Accelerating Quadratic Transform and WMMSE

Kaiming Shen, *Senior Member, IEEE*, Ziping Zhao, *Member, IEEE*, Yannan Chen, *Student Member, IEEE*, Zepeng Zhang, *Student Member, IEEE*, and Hei Victor Cheng, *Member, IEEE*

***Abstract*—Fractional programming (FP) arises in various communications and signal processing problems because several key quantities in the field are fractionally structured, e.g., the Cramér-Rao bound, the Fisher information, and the signal-to-interference-plus-noise ratio (SINR). A recently proposed method called the quadratic transform has been applied to the FP problems extensively. The main contributions of the present paper are two-fold. First, we investigate how fast the quadratic transform converges. To the best of our knowledge, this is the first work that analyzes the convergence rate for the quadratic transform as well as its special case the weighted minimum mean square error (WMMSE) algorithm. Second, we accelerate the existing quadratic transform via a novel use of Nesterov's extrapolation scheme [1]. Specifically, by generalizing the minorization-maximization (MM) approach in [2], we establish a nontrivial connection between the quadratic transform and the gradient projection, thereby further incorporating the gradient extrapolation into the quadratic transform to make it converge more rapidly. Moreover, the paper showcases the practical use of the accelerated quadratic transform with two frontier wireless applications: integrated sensing and communication (ISAC) and massive multiple-input multiple-output (MIMO).**

***Index Terms*—Fractional programming (FP), convergence rate, acceleration, minorizatioin-maximization (MM).**

I. Introduction

FRACTIONAL programming (FP) is a study of optimization aimed at ratio terms. With the matrix coefficients $\boldsymbol{A}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_n$ and the vector variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, this paper focuses on the following type of ratio term:

$$\big(\boldsymbol{A}\boldsymbol{x}_i\big)^{\mathrm{H}}\left(\sum_{j=1}^{n}\boldsymbol{B}_j\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_j^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}\boldsymbol{x}_i\big),$$

where $(\cdot)^{\mathrm{H}}$ stands for the conjugate transpose. A further matrix generalization of the scalar-valued ratio is also considered in the paper. The above ratio is of significant research interest not only because it is a natural extension of the Rayleigh quotient, but also because many key quantities in communications and signal processing are written in this form, e.g., the Cramér-Rao bound, the Fisher information, and the signal-to-interference-plus-noise ratio (SINR).

A. Motivation

The quadratic transform [3], [4] is a state-of-the-art tool for FP. Differing from the classic FP methods, i.e., the Charness-Cooper method [5] and Dinkelbach's method [6], that are limited to the single-ratio case, the quadratic transform is capable of decoupling the numerator and denominator for multiple ratios simultaneously and thus renders the multi-ratio FP problem easier to address. The quadratic transform has wide connecting links to other optimization methods, e.g., [3] shows that the quadratic transform boils down to a fixed-point iteration method when applied to the power control problem, while [4] interprets the quadratic transform as a minorization-maximization (MM) method [7], [8] in general. In particular, [9] shows that the quadratic transform encompasses the weighted minimum mean square error (WMMSE) algorithm for the beamforming problem [10], [11] as a special case. Furthermore, [9] proposes a better way to use the quadratic transform than WMMSE for the discrete FP problems.

Although the theory and applications of the quadratic transform have been considerably developed over the past few years, the convergence rate of the quadratic transform remains a complete mystery (which is unknown even for the special case of the quadratic transform, the WMMSE algorithm), with the following open problems:

- *How fast does the quadratic transform converge?*
- *How is it compared to the conventional gradient method?*
- *Can we further accelerate the quadratic transform?*

The analysis in this paper sheds light on all these questions. Roughly speaking, we show that: (i) if the starting point is sufficiently close to a strict local optimum, then the quadratic transform yields an error bound of $O(1/k)$, where k is the number of iterations; (ii) the quadratic transform converges faster than the gradient method in iterations, but more slowly in time due to its higher per-iteration complexity; (iii) the proposed accelerated quadratic transform has a much lower per-iteration complexity and reduces the error bound to $O(1/k^2)$.

B. Literature Review

The traditional study on FP focuses on the ratio between two scalar-valued functions [12], [13], the simplest case of which is the single-ratio problem. Despite the nonconvexity, the single-ratio problem can be efficiently solved either by the Charnes-Cooper method [5] or by Dinkelbach's method [6] so long as the concave-convex condition [3] holds. In contrast, the multi-ratio FP is much more challenging. Except for the max-min-ratios case that can be efficiently solved by a generalized Dinkelbach's method [14], it is difficult to extend the classic FP technique to the multi-ratio problems. It turns out that even the sum-of-ratios problem is NP-complete [15] so that

Manuscript accepted to IEEE Journal on Selected Areas in Communications in May 2024.

Kaiming Shen and Yannan Chen are with the School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), 518172 Shenzhen, China (e-mail: shenkaiming@cuhk.edu.cn; yannanchen@link.cuhk.edu.cn).

Ziping Zhao and Zepeng Zhang are with the School of Information Science and Technology, ShanghaiTech University, 201210 Shanghai, China (e-mail: zipingzhao@shanghaitech.edu.cn; zhangzp1@shanghaitech.edu.cn).

Hei Victor Cheng is with the Electrical and Computer Engineering Department, Aarhus Universrity, 8200 Aarhus, Denmark (e-mail: hvc@ieee.org).

the existing approaches mostly resort to the branch-and-bound algorithm [15]–[24] and thus incur exponential complexities. As such, the early applications of FP in communications and signal processing are restricted to the energy efficiency problem [25], namely the maximization of a single ratio of energy efficiency. The advent of the quadratic transform [3] has significantly extended the scope of the applications of FP by enabling efficient multi-ratio optimization. Aside from the sum-of-ratios problem, the quadratic transform works for (or can be extended to) the log-ratio problem [9], the matrix-ratio problem [4], and the mixed max-and-min FP problem [26].

As a special case of the quadratic transform, the WMMSE algorithm [10], [11] has been extensively studied for its own sake because of the critical role it plays in the beamforming design for multiple-input multiple-output (MIMO) wireless networks. The WMMSE algorithm entails computing the matrix inverse on a per-iteration basis—which is costly in modern and future wireless networks because the matrix size grows with the number of antennas. For this reason, the computational complexity has long been identified as a major bottleneck of the WMMSE algorithm. Assuming that the channel matrices are all full row-rank, the recent work [27] exploits the weighted sum-rate (WSR) problem structure to facilitate the matrix inverse operation. The more recent work in [2], [28] goes further. It does not require any channel assumptions and can sidestep the matrix inverse operation in all cases. In particular, [2] shows that its proposed beamforming algorithm without matrix inversion boils down to a gradient projection. We will show that the above results of [2] for the WSR problem carry over to a broad range of other FP problems. Moreover, another recent endeavor [29] suggests combining Nesterov's extrapolation and WMMSE in a heuristic fashion, but the resulting algorithm still incurs the matrix inverse operation and cannot give any performance guarantees.

C. Main Results

The main results of this paper divide into three parts. First, we propose two new FP methods with provable higher efficiency than the existing quadratic transform. By the nonhomogeneous bounding technique from the MM theory [8], we devise the *nonhomogeneous quadratic transform* to eliminate the matrix inverse operation. Geometric insight is provided into the advantage of the nonhomogeneous quadratic transform over the existing quadratic transform. Further, based on the connection with the gradient projection, we incorporate Nesterov's extrapolation to accelerate convergence, namely the *extrapolated quadratic transform.*

Of equal importance in this work is the convergence rate analysis for the quadratic transform, which, to the best of our knowledge, has never been considered in the literature to date. Under some mild condition for the nonconvex analysis purpose, we show that the existing quadratic transform (which encompasses WMMSE [10], [11] as a special case) yields faster convergence in iterations than the gradient projection and the nonhomogeneous quadratic transform, but possibly slower convergence in time, and yields an error bound of $O(1/k)$ after k iterations. In comparison, the extrapolated quadratic transform leads to a superior error bound of $O(1/k^2)$.

The last part of the main results in this paper concerns the practical applications of the accelerated quadratic transform. Because the new FP technique is free of matrix inversion and has fast convergence, it is particularly suited for those frontier wireless technologies involving large antenna arrays. We present two application cases: (i) the ISAC and (ii) the massive MIMO, neither of which can be efficiently handled by the existing quadratic transform [10], [11] due to the curse of dimensionality.

D. Paper Organization and Notation

The remainder of the paper is organized as follows. Section II states the sum-of-weighted-ratios FP problem. Section III reviews the quadratic transform in [3], connects it to the gradient projection, and then proposes its acceleration based on Nesterov's extrapolation scheme. Section IV gives the convergence rate analysis for the different quadratic transform methods. Section V discusses other FP problems. Section VI discusses the extension to the matrix ratio case. Two application cases are presented in Section VII. Finally, Section VIII concludes the paper.

Here and throughout, bold lower-case letters represent vectors while bold upper-case letters represent matrices. For a vector $\boldsymbol{a}$, $\boldsymbol{a}^c$ is its complex conjugate, $\boldsymbol{a}^{\mathrm{H}}$ is its conjugate transpose, and $\|\boldsymbol{a}\|_2$ is its ℓ_2 norm. For a matrix $\boldsymbol{A}$, $\boldsymbol{A}^c$ is its complex conjugate, $\boldsymbol{A}^\top$ is its transpose, $\boldsymbol{A}^{\mathrm{H}}$ is its conjugate transpose, $\lambda_{\max}(\boldsymbol{A})$ is its largest eigenvalue, and $\|\boldsymbol{A}\|_F$ is its Frobenius norm. For a square matrix $\boldsymbol{A}$, $\mathrm{tr}(A)$ is its trace. For a positive semi-definite matrix $\boldsymbol{A}$, $\boldsymbol{A}^{\frac{1}{2}}$ is its square root. Denote by $\boldsymbol{I}$ the identity matrix, $\mathbb{C}^\ell$ the set of $\ell \times 1$ vectors, $\mathbb{C}^{d\times m}$ the set of $d \times m$ matrices, and $\mathbb{S}^{d\times d}_{++}$ the set of $d \times d$ positive definite matrices. For a complex number $a \in \mathbb{C}$, $\Re\{a\}$ is its real part. The underlined letters represent the collections of the associated vectors or matrices, e.g., for $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{C}^d$ we write $\underline{\boldsymbol{a}} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n]^\top \in \mathbb{C}^{nd}$.

II. Problem Statement

Consider a total of n ratio terms, each written as

$$M_i(\underline{\boldsymbol{x}}) = \big(\boldsymbol{A}_i\boldsymbol{x}_i\big)^{\mathrm{H}}\left(\sum_{j=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}_i\boldsymbol{x}_i\big) \tag{1}$$

for $i = 1, \ldots, n$, where $\boldsymbol{A}_i \in \mathbb{C}^{\ell\times d}, \boldsymbol{B}_{ij} \in \mathbb{C}^{\ell\times d}$, and $\boldsymbol{x}_j \in \mathbb{C}^d$. Let $f_o(\underline{\boldsymbol{x}})$ be a positive linear combination of these ratios:

$$f_o(\underline{\boldsymbol{x}}) = \sum_{i=1}^{n}\omega_i M_i(\underline{\boldsymbol{x}}), \tag{2}$$

where each weight $\omega_i > 0$. We consider the following sum-of-weighted-ratios FP problem:

$$\underset{\underline{\boldsymbol{x}}}{\text{maximize}} \quad f_o(\underline{\boldsymbol{x}}) \tag{3a}$$

$$\text{subject to} \quad \boldsymbol{x}_i \in \mathcal{X}_i, \quad i = 1, \ldots, n, \tag{3b}$$

where $\mathcal{X}_i$ is a nonempty convex constraint set on $\boldsymbol{x}_i$. Accordingly, the Cartesian product $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n$ is the constraint set on $\underline{\boldsymbol{x}}$.

Notice that we can insert constant terms into the numerator and denominator of $M_i(\underline{\boldsymbol{x}})$ by using the dummy variable $\boldsymbol{x}_j = [1,\ldots,1]^\top$. Further, when each vector variable $\boldsymbol{x}_i \in \mathbb{C}^d$ is generalized to the matrix form $\boldsymbol{X}_i \in \mathbb{C}^{d\times m}$, the ratio term also becomes a matrix:

$$\boldsymbol{M}_i(\underline{\boldsymbol{X}}) = \big(\boldsymbol{A}_i\boldsymbol{X}_i\big)^{\mathrm{H}}\left(\sum_{j=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{X}_j\boldsymbol{X}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}_i\boldsymbol{X}_i\big). \quad (4)$$

The above matrix-FP case is dealt with in Section VI. We further consider other types of FP problems in Section V.

III. Quadratic Transform

We start by reviewing the quadratic transform [3], [9], then establish its connection to the gradient projection, and lastly exploit this connection to incorporate Nesterov's extrapolation scheme [1] into the quadratic transform and thereby accelerate the convergence.

A. Preliminary

The main idea of the quadratic transform is to decouple the numerator and denominator for each ratio term. The following proposition specializes the quadratic transform [3] to our problem case in (3).

Proposition 1: The sum-of-weighted-ratios FP problem (3) is equivalent to

$$\underset{\underline{\boldsymbol{x}},\,\underline{\boldsymbol{y}}}{\text{maximize}} \quad f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}}) \quad (5a)$$

$$\text{subject to} \quad \boldsymbol{x}_i \in \mathcal{X}_i,\ \boldsymbol{y}_i \in \mathbb{C}^{\ell}, \quad i = 1,\ldots,n \quad (5b)$$

with the new objective function

$$f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}}) = \sum_{i=1}^{n}\omega_i\left[2\Re\{\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i\} - \sum_{j=1}^{n}\boldsymbol{y}_i^{\mathrm{H}}\boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\boldsymbol{y}_i\right], \quad (6)$$

in the sense that $\underline{\boldsymbol{x}}^\star$ is a solution of problem (3) if and only if $(\underline{\boldsymbol{x}}^\star, \underline{\boldsymbol{y}}^\star)$ is a solution of problem (5) with some optimal auxiliary variable $\underline{\boldsymbol{y}}^\star$.

