Accelerating Quadratic Transform and WMMSE

Kaiming Shen[†], Ziping Zhao[§], Yannan Chen[†], Zepeng Zhang[§], and Hei Victor Cheng[‡]
[†]School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), China
[§]School of Information Science and Technology, ShanghaiTech University, China
[‡]Department of Electrical and Computer Engineering, Aarhus Universrity, Denmark
E-mail: shenkaiming@cuhk.edu.cn; zipingzhao@shanghaitech.edu.cn; hvc@ece.au.dk

***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 [2]. Specifically, by generalizing the minorization-maximization (MM) approach in [3], 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).**

I. Overview

Fractional programming (FP) aims at the optimization of ratio terms. This paper focuses on the following type of ratio:

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

with the variable $\underline{\boldsymbol{x}} = \{\boldsymbol{x}_j \in \mathbb{C}^d\}$ and the matrix coefficients $\{\boldsymbol{A}_i \in \mathbb{C}^{\ell\times d}, \boldsymbol{B}_{ij} \in \mathbb{C}^{\ell\times d}\}$, for $i,j=1,\ldots,n$. The above ratio term is of significant research interest not only because it is a natural extension of the Rayleigh quotient, but also because several key metrics in information science can be written in this form, e.g., the Cramér-Rao bound, Fisher information, and signal-to-interference-plus-noise ratio (SINR).

The quadratic transform [4], [5] is a state-of-the-art tool for FP. Its main idea is to decouple each ratio term and thereby reformulate the FP problem as a quadratic program that can be addressed efficiently (and often in closed form) in an iterative manner. As shown in [5], the quadratic transform has a connecting link to the minorization-maximization (MM) theory [6], [7], so it immediately follows that the quadratic transform method guarantees monotonic convergence to some stationary point provided that the original problem is differentiable. In particular, [8] shows that the well-known weighted minimum mean square error (WMMSE) algorithm [9], [10] boils down to a special case of the quadratic transform method; [8] further proposes a better way of applying the quadratic transform than WMMSE when dealing with discrete variables.

Despite the extensive studies on the quadratic transform, its convergence rate (even for the WMMSE case) remains a complete mystery, with the following open problems:

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

Our answers are: when the starting point is sufficiently close to a strict local optimum, the quadratic transform yields an objective-value error bound of $O(1/k)$, where k is the iteration index; it is faster than the gradient method in iterations, but slower in time; the error bound can be further reduced to $O(1/k^2)$ by incorporating Nesterov's extrapolation [2].

As a special case of the quadratic transform, the WMMSE algorithm [9], [10] has been extensively considered in the literature for its own sake because of the weighted sum-of-rates (WSR) maximization problem in wireless networks. The computational complexity is a major bottleneck of the WMMSE algorithm because it requires computing matrix inverse frequently. Assuming that the channel matrices are all full row-rank, the recent work [11] takes advantage of the WSR problem structure to facilitate the matrix inverse computation. The more recent work [3] goes further: it does not require any channel assumptions and yet can get rid of the matrix inverse operation completely. The main contribution of the present work is to extend the results in [3], [12] to a broad range of FP problems (not limited to the WSR problem). Moreover, another recent work [13] suggests combining Nesterov's extrapolation and WMMSE in a heuristic way, but its proposed algorithm still involves the matrix inverse operation and cannot provide any performance guarantee.

II. Preliminary

Consider a total of n ratio terms, each written as in (1). The weighted sum ratios problem is

$$\underset{\underline{\boldsymbol{x}}}{\text{maximize}} \quad f_o(\underline{\boldsymbol{x}}) := \sum_{i=1}^{n}\omega_i M_i \qquad (2a)$$

$$\text{subject to} \quad \boldsymbol{x}_i \in \mathcal{X}_i, \quad i=1,\ldots,n, \qquad (2b)$$

The work of Kaiming Shen was supported by NSFC under Grant 92167202. The work of Hei Victor Cheng was supported by the Aarhus Universitets Forskningsfond under Project AUFF 39001. The complete version [1] is available at https://kaimingshen.github.io/doc/accelerated_FP.pdf.

