Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-12
DOI:10.22034/cmde.2025.64854.2951

On The Efficiency of Algorithm for Solving Complex Quadratic Double–Ratio Minimax Optimization problem

Arezu Zare and Bahram Sadeghi Bigham[*,**]

Department of Computer Science, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran.

Abstract

Quadratic fractional optimization problems frequently arise in wireless communications. This paper introduces an enhanced semidefinite optimization relaxation approach for tackling signal design challenges associated with quadratic double–ratio minimax optimization in complex space. It results in two algorithms that offer a global optimum solution for the problem.

Keywords. Fractional programming, Minimax optimization, Quadratic programming, Semidefinite programming, Global optimization.

2010 Mathematics Subject Classification. 90C32, 90C20, 90C22, 90C26.

1. Introduction

Quadratic ratio minimax optimization problem is a significant challenge in wireless communications, as it seeks to maximize minimum network performance for fairness [11, 21, 24]. The signal–to–noise–and–interference ratio (SNIR) is a key performance metric in wireless communications. This paper examines an effective signal design strategy for the following NP–hard [17, 23] optimization problem:

$$(\mathcal{P}) \quad \min_{x\in\mathbf{C}^n} \max \quad \left\{\frac{p_1(x)}{p_2(x)}, \frac{p_3(x)}{p_4(x)}\right\},$$

where $p_i(x) = x^H A_i x - 2\mathrm{Re}(\mathrm{a_i^H x}) + \alpha_{\mathrm{i}}$, $A_i \in \mathbb{H}^n$ are Hermitian matrices, $a_i \in \mathbb{C}^n$ are complex vectors and $\alpha_i \in \mathbb{R}$ are constants for all $i = 1, 2, 3, 4$. "H" and "Re" denote the conjugate transpose operation and real part of $z \in \mathbb{C}$, respectively. Moreover, we assume that $p_2(x), p_4(x) > 0$ for all x in the feasible region.

In 1984, Datta and Bhatia [10] Investigated the complex minimax programming problem. Hereafter, many authors have explored linear and nonlinear fractional optimization problems in complex variables, employing diverse objective functions and developing various methods for addressing the design problem. For example, Lai and Liu [19] established optimality conditions for complex minimax programming problems based on generalized convexity. Lai and Haung [18] utilized Wolfe–type and Mond–Weir–type dual problems to find the optimal solution for the nondifferentiable minimax fractional programming problem in complex space. Gharanjik et al. [13] employed an iterative optimization framework with penalized reformulation to address the Max-Min beamforming design problem. Huang [16] formulated the second–order Mond–Weir type and Wolfe type dual models with respect to a complex minimax fractional programming problem. Recently, Hu et al. [9] studied a secure wireless communication system enhanced by an intelligent reflecting surface through complex quadratic fractional optimization.

It is worth noting that the structured total least squares (TSL) problem frequently occurs in wireless communications and signal processing in both complex and real spaces [5, 20, 25, 26]. In fact, the linear system $A\mathrm{Re}(x) \approx b$ approximate

Received: 28 November 2024 ; Accepted: 13 January 2025.
* Corresponding author. Email: b_sadeghi_b@alzahra.ac.ir.
** This research has been facilitated by the Data Science Lab of the Faculty of Mathematical Sciences at Alzahra University.

by solving

$$\min_{x\in\mathbb{C}^n} \quad \frac{\|A\mathrm{Re}(x)-b\|^2}{\|x\|^2+1},$$

where $A \in \mathbb{C}^{m\times n}$ is a complex matrix, $b \in \mathbb{C}^m$ and $\|\cdot\|$ denotes the Euclidean norm. Two approaches exist for stabilizing the ill–conditioned TLS: adding a quadratic constraint to limit the solution size [2, 3] or incorporating a quadratic penalty into the objective (Tikhonov regularization) [1, 6]. Accordingly, this paper aims to introduce a stabilization method using $(\mathcal{P})$ as follows:

$$\min_{x\in\mathbb{C}^n} \max \quad \left\{ \frac{\|A\mathrm{Re}(x)-b\|^2}{\|x\|^2+1}, \rho\|\mathrm{Im}(Lx)\|^2 \right\}, \tag{1.1}$$

where $\rho \in \mathbb{R}$ is a positive regularization parameter, $L \in \mathbb{C}^{k\times n}$ $(k \le n)$ is a regularization matrix of full row rank and "Im" denotes image part of $z \in \mathbb{C}^n$.

1.1. **Applications.** We examine an underlay cognitive radio network consisting of one primary and one secondary network, as illustrated in Figure 1 [22].