We consider optimizing $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ alternatingly in the new problem. When $\underline{\boldsymbol{x}}$ is held fixed, by completing the square for each $\boldsymbol{y}_i$ in (6), the optimal $\boldsymbol{y}_i$ can be obtained as

$$\boldsymbol{y}_i^\star = \left(\sum_{j=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}_i\boldsymbol{x}_i\big). \quad (7)$$

To solve for $\underline{\boldsymbol{x}}$ with $\underline{\boldsymbol{y}}$ held fixed, we regroup the terms in (6) according to $\boldsymbol{x}_i$ and thus rewrite $f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}})$ as

$$f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}}) = \sum_{i=1}^{n}\Big[2\Re\{\omega_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i\} - \boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{D}_i\boldsymbol{x}_i\Big], \quad (8)$$

where

$$\boldsymbol{D}_i = \sum_{j=1}^{n}\omega_j\boldsymbol{B}_{ji}^{\mathrm{H}}\boldsymbol{y}_j\boldsymbol{y}_j^{\mathrm{H}}\boldsymbol{B}_{ji}. \quad (9)$$

Each $\boldsymbol{x}_i$ can now be optimally determined as

$$\boldsymbol{x}_i^\star = \arg\min_{\boldsymbol{x}_i\in\mathcal{X}_i}\big\|\boldsymbol{D}_i^{\frac{1}{2}}\big(\boldsymbol{x}_i - \omega_i\boldsymbol{D}_i^{-1}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i\big)\big\|_2. \quad (10)$$

Algorithm 1 Conventional Quadratic Transform [3]

```
Initialize $\underline{\boldsymbol{x}}$ to a feasible value.
repeat
    Update each $\boldsymbol{y}_i$ according to (7).
    Update each $\boldsymbol{x}_i$ according to (10).
until the value of $f_o(\underline{\boldsymbol{x}})$ converges
```

Geometrically, the optimal solution requires projecting the vector $\omega_i\boldsymbol{D}_i^{-1}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i$ onto the set $\mathcal{X}_i$ under the linear transformation $\boldsymbol{D}_i^{\frac{1}{2}}$. In particular, if $\omega_i\boldsymbol{D}_i^{-1}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i \in \mathcal{X}_i$ already, then the optimal solution reduces to

$$\boldsymbol{x}_i^\star = \omega_i\boldsymbol{D}_i^{-1}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i. \quad (11)$$

Alternating between step (7) and step (10) constitutes the conventional quadratic transform method, as summarized in Algorithm 1.

Although $\underline{\boldsymbol{x}}$ can now be iteratively updated in closed form by Algorithm 1, the matrix inversion in step (10) becomes quite costly when $\boldsymbol{D}_i$ is a large matrix—which is the case for the applications involving large antenna arrays as discussed in Section VII.

B. Connection with Gradient Projection

The following result generalizes the WSR maximization algorithm in [2]. We first introduce two lemmas.

Lemma 1: After $\underline{\boldsymbol{y}}$ has been updated as in (7) for the current $\underline{\boldsymbol{x}}$, the partial derivative of each fractional function $M_i(\underline{\boldsymbol{x}})$ with respect to the complex conjugate[1] of $\boldsymbol{x}_j$ is given by

$$\frac{\partial M_i(\underline{\boldsymbol{x}})}{\partial\boldsymbol{x}_j^c} = \begin{cases}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i - \boldsymbol{B}_{ii}^{\mathrm{H}}\boldsymbol{y}_i\boldsymbol{y}_i^{\mathrm{H}}\boldsymbol{B}_{ii}\boldsymbol{x}_i & \text{if } j = i\\ -\boldsymbol{B}_{ji}^{\mathrm{H}}\boldsymbol{y}_j\boldsymbol{y}_j^{\mathrm{H}}\boldsymbol{B}_{ji}\boldsymbol{x}_i & \text{otherwise.}\end{cases}$$

Lemma 2 (Nonhomogeneous Bound [8]): Suppose that the two Hermitian matrices $\boldsymbol{L},\boldsymbol{K} \in \mathbb{C}^{d\times d}$ satisfy $\boldsymbol{L} \preceq \boldsymbol{K}$. Then for any two vectors $\boldsymbol{x},\boldsymbol{z} \in \mathbb{C}^d$, one has

$$\boldsymbol{x}^{\mathrm{H}}\boldsymbol{L}\boldsymbol{x} \le \boldsymbol{x}^{\mathrm{H}}\boldsymbol{K}\boldsymbol{x} + 2\Re\{\boldsymbol{x}^{\mathrm{H}}(\boldsymbol{L}-\boldsymbol{K})\boldsymbol{z}\} + \boldsymbol{z}^{\mathrm{H}}(\boldsymbol{K}-\boldsymbol{L})\boldsymbol{z}, \quad (12)$$

where the equality holds if $\boldsymbol{z} = \boldsymbol{x}$. This bound is so called due to the nonhomogeneous term $2\Re\{\boldsymbol{x}^{\mathrm{H}}(\boldsymbol{L}-\boldsymbol{K})\boldsymbol{z}\}$.

Treating $\boldsymbol{D}_i$ as $\boldsymbol{L}$ in (12), we let

$$\boldsymbol{K} = \lambda_i\boldsymbol{I}, \quad \text{where } \lambda_i \ge \lambda_{\max}(\boldsymbol{D}_i) \quad (13)$$

so that $\boldsymbol{L} \preceq \boldsymbol{K}$; one possible choice is $\lambda_i = \|\boldsymbol{D}_i\|_F$. Thus, by Lemma 2, we can bound $f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}})$ in (8) from below as

$$f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}}) \ge f_t(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}},\underline{\boldsymbol{z}}) \quad (14)$$

with

$$f_t(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}},\underline{\boldsymbol{z}}) = \sum_{i=1}^{n}\Big[2\Re\{\omega_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i + \boldsymbol{x}_i^{\mathrm{H}}(\lambda_i\boldsymbol{I}-\boldsymbol{D}_i)\boldsymbol{z}_i\} + \boldsymbol{z}_i^{\mathrm{H}}(\boldsymbol{D}_i-\lambda_i\boldsymbol{I})\boldsymbol{z}_i - \lambda_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{x}_i\Big]. \quad (15)$$

[1] The motivation of adopting $\partial M_i(\underline{\boldsymbol{x}})/\partial\boldsymbol{x}_j^c$ rather than $\partial M_i(\underline{\boldsymbol{x}})/\partial\boldsymbol{x}_j$ is that the corresponding differential is simpler. According to Theorem 2 in [30], the two types of partial derivatives are both feasible for computing the stationary point. In the rest of the paper, we shall always use the former type.

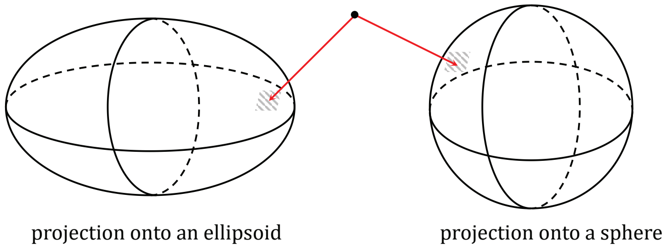


Fig. 1. The conventional quadratic transform amounts to the projection onto an ellipsoid and incurs matrix inverse operation. In contrast, the new quadratic transform avoids matrix inverse by computing the projection onto a sphere.

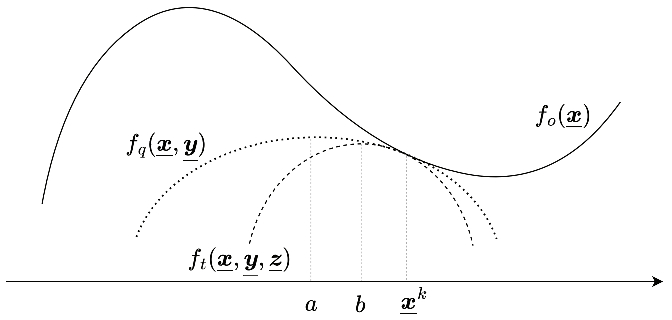

$f_o(\underline{\boldsymbol{x}})$
$f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$
$f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$
a
b
$\underline{\boldsymbol{x}}^k$

Fig. 2. Algorithm 1 approximates $f_o(\underline{\boldsymbol{x}})$ as $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$ while Algorithm 2 approximates $f_o(\underline{\boldsymbol{x}})$ as $f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$. By the MM procedure, for the current solution $\underline{\boldsymbol{x}}^k$, Algorithm 1 updates it to a, while Algorithm 2 updates it to b. Algorithm 1 converges faster in iterations because its approximation is tighter.

In particular, the equality in (14) holds if $\boldsymbol{z}_i = \boldsymbol{x}_i$ for all i.

We then use the lower bound $f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ to approximate the new objective function $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$. Observe that the three variables $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ can be iteratively optimized in closed form in $f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$. When $\underline{\boldsymbol{y}}$ and $\underline{\boldsymbol{x}}$ are both held fixed, the optimal update of $\underline{\boldsymbol{z}}$ follows by the equality condition in Lemma 2 as

$$\boldsymbol{z}_i^\star = \boldsymbol{x}_i, \quad \text{for } i = 1, \ldots, n. \tag{16}$$

After $\underline{\boldsymbol{z}}$ has been updated to $\underline{\boldsymbol{x}}$, the optimal $\boldsymbol{y}_i$ in (15) is still determined as in (7). Next, when $\underline{\boldsymbol{y}}$ and $\underline{\boldsymbol{z}}$ are both held fixed, the optimal $\boldsymbol{x}_i$ in (15) is given by

$$\begin{aligned} \boldsymbol{x}_i^\star &= \arg\min_{\boldsymbol{x}_i \in \mathcal{X}_i} \left\| \lambda_i \boldsymbol{x}_i - \omega_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i - \big(\lambda_i \boldsymbol{I} - \boldsymbol{D}_i\big) \boldsymbol{z}_i \right\|_2 \\ &= \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{z}_i + \frac{1}{\lambda_i} \big(\omega_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i - \boldsymbol{D}_i \boldsymbol{z}_i \big) \right), \end{aligned} \tag{17}$$

where $\mathcal{P}_{\mathcal{X}_i}(\cdot)$ is the Euclidean projection on $\mathcal{X}_i$. Algorithm 2 summarizes the above steps.

We are now ready to interpret the above iterative optimization as a gradient projection method. We use the superscript $k = 1, 2, \ldots$ to index the iteration, and assume that the three variables $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ are cyclically updated as

$$\underline{\boldsymbol{x}}^0 \to \cdots \to \underline{\boldsymbol{x}}^{k-1} \to \underline{\boldsymbol{z}}^k \to \underline{\boldsymbol{y}}^k \to \underline{\boldsymbol{x}}^k \to \underline{\boldsymbol{z}}^{k+1} \to \cdots.$$

With the optimal $\underline{\boldsymbol{y}}$ in (7) and the optimal $\underline{\boldsymbol{z}}$ in (16) substituted into (17), the optimal update of $\boldsymbol{x}_i$ in iteration k can be recognized as

$$\begin{aligned} \boldsymbol{x}_i^k &= \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{z}_i^k + \frac{1}{\lambda_i^k} \big(\omega_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i^k - \boldsymbol{D}_i^k \boldsymbol{z}_i^k \big) \right) \\ &\overset{(a)}{=} \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i^k} \big(\omega_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i^k - \boldsymbol{D}_i^k \boldsymbol{x}_i^{k-1} \big) \right) \\ &\overset{(b)}{=} \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i^k} \sum_{j=1}^{n} \left[\omega_j \cdot \frac{\partial M_j(\underline{\boldsymbol{x}}^{k-1})}{\partial \boldsymbol{x}_i^c} \right] \right) \\ &= \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i^k} \cdot \frac{\partial f_o(\underline{\boldsymbol{x}}^{k-1})}{\partial \boldsymbol{x}_i^c} \right), \end{aligned}$$

in which $\boldsymbol{D}_i$ is assigned the iteration index k because it has been updated by (9) for $\underline{\boldsymbol{y}}^k$, and λ_i is assigned the iteration index k because it is impacted by $\boldsymbol{D}_i^k$. Here, step (a) follows by (16), and step (b) follows by Lemma 1.

Algorithm 2 Nonhomogeneous Quadratic Transform

```
Initialize $\underline{\boldsymbol{x}}$ to a feasible value.
repeat
    Update each $\boldsymbol{z}_i$ according to (16).
    Update each $\boldsymbol{y}_i$ according to (7).
    Update each $\boldsymbol{x}_i$ according to (17).
until the value of $f_o(\underline{\boldsymbol{x}})$ converges
```

Remark 1 (Geometric Interpretation): The solution (10) in Algorithm 1 by the conventional quadratic transform amounts to the projection onto an ellipsoid, while the solution (17) in Algorithm 2 by the nonhomogeneous quadratic transform amounts to the projection onto a sphere, as shown in Fig. 1. The sphere projection is computationally much easier.

Remark 2: The connection of the quadratic transform to the gradient projection was initially proposed in [2] for the WSR problem (which is a log-ratio type of FP problem), as specified in Section V. This paper extends this connection considerably.

C. Accelerated Quadratic Transform

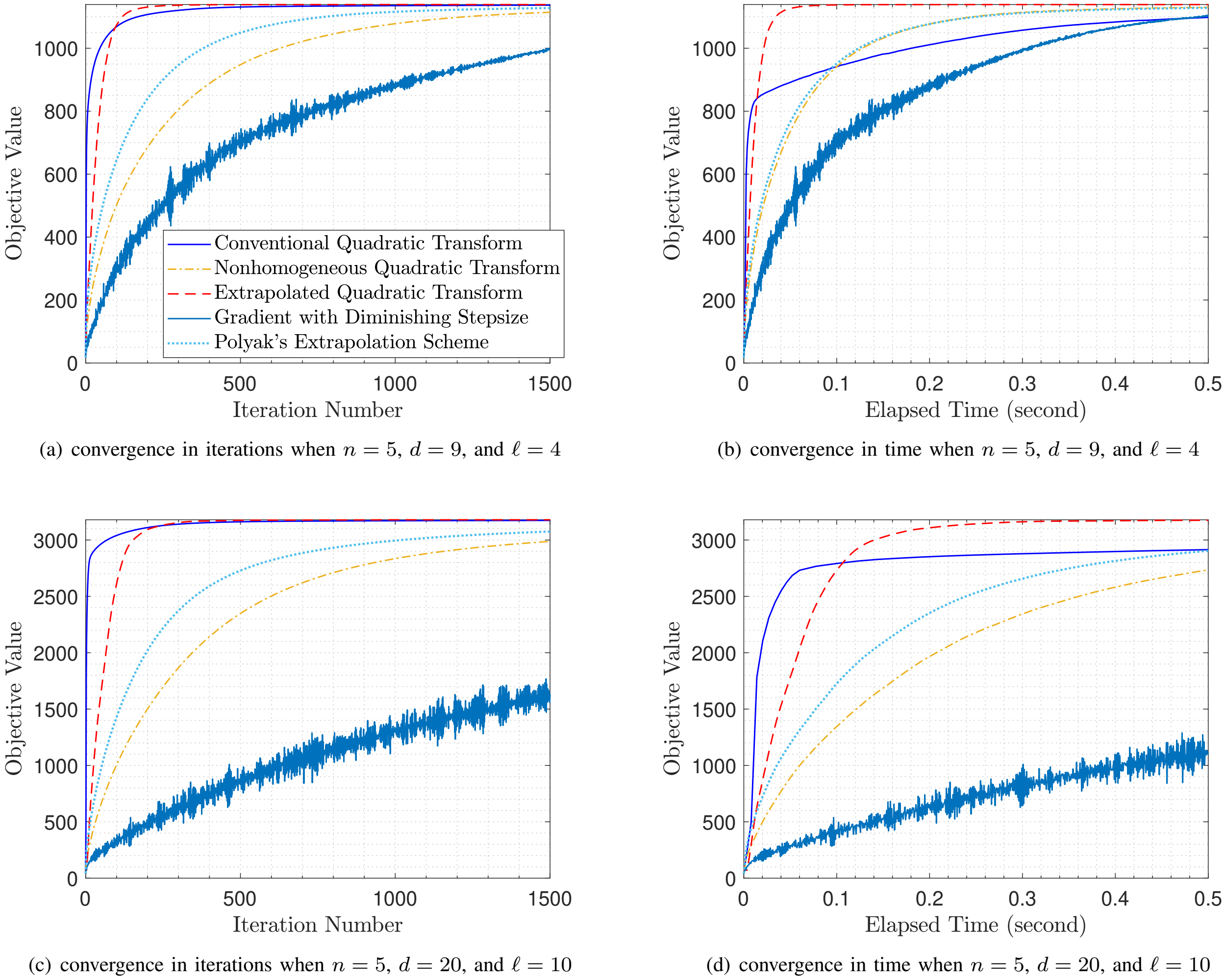
In terms of the per-iteration complexity, it is evident that Algorithm 2 is more efficient since it does not[2] require computing matrix inverse for the iterative update of $\underline{\boldsymbol{x}}$. The overall complexity is however much more difficult to examine because it also depends on how many iterations the algorithm entails to reach the convergence. Algorithm 1 uses $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$ to approximate the original objective $f_o(\underline{\boldsymbol{x}})$ from below, while Algorithm 2 further uses $f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ to approximate $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$ from below, i.e.,

$$f_o(\underline{\boldsymbol{x}}) \geq f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) \geq f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}).$$

Intuitively, Algorithm 1 has faster convergence than Algorithm 2 in iterations because it uses a tighter approximation of $f_o(\underline{\boldsymbol{x}})$, as illustrated in Fig. 2. A formal analysis of their convergence rates is provided in Section IV.