Algorithm 1 Conventional Quadratic Transform [4]

initialize $\underline{\boldsymbol{x}}$ to a feasible value
repeat
  update each $\boldsymbol{y}_i$ according to (3)
  update each $\boldsymbol{x}_i$ according to (4)
until the value of $f_o(\underline{\boldsymbol{x}})$ converges

Algorithm 2 Nonhomogeneous Quadratic Transform

initialize $\underline{\boldsymbol{x}}$ to a feasible value
repeat
  update each $\boldsymbol{z}_i$ according to (9)
  update each $\boldsymbol{y}_i$ according to (3)
  update each $\boldsymbol{x}_i$ according to (10)
until the value of $f_o(\underline{\boldsymbol{x}})$ converges

where each weight $\omega_i > 0$ and $\mathcal{X}_i$ is a nonempty convex set.

By the quadratic transform [4], the original objective $f_o(\underline{\boldsymbol{x}})$ can be recast to

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

where $\Re\{\cdot\}$ indicates the real part of a complex number. The benefit of adopting this new objective is that the primal variable $\underline{\boldsymbol{x}}$ and the auxiliary variable $\underline{\boldsymbol{y}}$ can be efficiently optimized in an alternating fashion. When $\underline{\boldsymbol{x}}$ is held fixed, each $\boldsymbol{y}_i$ is optimally updated as

$$\boldsymbol{y}_i^\star = \left(\sum_{j=1}^{n} \boldsymbol{B}_{ij} \boldsymbol{x}_j \boldsymbol{x}_j^{\mathrm{H}} \boldsymbol{B}_{ij}^{\mathrm{H}} \right)^{-1} (\boldsymbol{A}_i \boldsymbol{x}_i). \tag{3}$$

When $\underline{\boldsymbol{y}}$ is held fixed, the optimal $\boldsymbol{x}_i$ is given by

$$\boldsymbol{x}_i^\star = \arg\min_{\boldsymbol{x}_i \in \mathcal{X}} \left\| \boldsymbol{D}_i^{\frac{1}{2}} \left(\boldsymbol{x}_i - \omega_i \boldsymbol{D}_i^{-1} \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i \right) \right\|_2, \tag{4}$$

where

$$\boldsymbol{D}_i = \sum_{j=1}^{n} \omega_j \boldsymbol{B}_{ji}^{\mathrm{H}} \boldsymbol{y}_j \boldsymbol{y}_j^{\mathrm{H}} \boldsymbol{B}_{ji}. \tag{5}$$

Algorithm 1 summarizes the above steps. As shown in [5], Algorithm 1 guarantees a monotonically increasing convergence to a stationary point of problem (2).

III. Main Results

A. Matrix Inverse Elimination

As a long-standing issue with the above alternating method (and also with the WMMSE algorithm [9], [10]), the computation in (4) can be quite costly when $\boldsymbol{D}_i$ is a large matrix, e.g., when applied to the massive MIMO network. Our first result is to get rid of the matrix inverse for the quadratic transform-based iterative optimization.

Lemma 1 (Nonhomogeneous Bound [7]): Suppose that the two Hermitian matrices $\boldsymbol{L}, \boldsymbol{K} \in \mathbb{C}^{d \times d}$ satisfy the condition $\boldsymbol{L} \preceq \boldsymbol{K}$. Then for any two vectors $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{C}^d$, one has

$$\boldsymbol{x}^{\mathrm{H}} \boldsymbol{L} \boldsymbol{x} \le \boldsymbol{x}^{\mathrm{H}} \boldsymbol{K} \boldsymbol{x} + 2\Re\{\boldsymbol{x}^{\mathrm{H}} (\boldsymbol{L} - \boldsymbol{K}) \boldsymbol{z}\} + \boldsymbol{z}^{\mathrm{H}} (\boldsymbol{K} - \boldsymbol{L}) \boldsymbol{z}, \tag{6}$$

