherwise} \end{cases} \tag{40}$$

Furthermore, note that

$$\Psi(\tilde{\mathbf{z}}) = \begin{bmatrix} \nabla_{\tilde{\mathbf{x}}}\phi^{(n)}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \\ -\nabla_{\tilde{\mathbf{y}}}\phi^{(n)}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \end{bmatrix} = \begin{bmatrix} (\mathbf{C}^{(n)})^T\tilde{\mathbf{y}} \\ -(\mathbf{C}^{(n)}\tilde{\mathbf{x}} + \mathbf{d}^{(n)}) \end{bmatrix} \tag{41}$$

where the superscript n denotes the iteration index of the outer SCA loop. Finally, from (32), it can be verified that for a fixed n, we have $\beta_{11} = 0, \beta_{22} = 0, \beta_{12} = \beta_{21} = L$, where $L = \max_{m \in \mathcal{M}} \|\mathbf{c}_m^{(n)}\|_2$ is the Lipschitz constant of the functions $\mathbf{c}_m^{(n)T}\tilde{\mathbf{x}} + d_m^{(n)}, \forall\, m \in \mathcal{M}$. Thus, we obtain the step size $\alpha = \frac{1}{2L}$.

It now only remains to solve (17) according to the steps of the Mirror-Prox algorithm (Algorithm 3) with the mirror maps and saddle-point operator (along with the associated quantities) as defined above. The cost of each iteration of the Mirror-Prox algorithm is dominated by the formation of the saddle-point operator $\Psi(\tilde{\mathbf{z}})$, which requires only $O(MN)$ flops; all projections are again closed form operations. The overall SCA algorithm is now given by

Algorithm 4: Mirror-Prox SCA.

Initialization: Randomly generate a feasible starting point $\tilde{\mathbf{x}}^{(0)} \in \tilde{\mathcal{F}}$. Set $n := 0$.

Repeat

- Compute $\tilde{\mathbf{x}}^{(n+1)} \in \arg\min_{\tilde{\mathbf{x}} \in \tilde{\mathcal{F}}} \max_{\tilde{\mathbf{y}} \in \Delta_M} \phi^{(n)}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ using the Mirror-Prox algorithm.
- Set $n := n + 1$.

Until termination criterion is met.

C. Alternating Direction Method of Multipliers

We now propose an alternative low complexity method for solving each SCA subproblem. Define the indicator function of the constraint set $\tilde{\mathcal{F}}$ as

$$I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) := \begin{cases} 0, & \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \\ \infty, & \text{otherwise} \end{cases} \tag{42}$$

Then, consider the following equivalent reformulation of (13)

$$\min_{\tilde{\mathbf{x}} \in \mathbb{R}^{2N}} v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) \tag{43a}$$

$$= \min_{\tilde{\mathbf{x}} \in \mathbb{R}^{2N}} \omega(\mathbf{C}^{(n)}\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) \tag{43b}$$

where we have defined $\omega(\tilde{\mathbf{z}}, \tilde{\mathbf{x}}^{(n)}) := \max_{m \in \mathcal{M}}\{\tilde{z}_m + d_m^{(n)}\}$ and $\tilde{\mathbf{z}} \in \mathbb{R}^M$. Thus, the constrained minimization problem (13) is equivalent to minimizing the sum of two non-smoooth, convex functions. In order to ease the burden of notation, we suppress the explicit dependence of $\omega(.,.)$ on $\tilde{\mathbf{x}}^{(n)}$ and equivalently express (43b) as

$$\min_{\tilde{\mathbf{x}} \in \mathbb{R}^{2N}} \omega(\mathbf{C}^{(n)}\tilde{\mathbf{x}}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) \tag{44}$$

Defining $\tilde{\mathbf{z}} := \mathbf{C}^{(n)}\tilde{\mathbf{x}}$, we obtain the problem

$$\min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N},\tilde{\mathbf{z}}\in\mathbb{R}^{M}} \quad \omega(\tilde{\mathbf{z}}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) \tag{45a}$$

$$\text{s.t.} \quad \mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}} = \mathbf{0} \tag{45b}$$

which is in a form suitable for the Alternating Direction Method of Multipliers (ADMM); a method which combines the benefits of dual decomposition and augmented Lagrangian techniques into a simple but powerful algorithm. The advantage of ADMM is that it enables cost functions (which may be non-smooth) and constraints to be handled separately via variable splitting. This can yield very efficient updates that are amenable to distributed implementation, while requiring mild conditions for achieving convergence. The augmented Lagrangian of (45) is given by

$$\begin{aligned} &L_\rho(\tilde{\mathbf{x}},\tilde{\mathbf{z}},\boldsymbol{\lambda}) \\ &= \omega(\tilde{\mathbf{z}}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) + \lambda^T(\mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}}) + \frac{\rho}{2}\|\mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}}\|_2^2 \end{aligned} \tag{46}$$

where $\lambda \in \mathbb{R}^M$ is the dual variable and ρ is the penalty parameter of the augmented Lagrangian. The ADMM updates for a given subproblem (45) are then given by

$$\begin{aligned} \tilde{\mathbf{x}}_{j+1}^{(n)} &:= \arg\min_{\tilde{\mathbf{x}}} L_\rho(\tilde{\mathbf{x}},\tilde{\mathbf{z}}_j^{(n)},\lambda_j^{(n)}) \\ &= \arg\min_{\tilde{\mathbf{x}}} I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}) + \frac{\rho}{2}\|\mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}}_j^{(n)} + \tilde{\lambda}_j^{(n)}\|_2^2 \end{aligned} \tag{47a}$$

$$\begin{aligned} \tilde{\mathbf{z}}_{j+1}^{(n)} &:= \arg\min_{\tilde{\mathbf{z}}} L_\rho(\tilde{\mathbf{x}}_{j+1}^{(n)},\tilde{\mathbf{z}},\lambda_j^{(n)}) \\ &= \arg\min_{\tilde{\mathbf{z}}} \omega(\tilde{\mathbf{z}}) + \frac{\rho}{2}\|\tilde{\mathbf{z}} - \mathbf{C}^{(n)}\tilde{\mathbf{x}}_{j+1}^{(n)} - \tilde{\lambda}_j^{(n)}\|_2^2 \\ &= \text{prox}_{\frac{\omega}{\rho}}(\mathbf{C}^{(n)}\tilde{\mathbf{x}}_{j+1}^{(n)} + \tilde{\lambda}_j^{(n)}) \end{aligned} \tag{47b}$$

$$\tilde{\lambda}_{j+1}^{(n)} := \tilde{\lambda}_j^{(n)} + \mathbf{C}^{(n)}\tilde{\mathbf{x}}_{j+1}^{(n)} - \tilde{\mathbf{z}}_{j+1}^{(n)} \tag{47c}$$

where the subscript $j = 0, 1, \cdots$ is the ADMM iteration counter, the superscript n is the SCA iteration counter, $\tilde{\lambda} := \frac{1}{\rho}\lambda$ represents the scaled dual variable and in (47b), we have defined the proximal operator [45] of a convex, proper, closed function $f : \mathbb{R}^n \to \mathbb{R}$ as

$$\text{prox}_{\frac{f}{\rho}}(\mathbf{x}) = \arg\min_{\mathbf{y}} f(\mathbf{y}) + \frac{\rho}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \tag{48}$$