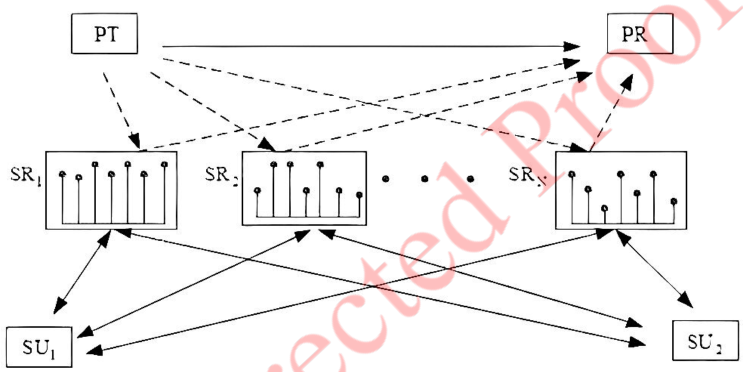

FIGURE 1. System model [22].

The primary network consists of a primary transmitter (PT) and a primary receiver (PR). The secondary network includes two secondary users (SU_1 and SU_2) and N relay nodes ($\mathrm{SR}_1,\dots,\mathrm{SR}_\mathrm{N}$). All terminals are equipped with a single antenna and operate in half–duplex mode, with channels subject to frequency selective Rayleigh fading and additive white Gaussian noise (AWGN). The primary network supports unidirectional transmission from PT to PR, while the secondary network enables bidirectional communication between SU_1 and SU_2 via the relay nodes, with no direct link between them. According to the network geometry in Figure 1, signals from SU_1 and SU_2 do not reach PR, but signals from SR_1 to SR_N can be detected by PR. In contrast, SU_1 and SU_2 cannot detect PT's signal, although they are influenced by it. The information exchange between SU_1 and SU_2 requires two–time slots: in the first, SU_1 and SU_2 transmit to the relay nodes while PT communicates with PR. Using the finite impulse response (FIR) filter model for the frequency selective channel, the received signal at the relay nodes in the first time slot can be expressed as follows:

$$r(n) = \sum_{i=1}^{2}\sum_{l=0}^{L-1} f_{i,l}x_{si}(n-l) + \sum_{l=0}^{L-1} g_{1,l}x_{p,1}(n-l) + v(n), \tag{1.2}$$

where $r(n)$ is an $N\times 1$ vector whose ith element denotes the received signals by SU_i and PT, respectively, $v(n)$ is the additive white Gaussian noise process, $f_{i,l} = [f_{i,l,1},\dots,f_{i,l,N}]^T$, where $f_{i,l,j}$ and $(\cdot)^T$ denote the lth channel coefficient between SU_i and the transpose, respectively. Moreover, relay $j, g_{1,l} = [g_{1,l,1}, \ldots, g_{1,l,N}]^T$, where $g_{1,l,j}$ denotes the lth channel coefficient from PT to relay j, and L denotes the number of frequency selective channel coefficients and is determined based on the multipath spread of the channel. The first term in the right–hand side of (1.2) is the received signal from SU_1 and SU_2, and the second term is the interfering signal received from PT. To combat the frequency selectivity of the channel, each relay employs an FIR filter to process the received signal. The resulting network–coded signal can be described as

$$t(n) = \sum_{l=0}^{L_w-1} U_l^H r(n-1),$$

where $t(n)$ is an $N \times 1$ vector whose ith element denotes the transmitted signal by relay by relay i, $U_l = \mathrm{diag}\left(u_{l,1}, u_{l,2}, \ldots, u_{l,N}\right)$ where $u_{l,i}$ denotes the lth tap of the FIR filter. In the second time slot, the relay nodes broadcast their network–coded signals, and at the same time, PT transmits its signal to PR. Under channel reciprocity assumption, the received signals at SU_i and PR in the second time slot can be formulated as

$$\begin{aligned} y_i(n) &= \sum_{l=0}^{L-1} f_{i,l}^T t(n-1) + n_{si}(n), \ \ i = 1, 2, \\ y_p(n) &= \sum_{l=0}^{L-1} h_{p,l} x_{p_2}(n-1) + \sum_{l=0}^{L-1} g_{2,l}^T t(n-l) + n_p(n), \end{aligned} \tag{1.3}$$

where $x_{p_2}(n)$ is the transmitted signal by PT in the second time slot, $g_{2,l} = [g_{2,l,1}, \ldots, g_{2,l,N}]^T$, where $g_{2,l,j}$ denotes the lth channel coefficient from relay j to PR, $h_{p,l}$ is the lth channel coefficient from PT to PR, and $n_{si}(n)$ and $n_p(n)$ are the AWGN processes at SU_i and PR, respectively. The first term on the right–hand side of (1.3) represents the signal received from PT, while the second term denotes the interference from the relay nodes. Define