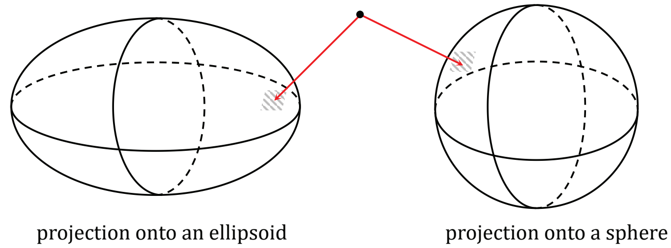

Fig. 1. The conventional quadratic transform amounts to the projection onto an ellipsoid and incurs matrix inverse operation. In contrast, the new quadratic transform avoids matrix inverse by computing the projection onto a sphere.

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

Treating $\boldsymbol{D}_i$ as $\boldsymbol{L}$ in (6), we let

$$\boldsymbol{K} = \lambda_i \boldsymbol{I} \quad \text{where} \quad \lambda_i \ge \lambda_{\max}(\boldsymbol{D}_i), \tag{7}$$

where $\lambda_{\max}(\boldsymbol{D}_i)$ is the largest eigenvalue of $\boldsymbol{D}_i$, in order to make $\boldsymbol{L} \preceq \boldsymbol{K}$. One possible choice is $\lambda_i = \|\boldsymbol{D}_i\|_F$. Thus, by virtue of Lemma 1, we can further recast $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$ to

$$f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}) = \sum_{i=1}^{n} \Big[2\Re\{ \omega_i \boldsymbol{x}_i^{\mathrm{H}} \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i + \boldsymbol{x}_i^{\mathrm{H}} (\lambda_i \boldsymbol{I} - \boldsymbol{D}_i) \boldsymbol{z}_i \} + \boldsymbol{z}_i^{\mathrm{H}} (\boldsymbol{D}_i - \lambda_i \boldsymbol{I}) \boldsymbol{z}_i - \lambda_i \boldsymbol{x}_i^{\mathrm{H}} \boldsymbol{x}_i \Big]. \tag{8}$$

In particular, $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = f_t(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ if $\boldsymbol{z}_i = \boldsymbol{x}_i$ for all i.

We now optimize $\underline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$, and $\underline{\boldsymbol{z}}$ iteratively in $f_q(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}})$. 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 1 as

$$\boldsymbol{z}_i^\star = \boldsymbol{x}_i, \quad \text{for } i = 1, 2, \ldots, n. \tag{9}$$

When $\underline{\boldsymbol{z}}$ and $\underline{\boldsymbol{x}}$ are both fixed, each $\boldsymbol{y}_i$ is still optimally determined as in (3). Next, when $\underline{\boldsymbol{y}}$ and $\underline{\boldsymbol{z}}$ are both held fixed, the optimal $\boldsymbol{x}_i$ in (8) is given by

$$\boldsymbol{x}_i^\star = \arg\min_{\boldsymbol{x}_i \in \mathcal{X}_i} \left\| \lambda_i \boldsymbol{x}_i - \omega_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{y}_i - (\lambda_i \boldsymbol{I} - \boldsymbol{D}_i) \boldsymbol{z}_i \right\|_2. \tag{10}$$

Most importantly, it no longer requires computing the inverse of the potentially large matrix $\boldsymbol{D}_i$. Algorithm 2 summarizes the above new iteration steps.

It is worth comparing (4) and (10) graphically. As shown in Fig. 1, the update of $\boldsymbol{x}_i$ in (4) by the conventional quadratic transform can be interpreted as the projection onto an ellipsoid, while the update of $\boldsymbol{x}_i$ in (10) by the nonhomogeneous quadratic transform can be interpreted as the projection onto a sphere. The ellipsoid projection is in general much more costly than the sphere projection in a high-dimensional space.