The update of $\tilde{\mathbf{z}}$ can be computed efficiently since the proximal operator of $\omega(.)$ can be evaluated via a bisection search (refer to Appendix B). Although the proximal operator of $I_{\tilde{\mathcal{F}}}(.)$ can be evaluated in closed form (being the Euclidean projection operator for the simple set $\tilde{\mathcal{F}}$, which can be computed in closed form), the update of $\tilde{\mathbf{x}}$ has to be solved numerically due to the presence of the matrix $\mathbf{C}^{(n)}$ (unless $\mathbf{C}^{(n)}$ is the identity matrix or is orthogonal, neither of which hold in our case), which is undesirable from a computational complexity standpoint. Thus, we propose to use an inexact version of ADMM, known as *Linearized* ADMM (L-ADMM)[45], [46], which is specifically designed to solve problems of the form (44) using the proximal operators of $\omega(.)$ and $I_{\tilde{\mathcal{F}}}(.)$ to update the primal variables. The variable updates for the L-ADMM algorithm are given by

$$\begin{aligned} \tilde{\mathbf{x}}_{j+1}^{(n)} &:= \text{prox}_{\eta I_{\tilde{\mathcal{F}}}}(\tilde{\mathbf{x}}_j^{(n)} - \eta\rho\mathbf{C}^{(n)T}(\mathbf{C}^{(n)}\tilde{\mathbf{x}}_j^{(n)} - \tilde{\mathbf{z}}_j^{(n)} + \tilde{\lambda}_j^{(n)})) \\ &:= \text{proj}_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}}_j^{(n)} - \eta\rho\mathbf{C}^{(n)T}(\mathbf{C}^{(n)}\tilde{\mathbf{x}}_j^{(n)} - \tilde{\mathbf{z}}_j^{(n)} + \tilde{\lambda}_j^{(n)})) \end{aligned} \tag{49a}$$

$$\tilde{\mathbf{z}}_{j+1}^{(n)} := \text{prox}_{\frac{\omega}{\rho}}(\mathbf{C}^{(n)}\tilde{\mathbf{x}}_{j+1}^{(n)} + \tilde{\lambda}_j^{(n)}) \tag{49b}$$

$$\tilde{\lambda}_{j+1}^{(n)} := \tilde{\lambda}_j^{(n)} + \mathbf{C}^{(n)}\tilde{\mathbf{x}}_{j+1}^{(n)} - \tilde{\mathbf{z}}_{j+1}^{(n)} \tag{49c}$$

where the parameters η and ρ are chosen to satisfy $0 < \eta\rho \leq \frac{1}{\|\mathbf{C}^{(n)}\|_2^2}$ [45, p. 158]. Note that the L-ADMM algorithm differs from standard ADMM in the update of $\tilde{\mathbf{x}}$ only, which now involves evaluating the projection onto the set $\tilde{\mathcal{F}}$ and can be computed in closed form. In L-ADMM, the standard update for $\tilde{\mathbf{x}}$ is modified by replacing the term $\frac{\rho}{2}\|\mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}}_j\|_2^2$ in the augmented Lagrangian $L_\rho(\tilde{\mathbf{x}},\tilde{\mathbf{z}}_j,\lambda_j)$ (46) by $\rho(\mathbf{C}^{(n)T}\mathbf{C}^{(n)}\tilde{\mathbf{x}}_j - \mathbf{C}^{(n)T}\tilde{\mathbf{z}}_j)^T\tilde{\mathbf{x}} + \frac{\eta}{2}\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_j\|_2^2$, i.e., linearization of $\frac{\rho}{2}\|\mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}}_j\|_2^2$ about $\tilde{\mathbf{x}}_j$ plus a quadratic regularization term. The result can be rearranged in the form of a proximal operator as in (49a).

In [47], the following convergence result of L-ADMM can be found. The authors reformulated the optimality condition of (45) into a variational inequality of the form

$$\text{find} \quad \mathbf{w}^* \in \Omega \tag{50a}$$

$$\text{s.t.} \quad \theta(\mathbf{u}) - \theta(\mathbf{u}^*) + (\mathbf{w} - \mathbf{w}^*)^T F(\mathbf{w}^*) \geq 0, \forall\, \mathbf{w} \in \Omega \tag{50b}$$

where we have defined $\mathbf{u} := [\tilde{\mathbf{x}}^T, \tilde{\mathbf{z}}^T]^T, \mathbf{w} := [\tilde{\mathbf{x}}^T, \tilde{\mathbf{z}}^T, \lambda^T]^T$, $\theta(\mathbf{u}) := \omega(\tilde{\mathbf{z}}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}})$, $\Omega := \mathbb{R}^{2N} \times \mathbb{R}^M \times \mathbb{R}^M$ and $F(\mathbf{w}) = [-(\mathbf{C}^{(n)T}\lambda)^T, \lambda^T, (\mathbf{C}^{(n)}\tilde{\mathbf{x}} - \tilde{\mathbf{z}})^T]^T$. Let $\bar{\mathbf{w}}_j := \frac{1}{T+1}\sum_{t=0}^{T}\mathbf{w}_j$ where $\mathbf{w}_j := [\tilde{\mathbf{x}}_{j+1}^T, \tilde{\mathbf{z}}_{j+1}^T, \lambda_{j+1}^T]^T$ (here we drop the superscript n for ease of notation). Then, the number of iterations required so that

$$\theta(\mathbf{u}) - \theta(\bar{\mathbf{u}}_j) + (\mathbf{w} - \bar{\mathbf{w}}_j)^T F(\bar{\mathbf{w}}_j) \geq -\epsilon, \forall\, \mathbf{w} \in \Omega \tag{51}$$

is $O(\frac{1}{\epsilon})$ (in an ergodic sense) in the worst case. Meanwhile, analysis of the per iteration cost of L-ADMM reveals that all the required matrix-vector multiplications incur a cost of $O(MN)$ flops. The update of $\tilde{\mathbf{x}}$ is in closed form, while for the update of $\tilde{\mathbf{z}}$, computing the proximal operator of $\omega(.)$ via bisection search requires $O(M\log_2(\frac{D}{\epsilon_b}))$ operations, where D is the initial bisection interval and ϵ_b is the desired length of the final interval. Hence, it follows that L-ADMM can be used to solve each SCA subproblem efficiently. The overall SCA algorithm is given by

Algorithm 5: L-ADMM SCA.

Initialization: Randomly generate a feasible starting point $\tilde{\mathbf{x}}^{(0)} \in \tilde{\mathcal{F}}$. Set $n := 0$.

Repeat

- Compute $\tilde{\mathbf{x}}^{(n+1)} \in \arg\min_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \omega(\mathbf{C}^{(n)}\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) + I_{\tilde{\mathcal{F}}}(\tilde{\mathbf{x}})$ according to the L-ADMM updates (49).
- Set $n := n + 1$.

Until termination criterion is met.

If we define $\tilde{\mathbf{p}}^{(n)} := \tilde{\mathbf{x}}^{(n)}$ (where the superscript n is the SCA iteration counter), then the nth L-ADMM subproblem can be warm-started by initializing $\tilde{\mathbf{x}}_1^{(n)} = \tilde{\mathbf{p}}^{(n)}, \tilde{\mathbf{z}}_1^{(n)} = \tilde{\mathbf{z}}_1^{(n-1)}$, $\tilde{\lambda}_1^{(n)} = \tilde{\lambda}_1^{(n-1)}$ (here the subscript 1 denotes the L-ADMM iteration counter). For the very first SCA iteartion, we use $\tilde{\mathbf{z}}_1^{(0)} = \mathbf{C}^{(0)}\tilde{\mathbf{p}}^{(0)}, \tilde{\lambda}_1^{(0)} = \mathbf{0}$.