To sum up, Algorithm 2 is more efficient per iteration, but in the meanwhile requires more iterations to attain convergence. One can find a balance between Algorithm 1 and Algorithm 2 via timesharing; the convergence to a stationary point is still guaranteed by the MM theory as shown in Section IV.

[2] Notice that Algorithm 2 still requires computing the $\ell \times \ell$ matrix inverse when updating $\underline{\boldsymbol{y}}$. Actually, this matrix inverse can be also eliminated by applying Lemma 2 one more time, and consequently it would introduce a new group of auxiliary variables. We do not consider this straightforward extension in this paper because ℓ is quite small in our application cases and thus it is unnecessary to eliminate the matrix inverse in (7).



(a) convergence in iterations when $n = 5$, $d = 9$, and $\ell = 4$

(b) convergence in time when $n = 5$, $d = 9$, and $\ell = 4$

(c) convergence in iterations when $n = 5$, $d = 20$, and $\ell = 10$

(d) convergence in time when $n = 5$, $d = 20$, and $\ell = 10$

Fig. 3. Average performance of solving 100 random examples of problem (3). Let each $\omega_i = 1$, let each $\mathcal{X}_i = \{\boldsymbol{X} \in \mathbb{C}^{d\times\ell} : \mathrm{tr}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{H}}) \leq 10\}$, and randomly generate each entry of $\boldsymbol{A}_i$ and $\boldsymbol{B}_{ij}$ i.i.d. according to $\mathcal{CN}(0,1)$; further, add $\boldsymbol{I}$ to each matrix denominator to ensure its positive definiteness.

A brand-new idea here is to reduce the number of iterations for Algorithm 2 by means of extrapolation. In principle, since the nonhomogeneous quadratic transform in essence utilizes the gradients, its convergence can be accelerated by momentum or heavy-ball method. Specifically, following Nesterov's extrapolation strategy [1], we propose to extrapolate each $\boldsymbol{x}_i$ along the direction of the difference between the preceding two iterates before the gradient projection, i.e.,

$$\boldsymbol{\nu}_i^{k-1} = \boldsymbol{x}_i^{k-1} + \eta_{k-1}(\boldsymbol{x}_i^{k-1} - \boldsymbol{x}_i^{k-2}) \tag{18}$$

$$\boldsymbol{x}_i^{k} = \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{\nu}_i^{k-1} + \frac{1}{\lambda_i^k}\cdot\frac{\partial f_o(\underline{\boldsymbol{\nu}}^{k-1})}{\partial \boldsymbol{x}_i^c}\right), \tag{19}$$

where the extrapolation step η_k is chosen as

$$\eta_k = \max\left\{\frac{k-2}{k+1}, 0\right\}, \quad \text{for } k \geq 1,$$

and the starting point is $\boldsymbol{x}^{-1} = \boldsymbol{x}^0$. The above gradient projection with extrapolation can be implemented with the assistance of the auxiliary variables $(\underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ as shown in Algorithm 3, which is referred to as the extrapolated quadratic transform.

Fig. 3 compares the above three algorithms numerically. Aside from Algorithms 1 to 3, we consider the gradient method with a diminishing stepsize $1/k$ and also a variant of Algorithm 3 by using Polyak's extrapolation [31] instead of Nesterov's. Observe that Algorithm 1 converges faster than Algorithm 3 in iterations according to Fig. 3(a) and Fig. 3(c), but is slower in time as shown in Fig. 3(b) and Fig. 3(d).

Algorithm 3 Extrapolated Quadratic Transform

```
Initialize $\underline{\boldsymbol{x}}$ to a feasible value.
repeat
    Update each $\boldsymbol{\nu}_i$ according to (18) and set $\boldsymbol{x}_i = \boldsymbol{\nu}_i$.
    Update each $\boldsymbol{z}_i$ according to (16).
    Update each $\boldsymbol{y}_i$ according to (7).
    Update each $\boldsymbol{x}_i$ according to (17).
until the value of $f_o(\underline{\boldsymbol{x}})$ converges
```

IV. Convergence Analysis

We have introduced three types of quadratic transform: the conventional quadratic transform (Algorithm 1), the nonhomogeneous quadratic transform (Algorithm 2), and the extrapolated quadratic transform (Algorithm 3). This section comprises two main results. First, we show that the three types of quadratic transform all guarantee convergence to a stationary-point solution. Second, we analyze the rate of convergence for them.

The convergence proof uses the MM interpretation of the quadratic transform from [2], [4]. Write the optimal update of $\underline{\boldsymbol{y}}$ in (7) as a function of $\underline{\boldsymbol{x}}$:

$$\mathcal{Y}(\underline{\boldsymbol{x}}) = \underline{\boldsymbol{y}} \text{ with each } \boldsymbol{y}_i = \left(\sum_{i=1}^{n} \boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}_i\boldsymbol{x}_i\big).$$

By Algorithm 1, after $\underline{\boldsymbol{y}}^k$ is optimally updated for the previous $\underline{\boldsymbol{x}}^{k-1}$, the current new objective function $f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}})$ can be rewritten as a function $r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1})$ of $\underline{\boldsymbol{x}}$ conditioned on $\underline{\boldsymbol{x}}^{k-1}$:

$$r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}) = f_q(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}^{k-1})), \tag{20}$$

and accordingly the update of $\underline{\boldsymbol{x}}$ in (10) can be rewritten as

$$\underline{\boldsymbol{x}}^k = \arg\max_{\underline{\boldsymbol{x}}\in\mathcal{X}} r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}). \tag{21}$$

Importantly, it always holds that

$$r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}) \le f_o(\underline{\boldsymbol{x}}) \;\text{ and }\; r_q(\underline{\boldsymbol{x}}^{k-1}|\underline{\boldsymbol{x}}^{k-1}) = f_o(\underline{\boldsymbol{x}}^{k-1}),$$

so updating $\underline{\boldsymbol{y}}$ for $\underline{\boldsymbol{x}}^{k-1}$ is equivalent to constructing a surrogate function $r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1})$ for $f_o(\underline{\boldsymbol{x}})$ at $\underline{\boldsymbol{x}}^{k-1}$, namely the *minorization* step. Moreover, (21) can be recognized as the *maximization* step. As such, Algorithm 1 turns out to be an MM method, and hence it guarantees convergence to a stationary point of problem (3). By a similar argument, we can also interpret Algorithm 2 as an MM method, with the surrogate function

$$r_t(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}) = f_t(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}^{k-1}),\underline{\boldsymbol{x}}^{k-1}). \tag{22}$$

Besides, the tradeoff between Algorithm 1 and Algorithm 2 via timesharing constitutes an MM algorithm as well and hence preserves the stationary-point convergence. Furthermore, recall that Algorithm 2 can also be interpreted as a gradient projection method; since it has provable convergence to a stationary point, so does its accelerated version Algorithm 3. The following proposition summarizes the above results based on the MM interpretation.

Proposition 2: For the sum-of-weighted-ratios FP problem (3), the conventional quadratic transform in Algorithm 1, the nonhomogeneous quadratic transform in Algorithm 2, and the extrapolated quadratic transform in Algorithm 3 all guarantee convergence to some stationary point, with the original objective value increasing monotonically.

We then analyze the rate of convergence for the various quadratic transform methods. Due to the nonconvexity of the FP problem, the global analysis (assuming that the starting point is far from any stationary point) is intractable. We would like to give a local analysis by restricting the constraint set to a small neighborhood of a strict local optimum (so that the starting point is not far away), i.e.,

$$\mathcal{X} = \big\{\underline{\boldsymbol{x}} : \|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^*\|_2 \le R\big\}, \tag{23}$$

where $\underline{\boldsymbol{x}}^*$ is a strict local optimum of (3) satisfying

$$\nabla^2 f_o(\underline{\boldsymbol{x}}^*) \preceq -\xi\boldsymbol{I} \prec \boldsymbol{0}$$

for some strictly positive constant $\xi > 0$, and the radius $R > 0$ is sufficiently small so that $f_o(\underline{\boldsymbol{x}})$ is concave on $\mathcal{X}$. Assume that the Hessian of $f_o(\underline{\boldsymbol{x}})$ is L-Lipschitz continuous on $\mathcal{X}$, i.e.,

$$\|\nabla^2 f_o(\underline{\boldsymbol{x}}) - \nabla^2 f_o(\underline{\boldsymbol{x}}')\|_2 \le L\|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}'\|_2$$

for any $\underline{\boldsymbol{x}},\underline{\boldsymbol{x}}' \in \mathcal{X}$. By Corollary 1.2.2 of [1], we have

$$\nabla^2 f_o(\underline{\boldsymbol{x}}) \preceq \nabla^2 f_o(\underline{\boldsymbol{x}}^*) + L\|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^*\|_2\boldsymbol{I},$$

so it suffices to require $R \le \xi/L$ in order to ensure that $f_o(\underline{\boldsymbol{x}})$ is concave on $\mathcal{X}$.

The convergence rate analysis also uses the MM interpretation. Conditioned on $\underline{\boldsymbol{x}}' \in \mathcal{X}$, write the difference between $f_o(\underline{\boldsymbol{x}})$ and each surrogate function as a function of $\underline{\boldsymbol{x}} \in \mathcal{X}$:

$$\delta_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}') = f_o(\underline{\boldsymbol{x}}) - f_q(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}'))$$
$$\delta_t(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}') = f_o(\underline{\boldsymbol{x}}) - f_t(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}'),\underline{\boldsymbol{x}}').$$

It can be readily shown that

$$\delta_q(\underline{\boldsymbol{x}}^k|\underline{\boldsymbol{x}}^k) = \delta_t(\underline{\boldsymbol{x}}^k|\underline{\boldsymbol{x}}^k) = 0 \tag{24a}$$
$$\nabla\delta_q(\underline{\boldsymbol{x}}^k|\underline{\boldsymbol{x}}^k) = \nabla\delta_t(\underline{\boldsymbol{x}}^k|\underline{\boldsymbol{x}}^k) = 0. \tag{24b}$$

Moreover, define the following quantities to be the largest eigenvalues of two Hessian matrices:

$$\Lambda_q = \max_{\underline{\boldsymbol{x}}\in\mathcal{X}} \lambda_{\max}\big(\nabla^2\delta_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}})\big)$$
$$\Lambda_t = \max_{\underline{\boldsymbol{x}}\in\mathcal{X}} \lambda_{\max}\big(\nabla^2\delta_t(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}})\big).$$

Recall each ratio $M_i(\underline{\boldsymbol{x}})$ is finite with nonsingular denominator matrix $\sum_{i=1}^n \boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}$, so each entry of $\underline{\boldsymbol{y}}$ is finite and hence each $\lambda_{\max}(\boldsymbol{D}_i) < \infty$ according to (9). As a result, $\lambda_{\max}(\nabla^2_{\underline{\boldsymbol{x}}} f_t(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}},\underline{\boldsymbol{z}}) = -\max_i 2\lambda_{\max}(\boldsymbol{D}_i) > -\infty$. Further, $\Lambda_t \le \lambda_{\max}(\nabla^2_{\underline{\boldsymbol{x}}} f_o(\underline{\boldsymbol{x}})) - \lambda_{\max}(\nabla^2_{\underline{\boldsymbol{x}}} f_t(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}'),\underline{\boldsymbol{x}}')) < \infty$. Moreover, because $\nabla^2_{\underline{\boldsymbol{x}}} f_t(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}'),\underline{\boldsymbol{x}}') \preceq \nabla^2_{\underline{\boldsymbol{x}}} f_o(\underline{\boldsymbol{x}},\mathcal{Y}(\underline{\boldsymbol{x}}'))$, we must have $\Lambda_q \le \Lambda_t < \infty$. We are now ready to derive the (local) convergence rates of Algorithm 1 and Algorithm 2, as stated in the following proposition.