$$\begin{aligned}
&\widetilde{v}(n) = \left[v^T(n), \ldots, v^T(n-L_w+1)\right]^T, && \widetilde{x}_{s_i}(n) = \left[x_{s_i}(n), \ldots, x_{s_i}(n-L-L_w+2)\right]^T, \\
&\widetilde{x}_{p_1}(n) = \left[x_{p_1}(n), \ldots, x_{p_1}(n-L-L_w+2)\right]^T, && \widetilde{F}_i = \left[\overline{F}_{i,0}^T, \ldots, \overline{F}_{i,L_w-1}\right]^T, \\
&\widetilde{G}_i = \left[\overline{G}_{i,0}^T, \ldots, \overline{G}_{i,L_w-1}\right]^T, && \overline{G}_i = \left[g_{i,0}, \ldots g_{i,L-1}\right], \\
&\widetilde{F}_{i,l} = \left[0_{N \times l}, \overline{F}_i, 0_{N \times (L_w-l-1)}\right], && \overline{F}_i = \left[f_{i,0}^T, \ldots, f_{i,L-1}\right], \\
&\overline{G}_{i,l} = \left[0_{N \times l}, \overline{G}_i, 0_{N \times (L_w-l-1)}\right], && u_i = \mathrm{diag}\{\mathrm{U_l}\}, \\
&G_{i,1} = \mathrm{diag}\{\mathrm{g_{i,1}}\}, && F_{i,1} = \mathrm{diag}\{\mathrm{f_{i,1}}\}, \\
&u = \left[u_0^T, \ldots, u_{L_w-1}^T\right]^T, && i = 1, 2, \quad l = 0, \ldots, L-1.
\end{aligned}$$

Based on the above definitions, $y_i(n)$ can be expressed as

$$y_i(n) \ = \ \sum_{l=0}^{L-1} \left\{ u^H \left(I_{L_w} \otimes F_{i,l}\right) \left[\widetilde{F}_1 \widetilde{x}_{s_1}(n-1) + \widetilde{F}_2 \widetilde{x}_{s_2}(n-l) + \widetilde{G}_1 \widetilde{x}_{p_1}(n-1)\right] + \widetilde{v}(n-1) \right\} + n_{s_i}(n), \quad i = 1, 2,$$

where $\otimes$ denotes the Kronecker product. Similarly, $y_p(n)$ can be rewritten as

$$y_p(n) \ = \ u^H \Psi_{g_2} \left[\widehat{F}_1 \widehat{x}_{s_1}(n) + \widehat{F}_2 \widehat{x}_{s_2}(n) + \widehat{G}_1 \widehat{x}_{p_1}(n) + \widehat{I} \widehat{v}(n)\right] + \sum_{l=0}^{L-1} h_{p,l} x_{p_2}(n-1) + n_p(n), \tag{1.4}$$

where

$$\widehat{F}i,l = \left[0_{NL_w \times l}, \widetilde{F}_i, 0_{NL_w \times (L-l-1)}\right], \qquad \widehat{F}_i = \left[\widehat{F}_{i,0}^T, \dots, \widehat{F}_{i,L-1}^T\right]^T,$$

$$\widehat{G}_{i,l} = \left[0_{NL_w \times l}, \widetilde{G}_{i,L-1}^T\right], \qquad \widehat{G}_i = \left[\widehat{G}_{i,0}^T, \dots, \widehat{G}_{i,L-1}^T\right]^T,$$

$$\Psi_{g_i} = \left[I_{L_w} \otimes G_{i,0}, \dots, I_{L_w} \otimes G_{i,L-1}\right], \qquad \Psi_{f_i} = \left[I_{L_w} \otimes F_{i,0}, \dots, I_{L_w} \otimes F_{i,L-1}\right],$$

$$\widehat{v} = \left[v^T(n), \dots, v^T(n-L-L_w+2)\right]^T, \qquad \widehat{x}_{s_i}(n) = \left[x_{s_i}(n), \dots, x_{s_i}(n-2L-L_w+3)\right]^T,$$

$$\widehat{x}_{p_1} = \left[x_{p_1}(n), \dots, x_{p_1}(n-2L-L_w+3)\right]^T, \qquad \widehat{I} = \left[0_{NL_w \times Nl}, I_{NL_w \times N(L-l-1)}\right],$$

$$\widehat{I} = \left[\widehat{I}_0^T, \dots, \widehat{I}_{L-1}^T\right]^T.$$

SU_1 and SU_2 are assumed to have perfect channel state information (CSI), enabling complete cancellation of self–interference from the relayed signal. After this cancellation, SU_i obtains

$$\widetilde{y}_i(n) = u^H \Psi_{f_i}\left[\widehat{F}_j \widehat{x}_{s_j}(n) + \widehat{G}_1 \widehat{x}_{p_1}(n) + \widehat{I}\widehat{v}(n)\right] + n_{s_i}(n), \quad i,j = 1,2, \ i \neq j, \tag{1.5}$$

Which can be used for the detection of the desired symbols.The tap weights of the FIR filters provide NL_w degrees of freedom for beamforming at relay nodes, optimizing secondary network performance while keeping interference at the PR within an allowable level.