VI. Numerical Results

In order to benchmark the performance of our proposed low-complexity SCA algorithms, we implemented a standard SCA algorithm where each subproblem was cast as a SoCP problem and solved with the MOSEK solver [48] in MATLAB using the modeling language YALMIP [49] as a parser. We implemented the Nesterov SCA algorithm in MATLAB using the optimization toolbox TFOCS [50] to solve each SCA subproblem via the accelerated FOM described in [36]. The Mirror-Prox SCA and L-ADMM SCA algorithms were implemented in MATLAB by straightforward coding. The Nesterov SCA and L-ADMM SCA make use of the warm-starting strategies described previously. For Nesterov SCA, we set the smoothing parameter $\mu = 1e^{-4}$, while in L-ADMM, we set $\epsilon_b = 1e^{-6}$, and, in each SCA iteration, we let $\eta = \frac{1}{\rho\|\mathbf{C}^{(n)}\|_2^2}$. The value of ρ used depended on the scenario under consideration (i.e., traditional multicast or Massive MIMO multicast). In both scenarios, the downlink channels $\{\mathbf{h}_m\}_{m=1}^M$ were modeled as random vectors drawn from a complex, circularly symmetric, normal distribution with zero mean and identity covariance matrix and the noise variance was set to be 1 for all users. The SCA algorithms were all initialized from the same starting point and run for a maximum of 20 iterations. For the FOM-based SCA algorithms, each subproblem was solved using 1000 iterations. All experiments were carried out on a Windows desktop with 4 Intel i7 cores and 8 GB of RAM.

In a preliminary simulation, we considered a traditional multicasting scenario with $N = 10$ transmit antennas and $M = 200$ users. In this case, we set $\rho = 0.1$ in the L-ADMM method. For initializing our SCA algorithms, we considered the problem of maximizing the *average* SNR, which can serve as a reasonable starting point for further refinement [21]. In [51], Lopez demonstrated that the average SNR maximization problem in a multicasting scenario reduces to computing the principal eigenvector of the normalized channel correlation matrix $\mathbf{H} = \sum_{m=1}^M \frac{\mathbf{h}_m \mathbf{h}_m^H}{\sigma_m^2}$, and can be determined via the power method. Using the Lopez initialization as a starting point for our SCA algorithms, the results obtained after averaging over 200 channel realizations are depicted in Fig. 1, which plots the average minimum SNR in dB as a function of the SCA iteration index. It is observed that the FOM-based SCA algorithms attain the same performance as the standard SCA algorithm which uses the MOSEK solver to solve each subproblem. The timing results are summarized in Table I. Taken together, we observe that the Nemirovski SCA algorithm, which uses the Mirror-Prox algorithm to solve each SCA subproblem, exhibits the best overall performance in terms of speed and max-min SNR objective function. The Nesterov SCA and L-ADMM algorithms exhibit slightly improved performance in terms of the objective value attained, but are more expensive compared to Nemirovski SCA; however, they are still less expensive compared to the standard SCA algorithm.

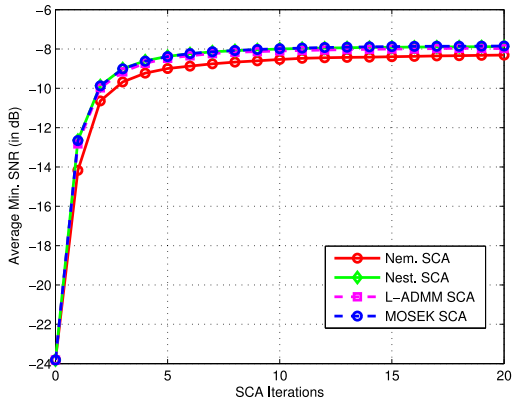


Fig. 1. Comparison of average max-min SNR attained for $N = 10$ antennas, $M = 200$ users (traditional multicasting).

TABLE I
Timing Results for Traditional Multicasting

Method	Stand. SCA	Nest. SCA	Nem. SCA	L-ADMM SCA
Avg. Time	10.24s	6.77s	1.94s	4.91s

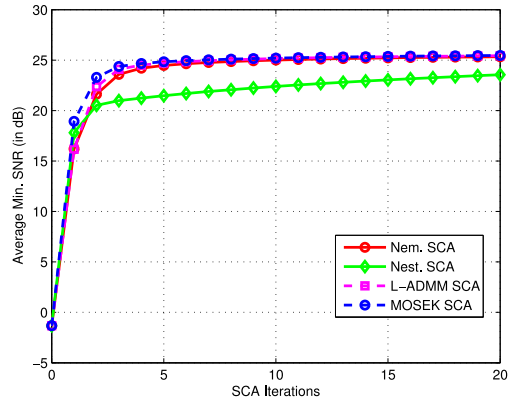
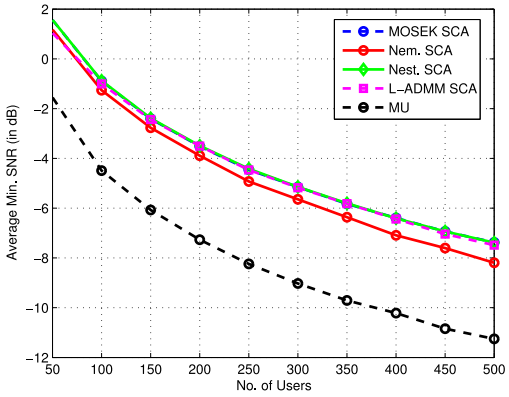


Fig. 2. Comparison of average max-min SNR attained for $N = 200$ antennas, $M = 50$ users (Massive MIMO multicasting).

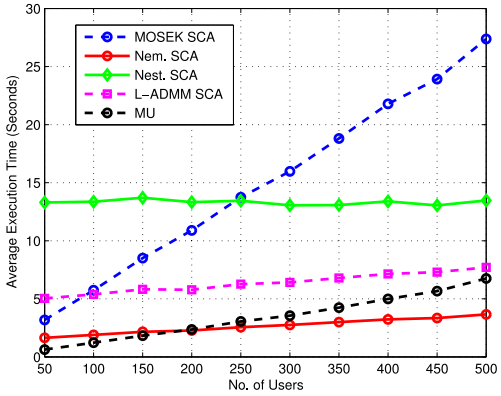
We also carried out a similar experiment for Massive MIMO multicasting, with $N = 200$ antennas and 50 users. In this scenario, we replaced the sum power constraint by the PAPCs. We set $P_i = 0.33, \forall\, i \in \{1, \cdots, N\}$. A starting point that satisfies the PAPCs was randomly generated and was used to initialize the SCA algorithms. The value of ρ in the L-ADMM method was set to 0.01. All results were averaged over 200 channel realizations. The performance with respect to the value of the objective function attained is shown in Fig. 2, while the timing results are depicted in Table II. It can be seen that Nemirovski SCA and L-ADMM SCA match the performance of standard SCA in terms of average minimum SNR attained, but at much lower complexity (Nemirovski SCA in particular). Nesterov SCA did not perform as well in this regime with respect to

TABLE II
TIMING RESULTS FOR MASSIVE MIMO MULTICASTING

Method	Stand. SCA	Nest. SCA	Nem. SCA	L-ADMM SCA
Avg. Time	33.10s	16.70s	4.51s	8.64s