Proposition 3 (Convergence Rates of Algorithm 1 and Algorithm 2): Suppose that the Hessian of the objective function $f_o(\underline{\boldsymbol{x}})$ of problem (3) is L-Lipschitz continuous and that the radius of the constraint set $\mathcal{X}$ is sufficiently small as defined in (23). Then the local convergence rate of Algorithm 1 or Algorithm 2 is

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^1) \le \frac{\Lambda R^2}{2} + \frac{LR^3}{6} \tag{25}$$

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^k) \le \frac{2\Lambda R^2 + 2LR^3/3}{k+3}, \quad \text{for } k \ge 2, \tag{26}$$

where

$$\Lambda = \begin{cases} \Lambda_q & \text{for Algorithm 1} \\ \Lambda_t & \text{for Algorithm 2.} \end{cases} \tag{27}$$

Proof: See Appendix A. ■

Because $0 \le \Lambda_q \le \Lambda_t$, Algorithm 1 converges faster than Algorithm 2 in iterations according to Proposition 3. Notice that Λ_q and Λ_t in essence characterize how well their corresponding surrogate functions approximate the second-order profile of $f_o(\underline{\boldsymbol{x}})$. In the ideal case, the surrogate function and $f_o(\underline{\boldsymbol{x}})$ have exactly the same second-order profile so that $\Lambda = 0$, then the objective-value error bound in Proposition 3 becomes

$$f_o(\underline{\boldsymbol{x}}) - f_o(\underline{\boldsymbol{x}}^k) \le \frac{L}{6}\|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^{k-1}\|_2^3, \tag{28}$$

which also holds for the *cubically regularized Newton's method* due to Nesterov as shown in [1]. Equipped with the error bound (28), it immediately follows from Theorem 4.1.4 in [1] that

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^1) \le \frac{LR^3}{6} \tag{29}$$

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^k) \le \frac{LR^3}{2(1+k/3)^2}, \quad \text{for } k \ge 2. \tag{30}$$

We now show that the extrapolated quadratic transform method in Algorithm 3 can achieve fairly close to the ideal case stated in (29) and (30). Since Algorithm 2 is a gradient projection method and Algorithm 3 accelerates it by Nesterov's extrapolation, we immediately obtain the following convergence rate from Proposition 6.2.1 of [32].

Proposition 4 (Convergence Rate of Algorithm 3): Suppose that the gradient of $f_o(\underline{\boldsymbol{x}})$ is C-Lipschitz continuous and let $\lambda_i^k = 1/(2C)$. Then Algorithm 3 yields

$$f(\underline{\boldsymbol{x}}^*) - f(\underline{\boldsymbol{x}}) \le \frac{2C \cdot [f(\underline{\boldsymbol{x}}^*) - f(\underline{\boldsymbol{x}}^0)]}{(k+1)^2}, \quad \text{for } k \ge 1. \tag{31}$$

In summary, as compared to Algorithm 1 and Algorithm 2 that both yield an objective-value error bound of $O(1/k)$, Algorithm 3 yields a smaller error bound of $O(1/k^2)$.

V. Extension to Other FP Problems

The discussion thus far is limited to the sum-of-weighted-ratios problem in (3). Our goal here is to extend the above results to other FP problems. We begin with a recent discovery in [2] that the joint application of the nonhomogeneous quadratic transform and the Lagrangian dual transform [9] can be interpreted as a gradient projection method. We then generalize the above result for a general utility FP problem under certain conditions.

A. Log-Ratio Case

We first focus on the WSR problem and aim to rederive the main result in [2]. Consider n interfering wireless links, each having d transmit antennas and ℓ receive antennas. Let $\boldsymbol{x}_i$ be the transmit beamformer of link i, let ρ be the power constraint, and let μ_i be the rate weight of link i. Assume that one data stream is transmitted on each link. With $M_i(\underline{\boldsymbol{x}})$ interpreted as the SINR of link i, the WSR problem has the log-ratio form:

$$\underset{\underline{\boldsymbol{x}}}{\text{maximize}} \quad \sum_{i=1}^n \mu_i \log\big(1 + M_i(\underline{\boldsymbol{x}})\big) \tag{32a}$$

$$\text{subject to} \quad \|\boldsymbol{x}_i\|_2^2 \le \rho, \quad i = 1, \ldots, n. \tag{32b}$$

By the Lagrangian dual transform [9], the above problem can be converted to the sum-of-weighted-ratios problem:

$$\underset{\underline{\boldsymbol{x}},\,\underline{\gamma}}{\text{maximize}} \quad h(\underline{\boldsymbol{x}}, \underline{\gamma}) \tag{33a}$$

$$\text{subject to} \quad \|\boldsymbol{x}_i\|_2^2 \le \rho, \quad i = 1, \ldots, n \tag{33b}$$

$$\gamma_i \in \mathbb{R}, \quad i = 1, \ldots, n, \tag{33c}$$

where

$$h(\underline{\boldsymbol{x}}, \underline{\gamma}) = \sum_{i=1}^n \Big[\mu_i(1+\gamma_i)\cdot \hat{M}_i(\underline{\boldsymbol{x}})\Big] + \sum_{i=1}^n \Big[\mu_i \log(1+\gamma_i) - \mu_i\gamma_i\Big] \tag{34}$$

and

$$\hat{M}_i(\underline{\boldsymbol{x}}) = \big(\boldsymbol{A}_i\boldsymbol{x}_i\big)^{\mathrm{H}}\left(\boldsymbol{A}_i\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}} + \sum_{j=1}^n \boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}_i\boldsymbol{x}_i\big). \tag{35}$$

Observe that the optimization of $\underline{\boldsymbol{x}}$ in (33) amounts to a sum-of-weighted-ratios problem, so we can use the nonhomogeneous quadratic transform to further recast $h(\underline{\boldsymbol{x}}, \underline{\gamma})$ to

$$\begin{aligned} f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\gamma}) = \sum_{i=1}^n \Big[&2\Re\{\mu_i(1+\gamma_i)\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i \\ &+ \boldsymbol{x}_i^{\mathrm{H}}(\lambda_i\boldsymbol{I} - \boldsymbol{D}_i)\boldsymbol{z}_i\} + \boldsymbol{z}_i^{\mathrm{H}}(\boldsymbol{D}_i - \lambda_i\boldsymbol{I})\boldsymbol{z}_i - \lambda_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{x}_i\Big] \\ &+ \sum_{i=1}^n \Big[\mu_i\log(1+\gamma_i) - \mu_i\gamma_i\Big], \end{aligned} \tag{36}$$

where

$$\boldsymbol{D}_i = \mu_i(1+\gamma_i)\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i\boldsymbol{y}_i^{\mathrm{H}}\boldsymbol{A}_i + \sum_{j=1}^n \mu_j(1+\gamma_j)\boldsymbol{B}_{ji}^{\mathrm{H}}\boldsymbol{y}_j\boldsymbol{y}_j^{\mathrm{H}}\boldsymbol{B}_{ji}$$

and

$$\lambda_i \ge \lambda_{\max}(\boldsymbol{D}_i).$$

We optimize $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\gamma})$ in (36) iteratively. Again, by Lemma 2, each $\boldsymbol{z}_i$ is optimally determined as

$$\boldsymbol{z}_i^\star = \boldsymbol{x}_i. \tag{37}$$

By completing the square for each $\boldsymbol{y}_i$ in (36), we obtain the optimal $\boldsymbol{y}_i$ as

$$\boldsymbol{y}_i^\star = \left(\boldsymbol{A}_i\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}} + \sum_{i=1}^n \boldsymbol{B}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\boldsymbol{A}_i\boldsymbol{x}_i. \tag{38}$$

After the above $\boldsymbol{z}_i^\star$ and $\boldsymbol{y}_i^\star$ are plugged in (36), by solving $\partial f_t/\partial\gamma_i = 0$, each optimal γ_i can be obtained as

$$\gamma_i^\star = M_i(\underline{\boldsymbol{x}}). \tag{39}$$

Moreover, when $(\underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\gamma})$ are all held fixed, we complete the square for each $\boldsymbol{x}_i$ in $g_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\gamma})$ and solve for $\boldsymbol{x}_i$ as

$$\boldsymbol{x}_i^\star = \begin{cases} \hat{\boldsymbol{x}}_i & \text{if } \|\hat{\boldsymbol{x}}_i\|_2^2 \le \rho \\ \big(\sqrt{\rho}/\|\hat{\boldsymbol{x}}_i\|_2\big)\hat{\boldsymbol{x}}_i & \text{otherwise,} \end{cases} \tag{40}$$

where

$$\hat{\boldsymbol{x}}_i = \boldsymbol{z}_i + \frac{1}{\lambda_i}\Big(\mu_i(1+\gamma_i)\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i - \boldsymbol{D}_i\boldsymbol{z}_i\Big). \tag{41}$$

Putting (38), (39), (40), and (41) together gives rise to the beamforming algorithm proposed in [2]. Importantly, [2] points out that its proposed beamforming algorithm amounts to a gradient projection method, as specified in the following proposition.

Proposition 5: For the WSR problem (32), updating $\underline{\boldsymbol{z}}$, $\underline{\boldsymbol{y}}$, $\underline{\gamma}$, and $\underline{\boldsymbol{x}}$ by (37), (38), (39), and (40), respectively, in an iterative fashion is equivalent to the gradient projection:

$$\boldsymbol{x}_i^k = \mathcal{P}_{\|\boldsymbol{x}_i\|_2^2\le\rho}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i}\cdot\frac{\partial f_o(\underline{\boldsymbol{x}}^{k-1})}{\partial \boldsymbol{x}_i^c}\right),$$

where $f_o(\underline{\boldsymbol{x}}) = \sum_{i=1}^n \mu_i \log\big(1+M_i(\underline{\boldsymbol{x}})\big)$ is the optimization objective of the WSR problem.

Remark 3: For the log-ratio case, we treat each $\mu_j(1+\gamma_j)$ in (34) as the weight of the ratio $\hat{M}_i(\underline{\boldsymbol{x}})$, and then applying the conventional quadratic transform recovers the WMMSE algorithm [10], [11]. Alternatively, we could have let $\hat{M}_i(\underline{\boldsymbol{x}})$ absorb $\mu_i(1+\gamma_i)$ and then treat $\mu_i(1+\gamma_i)\hat{M}_i(\underline{\boldsymbol{x}})$ as the ratio term with weight one; [9] shows the above type of ratio term is more suited for the discrete FP solving. In fact, there are infinitely many ways of deciding which part is the ratio term and which part is the weight. The resulting quadratic transform method can be accelerated as in Algorithm 3, regardless.

B. General Utility Case

We now consider the following FP problem whose objective function is a general utility function of multiple ratios:

$$\underset{\underline{\boldsymbol{x}}}{\text{maximize}} \quad \mathcal{G}\big(M_1(\underline{\boldsymbol{x}}),\ldots,M_n(\underline{\boldsymbol{x}})\big) \tag{42a}$$

$$\text{subject to} \quad \boldsymbol{x}_i \in \mathcal{X}_i, \quad i=1,\ldots,n, \tag{42b}$$

where $\mathcal{G}:\mathbb{R}^n \to \mathbb{R}$ is a differentiable function with n ratio arguments. Our goal is to show that the gradient projection interpretation continues to hold for this general utility FP problem under certain condition; the previous log-ratio WSR problem case [2] turns out to be its special case.

Assume that there exists a surrogate function $h(\underline{\boldsymbol{x}},\boldsymbol{g})$ of $\mathcal{G}\big(M_1(\underline{\boldsymbol{x}}),\ldots,M_n(\underline{\boldsymbol{x}})\big)$ with the form of

$$h(\underline{\boldsymbol{x}},\boldsymbol{g}) = \sum_{i=1}^n \Big[\alpha_i(\boldsymbol{g})\cdot \hat{M}_i(\underline{\boldsymbol{x}})\Big] + \beta(\boldsymbol{g}), \tag{43}$$

where $\boldsymbol{g}$ is an auxiliary variable, $\alpha_i(\boldsymbol{g})\ge 0$, $i=1,\ldots,n$, and $\beta(\boldsymbol{g})$ are differentiable scalar-valued functions of $\boldsymbol{g}$, and

$$\hat{M}_i(\underline{\boldsymbol{x}}) = \big(\hat{\boldsymbol{A}}_i\boldsymbol{x}_i\big)^{\mathrm{H}}\left(\sum_{j=1}^n \hat{\boldsymbol{B}}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\hat{\boldsymbol{B}}_{ij}^{\mathrm{H}}\right)^{-1}\big(\hat{\boldsymbol{A}}_i\boldsymbol{x}_i\big). \tag{44}$$

Note that $\hat{M}_i(\underline{\boldsymbol{x}}) \ne M_i(\underline{\boldsymbol{x}})$ in general. Since $h(\underline{\boldsymbol{x}},\boldsymbol{g})$ is a surrogate function of $f_o(\underline{\boldsymbol{x}})$, we have

$$\mathcal{G}\big(M_1(\underline{\boldsymbol{x}}),\ldots,M_n(\underline{\boldsymbol{x}})\big) = \max_{\boldsymbol{g}} h(\underline{\boldsymbol{x}},\boldsymbol{g}). \tag{45}$$

When $\boldsymbol{g}$ is held fixed, $h(\underline{\boldsymbol{x}},\boldsymbol{g})$ can be recognized as a sum-of-weighted-ratios objective function of $\underline{\boldsymbol{x}}$, so we further use the nonhomogeneous quadratic transform to reformulate $h(\underline{\boldsymbol{x}},\boldsymbol{g})$ as

$$\begin{aligned} f_t(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}},\underline{\boldsymbol{z}},\boldsymbol{g}) = \sum_{i=1}^n \Big[&2\Re\{\alpha_i(\boldsymbol{g})\boldsymbol{x}_i^{\mathrm{H}}\hat{\boldsymbol{A}}_i^{\mathrm{H}}\boldsymbol{y}_i + \boldsymbol{x}_i^{\mathrm{H}}(\lambda_i\boldsymbol{I}-\boldsymbol{D}_i)\boldsymbol{z}_i\} \\ &+ \boldsymbol{z}_i^{\mathrm{H}}(\boldsymbol{D}_i-\lambda_i\boldsymbol{I})\boldsymbol{z}_i - \lambda_i\boldsymbol{x}_i^{\mathrm{H}}\boldsymbol{x}_i\Big] + \beta(\boldsymbol{g}), \end{aligned} \tag{46}$$

where

$$\boldsymbol{D}_i = \sum_{j=1}^n \alpha_j(\boldsymbol{g})\hat{\boldsymbol{B}}_{ji}^{\mathrm{H}}\boldsymbol{y}_j\boldsymbol{y}_j^{\mathrm{H}}\hat{\boldsymbol{B}}_{ji}$$

and

$$\lambda_i \ge \lambda_{\max}(\boldsymbol{D}_i).$$

Again, we then optimize the variables $(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}},\underline{\boldsymbol{z}},\boldsymbol{g})$ iteratively in (46). According to Lemma 2, the optimal update of $\boldsymbol{z}_i$ is

$$\boldsymbol{z}_i^\star = \boldsymbol{x}_i. \tag{47}$$

By completing the square for each $\boldsymbol{y}_i$ in (46), the optimal $\boldsymbol{y}_i$ is obtained as

$$\boldsymbol{y}_i^\star = \left(\sum_{j=1}^n \hat{\boldsymbol{B}}_{ij}\boldsymbol{x}_j\boldsymbol{x}_j^{\mathrm{H}}\hat{\boldsymbol{B}}_{ij}^{\mathrm{H}}\right)^{-1}\big(\hat{\boldsymbol{A}}_i\boldsymbol{x}_i\big). \tag{48}$$

Subsequently, the optimal $\boldsymbol{g}$ is given by

$$\boldsymbol{g}^\star = \arg\max_{\boldsymbol{g}} h(\underline{\boldsymbol{x}},\boldsymbol{g}). \tag{49}$$

When the above variables are held fixed, we update each $\boldsymbol{x}_i$ optimally as

$$\boldsymbol{x}_i^\star = \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{z}_i + \frac{1}{\lambda_i}\big(\alpha_i(\boldsymbol{g})\hat{\boldsymbol{A}}_i^{\mathrm{H}}\boldsymbol{y}_i - \boldsymbol{D}_i\boldsymbol{z}_i\big)\right). \tag{50}$$

We now show that the result of Proposition 5 carries over to the general utility case.