B. Accelerated Quadratic Transform

We now interpret the above proposed iterative optimization as gradient projection. 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

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

Algorithm 3 Extrapolated Quadratic Transform

1: initialize $\underline{\boldsymbol{x}}$ to a feasible value
2: **repeat**
3: update each $\boldsymbol{\nu}_i$ according to (12) and let $\boldsymbol{x}_i = \boldsymbol{\nu}_i$
4: update each $\boldsymbol{z}_i$ according to (9)
5: update each $\boldsymbol{y}_i$ according to (3)
6: update each $\boldsymbol{x}_i$ according to (13)
7: **until** the value of $f_o(\underline{\boldsymbol{x}})$ converges

We rewrite the optimal update of $\boldsymbol{x}_i$ in (10) in a sphere-projection form:

$$\boldsymbol{x}_i^k = \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{z}_i^k + \frac{1}{\lambda_i}\Big(\omega_i \boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{y}_i^k - \boldsymbol{D}_i\boldsymbol{z}_i^k\Big)\right). \tag{11}$$

Recall that $\underline{\boldsymbol{y}}^k$ depends on $\underline{\boldsymbol{x}}^{k-1}$ according to (3), while $\underline{\boldsymbol{z}}^k$ depends on $\underline{\boldsymbol{x}}^{k-1}$ according to (9). Expressing $\underline{\boldsymbol{y}}^k$ and $\underline{\boldsymbol{z}}^k$ in terms of $\underline{\boldsymbol{x}}^{k-1}$ rewrites the sphere projection of $\underline{\boldsymbol{x}}_i^k$ as

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

which can be recognized as a gradient projection update.

The fact that the new quadratic transform method is a type of gradient projection motivates us to use Nesterov's extrapolation scheme [2] to accelerate the convergence. Specifically, following the heavy-ball intuition, we extrapolate each $\boldsymbol{x}_i$ along the direction of the difference between the preceding two iterates before the gradient projection, i.e.,

$$\boldsymbol{\nu}_i^{k-1} = \boldsymbol{x}_i^{k-1} + \eta_{k-1}(\boldsymbol{x}_i^{k-1} - \boldsymbol{x}_i^{k-2}), \tag{12}$$

$$\boldsymbol{x}_i^k = \mathcal{P}_{\mathcal{X}_i}\left(\boldsymbol{\nu}_i^{k-1} + \frac{1}{\lambda_i^k}\cdot\frac{\partial f_o(\underline{\boldsymbol{\nu}}^{k-1})}{\partial \boldsymbol{x}_i^c}\right), \tag{13}$$

where the extrapolation step η_k is chosen as

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

and the starting point is $\boldsymbol{x}^{-1} = \boldsymbol{x}^0$ as in [2]. The implementation details are summarized in Algorithm 3, referred to as the extrapolated quadratic transform.

C. Convergence Analysis

In this subsection, we first show that the various quadratic transform methods all guarantee convergence to a stationary point of the FP problem in (2), and then analyze their rates of convergence. All the proofs are omitted here and can be found in the complete version [1] of this work.

The proof of the stationary-point convergence is based on the MM theory [6], [7]. Write the optimal update of $\underline{\boldsymbol{y}}$ in (3) as a function of $\underline{\boldsymbol{x}}$:

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

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 $\underline{\boldsymbol{x}}^{k-1}$:

$$r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}) = f_q(\underline{\boldsymbol{x}}, \mathcal{Y}(\underline{\boldsymbol{x}}^{k-1})), \tag{14}$$

and accordingly the update of $\underline{\boldsymbol{x}}$ in (4) can be rewritten as

$$\underline{\boldsymbol{x}}^k = \arg\max_{\underline{\boldsymbol{x}}\in\mathcal{X}} r_q(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}). \tag{15}$$

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, (15) 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 (2). By a similar argument, we can also interpret Algorithm 2 as an MM method, with the surrogate function

$$r_t(\underline{\boldsymbol{x}}|\underline{\boldsymbol{x}}^{k-1}) = f_t(\underline{\boldsymbol{x}}, \mathcal{Y}(\underline{\boldsymbol{x}}^{k-1}), \underline{\boldsymbol{x}}^{k-1}). \tag{16}$$

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.