(a) Average minimum SNR vs M ($N = 25$)

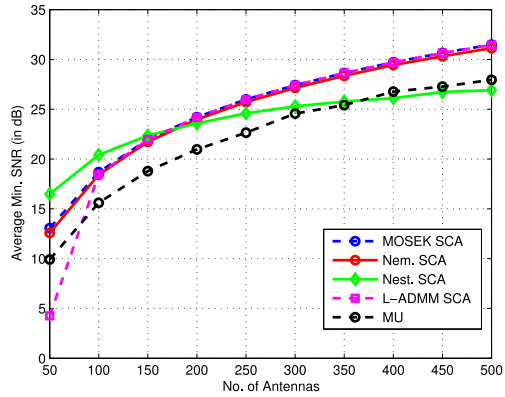


(b) Average Execution Time vs M ($N = 25$)

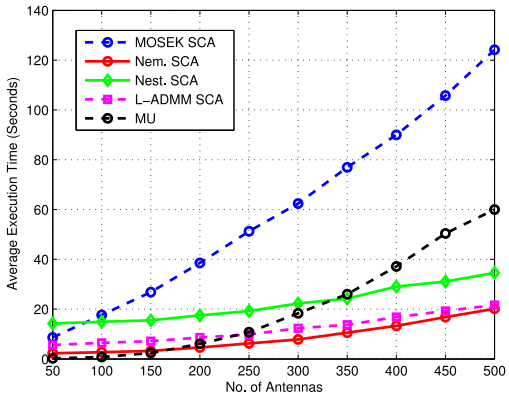
Fig. 3. Traditional multicasting.

the objective value attained, and was also the most expensive amongst the FOM-based SCA algorithms. From these initial experiments, it is evident that using fast FOMs to solve the SCA subproblems allows us to effect a very favorable performance-complexity tradeoff, i.e., we attain the same performance as that of an interior-point method based SCA algorithm, but at much lower complexity.

We also carried out a more comprehensiveexperiment for both multicasting scenarios. First, we considered a traditional multicasting scenario where we fixed the number of transmit antennas $N = 25$ and increased the number of users M from 50 to 500. The algorithm parameters were set to be the same as previously indicated. We also added the MU algorithm in [23], which uses proportional fairness as a surrogate for max-min fairness, for comparison. Lopez initialization was again used for all algorithms. The MU algorithm was run for 1000 iterations. All results were obtained by averaging across 200 channel realizations for each value of M. The average minimum SNR attained (in dB) as a function of the number of users M is shown in Fig. 3(a) while the timing results are depicted in



(a) Average minimum SNR vs N ($M = 50$)



(b) Average Execution Time vs N ($M = 50$)

Fig. 4. Massive MIMO multicasting.

Fig. 3(b). From the figures, it is observed that Nesterov SCA and L-ADMM SCA methods always attain the same average minimum SNR as standard SCA, with Nemirovski SCA being only slightly worse. In terms of execution time, it is observed that as the number of users is increased, the time taken by standard SCA increases considerably (by almost an order of magnitude), while the execution times of the FOM-based SCA methods remains relatively constant.

Next, a Massive MIMO multicasting scenario was considered with $M = 50$ users, and the number of transmit antennas N was increased from 50 to 500. The power budget of each antenna was set to be $P_i = 0.25, \forall\, i = \{1, \cdots, N\}$. We used the same choice of algorithm parameters for Massive MIMO multicast as described previously. As an additional performance benchmark, we appropriately modified the MU algorithm to handle PAPCs (see Appendix C for details). A randomly generated, feasible, starting point was used to initialize all the algorithms. The MU algorithm was run for 200 iterations in this case. All generated results were averaged across 200 channel realizations. The average minimum SNR attained is shown in Fig. 4a, while the average execution times are depicted in Fig. 4b. The figures demonstrate the state-of-the-art performance and computational gains offered by our proposed algorithms. The Nemirovski SCA and L-ADMM SCA algorithms attain the same performance as that of standard SCA but at significantly lower complexity.

While the Nesterov SCA algorithm initially exhibits the best performance (when $N \leq 150$), it falls off as N is increased further. In terms of complexity, it is also slower than the other FOM-based SCA algorithms. Overall, as the number of antennas is increased, the timing curves for the proposed SCA algorithms increase very gracefully compared to that of the standard SCA (showcasing the drawback of using interior-point methods for solving large problems) and the MU algorithms. The Nemirovski SCA algorithm effects the best performance-complexity tradeoff in this regime.[4]

VII. Conclusions

We considered a special class of non-convex QCQP problems which can be expressed as maximizing the point-wise minimum of homogeneous, convex quadratics over a simple convex set. The development of SCA algorithms was pursued for obtaining high quality approximate solutions of this problem at low complexity. Our approach involves iteratively solving a sequence of convex approximations of the non-convex problem. Each subproblem is formulated as a non-smooth convex optimization problem, and solved using specialized FOMs which leverage the structure inherent in each subproblem to efficiently compute solutions at low overall complexity. This endows the algorithms with the ability to scale well to problems in high dimensions with a large number of constraints. The proposed algorithms were applied to the problem of single-group multicast beamforming. Simulations demonstrated that the algorithms offer substantial computational savings while attaining the same performance as standard SCA algorithms using interior-point methods to solve each SCA subproblem. These results are borne out of theoretical worst-case complexity considerations, which prove complexity reduction. Careful implementation of these algorithms in an appropriate lower-level programming language has the potential for deployment in real-time applications.

Appendix A
Proof of Proposition 1

In this section, we discuss the convergence properties of Algorithm 1. Our first goal is to establish that every limit point $\tilde{\mathbf{x}}_l$ of the iterates $\{\tilde{\mathbf{x}}^{(n)}\}_{n\in\mathbb{N}}$ generated by Algorithm 1 is a d-stationary point of (11); i.e., $f'(\tilde{\mathbf{x}}_l, \tilde{\mathbf{d}}) \geq 0$ for all $\tilde{\mathbf{d}}$ such that $\tilde{\mathbf{x}} + \tilde{\mathbf{d}} \in \tilde{\mathcal{F}}$. In order to do so, we will resort to [7, Theorem 1]. However, we first have to verify that the non-convex cost function $f(.)$ and its convex surrogate $v(.,.)$ satisfy the four conditions laid out in [7, Assumption 1]. By virtue of properties (A2-A4), simple inspection reveals that all but one of these conditions are apparently satisfied; the condition in question being [7, Assumption A3], which requires that the directional derivatives of $f(.)$ and $v(.,.)$ are equal at the point of approximation. This condition is hard to check in general, and a sufficient condition is proposed in [7, Proposition 1] under which it is automatically satisfied. Unfortunately, this sufficient condition does not hold in our case, which complicates matters. Nevertheless, by relying upon a different set of results borrowed from variational analysis [11], it is indeed possible to verify that [7, Assumption A3] is satisfied in our case, as we now show.