Proposition 6: For problem (42), suppose that there exists a surrogate function $h(\underline{\boldsymbol{x}},\boldsymbol{g})$ satisfying (43), (44), and (45). Then updating $\underline{\boldsymbol{z}}$, $\underline{\boldsymbol{y}}$, $\boldsymbol{g}$, and $\underline{\boldsymbol{x}}$ by (47), (48), (49), and (50), respectively, in an iterative fashion is equivalent to the gradient projection:

$$\boldsymbol{x}_i^k = \mathcal{P}_{\boldsymbol{x}_i\in\mathcal{X}_i}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i}\cdot\frac{\partial\mathcal{G}\big(M_1(\underline{\boldsymbol{x}}),\ldots,M_n(\underline{\boldsymbol{x}})\big)}{\partial\boldsymbol{x}_i^c}\right),$$

Proof: See Appendix B. ■

Again, we could have adopted the extrapolated quadratic transform in place of the nonhomogeneous quadratic transform for the above method, and thereby accelerate the convergence.

VI. Matrix Ratio Case

This section extends the preceding results to the generalized matrix ratios with the matrix variables $\boldsymbol{X}_i \in \mathbb{C}^{d\times m}$ as in (4). The sum-of-weighted-ratios FP problem in (3) now becomes

$$\underset{\underline{\boldsymbol{X}}}{\text{maximize}} \quad \sum_{i=1}^n \Big[\omega_i\cdot\operatorname{tr}\big(\boldsymbol{M}_i(\underline{\boldsymbol{X}})\big)\Big] \tag{51a}$$

$$\text{subject to} \quad \boldsymbol{X}_i\in\mathcal{X}_i, \quad \text{for } i=1,\ldots,n. \tag{51b}$$

$$f_q(\underline{\boldsymbol{X}},\underline{\boldsymbol{Y}})=\sum_{i=1}^{n}\left[\omega_i\cdot\mathrm{tr}\left(2\Re\{\boldsymbol{X}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{Y}_i\}-\boldsymbol{Y}_i^{\mathrm{H}}\left(\sum_{j=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{X}_j\boldsymbol{X}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)\boldsymbol{Y}_i\right)\right] \tag{52}$$

$$f_t(\underline{\boldsymbol{X}},\underline{\boldsymbol{Y}},\underline{\boldsymbol{Z}})=\sum_{i=1}^{n}\left[\mathrm{tr}\Big(2\Re\{\omega_i\boldsymbol{X}_i^{\mathrm{H}}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{Y}_i+\boldsymbol{X}_i^{\mathrm{H}}(\lambda_i\boldsymbol{I}-\boldsymbol{D}_i)\boldsymbol{Z}_i\}+\boldsymbol{Z}_i^{\mathrm{H}}(\boldsymbol{D}_i-\lambda_i\boldsymbol{I})\boldsymbol{Z}_i-\lambda_i\boldsymbol{X}_i^{\mathrm{H}}\boldsymbol{X}_i\Big)\right] \tag{56}$$

The new objective function $f_q(\underline{\boldsymbol{X}},\underline{\boldsymbol{Y}})$ by the quadratic transform is shown in (52), where an auxiliary variable $\boldsymbol{Y}_i\in\mathbb{C}^{d\times M}$ is introduced for each matrix ratio $\boldsymbol{M}_i(\underline{\boldsymbol{X}})$.

Optimizing $\underline{\boldsymbol{X}}$ and $\underline{\boldsymbol{Y}}$ alternatingly in $f_q(\underline{\boldsymbol{X}},\underline{\boldsymbol{Y}})$ leads us to the matrix-ratio version of Algorithm 1, wherein

$$\boldsymbol{Y}_i^\star=\left(\sum_{i=1}^{n}\boldsymbol{B}_{ij}\boldsymbol{X}_j\boldsymbol{X}_j^{\mathrm{H}}\boldsymbol{B}_{ij}^{\mathrm{H}}\right)^{-1}\big(\boldsymbol{A}_i\boldsymbol{X}_i\big) \tag{53}$$

and

$$\boldsymbol{X}_i^\star=\arg\min_{\boldsymbol{X}_i\in\mathcal{X}_i}\left\|\boldsymbol{D}_i^{\frac{1}{2}}\left(\boldsymbol{X}_i-\omega_i\boldsymbol{D}_i^{-1}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{Y}_i\right)\right\|_F \tag{54}$$

with

$$\boldsymbol{D}_i=\sum_{j=1}^{n}\omega_j\boldsymbol{B}_{ji}^{\mathrm{H}}\boldsymbol{Y}_j\boldsymbol{Y}_j^{\mathrm{H}}\boldsymbol{B}_{ji}. \tag{55}$$

We further extend the new objective function $f_t(\underline{\boldsymbol{X}},\underline{\boldsymbol{Y}},\underline{\boldsymbol{Z}})$ of the nonhomogeneous quadratic transform as shown in (56), with an auxiliary variable $\boldsymbol{Z}_i\in\mathbb{C}^{d\times m}$ introduced for each $\boldsymbol{D}_i$. Again, we optimize the variables of $f_t(\underline{\boldsymbol{X}},\underline{\boldsymbol{Y}},\underline{\boldsymbol{Z}})$ in an iterative fashion: $\underline{\boldsymbol{Z}}$ is optimally updated to $\underline{\boldsymbol{X}}$, $\underline{\boldsymbol{Y}}$ is optimally updated as in (53), and $\underline{\boldsymbol{X}}$ is optimally updated as

$$\boldsymbol{X}_i^\star=\mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{Z}_i+\frac{1}{\lambda_i}\Big(\omega_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{Y}_i-\boldsymbol{D}_i\boldsymbol{Z}_i\Big)\right).$$

Combining the above steps gives the matrix-ratio version of Algorithm 2. Its connection with the gradient projection continues to hold:

$$\boldsymbol{X}^k=\mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{X}_i^{k-1}+\frac{1}{\lambda_i^k}\cdot\frac{\partial f_o(\underline{\boldsymbol{X}}^{k-1})}{\partial\boldsymbol{X}_i^c}\right). \tag{57}$$

Equipped with (57), Algorithm 3 can be immediately extended to the matrix ratio case as well.

VII. Two Application Cases

A. Integrated Sensing and Communications (ISAC)

Consider two base-stations (BSs) as depicted in Fig. 4. BS 1 performs ISAC while BS 2 only performs downlink transmission. The two BSs have M transmit antennas each, the two downlink users have N antennas each, and BS 1 has N_r radar receive antennas. Denote by $\boldsymbol{H}_{ij}\in\mathbb{C}^{N\times M}$ the channel from BS j to downlink user i, where $i,j\in\{1,2\}$, $\boldsymbol{G}\in\mathbb{C}^{N_r\times M}$ the channel from BS 2 to the radar receiver at BS 1, σ_i^2 the background noise power at downlink user i, and σ_r^2 the background noise power at the radar antenna array of BS 1. Let $\boldsymbol{v}_i\in\mathbb{C}^M$ be the transmit precoder at BS i subject to the power constraint $P_{\max}$, i.e., $\|\boldsymbol{v}_i\|_2^2\le P_{\max}$.

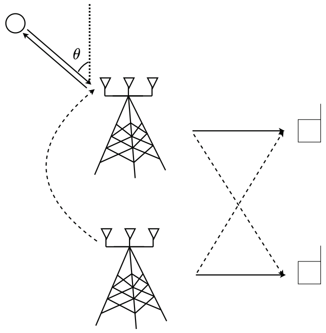


Fig. 4. Two BSs serve one downlink user each. One BS performs ISAC while the other BS only performs transmission; the aim of sensing is to recover the angle θ. The dashed arrows represent the interference.

Moreover, for BS 1, consider the transmit steering vector $\boldsymbol{a}_t(\theta)\in\mathbb{C}^M$ and the receive steering vector $\boldsymbol{a}_r(\theta)\in\mathbb{C}^{N_r}$, both dependent on the target angle θ as shown in Fig. 4:

$$\boldsymbol{a}_t(\theta)=\left[1,e^{-j\pi\sin\theta},\ldots,e^{-j\pi(M-1)\sin\theta}\right]^\top$$
$$\boldsymbol{a}_r(\theta)=\left[1,e^{-j\pi\sin\theta},\ldots,e^{-j\pi(N_r-1)\sin\theta}\right]^\top.$$

Thus, for the complex Gaussian symbol $\boldsymbol{s}_i\sim\mathcal{CN}(\boldsymbol{0},\boldsymbol{I})$ from BS $i\in\{1,2\}$, the received echo signal at BS 1 is given by

$$r=\xi\boldsymbol{a}_r(\theta)\boldsymbol{a}_t(\theta)^\top\boldsymbol{v}_1\boldsymbol{s}_1+\boldsymbol{G}\boldsymbol{v}_2\boldsymbol{s}_2+\boldsymbol{z}, \tag{58}$$

where $\xi\in\mathbb{C}$ is the reflection coefficient, and $\boldsymbol{z}\sim\mathcal{CN}(\boldsymbol{0},\sigma_r^2\boldsymbol{I})$ is the background noise at BS 1. Let $\hat{\boldsymbol{F}}=\boldsymbol{G}\boldsymbol{v}_2\boldsymbol{s}_2+\boldsymbol{z}$ and define the interference-plus-noise covariance matrix to be

$$\boldsymbol{Q}=\mathbb{E}[\hat{\boldsymbol{F}}\hat{\boldsymbol{F}}^{\mathrm{H}}]=\sigma_r^2\boldsymbol{I}+\boldsymbol{G}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{G}^{\mathrm{H}}. \tag{59}$$

The Fisher information about the target angle θ in Fig. 4 is

$$J_\theta=\alpha\boldsymbol{v}_1^{\mathrm{H}}\dot{\boldsymbol{A}}^{\mathrm{H}}\boldsymbol{Q}^{-1}\dot{\boldsymbol{A}}\boldsymbol{v}_1, \tag{60}$$

where $\boldsymbol{A}=\boldsymbol{a}_r(\theta)\boldsymbol{a}_t(\theta)^\top$, $\dot{\boldsymbol{A}}=\partial\boldsymbol{A}/\partial\theta$, and $\alpha=2|\xi|^2$. The SINRs of the two downlink users are given by

$$\mathrm{SINR}_1=\boldsymbol{v}_1^{\mathrm{H}}\boldsymbol{H}_{11}^{\mathrm{H}}\Big(\sigma_1^2\boldsymbol{I}+\boldsymbol{H}_{12}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{H}_{12}^{\mathrm{H}}\Big)^{-1}\boldsymbol{H}_{11}\boldsymbol{v}_1 \tag{61}$$

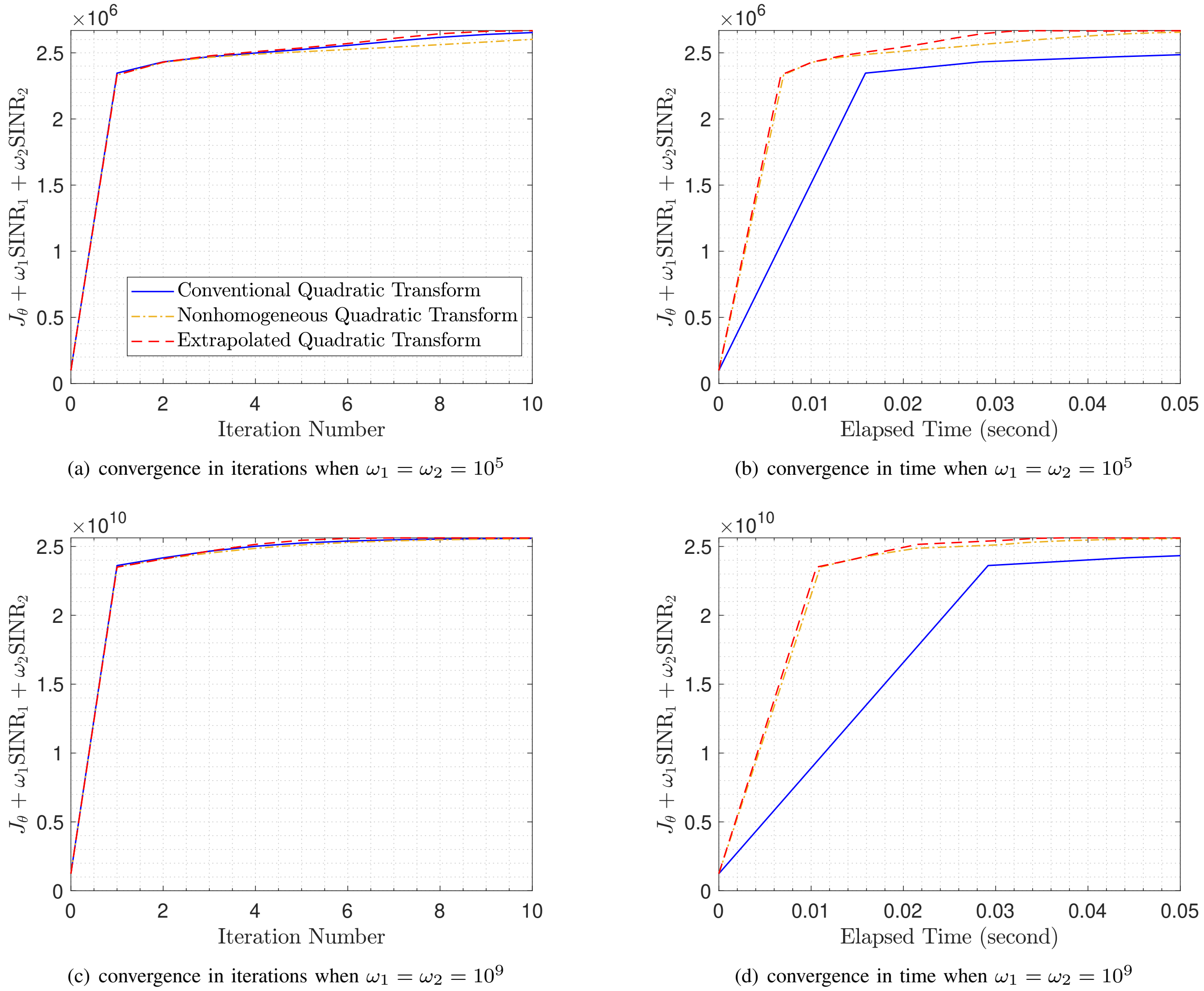
$$\mathrm{SINR}_2=\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{H}_{22}^{\mathrm{H}}\Big(\sigma_2^2\boldsymbol{I}+\boldsymbol{H}_{21}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{H}}\boldsymbol{H}_{21}^{\mathrm{H}}\Big)^{-1}\boldsymbol{H}_{22}\boldsymbol{v}_2. \tag{62}$$

We seek the optimal precoders $\underline{\boldsymbol{v}}=\{\boldsymbol{v}_1,\boldsymbol{v}_2\}$ to maximize a linear combination of the Fisher information and the SINRs:

$$\underset{\underline{\boldsymbol{v}}}{\text{maximize}}\quad J_\theta+\omega_1\mathrm{SINR}_1+\omega_2\mathrm{SINR}_2 \tag{63a}$$

$$\text{subject to}\quad \|\boldsymbol{v}_i\|_2^2\le P_{\max},\quad i\in\{1,2\}, \tag{63b}$$

where $\omega_i>0$ reflects the priority of SINR_i.