Proposition 1: Algorithms 1 and 2 are both the MM methods. Algorithms 1, 2, and 3 all guarantee convergence to some stationary point of the FP problem 2.

We now examine the rates 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.,

$$\mathcal{X} = \big\{\underline{\boldsymbol{x}} : \|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^*\|_2 \le R\big\}, \tag{17}$$

where $\underline{\boldsymbol{x}}^*$ is a strict local optimum of (2) so that $\nabla^2 f_o(\underline{\boldsymbol{x}}^*) \preceq -\xi\boldsymbol{I} \prec \boldsymbol{0}$ holds 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 also 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 [2], 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 let $R \le \xi/L$ to render $f_o(\underline{\boldsymbol{x}})$ concave on $\mathcal{X}$.

Conditioned on $\underline{\boldsymbol{x}}' \in \mathcal{X}$, define the gaps between $f_o(\underline{\boldsymbol{x}})$ and the two surrogate functions to be two functions of $\underline{\boldsymbol{x}} \in \mathcal{X}$ as

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

Moreover, define the two quantities:

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

It can be shown that $\Lambda_q \le \Lambda_t < \infty$. We are now ready to show the (local) convergence rates of Algorithm 1 and Algorithm 2.

Proposition 2 (Convergence Rates of Algorithm 1 and Algorithm 2): For the FP problem (2), the local convergence rate of Algorithm 1 or Algorithm 2 is

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^1) \le \frac{\Lambda R^2}{2} + \frac{LR^3}{6}, \tag{18}$$

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^k) \le \frac{2\Lambda R^2 + 2LR^3/3}{k+3}, \text{ for } k \ge 2, \tag{19}$$

where

$$\Lambda = \begin{cases} \Lambda_q & \text{for Algorithm 1;} \\ \Lambda_t & \text{for Algorithm 2.} \end{cases} \tag{20}$$

Because $0 \le \Lambda_q \le \Lambda_t$, Algorithm 1 converges faster than Algorithm 2 in iterations according to Proposition 2. Notice that Λ_q and Λ_t 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 2 becomes

$$f_o(\underline{\boldsymbol{x}}) - f_o(\underline{\boldsymbol{x}}^k) \le \frac{L}{6}\|\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^{k-1}\|_2^3, \tag{21}$$

which also holds for the *cubically regularized Newton's method* due to Nesterov as shown in [2]. Equipped with the error bound (21), it immediately follows from Theorem 4.1.4 in [2] that

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^1) \le \frac{LR^3}{6}, \tag{22}$$

$$f_o(\underline{\boldsymbol{x}}^*) - f_o(\underline{\boldsymbol{x}}^k) \le \frac{LR^3}{2(1+k/3)^2}, \text{ for } k \ge 2. \tag{23}$$

We now show that the extrapolated quadratic transform method in Algorithm 3 can achieve fairly close to the ideal case stated in (22) and (23). 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 [14].

Proposition 3 (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

$$f(\underline{\boldsymbol{x}}^*) - f(\underline{\boldsymbol{x}}) \le \frac{2C\cdot[f(\underline{\boldsymbol{x}}^*) - f(\underline{\boldsymbol{x}}^0)]}{(k+1)^2}, \text{ for } k \ge 1. \tag{24}$$

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 $O(1/k^2)$.

IV. Application Case: Massive MIMO Beamforming

Consider a downlink multi-cell network with L cells. In each cell, one BS with M antennas sends independent messages towards 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 ℓ or $i = 1, \ldots, L$ to index the cells and the corresponding BSs and use $q, j = 1, \ldots, Q$ 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 as

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

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:

$$\underset{\underline{\boldsymbol{v}}}{\text{maximize}} \quad \sum_{\ell=1}^{L}\sum_{q=1}^{Q}\mu_{\ell q}\log\big(1+\mathrm{SINR}_{\ell q}\big) \tag{25a}$$

$$\text{subject to} \quad \sum_{q=1}^{Q}\|\boldsymbol{v}_{\ell q}\|_2^2 \le P, \quad \ell = 1,\ldots,L, \tag{25b}$$

where the constraint (25b) states that the total transmit power at each BS cannot exceed the power budget P.