First, for ease of notation, we first introduce the definition $l_m(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) := \mathbf{c}_m^{(n)T}\tilde{\mathbf{x}} + d_m^{(n)}, \forall\, m \in \mathcal{M}$. Now, consider the directional derivative of $v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) = \max\limits_{m\in\mathcal{M}} l_m(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})$, which, being a convex function, admits the following representation

$$v'(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}; \tilde{\mathbf{d}}) = \max_{\tilde{\mathbf{w}}\in\partial v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})} \tilde{\mathbf{w}}^T \tilde{\mathbf{d}} \tag{52}$$

where $\partial v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) = \mathrm{conv}(\nabla l_i(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) | \forall\, i \in \mathcal{M}(\tilde{\mathbf{x}}))$ and $\mathcal{M}(\tilde{\mathbf{x}}) := \{i \,|\, l_i(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}) = v(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)})\} \subseteq \mathcal{M}$. We then have that

$$v'(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^{(n)}; \tilde{\mathbf{d}})\Big|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}^{(n)}} = \max_{\tilde{\mathbf{w}}\in\partial v(\tilde{\mathbf{x}}^{(n)}, \tilde{\mathbf{x}}^{(n)})} \tilde{\mathbf{w}}^T \tilde{\mathbf{d}} \tag{53}$$

where by construction of the surrogate function $v(.,.)$, we now have $\partial v(\tilde{\mathbf{x}}^{(n)}, \tilde{\mathbf{x}}^{(n)}) = \mathrm{conv}(\nabla u_i(\tilde{\mathbf{x}}^{(n)}) | \forall\, i \in \mathcal{M}(\tilde{\mathbf{x}}))$ and $\mathcal{M}(\tilde{\mathbf{x}}) := \{i \,|\, u_i(\tilde{\mathbf{x}}^{(n)}) = f(\tilde{\mathbf{x}}^{(n)})\}$. At this stage, it is fairly obvious that if we can establish a similar relationship for the directional derivative of $f(.)$ at $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(n)}$, the proof is complete. However, for non-convex functions, the representation (52) is not valid in general, which prevents us from establishing the desired result via the aforementioned arguments. Instead, under additional assumptions on f (which will be shown to be implicitly satisfied), and by exploiting the fact that f is the point-wise maximum of a finite number of smooth functions, we will utilize a different line of reasoning to derive an expression for $f'(\tilde{\mathbf{x}}; \tilde{\mathbf{d}})$, which, interestingly, will turn out to be the same as (53) at the point $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(n)}$.

Before we describe our approach in detail, we will require the following definitions. Adopting the exposition of [11], the *difference quotient function* associated with f at a point $\tilde{\mathbf{x}}$ (where $f(\tilde{\mathbf{x}})$ is finite) and a direction $\tilde{\mathbf{d}}$, is defined as

$$\Delta_\tau f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}) := \frac{f(\tilde{\mathbf{x}} + \tau\tilde{\mathbf{d}}) - f(\tilde{\mathbf{x}})}{\tau} \tag{54}$$

Clearly, we have

$$f'(\tilde{\mathbf{x}}; \tilde{\mathbf{d}}) := \lim_{\tau\downarrow 0} \Delta_\tau f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}) \tag{55}$$

The above definition can be generalized to define a *semiderivative* of f at $\tilde{\mathbf{x}}$ for $\tilde{\mathbf{d}}$ [11, Definition 7.20], which is given by

$$f_s(\tilde{\mathbf{x}}; \tilde{\mathbf{d}}) := \lim_{\substack{\tau\downarrow 0,\\ \tilde{\mathbf{d}}'\to\tilde{\mathbf{d}}}} \Delta_\tau f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}') \tag{56}$$

If the above limit exists, f is said to be semidifferentiable at $\tilde{\mathbf{x}}$ for $\tilde{\mathbf{d}}$. If it holds for all $\tilde{\mathbf{d}}$, then f is said to be semidifferentiable at $\tilde{\mathbf{x}}$. While $f'(\tilde{\mathbf{x}}; \tilde{\mathbf{d}})$ is only concerned with the limiting behavior of $\Delta_\tau f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}})$ along the ray $\{\tilde{\mathbf{x}} + \tau\tilde{\mathbf{d}} | \tau \in \mathbb{R}_+\}$, the semiderivative, loosely speaking, tests the behavior of $\Delta_\tau f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}})$ along all curves from $\tilde{\mathbf{x}}$ in the direction of $\tilde{\mathbf{d}}$. Clearly, if $f_s(\tilde{\mathbf{x}}; \tilde{\mathbf{d}})$ exists and is finite, then $f'(\tilde{\mathbf{x}}; \tilde{\mathbf{d}}) = f_s(\tilde{\mathbf{x}}; \tilde{\mathbf{d}})$. However, the converse is not true in general. When the existence of $f_s(\tilde{\mathbf{x}}; \tilde{\mathbf{d}})$ is not guaranteed, it is useful to work with *subderivatives* of $f(\tilde{\mathbf{x}})$ [11,

[4] In this case, the MU algorithm does not scale very well to large dimensions. Hence, we defer from using it for last-mile refinement using Nemirovski SCA.

Definition 8.1], which always exist and are defined as

$$\mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}) = \liminf_{\substack{\tau \downarrow 0, \\ \tilde{\mathbf{d}}' \to \tilde{\mathbf{d}}}} \Delta_\tau f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}') \tag{57}$$

It is again evident that when $f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}})$ exists, we must have $f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}}) = \mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}})$ (since $\lim$ and $\liminf$ coincide in this case). In addition, if $f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}})$ is also finite, we obtain the following series of equalities

$$f'(\tilde{\mathbf{x}};\tilde{\mathbf{d}}) = f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}}) = \mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}) \tag{58}$$

From (58), it can be inferred that it in order to obtain an expression for $f'(\tilde{\mathbf{x}};\tilde{\mathbf{d}})$, it suffices to show that $f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}})$ exists, compute $f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}})$ (or possibly $\mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}})$) and verify that it is finite valued. This is precisely what we now set out to establish via the following claims.

Claim 1: *The non-convex function* $f(\tilde{\mathbf{x}}) = \max_{m\in\mathcal{M}} u_m(\tilde{\mathbf{x}})$ *is semidifferentiable for all* $\tilde{\mathbf{x}} \in \mathbb{R}^{2N}$.

Proof: Follows directly from [11, Exercise 10.27(c)] ■

Hence, although $f(\tilde{\mathbf{x}})$ is non-differentiable, the fact that it is the point-wise maximum of a finite number of smooth functions $\{u_m(\tilde{\mathbf{x}})\}_{m\in\mathcal{M}}$ ensures that it is semidifferentiable. While this result establishes the existence of semiderivatives of $f(\tilde{\mathbf{x}})$, since we are interested in minimizing f over a convex, compact set $\tilde{\mathcal{F}}$, we require certain regularity assumptions on f [5]and $\tilde{\mathcal{F}}$ [6] being satisfied in order to proceed towards deriving an expression for the semiderivatives of f. The following result establishes that these regularity conditions are automatically satisfied in our case.

Claim 2: *The set* $\tilde{\mathcal{F}}$ *is Clarke regular while* f *is subdifferentially regular for all* $\tilde{\mathbf{x}} \in \tilde{\mathcal{F}}$.

Proof: The first part follows directly from the fact that $\tilde{\mathcal{F}}$ is convex and then invoking [11, Theorem 6.4], while the second part holds due to the point-wise max structure of f which enables us to appeal to [11, Example 7.28]. ■

Thanks to the above result, our overall convergence claims only depend upon the aforementioned regularity conditions *implicitly*; i.e., one does not have to check to see if they are verified; they are automatically guaranteed to hold by Claim 2. With these results in hand, we are now ready to state the main claim.