(a) convergence in iterations when $\omega_1 = \omega_2 = 10^5$

(b) convergence in time when $\omega_1 = \omega_2 = 10^5$

(c) convergence in iterations when $\omega_1 = \omega_2 = 10^9$

(d) convergence in time when $\omega_1 = \omega_2 = 10^9$

Fig. 5. Maximizing a weighted sum of the Fisher information and the SINRs, $J_\theta + \omega_1\mathrm{SINR}_1 + \omega_2\mathrm{SINR}_2$, for an ISAC system.

By the conventional quadratic transform, the original objective function can be recast to

$$\begin{aligned} f_q(\underline{\boldsymbol{v}}, \underline{\boldsymbol{y}}) &= 2\Re\{\alpha \boldsymbol{v}_1^{\mathrm{H}} \dot{\boldsymbol{A}}^{\mathrm{H}} \boldsymbol{y}_r\} - \alpha \boldsymbol{y}_r^{\mathrm{H}}\Big(\sigma_r^2 \boldsymbol{I} + \boldsymbol{G}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{G}^{\mathrm{H}}\Big)\boldsymbol{y}_r \\ &+ 2\Re\{\omega_1 \boldsymbol{v}_1^{\mathrm{H}} \boldsymbol{H}_{11}^{\mathrm{H}} \boldsymbol{y}_1\} - \omega_1 \boldsymbol{y}_1^{\mathrm{H}}\Big(\sigma_1^2 \boldsymbol{I} + \boldsymbol{H}_{12}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{H}_{12}^{\mathrm{H}}\Big)\boldsymbol{y}_1 \\ &+ 2\Re\{\omega_2 \boldsymbol{v}_2^{\mathrm{H}} \boldsymbol{H}_{22}^{\mathrm{H}} \boldsymbol{y}_2\} - \omega_2 \boldsymbol{y}_2^{\mathrm{H}}\Big(\sigma_2^2 \boldsymbol{I} + \boldsymbol{H}_{21}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{H}}\boldsymbol{H}_{21}^{\mathrm{H}}\Big)\boldsymbol{y}_2, \end{aligned} \tag{64}$$

where the auxiliary variables $\boldsymbol{y}_r \in \mathbb{C}^{N_r}$, $\boldsymbol{y}_1 \in \mathbb{C}^N$, and $\boldsymbol{y}_2 \in \mathbb{C}^N$ are introduced for J_θ, SINR_1, and SINR_2, respectively. We optimize the precoders $\underline{\boldsymbol{v}}$ and the auxiliary variables $\underline{\boldsymbol{y}} = (\boldsymbol{y}_r, \boldsymbol{y}_1, \boldsymbol{y}_2)$ alternatingly as

$$\underline{\boldsymbol{v}}^0 \to \cdots \to \underline{\boldsymbol{v}}^{k-1} \to \underline{\boldsymbol{y}}^k \to \underline{\boldsymbol{v}}^k \to \cdots .$$

For fixed $\underline{\boldsymbol{v}}$, the optimal $\underline{\boldsymbol{y}}$ in $f_q(\underline{\boldsymbol{v}}, \underline{\boldsymbol{y}})$ is given by

$$\boldsymbol{y}_r^\star = \Big(\sigma_r^2 \boldsymbol{I} + \boldsymbol{G}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{G}^{\mathrm{H}}\Big)^{-1} \dot{\boldsymbol{A}}\boldsymbol{v}_1 \tag{65a}$$

$$\boldsymbol{y}_1^\star = \Big(\sigma_1^2 \boldsymbol{I} + \boldsymbol{H}_{12}\boldsymbol{v}_2\boldsymbol{v}_2^{\mathrm{H}}\boldsymbol{H}_{12}^{\mathrm{H}}\Big)^{-1} \boldsymbol{H}_{11}\boldsymbol{v}_1 \tag{65b}$$

$$\boldsymbol{y}_2^\star = \Big(\sigma_2^2 \boldsymbol{I} + \boldsymbol{H}_{21}\boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{H}}\boldsymbol{H}_{21}^{\mathrm{H}}\Big)^{-1} \boldsymbol{H}_{22}\boldsymbol{v}_2. \tag{65c}$$

Next, we further find the optimal $\underline{\boldsymbol{v}}$ in closed form as

$$\boldsymbol{v}_1^\star = \Big(\eta_1 \boldsymbol{I} + \boldsymbol{D}_1\Big)^{-1}\Big(\alpha \dot{\boldsymbol{A}}^H \boldsymbol{y}_r + \omega_1 \boldsymbol{H}_{11}^{\mathrm{H}} \boldsymbol{y}_1\Big) \tag{66a}$$

$$\boldsymbol{v}_2^\star = \Big(\eta_2 \boldsymbol{I} + \boldsymbol{D}_2\Big)^{-1} \omega_2 \boldsymbol{H}_{22}^{\mathrm{H}} \boldsymbol{y}_2, \tag{66b}$$

where

$$\boldsymbol{D}_1 = \omega_2 \boldsymbol{H}_{21}^{\mathrm{H}} \boldsymbol{y}_2 \boldsymbol{y}_2^{\mathrm{H}} \boldsymbol{H}_{21} \tag{67a}$$

$$\boldsymbol{D}_2 = \alpha \boldsymbol{G}^{\mathrm{H}} \boldsymbol{y}_r \boldsymbol{y}_r^{\mathrm{H}} \boldsymbol{G} + \omega_1 \boldsymbol{H}_{12}^{\mathrm{H}} \boldsymbol{y}_1 \boldsymbol{y}_1^{\mathrm{H}} \boldsymbol{H}_{12}, \tag{67b}$$

and the Lagrange multipliers (η_1, η_2) for the power constraint are optimally determined as

$$\eta_i^\star = \min\big\{\eta_i \ge 0 : \|\boldsymbol{v}_i\|_2^2 \le P_{\max}\big\}. \tag{68}$$

In practice, we may first try $\eta_i^\star = 0$; if $\|\boldsymbol{v}_i\|_2^2 > P_{\max}$, then we tune $\eta^\star$ via bisection search to render $\|\boldsymbol{v}_i\|_2^2 = P_{\max}$.

Differing from the above conventional quadratic transform, the nonhomogeneous quadratic transform in Algorithm 3 recasts the original objective $f_o(\underline{\boldsymbol{v}})$ to $f_t(\underline{\boldsymbol{v}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ as shown in (69). We optimize the variables in $f_t(\underline{\boldsymbol{v}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ iteratively:

$$\underline{\boldsymbol{v}}^0 \to \cdots \to \underline{\boldsymbol{v}}^{k-1} \to \underline{\boldsymbol{z}}^k \to \underline{\boldsymbol{y}}^k \to \underline{\boldsymbol{v}}^k \to \underline{\boldsymbol{z}}^{k+1} \to \cdots .$$

$$\begin{aligned} f_t(\underline{\boldsymbol{v}},\underline{\boldsymbol{y}},\underline{\boldsymbol{z}}) = 2\Re\Big\{\boldsymbol{v}_1^{\mathrm{H}}\Big[\alpha\dot{\boldsymbol{A}}^{\mathrm{H}}\boldsymbol{y}_r+\omega_1\boldsymbol{H}_{11}^{\mathrm{H}}\boldsymbol{y}_1+(\lambda_1\boldsymbol{I}-\boldsymbol{D}_1)\boldsymbol{z}_1\Big]+\boldsymbol{v}_2^{\mathrm{H}}\Big[\omega_2\boldsymbol{H}_{22}^{\mathrm{H}}\boldsymbol{y}_2+(\lambda_2\boldsymbol{I}-\boldsymbol{D}_2)\boldsymbol{z}_2\Big]\Big\} \\ +\boldsymbol{z}_1^{\mathrm{H}}(\boldsymbol{D}_1-\lambda_1\boldsymbol{I})\boldsymbol{z}_1+\boldsymbol{z}_2^{\mathrm{H}}(\boldsymbol{D}_2-\lambda_2\boldsymbol{I})\boldsymbol{z}_2-\lambda_1\|\boldsymbol{v}_1\|_2^2-\lambda_2\|\boldsymbol{v}_2\|_2^2-\alpha\sigma_r^2\|\boldsymbol{y}_r\|_2^2-\omega_1\sigma_1^2\|\boldsymbol{y}_1\|_2^2-\omega_2\sigma_2^2\|\boldsymbol{y}_2\|_2^2 \quad (69)\end{aligned}$$

When $\underline{\boldsymbol{v}}$ and $\underline{\boldsymbol{y}}$ are both held fixed, $\underline{\boldsymbol{z}}$ is optimally updated as $\boldsymbol{z}_1^\star=\boldsymbol{v}_1$, and $\boldsymbol{z}_2^\star=\boldsymbol{v}_2$. The optimal update of $\underline{\boldsymbol{y}}$ is the same as in (65a), (65b), and (65c). When $\underline{\boldsymbol{y}}$ and $\underline{\boldsymbol{z}}$ are both held fixed, we first compute

$$\hat{\boldsymbol{v}}_1=\boldsymbol{z}_1+\frac{1}{\lambda_1}\Big(\alpha\dot{\boldsymbol{A}}^{\mathrm{H}}\boldsymbol{y}_r+\omega_1\boldsymbol{H}_{11}^{\mathrm{H}}\boldsymbol{y}_1-\boldsymbol{D}_1\boldsymbol{z}_1\Big) \quad (70)$$

and

$$\hat{\boldsymbol{v}}_2=\boldsymbol{z}_2+\frac{1}{\lambda_2}\Big(\omega_2\boldsymbol{H}_{22}^{\mathrm{H}}\boldsymbol{y}_2-\boldsymbol{D}_2\boldsymbol{z}_2\Big), \quad (71)$$

and then update $\underline{\boldsymbol{v}}$ optimally as

$$\boldsymbol{v}_i^\star=\begin{cases}\hat{\boldsymbol{v}}_i & \text{if } \|\hat{\boldsymbol{v}}_i\|_2^2\le P_{\max}\\ \big(\sqrt{P_{\max}}/\|\hat{\boldsymbol{v}}_i\|_2\big)\hat{\boldsymbol{v}}_i & \text{otherwise.}\end{cases} \quad (72)$$

We remark that other settings of ISAC have been considered in the literature with the Fisher information (or the Cramér-Rao bound) written in a different form from our case, but the ISAC beamforming problem can still be handled by FP, e.g., as discussed in [33]–[35].

We now compare the various quadratic transform methods in simulations. Let $M=64$, $N=2$, $N_r=72$, $\sigma_1^2=\sigma_2^2=-80$ dBm, $\sigma_r^2=-80$ dBm, and $P_{\max}=20$ dBm. The path loss (in dB) is computed as $32.6+36.7\log_{10}d$, where d is the distance in meters; the position coordinates of BS 1, BS 2, user 1, user 2, and the sensed object are $(0,0)$, $(250,0)$, $(-10,100)$, $(350,100)$, and $(200,200)$, respectively, all in meters. The Rayleigh fading model is adopted. Algorithm 1, Algorithm 2, and Algorithm 3 are tested. As shown in Fig. 5(a), if the convergence is considered in terms of iterations, then all these algorithms yield almost the same convergence rate. The objective value is monotonically increasing with the iteration number by all these algorithms. If we instead evaluate convergence in terms of the elapsed time as displayed in Fig. 5(b), then the proposed two accelerated methods, Algorithm 2 and Algorithm 3, become much faster than Algorithm 1; the former two algorithms attain convergence after 0.02 seconds, whereas the latter algorithm still does not converge after 0.05 seconds. Algorithm 3 outperforms Algorithm 2, but their gap is marginal. Moreover, as shown in Fig. 5(c) and Fig. 5(d), the above results still hold when the SINR weights are altered.

B. Massive Multiple-Input Multiple-Output (MIMO)

Consider a downlink network with L cells as depicted in Fig. 6. In each cell, the BS with M antennas sends independent messages to Q downlink user terminals simultaneously by spatial multiplexing; it shall be well understood that $Q\le M$. Assume also that each user terminal has N receive antennas. In particular, $M\gg N$ under the massive MIMO setting.

Moreover, we use $\ell,i=1,\dots,L$ to index the cells and the corresponding BSs, and use $q,j=1,\dots,Q$ to index the users in each cell. Denote by $\boldsymbol{H}_{\ell q,i}\in\mathbb{C}^{N\times M}$ the channel from BS i to the qth user in cell ℓ, denote by $\boldsymbol{v}_{\ell q}\in\mathbb{C}^M$ the transmit precoder of BS ℓ for its qth associated user, and denote by σ^2 the background noise power. The SINR of the qth user in cell ℓ, denoted by $\mathrm{SINR}_{\ell q}$, is computed[3] as

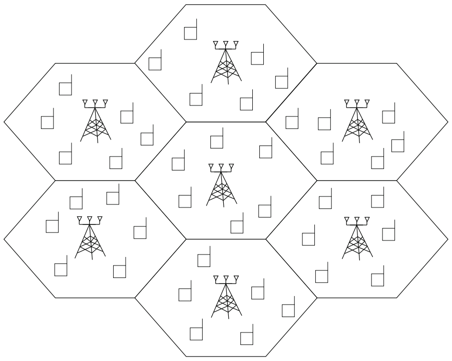


Fig. 6. A 7-cell downlink wrapped-around massive MIMO network. We seek the optimal beamformers to maximize the sum of weighted rates across cells.

$$\mathrm{SINR}_{\ell q}=\boldsymbol{v}_{\ell q}^{\mathrm{H}}\boldsymbol{H}_{\ell q,\ell}^{\mathrm{H}}\Bigg(\sigma^2\boldsymbol{I}+\sum_{(i,j)\neq(\ell,q)}\boldsymbol{H}_{\ell q,i}\boldsymbol{v}_{ij}\boldsymbol{v}_{ij}^{\mathrm{H}}\boldsymbol{H}_{\ell q,i}^{\mathrm{H}}\Bigg)^{-1} \boldsymbol{H}_{\ell q,\ell}\boldsymbol{v}_{\ell q}. \quad (73)$$

Assigning a positive weight $\mu_{\ell q}>0$ for each user q in cell ℓ, we seek the optimal set of precoding vectors $\underline{\boldsymbol{v}}=\{\boldsymbol{v}_{\ell q}\}$ to maximize the weighted sum-of-rates throughout the network:

$$\underset{\underline{\boldsymbol{v}}}{\text{maximize}}\quad \sum_{\ell=1}^{L}\sum_{q=1}^{Q}\mu_{\ell q}\log\big(1+\mathrm{SINR}_{\ell q}\big) \quad (74a)$$

$$\text{subject to}\quad \sum_{q=1}^{Q}\|\boldsymbol{v}_{\ell q}\|_2^2\le P_{\max},\ \text{ for } \ell=1,\dots,L, \quad (74b)$$

where the constraint (74b) states that the total transmit power at each BS cannot exceed the power budget $P_{\max}$.