At first glance, the FP methods do not apply to the above problem because all the ratios (i.e., the SINRs) are now nested in logarithms. Nevertheless, we can move the ratios to the outside of logarithms by the Lagrangian dual transform [8], so that problem (25) is converted to

$$\underset{\underline{\boldsymbol{v}},\,\underline{\gamma}}{\text{maximize}} \quad h(\underline{\boldsymbol{v}},\underline{\gamma}) \tag{26a}$$

$$\text{subject to} \quad \sum_{q=1}^{Q}\|\boldsymbol{v}_{\ell q}\|_2^2 \le P, \quad \ell = 1,\ldots,L, \tag{26b}$$

where $\underline{\gamma} = \{\gamma_{\ell q}\}$ is a set of auxiliary variables and the new objective function is given by

$$h(\underline{\boldsymbol{v}},\underline{\gamma}) = \sum_{i=1}^{n}\mu_{\ell q}\Big[\log(1+\gamma_{\ell q}) - \gamma_{\ell q} + (1+\gamma_{\ell q})M_{\ell q}\Big]$$

with the shorthand

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

Notice the distinction between $\mathrm{SINR}_{\ell q}$ and $M_{\ell q}$.

We propose optimizing $\underline{\boldsymbol{v}}$ and $\underline{\gamma}$ alternatingly in the new problem (26). When $\underline{\boldsymbol{v}}$ is fixed, the optimal $\gamma_{\ell q}$ equals $\mathrm{SINR}_{\ell q}$ exactly. When $\underline{\gamma}$ is fixed, we only need to consider the first term inside the square parentheses of $h(\underline{\boldsymbol{v}},\underline{\gamma})$; this subproblem can be recognized as a weighted sum ratios problem, so all the FP methods are applicable immediately.

We only show the use of Algorithm 2 in what follows; Algorithm 3 can be used similarly. The nonhomogeneous quadratic transform recasts $h(\underline{\boldsymbol{v}}, \underline{\gamma})$ to

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

for which the iterative updates are carried out as

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

Recall that the optimal update of $\underline{\gamma}$ is to let each $\gamma_{\ell q}^\star = \mathrm{SINR}_{\ell q}$. The optimal update of $\underline{\boldsymbol{y}}$ is

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

and the optimal update of $\underline{\boldsymbol{z}}$ is to let $\underline{\boldsymbol{z}}^\star = \underline{\boldsymbol{x}}$. To update $\underline{\boldsymbol{v}}$, we first compute

$$\boldsymbol{v}_{\ell q} = \boldsymbol{z}_{\ell q} + \frac{1}{\lambda_\ell}\Big(\mu_{\ell q}(1+\lambda_{\ell q})\boldsymbol{H}_{\ell q,\ell}^{\mathrm{H}}\boldsymbol{y}_{\ell q} - \boldsymbol{D}_\ell \boldsymbol{z}_{\ell q}\Big), \tag{27}$$

where

$$\boldsymbol{D}_\ell = \sum_{i=1}^{L}\sum_{j=1}^{Q}\mu_{ij}(1+\gamma_{ij})\boldsymbol{H}_{ij,\ell}^{\mathrm{H}}\boldsymbol{y}_{ij}\boldsymbol{y}_{ij}^{\mathrm{H}}\boldsymbol{H}_{ij,\ell}, \tag{28}$$

and then optimally enforce the power constraint as

$$\boldsymbol{v}_{\ell q}^\star = \begin{cases} \hat{\boldsymbol{v}}_{\ell q} & \text{if } \sum_{j=1}^{Q}\|\hat{\boldsymbol{v}}_{\ell j}\|_2^2 \le P \\ \sqrt{P/\sum_{j=1}^{Q}\|\hat{\boldsymbol{v}}_{\ell j}\|_2^2}\cdot \hat{\boldsymbol{v}}_{\ell q} & \text{otherwise.} \end{cases}$$

Interestingly, if we instead use the conventional quadratic transform of Algorithm 1 to optimize $\underline{\boldsymbol{v}}$ in $h(\underline{\boldsymbol{v}}, \underline{\gamma})$, then the iterative optimization boils down to the WMMSE algorithm [9], [10]. However, WMMSE requires computing the inverse of an $M \times M$ matrix many times, so its complexity is high in the massive MIMO scenario in which M is a large number.