Claim 3: *The subderivative of* f *for all* $\tilde{\mathbf{x}} \in \tilde{\mathcal{F}}$ *can be expressed as*

$$\mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}) = \max_{i\in\mathcal{M}(\tilde{\mathbf{x}})} \nabla u_i(\tilde{\mathbf{x}})^T \tilde{\mathbf{d}} \tag{59}$$

where $\mathcal{M}(\tilde{\mathbf{x}}) := \{i \,\lfloor\, u_i(\tilde{\mathbf{x}}) = \underline{f}(\tilde{\mathbf{x}})\}$. *Furthermore,* $\mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}}) < \infty$, $\forall\, \tilde{\mathbf{x}} \in \tilde{\mathcal{F}}$ *and* $\tilde{\mathbf{d}}$ *s.t.* $\tilde{\mathbf{x}} + \tilde{\mathbf{d}} \in \tilde{\mathcal{F}}$.

Proof: The first part of the claim follows from the regularity of $f(\tilde{\mathbf{x}})$, $\forall\, \tilde{\mathbf{x}} \in \tilde{\mathcal{F}}$, and then invoking [11, Exercise 8.31]. In order to show the second part, by our assumption that $\tilde{\mathcal{F}}$ is compact, we have that $\mathrm{diam}(\tilde{\mathcal{F}}) := \sup_{\mathbf{x},\mathbf{y}\in\tilde{\mathcal{F}}} \|\mathbf{x}-\mathbf{y}\|_2 = D < \infty$ for some $D \in \mathbb{R}_+$. This enables us to write

$$\begin{aligned} \nabla u_i(\tilde{\mathbf{x}})^T \tilde{\mathbf{d}} &\leq \|\nabla u_i(\tilde{\mathbf{x}})\|_2 \|\tilde{\mathbf{d}}\|_2 = \|\bar{\mathbf{A}}_i \tilde{\mathbf{x}}\|_2 \|\tilde{\mathbf{d}}\|_2 \\ &\leq \|\bar{\mathbf{A}}_i\|_2 \|\tilde{\mathbf{x}}\|_2 \|\tilde{\mathbf{d}}\|_2 \\ &\leq \|\bar{\mathbf{A}}_i\|_2 D^2, \forall\, i \in \mathcal{M}(\tilde{\mathbf{x}}) \\ \Rightarrow \max_{i\in\mathcal{M}(\tilde{\mathbf{x}})} \nabla u_i(\tilde{\mathbf{x}})^T \tilde{\mathbf{d}} &\leq D^2 \max_{i\in\mathcal{M}(\tilde{\mathbf{x}})} \|\bar{\mathbf{A}}_i\|_2 < \infty \end{aligned} \tag{60}$$

■

Taken together, our claims guarantee that $f_s(\tilde{\mathbf{x}};\tilde{\mathbf{d}})$ always exists (Claim 1), establish that all requisite regularity conditions are automatically satisfied by f and $\tilde{\mathcal{F}}$ (Claim 2), and also provide a characterization of $\mathrm{d}f(\tilde{\mathbf{x}})(\tilde{\mathbf{d}})$, which is finite-valued over $\tilde{\mathcal{F}}$ (Claim 3). Thus, it follows that the chain of equalities (58) hold in our case, which allows us to directly write

$$\begin{aligned} f(\tilde{\mathbf{x}};\tilde{\mathbf{d}}) &= \max_{i\in\mathcal{M}(\tilde{\mathbf{x}})} \nabla u_i(\tilde{\mathbf{x}})^T \tilde{\mathbf{d}} \\ &= \max_{\substack{\tilde{\mathbf{w}}\in\mathrm{conv}(\nabla u_i(\tilde{\mathbf{x}})| \\ \forall\, i\in\mathcal{M}(\tilde{\mathbf{x}}))}} \tilde{\mathbf{w}}^T \tilde{\mathbf{d}} \end{aligned} \tag{61}$$

Comparing the above expression with (53), it directly follows that we have

$$f'(\tilde{\mathbf{x}};\tilde{\mathbf{d}})\Big|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}^{(n)}} = v'(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{(n)};\tilde{\mathbf{d}})\Big|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}^{(n)}} \tag{62}$$

which is the condition that we set out to verify. Note that, by the feasibility of the iterates $\{\tilde{\mathbf{x}}^{(n)}\}_{n\in\mathbb{N}}$ of Algorithm 1, (61) always holds for *every* SCA iteration n.

Now that we have verified all four conditions listed in [7, Assumption 1], it only remains to invoke [7, Theorem 1] to claim that every limit point $\tilde{\mathbf{x}}_l$ of $\{\tilde{\mathbf{x}}^{(n)}\}_{n\in\mathbb{N}}$ is a d-stationary point of (11). Of course, this is a weaker claim compared to what we stated in Proposition 1, as it only guarantees convergence along a subsequence of the iterates (provided that a convergent subsequence exists in the first place). While the compactness of $\tilde{\mathcal{F}}$ guarantees the existence of such a convergent subsequence, it also allows us to strengthen our result to achieve the desired outcome via the following claim.

Claim 4: *If* $\tilde{\mathcal{F}}$ *is a compact set, then, under [7, Assumption 1], the sequence of iterates* $\{\tilde{\mathbf{x}}^{(n)}\}_{n\in\mathbb{N}}$ *satisfy*

$$\lim_{n\to\infty} d(\tilde{\mathbf{x}}^{(n)}, \tilde{\mathcal{F}}^*) = 0,$$

where $\tilde{\mathcal{F}}^*$ *is the set of d-stationary solutions of (11).*

Proof: Follows directly from [7, Corollary 1]. ■

This concludes the proof of Proposition 1.

Remark 1: While the proof of convergence requires computing an exact solution to each SCA subproblem, whether solving the inner subproblems inaccurately via FOMs still results in provable convergence is an open question. We point out that we have used different FOMs (varying from primal methods to primal-dual methods) to solve each SCA subproblem and thus, each algorithm traces a different path through solution space. This complicates matters in developing a unified convergence

[5]Here, by regularity of f, we mean that f satisfies the notion of *subdifferential regularity* as defined in [11, Definition 7.25].

[6]By regularity of the set $\tilde{\mathcal{F}}$, we mean that $\tilde{\mathcal{F}}$ satisfies *Clarke regularity* as defined in [11, Definition 6.4].

analysis of the various FOM based SCA algorithms. In [9], the authors consider a proximal regularized version of Algorithm 1, which is also provably convergent while allowing for inexact solution of the inner subproblems. We speculate that these results can be adapted to our algorithmic setting as well (i.e., SCA without the proximal term) to obtain iteration complexity estimates required by each FOM to guarantee overall convergence of the sequence of SCA iterates while using inexact solutions. However, formally proving this conjecture is beyond the scope of this paper. Nevertheless, as evidenced by our experiments, the proposed algorithms with the prescribed stopping criteria perform very admirably and offer state-of-the-art performance (in terms of multicast rate) at low complexity, which make them very appealing for use in practice.