The traditional WMMSE method [10], [11] addresses the above problem by performing the following iterative updates:

$$\underline{\boldsymbol{v}}^0\to\cdots\to\underline{\boldsymbol{v}}^{k-1}\to\underline{\boldsymbol{y}}^k\to\underline{\gamma}^k\to\underline{\boldsymbol{v}}^k\to\cdots,$$

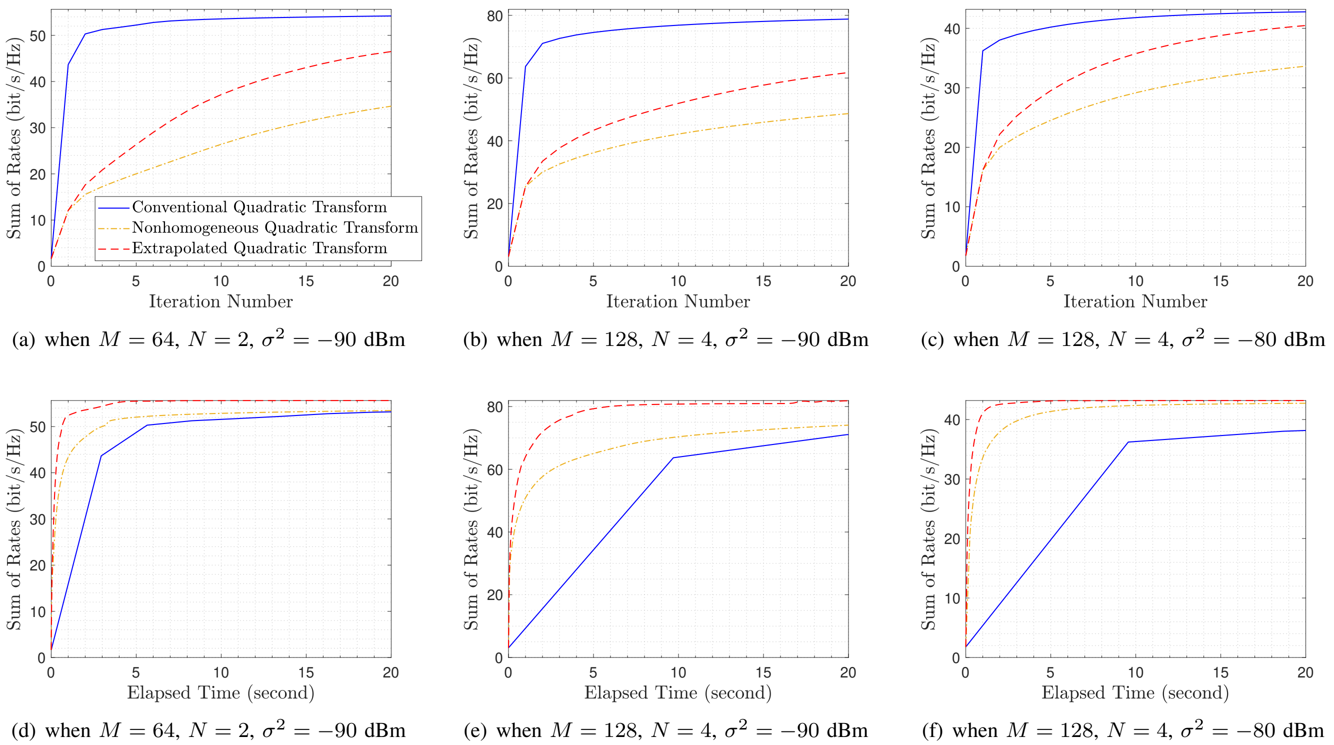
where the auxiliary variable $\underline{\gamma}$ is updated as

$$\gamma_{\ell q}^\star=\mathrm{SINR}_{\ell q} \quad (75)$$

for the current $\underline{\boldsymbol{v}}$, and the auxiliary variable $\underline{\boldsymbol{y}}$ is updated as

$$\boldsymbol{y}_{\ell q}^\star=\Bigg(\sigma^2\boldsymbol{I}+\sum_{i=1}^{L}\sum_{j=1}^{Q}\boldsymbol{H}_{\ell q,i}\boldsymbol{v}_{ij}\boldsymbol{v}_{ij}^{\mathrm{H}}\boldsymbol{H}_{\ell q,i}^{\mathrm{H}}\Bigg)^{-1}\boldsymbol{H}_{\ell q,\ell}\boldsymbol{v}_{\ell q}. \quad (76)$$

[3] The SINR in (73) can be achieved by using the MMSE receiver.



(a) when $M = 64$, $N = 2$, $\sigma^2 = -90$ dBm
(b) when $M = 128$, $N = 4$, $\sigma^2 = -90$ dBm
(c) when $M = 128$, $N = 4$, $\sigma^2 = -80$ dBm
(d) when $M = 64$, $N = 2$, $\sigma^2 = -90$ dBm
(e) when $M = 128$, $N = 4$, $\sigma^2 = -90$ dBm
(f) when $M = 128$, $N = 4$, $\sigma^2 = -80$ dBm

Fig. 7. Maximizing the sum of rates in a multi-cell downlink massive MIMO network. On the top row the three figures show the convergence in iterations, while on the bottom row the three figures show the convergence in time.

With the auxiliary variables held fixed, the precoding vectors are optimally updated as

$$\boldsymbol{v}_{\ell q}(\eta_\ell) = \left(\eta_\ell \boldsymbol{I} + \sum_{i=1}^{L}\sum_{j=1}^{Q} \mu_{ij}(1+\gamma_{ij})\boldsymbol{H}^{\mathrm{H}}_{ij,\ell}\boldsymbol{y}_{ij}\boldsymbol{y}^{\mathrm{H}}_{ij}\boldsymbol{H}_{ij,\ell}\right)^{-1} \mu_{\ell q}(1+\gamma_{\ell q})\boldsymbol{H}^{\mathrm{H}}_{\ell q,\ell}\boldsymbol{y}_{\ell q}, \quad (77)$$

where the Lagrange multiplier η_ℓ accounts for the power constraint at BS ℓ and is optimally determined as

$$\eta^\star_\ell = \min\left\{\eta \geq 0 : \sum_{q=1}^{Q} \|\boldsymbol{v}_{\ell q}(\eta)\|_2^2 \leq P_{\max}\right\}. \quad (78)$$

In practice, the above η_ℓ can be obtained via bisection search. The WMMSE algorithm frequently requires inverting an $M \times M$ matrix, so it can be quite costly when $M \gg N$.

In contrast, the nonhomogeneous quadratic transform reformulates the objective function as

$$f_t(\underline{\boldsymbol{v}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \underline{\gamma}) = \sum_{\ell=1}^{L}\sum_{q=1}^{Q}\Big[2\Re\{\mu_{\ell q}(1+\gamma_{\ell q})\boldsymbol{v}^{\mathrm{H}}_{\ell q}\boldsymbol{H}^{\mathrm{H}}_{\ell q,\ell}\boldsymbol{y}_{\ell q} + \boldsymbol{v}^{\mathrm{H}}_{\ell q}(\lambda_\ell \boldsymbol{I} - \boldsymbol{D}_\ell)\boldsymbol{z}_{\ell q}\} + \boldsymbol{z}^{\mathrm{H}}_{\ell q}(\boldsymbol{D}_\ell - \lambda_\ell \boldsymbol{I})\boldsymbol{z}_{\ell q} - \lambda_\ell \boldsymbol{v}^{\mathrm{H}}_{\ell q}\boldsymbol{v}_{\ell q} - \mu_{\ell q}(1+\gamma_{\ell q})\sigma^2 \boldsymbol{y}^{\mathrm{H}}_{\ell q}\boldsymbol{y}_{\ell q} + \mu_{\ell q}\log(1+\gamma_{\ell q}) - \mu_{\ell q}\gamma_{\ell q}\Big], \quad (79)$$

for which the iterative updates are carried out as

$$\underline{\boldsymbol{v}}^0 \rightarrow \cdots \rightarrow \underline{\boldsymbol{v}}^{k-1} \rightarrow \underline{\boldsymbol{z}}^k \rightarrow \underline{\boldsymbol{y}}^k \rightarrow \underline{\gamma}^k \rightarrow \underline{\boldsymbol{v}}^k \rightarrow \cdots,$$

where the auxiliary variable $\underline{\boldsymbol{z}}$ is updated as $\boldsymbol{z}_{\ell q} = \boldsymbol{v}_{\ell q}$, and the other two auxiliary variables $\underline{\boldsymbol{y}}$ and $\underline{\gamma}$ are updated as in (76) and (75), respectively. To update $\underline{\boldsymbol{v}}$, we first compute

$$\hat{\boldsymbol{v}}_{\ell q} = \boldsymbol{z}_{\ell q} + \frac{1}{\lambda_\ell}\Big(\mu_{\ell q}(1+\gamma_{\ell q})\boldsymbol{H}^{\mathrm{H}}_{\ell q,\ell}\boldsymbol{y}_{\ell q} - \boldsymbol{D}_\ell \boldsymbol{z}_{\ell q}\Big), \quad (80)$$

where

$$\boldsymbol{D}_\ell = \sum_{i=1}^{L}\sum_{j=1}^{Q} \mu_{ij}(1+\gamma_{ij})\boldsymbol{H}^{\mathrm{H}}_{ij,\ell}\boldsymbol{y}_{ij}\boldsymbol{y}^{\mathrm{H}}_{ij}\boldsymbol{H}_{ij,\ell}, \quad (81)$$

and then optimally enforce the power constraint as

$$\boldsymbol{v}^\star_{\ell q} = \begin{cases} \hat{\boldsymbol{v}}_{\ell q} & \text{if } \sum_{j=1}^{Q}\|\hat{\boldsymbol{v}}_{\ell j}\|_2^2 \leq P_{\max} \\ \sqrt{\frac{P_{\max}}{\sum_{j=1}^{Q}\|\hat{\boldsymbol{v}}_{\ell j}\|_2^2}}\,\hat{\boldsymbol{v}}_{\ell q} & \text{otherwise.} \end{cases}$$

As opposed to the updating formula (77) of the WMMSE algorithm, the update of $\underline{\boldsymbol{v}}$ in (80) no longer incurs any matrix inverse. Even though Algorithm 2 still requires computing matrix inverse for updating the auxiliary variable $\underline{\boldsymbol{y}}$ as in (76), the matrix size is just $N \times N$ with $N \ll M$ and thus can be neglected. Moreover, the above beamforming method for massive MIMO can be accelerated via extrapolation as in Algorithm 3.

We now test the various quadratic transform methods for massive MIMO in a simulated 7-hexagonal-cell wrapped-around network as considered in [3]. Within each cell, the BS is located at the center and the 6 downlink users are randomly placed. Each BS has 128 antennas and each user has 4 antennas. The BS-to-BS distance equals 0.8 km. The maximum transmit power level at the BS side is 20 dBm, and the background noise power level is -90 dBm. The downlink distance-dependent path-loss is simulated by $128.1 + 37.6\log_{10} d + \tau$ (in dB), where d represents the BS-to-user distance in km, and τ is a zero-mean Gaussian random variable with 8 dB standard deviation for the shadowing effect. We consider sum rate maximization by setting all the weights to 1. Again, Algorithm 1, Algorithm 2, and Algorithm 3 are the competitors. As shown in Fig. 7(a), Algorithm 1 converges faster than the other two methods in terms of iterations; this result agrees with the former discussion below Proposition 3. When it comes to the convergence evaluated by time, as shown in Fig. 7(d), the two accelerated quadratic transform methods are much more efficient than the conventional method in Algorithm 1. In particular, observe that Algorithm 3 is also much faster than Algorithm 2, as opposed to the ISAC case in Fig. 5. There are two reasons. First, there are more matrix ratio terms in the massive MIMO problem case; second, the FP of massive MIMO has a more complicated structure (with ratios nested in logarithms). Moreover, as shown in Fig. 7(b) and Fig. 7(e), the advantage of Algorithm 3 over the other two methods becomes larger when more antennas are deployed. Nevertheless, when the noise level is raised, as shown in Fig. 7(c) and Fig. 7(d), the advantage of Algorithm 3 shrinks. The reason is that the FP-based coordination between the different beamformers becomes less important in the low-SNR regime; the maximum ratio transmission (MRT), i.e., aligning each beamformer with its corresponding channel, now tends to be a better choice [36].

VIII. Conclusion

This work significantly develops the existing theory and algorithm of the quadratic transform—which is a state-of-the-art tool for FP, with an emphasis on the applications for wireless communications. To start with, we establish a connection between the quadratic transform and the gradient projection, thus generalizing the result in [2] about the WSR problem to a much wider range of FP problems. The above connection leads us to the nonhomogeneous quadratic transform that eliminates the matrix inverse operation from the FP solving. Next, we propose a novel idea of accelerating the quadratic transform by means of Nesterov's extrapolation [1]. We then provide convergence rate analysis for the quadratic transform (including the conventional version and the accelerated ones). Since the quadratic transform encompasses the WMMSE algorithm [10], [11] as a special case, the convergence rate analysis accounts for WMMSE too. To the best of our knowledge, this is the first work that gives a formal analysis as to how fast the quadratic transform or WMMSE converges. Moreover, we demonstrate the practical usefulness of the accelerated quadratic transform through two application cases, ISAC and massive MIMO, both of which are envisioned to be the key components of the next-generation wireless networks.