We now test the various quadratic transform methods for massive MIMO in a simulated 7-hexagonal-cell wrapped-around network as considered in [4]. 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 is set to be 0.8 km. The maximum transmit power level at the BS side is set to be 20 dBm, and the AWGN power level is set to be -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. 2(a), Algorithm 1 converges faster than the other two methods in terms of iterations; this result agrees with the former discussion below Proposition 2.

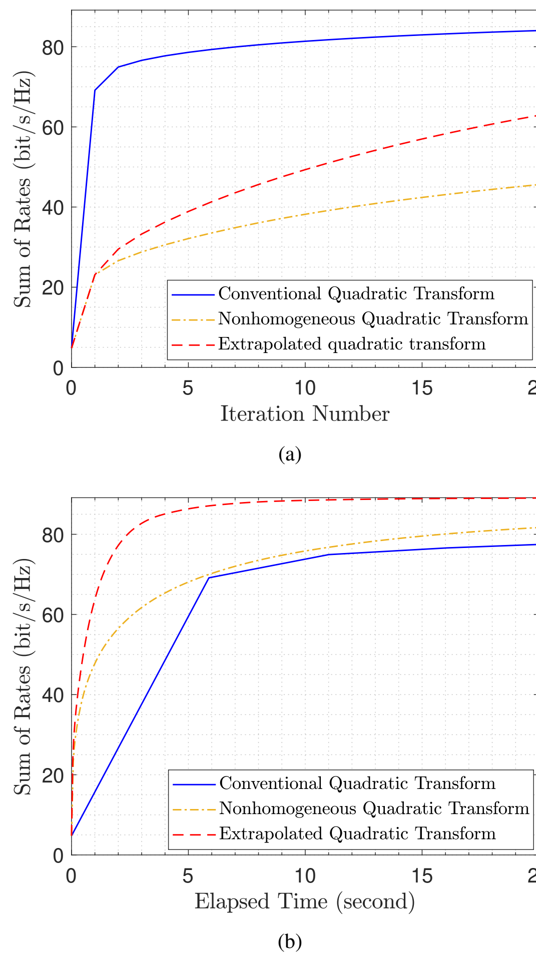

Fig. 2. Maximizing sum rates in massive MIMO network. Panel (a) shows the convergence in iterations, while panel (b) shows the convergence in time.

When it comes to the convergence evaluated by time, as shown in Fig. 2(b), the two accelerated quadratic transform methods are much more efficient than Algorithm 1.

V. Conclusion

This work develops the existing theory and algorithm of FP, focusing on their applications in wireless networks. The quadratic transform is a state-of-the-art tool in the FP area. As a starting point, we establish a connection between the quadratic transform and the gradient projection; this connection turns out to be fairly useful in that it enables the iterative optimization to get rid of matrix inverses. We then propose further accelerating the quadratic transform via extrapolation. Of fundamental importance is the convergence rate analysis that follows. To the best of our knowledge, this is the very first work that examines how fast the quadratic transform (including its special case the WMMSE algorithm) converges and also how to render it even faster. Moreover, we demonstrate the practical usefulness of the accelerated quadratic transform through the application case of massive MIMO beamforming.

REFERENCES

[1] K. Shen, Z. Zhao, Y. Chen, Z. Zhang, and H. Cheng, "Accelerating quadratic transform and WMMSE," Dec. 2023, [Online]. Available: https://kaimingshen.github.io/doc/accelerated_FP.pdf.

[2] Y. Nesterov, "Lectures on convex optimization (second edition)." Springer, 2018.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] D. P. Bertsekas, "Convex optimization algorithms." Athena Scientific, 2015.