Appendix B
Proximal Operator of $\omega(.)$

The results of this appendix are derived in a manner similar to that in [45, Section 6.4]. Determining the proximal operator $\text{prox}_{\frac{\omega}{\rho}}(\mathbf{x})$ of the function $\omega(\mathbf{y}) = \max_{m\in\mathcal{M}}\{y_m + b_m\}$, where $\mathcal{M} = \{1, \cdots, M\}$, requires one to solve the following convex optimization problem

$$\min_{\mathbf{y}} \omega(\mathbf{y}) + \frac{\rho}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \tag{63}$$

which can be represented in its epigraph form as the following smooth optimization problem

$$\min_{\mathbf{y},t} \quad t + \frac{\rho}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \tag{64a}$$
$$\text{s.t.} \quad y_m + b_m \le t,\ m = 1, \cdots, M \tag{64b}$$

The KKT optimality conditions for (64) are given by

$$y_m^* + b_m \le t^*, \tag{65a}$$
$$\eta_m^* \ge 0, \tag{65b}$$
$$\eta_m^*(y_m^* + b_m - t^*) = 0, \tag{65c}$$
$$\rho(y_m^* - x_m) + \eta_m^* = 0, \tag{65d}$$
$$\sum_{m=1}^{M} \eta_m^* = 1 \tag{65e}$$

where $m \in \mathcal{M}$ and $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_M]^T$ denotes the vector of dual variables. If $y_m^* + b_m < t^*$, then from the third condition, we have $\eta_m^* = 0$. Otherwise, if $y_m^* + b_m = t^*$, then from the fourth condition we obtain $\eta_m^* = \rho(x_m + b_m - t^*)$. Since $\eta_m^* \ge 0$, we must have that

$$\eta_m^* = \rho \max\{x_m + b_m - t^*, 0\} \tag{66}$$

Substituting for η_m^* in the final KKT condition, we obtain the equation

$$\rho \sum_{m=1}^{M} \max\{x_m + b_m - t^*, 0\} = 1 \tag{67}$$

which can be solved for t^* via bisection using the initial interval $[\min_m\{x_m + b_m\} - (1/\rho M), \max_m\{x_m + b_m\}]$. Once t^* is determined, we can solve for $\mathbf{y}^* = [y_1^*, \cdots, y_M^*]^T$ as

$$\begin{aligned} y_m^* &= x_m - \max\{x_m + b_m - t^*, 0\} \\ &= \min\{t^* - b_m, x_m\}, \forall\, m \in \mathcal{M} \end{aligned} \tag{68}$$

The proximal operator of $\omega(.)$ is then given by

$$\text{prox}_{\frac{\omega}{\rho}}(\mathbf{x}) = \mathbf{y}^* \tag{69}$$

Appendix C
MU Algorithm for Massive MIMO Multicasting

In this appendix, we derive a variant of the MU algorithm described in [23] for handling PAPCs in Massive MIMO multicasting. As a surrogate for the max-min fair problem (10), consider the following proportionally-fair formulation

$$\max_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \quad \sum_{m=1}^{M} \log\left(\tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} + \delta\right) \tag{70a}$$
$$\text{s.t.} \quad \tilde{\mathbf{x}} \in \tilde{\mathcal{F}} \tag{70b}$$

where $\delta \in \mathbb{R} > 0$. After expressing the set $\tilde{\mathcal{F}}$ in terms of PAPCs, we obtain the following non-convex problem

$$\max_{\tilde{\mathbf{x}}\in\mathbb{R}^{2N}} \quad \sum_{m=1}^{M} \log\left(\tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} + \delta\right) \tag{71a}$$
$$\text{s.t.} \quad \tilde{x}^2(i) + \tilde{x}^2(i+N) \le P_i, \forall\, i = 1, \cdots, N \tag{71b}$$

The MU algorithm proposes to solve (71) in the following iterative manner. Starting from an initial feasible point $\tilde{\mathbf{x}}^{(0)}$, at each iteration $n \ge 0$, we construct the following first order surrogate of $h(\tilde{\mathbf{x}}) := \sum_{m=1}^{M} \log\left(\tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} + \delta\right)$ about the current iterate $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(n)}$

$$h(\tilde{\mathbf{x}}) \approx h(\tilde{\mathbf{x}}) + \nabla h(\tilde{\mathbf{x}}^{(n)})^T(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}^{(n)}) \tag{72}$$

by determining the first-order Taylor series expansion of $h(\tilde{\mathbf{x}})$ about $\tilde{\mathbf{x}}^{(n)}$. The gradient of $h(\tilde{\mathbf{x}})$ is given by

$$\nabla h(\tilde{\mathbf{x}}) := \sum_{m=1}^{M} \frac{2\bar{\mathbf{A}}_m \tilde{\mathbf{x}}}{\tilde{\mathbf{x}}^T \tilde{\mathbf{A}}_m \tilde{\mathbf{x}} + \delta} \tag{73}$$

which corresponds to taking an inversely weighted combination of the gradients of the functions $u_m(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^T \bar{\mathbf{A}}_m \tilde{\mathbf{x}}, \forall\, m \in \mathcal{M}$, with more emphasis placed on the gradients of those functions $u_m(\tilde{\mathbf{x}})$ which are small. This intuitively suggests that $\nabla h(\tilde{\mathbf{x}})$ corresponds to a good search direction for attaining max-min fairness. The update rule of our algorithm at each iteration n is then given by

$$\begin{aligned} \tilde{\mathbf{x}}^{(n+1)} &= \arg\max_{\tilde{\mathbf{x}}\in\tilde{\mathcal{F}}} \nabla h(\tilde{\mathbf{x}}^{(n)})^T(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}^{(n)}) && \text{(74a)} \\ &= \arg\max_{\substack{\tilde{x}^2(i)+\tilde{x}^2(i+N)\le P_i, \\ \forall\, i=1,\cdots,N}} \nabla h(\tilde{\mathbf{x}}^{(n)})^T \tilde{\mathbf{x}} && \text{(74b)} \end{aligned}$$

Define $\tilde{\mathbf{g}}^{(n)} := \nabla h(\tilde{\mathbf{x}}^{(n)})$. Then, the objective function of (74b) can be expressed as

$$\tilde{\mathbf{g}}^{(n)T} \tilde{\mathbf{x}} = \sum_{i=1}^{2N} \tilde{g}^{(n)}(i)\tilde{x}(i) = \sum_{i=1}^{N} \bar{\mathbf{g}}_i^{(n)T} \bar{\mathbf{x}}_i \tag{75}$$

where $\bar{\mathbf{g}}_i^{(n)} := [\tilde{g}^{(n)}(i), \tilde{g}^{(n)}(i+N)]^T$, $\bar{\mathbf{x}}_i := [\tilde{x}(i), \tilde{x}(i+N)]^T$, $\forall\, i = 1, \cdots, N$. With the objective function represented in this form, it is obvious that (74b) decomposes into N parallel problems of the form

$$\bar{\mathbf{x}}_i^{(n+1)} = \arg\max_{\|\bar{\mathbf{x}}_i\|_2^2 \le P_i} \bar{\mathbf{g}}_i^{(n)T} \bar{\mathbf{x}}_i = \sqrt{P_i}\, \frac{\bar{\mathbf{g}}_i}{\|\bar{\mathbf{g}}_i\|_2}, \forall\, i = 1, \cdots, N \tag{76}$$

From the vectors $\{\bar{\mathbf{x}}_i^{(n+1)}\}_{i=1}^N$, the update vector $\tilde{\mathbf{x}}_i^{(n+1)}$ can be easily synthesized.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor G. Scutari for pointing out that reference [11] can be used to prove Proposition 1 . The authors would also like to thank M. Razaviyayn and the anonymous referees, whose feedback and suggestions helped improve the quality of the paper considerably.