Appendix A
Proof of Proposition 3

We focus on the convergence rate of Algorithm 1; the convergence rate of Algorithm 2 can be obtainedd similarly. Lemma 1.2.4 in [1] states that for any twice-differentiable function $f(\boldsymbol{x})$ with L-Lipschitz continuous gradient, we have

$$\left|f(\boldsymbol{x}') - f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})^{\mathrm{H}}(\boldsymbol{x}'-\boldsymbol{x}) - \frac{1}{2}(\boldsymbol{x}'-\boldsymbol{x})^{\mathrm{H}}\nabla^2 f(\boldsymbol{x})(\boldsymbol{x}'-\boldsymbol{x})\right| \le \frac{L}{6}\|\boldsymbol{x}'-\boldsymbol{x}\|_2^3$$

given any two feasible $\boldsymbol{x}$ and $\boldsymbol{x}'$. Applying the above lemma to the function $\delta_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1})$ and using the results in (24) give

$$\begin{aligned}
&\frac{L}{6}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^3\\
&\ge \delta_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1})\\
&\quad -\frac{1}{2}(\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1})^{\mathrm{H}}\nabla^2\delta_q(\underline{\boldsymbol{x}}^{k-1}|\underline{\boldsymbol{x}}^{k-1})(\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1})\\
&\ge \delta_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}) - \frac{\Lambda_q}{2}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^2\\
&= f_o(\underline{\boldsymbol{x}}) - f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}}^k) - \frac{\Lambda_q}{2}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^2\\
&\overset{(a)}{\ge} f_o(\underline{\boldsymbol{x}}) - f_q(\underline{\boldsymbol{x}}^k,\underline{\boldsymbol{y}}^k) - \frac{\Lambda_q}{2}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^2\\
&\overset{(b)}{\ge} f_o(\underline{\boldsymbol{x}}) - f_q(\underline{\boldsymbol{x}}^k,\underline{\boldsymbol{y}}^{k+1}) - \frac{\Lambda_q}{2}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^2\\
&\overset{(c)}{=} f_o(\underline{\boldsymbol{x}}) - f_o(\underline{\boldsymbol{x}}^k) - \frac{\Lambda_q}{2}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^2,
\end{aligned}\tag{82}$$

where step (a) follows since $\underline{\boldsymbol{x}}^k$ maximizes $f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}})$ for the current $\underline{\boldsymbol{y}} = \underline{\boldsymbol{y}}^k$, step (b) follows since $\underline{\boldsymbol{y}}^{k+1}$ maximizes $f_q(\underline{\boldsymbol{x}},\underline{\boldsymbol{y}})$ for the current $\underline{\boldsymbol{x}} = \underline{\boldsymbol{x}}^k$, and step (c) follows by the property of the surrogate function. Following Nesterov's proof technique in [1], we let

$$\underline{\boldsymbol{x}} = \pi\underline{\boldsymbol{x}}^* + (1-\pi)\underline{\boldsymbol{x}}^{k-1},\tag{83}$$

where $\pi\in[0,1]$. Then the concavity of $f_o(\underline{\boldsymbol{x}})$ on $\mathcal{X}$ gives

$$f_o(\underline{\boldsymbol{x}}) \le \pi f_o(\underline{\boldsymbol{x}}^*) + (1-\pi)f_o(\underline{\boldsymbol{x}}^{k-1}).\tag{84}$$

Denote the gap in the objective value as

$$v_k = f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^k).\tag{85}$$

Substituting (83) and (84) into (82) gives rise to

$$\begin{aligned}
v_k &\le (1-\pi)v_{k-1} + \frac{\pi^2\Lambda_q}{2}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^2\\
&\quad + \frac{\pi^3 L}{6}\|\underline{\boldsymbol{x}}-\underline{\boldsymbol{x}}^{k-1}\|_2^3\\
&\le (1-\pi)v_{k-1} + \pi^2\left(\frac{\Lambda_q R^2}{2} + \frac{LR^3}{6}\right),
\end{aligned}\tag{86}$$

where the second inequality follows by (23) and $0\le\pi\le1$. The choice of π depends on k.

When $k=1$, we let $\pi=1$ in (86) and obtain

$$v_1 \le \frac{\Lambda_q R^2}{2} + \frac{LR^3}{6}.\tag{87}$$

When $k\ge2$, we let

$$\pi = \frac{v_{k-1}}{\Lambda_q R^2 + LR^3/3}.\tag{88}$$

It can be shown by induction that the above π is always feasible (i.e., $0 \leq \pi \leq 1$) for all $k \geq 2$. Plugging (88) in (86) yields

$$v_k \leq v_{k-1}\left(1 - \frac{v_{k-1}}{2\Lambda_q R^2 + 2LR^3/3}\right), \tag{89}$$

which can be further rewritten as

$$\begin{aligned}\frac{1}{v_k} &\geq \frac{1}{v_{k-1}} \cdot \left(1 - \frac{v_{k-1}}{2\Lambda_q R^2 + 2LR^3/3}\right)^{-1}\\ &\geq \frac{1}{v_{k-1}} \cdot \left(1 + \frac{v_{k-1}}{2\Lambda_q R^2 + 2LR^3/3}\right)\\ &= \frac{1}{v_{k-1}} + \frac{1}{2\Lambda_q R^2 + 2LR^3/3},\end{aligned} \tag{90}$$

where the second inequality follows since $(1-a)^{-1} > 1 + a$ for any $0 \leq a \leq 1$. The result of (90) immediately gives

$$\begin{aligned}\frac{1}{v_k} &\geq \frac{1}{v_1} + \frac{k-1}{2\Lambda_q R^2 + 2LR^3/3}\\ &\geq \frac{k+3}{2\Lambda_q R^2 + 2LR^3/3},\end{aligned} \tag{91}$$

where the second inequality is due to (87). The proof is then completed for Algorithm 1. The case of Algorithm 2 can be verified similarly.

Appendix B
Proof of Proposition 6

Because the optimal value of $\boldsymbol{g}$ depends on the current $\underline{\boldsymbol{x}}$, it can be written as a function of $\underline{\boldsymbol{x}}$, i.e.,

$$\boldsymbol{g}^\star = \arg\max_{\boldsymbol{g}} h(\underline{\boldsymbol{x}}, \boldsymbol{g}) \triangleq \mathcal{T}(\underline{\boldsymbol{x}}). \tag{92}$$

Computing $\boldsymbol{g}$ in (92) is an unconstrained differentiable problem, so the optimal $\boldsymbol{g}^\star$ must satisfy the first-order condition

$$\frac{d\beta(\boldsymbol{g}^\star)}{d\boldsymbol{g}} + \sum_{j=1}^{n}\left[\frac{d\alpha_j(\boldsymbol{g}^\star)}{d\boldsymbol{g}} \cdot \hat{M}_j(\underline{\boldsymbol{x}})\right] = 0. \tag{93}$$

As a result, the partial derivative of $\mathcal{G}\big(M_1(\underline{\boldsymbol{x}}), \ldots, M_n(\underline{\boldsymbol{x}})\big)$ with respect to each $\boldsymbol{x}_i^c$ can be simplified into

$$\begin{aligned}&\frac{\partial \mathcal{G}\big(M_1(\underline{\boldsymbol{x}}), \ldots, M_n(\underline{\boldsymbol{x}})\big)}{\partial \boldsymbol{x}_i^c}\\ &= \frac{\partial h(\underline{\boldsymbol{x}}, \mathcal{T}(\underline{\boldsymbol{x}}))}{\partial \boldsymbol{x}_i^c}\\ &= \sum_{j=1}^{n}\left[\left(\frac{\partial \mathcal{T}(\underline{\boldsymbol{x}})}{\partial \boldsymbol{x}_i^c}\right)^\top \frac{d\alpha_j(\boldsymbol{g}^\star)}{d\boldsymbol{g}} \cdot \hat{M}_j(\underline{\boldsymbol{x}}) + \alpha_j(\boldsymbol{g}^\star) \cdot \frac{\partial \hat{M}_j(\underline{\boldsymbol{x}})}{\partial \boldsymbol{x}_i^c}\right]\\ &\qquad + \left(\frac{\partial \mathcal{T}(\underline{\boldsymbol{x}})}{\partial \boldsymbol{x}_i^c}\right)^\top \cdot \frac{d\beta(\boldsymbol{g}^\star)}{d\boldsymbol{g}}\\ &= \sum_{j=1}^{n}\left[\alpha_j(\boldsymbol{g}^\star)\frac{\partial \hat{M}_j(\underline{\boldsymbol{x}})}{\partial \boldsymbol{x}_i^c}\right].\end{aligned} \tag{94}$$

We now optimize the variables of $f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}, \boldsymbol{g})$ in (46) iteratively as

$$\underline{\boldsymbol{x}}^0 \to \cdots \to \underline{\boldsymbol{x}}^{k-1} \to \underline{\boldsymbol{z}}^k \to \underline{\boldsymbol{y}}^k \to \boldsymbol{g}^k \to \underline{\boldsymbol{x}}^k \to \cdots.$$

The optimal update of $\boldsymbol{x}_i^k$ is

$$\begin{aligned}\boldsymbol{x}_i^k &\overset{(a)}{=} \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i^k}\Big(\alpha_i(\boldsymbol{g}^k)\hat{\boldsymbol{A}}_i^{\mathrm{H}}\boldsymbol{y}_i^k - \boldsymbol{D}_i^k\boldsymbol{x}_i^{k-1}\Big)\right)\\ &\overset{(b)}{=} \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{x}_i^{k-1} + \frac{1}{\lambda_i^k}\sum_{j=1}^{n}\left[\alpha_j(\boldsymbol{g}^k) \cdot \frac{\partial \hat{M}_j(\underline{\boldsymbol{x}}^{k-1})}{\partial \boldsymbol{x}_i^c}\right]\right),\end{aligned} \tag{95}$$

where step (a) follows since each $\boldsymbol{z}_i^k$ has been updated to $\boldsymbol{x}_i^{k-1}$, and step (b) follows by Lemma 1. Substituting (94) in the above equation completes the proof.

References

[1] Y. Nesterov, *Lectures on Convex Optimization (Second Edition)*. Springer, 2018.

[2] Z. Zhang, Z. Zhao, K. Shen, D. P. Palomar, and W. Yu, "Discerning and enhancing the weighted sum-rate maximization algorithms in communications," Nov. 2023, [Online]. Available: https://arxiv.org/pdf/2311.04546.

[3] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, Mar. 2018.

[4] K. Shen, W. Yu, L. Zhao, and D. P. Palomar, "Optimization of MIMO device-to-device networks via matrix fractional programming: A minorization–maximization approach," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 2164–2177, Oct. 2019.

[5] A. Charnes and W. W. Cooper, "Programming with linear fractional functionals," *Nav. Res. Logist.*, vol. 9, no. 3, pp. 181–186, Dec. 1962.

[6] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[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] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Aug. 2016.

[9] K. Shen and W. Yu, "Fractional programming for communication systems—Part II: Uplink scheduling via matching," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2631–2644, Mar. 2018.

[10] S. S. Christensen, R. Agarwal, E. D. Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, Dec. 2008.

[11] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Apr. 2011.

[12] I. M. Stancu-Minasian, *Fractional programming: Theory, methods and applications*. Norwell, MA, USA: Kluwer, 2012.

[13] Y.-F. Liu, T.-H. Chang, M. Hong, Z. Wu, A. M.-C. So, E. A. Jorswieck, and W. Yu, "A survey of recent advances in optimization methods for wireless communications," Apr. 2024, [Online]. Available: https://arxiv.org/abs/2401.12025.

[14] J. P. Crouzeix, J. A. Ferland, and S. Schaible, "An algorithm for generalized fractional programs," *J. Optim. Theory Appl.*, vol. 47, no. 1, pp. 35–49, Sep. 1985.

[15] R. W. Freund and F. Jarre, "Solving the sum-of-ratios problem by an interior-point method," *J. Global Optim.*, vol. 19, no. 1, pp. 83–102, Jan. 2001.

[16] N. T. H. Phuong and H. Tuy, "A unified monotonic approach to generalized linear fractional programming," *J. Global Optim.*, vol. 26, pp. 229–259, July 2003.

[17] H. Konno and K. Fukaishi, "A branch and bound algorithm for solving low rank linear multiplicative and fractional programming problems," *J. Global Optim.*, vol. 18, pp. 283–299, Nov. 2000.

[18] H. P. Benson, "Global optimization of nonlinear sums of ratios," *J. Math. Anal. Appl.*, vol. 263, no. 1, pp. 301–315, Nov. 2001.

[19] S. Qu, K. Zhang, and J. Zhao, "An efficient algorithm for globally minimizing sum of quadratic ratios problem with nonconvex quadratic constraints," *Appl. Math. Comput.*, vol. 189, no. 2, pp. 1624–1636, June 2007.

[20] T. Kuno, "A branch-and-bound algorithm for maximizing the sum of several linear ratios," *J. Global Optim.*, vol. 22, pp. 155–174, Jan. 2002.

[21] X. Liu, Y. Gao, B. Zhang, and F. Tian, "A new global optimization algorithm for a class of linear fractional programming," *Mathematics*, vol. 7, no. 9, p. 867, Sep. 2019.

[22] H. P. Benson, "Solving sum of ratios fractional programs via concave minimization," *J. Optim. Theory Appl.*, vol. 135, no. 1, pp. 1–17, June 2007.

[23] ——, "Global optimization algorithm for the nonlinear sum of ratios problem," *J. Optim. Theory Appl.*, vol. 112, pp. 1–29, Jan. 2002.

[24] ——, "Using concave envelopes to globally solve the nonlinear sum of ratios problem," *J. Global Optim.*, vol. 22, pp. 343–364, Jan. 2002.

[25] A. Zappone and E. Jorswieck, "Energy efficiency in wireless networks via fractional programming theory," *Found. Trends Commun. Inf. Theory*, vol. 11, no. 3-4, pp. 185–396, 2015.

[26] Y. Chen, L. Zhao, and K. Shen, "Mixed max-and-min fractional programming for wireless networks," *IEEE Trans. Signal Process.*, vol. 72, pp. 337–351, Dec. 2023.

[27] X. Zhao, S. Lu, Q. Shi, and Z.-Q. Luo, "Rethinking WMMSE: Can its complexity scale linearly with the number of bs antennas?" *IEEE Trans. Signal Process.*, vol. 71, pp. 433–446, Feb. 2023.

[28] Z. Zhang, Z. Zhao, and K. Shen, "Enhancing the efficiency of WMMSE and FP for beamforming by minorization-maximization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, June 2023.

[29] K. Zhou, Z. Chen, G. Liu, and Z. Chen, "A novel extrapolation technique to accelerate WMMSE," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, June 2023.

[30] A. Hjørungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2740–2746, May 2007.

[31] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[32] D. P. Bertsekas, *Convex Optimization Algorithms*. Athena Scientific, 2015.

[33] W. Fan, J. Liang, and J. Li, "Constant modulus mimo radar waveform design with minimum peak sidelobe transmit beampattern," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4207–4222, Aug. 2018.

[34] B. Guo, J. Liang, G. Wang, B. Tang, and H. So, "Bistatic MIMO DFRC system waveform design via fractional programming," *IEEE Trans. Signal Process.*, vol. 71, pp. 1952–1967, May 2023.

[35] B. Guo, J. Liang, B. Tang, L. Li, and H. C. So, "Bistatic MIMO DFRC system waveform design via symbol distance/direction discrimination," *IEEE Trans. Signal Process.*, vol. 71, pp. 3996–4010, Oct. 2023.

[36] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure," *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 142–148, June 2014.