REFERENCES

[1] A. Konar and N. D. Sidiropoulos, "A fast approximation algorithm for single-group multicast beamforming with large antenna arrays," in *Proc. IEEE 17th Int. Workshop Signal Process. Adv. Wireless Commun.*, Edinburgh, U.K., Jul. 2016.

[2] Z.-Q. Luo and T.-H. Chang, "SDP relaxation of homogeneous quadratic optimization: Approximation bounds and applications," in *Convex Optimization in Signal Processing and Communications*, D. Palomar and Y. Eldar, Eds. Cambridge U.K.: Cambridge Univ. Press, 2010, pp. 117–165.

[3] A. d'Aspremont and S. Boyd, *Relaxations and Randomized Methods for Nonconvex QCQPs* (EE392 Lecture Notes). Stanford, CA, USA: Stanford Univ., 2003.

[4] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006.

[5] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Oper. Res.*, vol. 26, no. 4, pp. 681–683, 1978.

[6] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, 2010.

[7] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[8] G. Scutari, F. Facchinei, L. Lampariello, and P. Song, "Parallel and distributed methods for nonconvex optimization–Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.

[9] F. Facchinei, L. Lampariello, and G. Scutari, "Feasible methods for nonconvex nonsmooth problems with applications in green communications," *Math. Program. Ser. A*, pp. 1–36, 2016, doi: 10.1007/s10107-016-1072-9.

[10] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[11] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*. New York, NY, USA: Springer, vol. 317, 2009.

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

[13] 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.

[14] R. Glowinski and A. Marrocco, "Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualite, d'une classe de problems de Dirichlet non lineares," *Revue Francaise d'Automatique, Informatique, et Recherche Operationelle*, vol. 9, pp. 41–76, 1975.

[15] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Comput. Math. Appl.*, vol. 2, pp. 17–40, 1976.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[17] O. T. Demir and T. E. Tuncer, "Alternating maximization algorithm for the broadcast beamforming," in *Proc. 22nd Eur. Signal Process. Conf.*, Lisbon, Portugal, Sep. 1–5, 2014, pp. 1915–1919.

[18] N. Bornhorst and M. Pesavento, "An iterative convex approximation approach for transmit beamforming in multi-group multicasting," in *Proc. IEEE 12th Int. Workshop Signal Process. Adv. Wireless Commun.*, San Francisco, CA, USA, Jun. 2011, pp. 411–415.

[19] 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 Process. Lett.*, vol. 21, no. 1, pp. 114–117, Jan. 2014.

[20] A. Lozano, "Long-term transmit beamforming for wireless multicasting," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Honolulu, HI, USA, Apr. 15–20, 2007, pp. 417–420.

[21] E. Matskani, N. D. Sidiropoulos, Z.-Q. Luo, and L. Tassiulas, "Efficient batch and adaptive approximation algorithms for joint multicast beamforming and admission control," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4882–4894, Dec. 2009.

[22] A. Abdelkader, A. B. Gershman, and N. D. Sidiropoulos, "Multiple-antenna multicasting using channel orthogonalization and local refinement," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3922–3927, Jul. 2010.

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

[24] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large scale arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[25] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and Challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[26] J. G. Andrews *et al.*, "What will 5G be?," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[27] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[29] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Sci., 1999.

[30] D. Christopoulos, S. Chatzinotas, and B. Ottersten, "Multicast multigroup beamforming for per-antenna power constrained large-scale arrays," in *Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun.*, Stockholm, Sweden, Jun. 28–Jul. 1, 2015, pp. 271–275.

[31] Y. Nesterov and A. Nemirovski, *Interior Point Polynomial methods in Convex Programming: Theory and Applications*. Philadelphia, PA, USA: SIAM, 1994.

[32] B. T. Polyak, *Introduction to optimization*. New York, NY, USA: Optim. Software, Publ. Division, 1987.

[33] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions* (Springer Series in Computational Mathematics), vol. 3, Berlin, Germany: Springer-Verlag, 1985.

[34] A. Nemirovski and D. Yudin, *Problem complexity and method efficiency in Optimization.* New York, NY, USA: Wiley, 1983.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] Y. Nesterov, "A method for solving a convex programming problem with rate of convergence o$(1/k^2)$," *Soviet Math. Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[37] L. Vandenberghe, "Smoothing," EE236C Lecture Notes, UCLA, 2010. [Online]. Available: http://www.seas.ucla.edu/∼vandenbe/236C/lectures/smoothing.pdf

[38] M. Sion, "On general minimax theorems," *Pacific J. Math.*, vol. 8, no. 1, pp. 171–176, 1958.

[39] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems* (Springer Series in Operations Research). New York, NY, USA: Springer-Verlag, 2003.

[40] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," vol. 15, pp. 3–43, 2016. [Online]. Available: http://web.stanford.edu/∼eryu/papers/monotone_notes.pdf
[41] G. M. Korpelevich, "The extragradient method for finding saddle points and other problems," *Ekonomika i Matematicheskie Metody*, vol. 12, pp. 747–756, 1976.
[42] A. Nemirovski and D. Yudin, "On Cesaro's convergence of the gradient descent method for finding saddle points of convex-concave functions," *Dokl. Akad. Nauk SSSR*, vol. 239, no. 4, 1978.
[43] A. Beck and M. Teboulle, "Mirror Descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, May 2003.
[44] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3/4, pp. 231–357, 2015.
[45] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2014.
[46] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.
[47] B. He and X. Yuan, "On the O(1/n) convergence of the Douglas-Rachford alternating direction method," *SIAM J. Numer. Anal.*, vol. 50, no. 2, pp. 700–709, Apr. 2012.
[48] APS Mosek, "The MOSEK optimization software," 2010. [Online]. Available: http://www.mosek.com
[49] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Symp. Comput. Aided Control Syst. Des.*, Taipei, Taiwan, Sep. 4, 2004, pp. 284–289.
[50] S. R. Becker, E. J. Candes, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Math. Program. Comput.*, vol. 3, pp. 165–218, 2011.
[51] M. J. Lopez, "Multiplexing, scheduling, and multicasting strategies for antenna arrays in wireless networks," Ph.D. dissertation, Elect. Eng. Comp. Sci. Dept., MIT, Cambridge, MA, USA, 2002.

Aritra Konar (S'14) received the B.Tech. degree in electronics and communications engineering from the West Bengal University of Technology, West Bengal, India, in 2011, and the M.S. degree in electrical engineering in 2014 from the University of Minnesota, Minneapolis, MN, USA, where he is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering. His research interests include statistical signal processing, communications, nonlinear optimization, and data analytics.

Nicholas D. Sidiropoulos (F'09) received the Diploma degree in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 1988, 1990, and 1992, respectively. He was an Assistant Professor in the University of Virginia, an Associate Professor in the University of Minnesota, and a Professor in TU Crete, Greece. Since 2011, he has been with the University of Minnesota, where he currently holds an ADC Chair in digital technology. His research spans topics in signal processing theory and algorithms, optimization, communications, and factor analysis—with a long-term interest in tensor decomposition and its applications. His current focus is primarily on signal and tensor analytics for learning from big data. He received the NSF/CAREER Award in 1998, and the IEEE Signal Processing (SP) Society Best Paper Award in 2001, 2007, and 2011. He served as the IEEE SP Society Distinguished Lecturer (2008–2009), and as the Chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He received the 2010 IEEE SP Society Meritorious Service Award, and the 2013 Distinguished Alumni Award from the Department of Electrical and Computer Engineering, University of Maryland. He is a Fellow of EURASIP (2014).