Now, the transmitted power of the mth relay is given by

$$\widetilde{p}_m = u^H \left(p_1 D_{1,m} + p_2 D_{2,m} + p_p D_{3,m} + \sigma_v^2 D_{N,m}\right) u,$$

where σ_v^2 is the noise variance at the relay nodes, p_1, p_2 and p_p are the transmitted power by SU_1 and SU_2 and PT, respectively, and $D_{1,m}$, $D_{2,m}$, $D_{3,m}$ and $D_{N,m}$ are given by

$$\begin{aligned} D_{N,m} &= \left(I_{L_w} \otimes E_m\right)\left(I_{L_w} \otimes E_m\right)^H, \\ D_{3,m} &= \left(I_{L_w} \otimes E_m\right)\overline{G}_1\overline{G}_1^H\left(I_{L_w} \otimes E_m\right)^H, \\ D_{i,m} &= \left(I_{L_w} \otimes E_m\right)\overline{F}_i\overline{F}_i^H\left(I_{L_w} \otimes E_m\right)^H, \quad i = 1,2 \end{aligned}$$

where $E_m = \mathrm{diag}\{e_m\}$, and e_m is the mth row of the identity matrix. So, the total power transmitted by the relay nodes can be calculated as follows:

$$p_r = \sum_{m=1}^{N} \widetilde{p}_m = u^H \left(p_1 D_1 + p_2 D_2 + p_P D_3 + \sigma_v^2 I_{NL_W}\right) u,$$

where $D_i = \sum_{m=1}^{N} D_{i,m}$, for all $i = 1,2,3$. The signals from SU_1 and SU_2, as well as those from each user at different time indexes, are assumed to be independent. Using substituting $\Psi_{f_i}\widehat{F}_j = \left[h_s, \overline{H}_s\right]$ and $\widehat{x}_{s_j}(n) = \left[x_{s_j}(n), \widetilde{x}_{s_j}(n)\right]^T$ into (1.5), $\widetilde{y}_i(n)$ can be rewritten as

$$\widetilde{y}_i(n) = u^H h_s x_{s_j}(n) + u^H \overline{H}_s \widetilde{x}_{s_j}(n) + u^H \Psi_{f_i}\widehat{G}_1 \widehat{x}_{p_1}(n) + u^H \Psi_{f_i}\widehat{I}\widehat{v}(n) + n_{s_i}(n), \tag{1.6}$$

where $x_{s_j}(n)$ and $\widetilde{x}_{s_j}(n)$ are the desired signal and ISI components at SU_i, respectively. From (1.6), the powers of the desired signal, the ISI component, PT interference, and AWGN process at SU_i can be calculated as

$$\begin{aligned} P_{s_i} &= E\{\mid u^H h_s x_{s_j}(n) \mid^2\} = p_j u^H A_s u, \\ P_{\mathrm{ISI}} &= E\{\mid u^H \overline{H}_s \widetilde{x}_{s_j}(n) \mid^2\} = p_j u^H B_{\mathrm{ISI}} u, \\ P_{PI_i} &= E\{\mid u^H \Psi_{f_i}\widehat{G}_1 \widehat{x}_{p_1}(n) \mid^2\} = p_p u^H B_{P_i} u, \\ P_{N_i} &= E\{\mid u^H \Psi_{f_i}\widehat{I}\widehat{v}(n) + n_{s_i}(n) \mid^2\} = \sigma_v^2 u^H \Psi_{f_i}\widehat{I}\widehat{I}^H \Psi_{f_i}^H u + \sigma_{s_i}^2, \end{aligned}$$

where $A_s = h_s h_s^H$, $B_{\text{ISI}} = \overline{H}_s \overline{H}_s^H$ and $B_{p_i} = \Psi_{f_i} \overline{G}_1 \Psi_{f_i}^H \overline{G}_1^H$. Thus, the signal–to–interference–plus–noise ratio (SIPNR) at SU_i is given by

$$\text{SIPNR}_i = \frac{P_{s_i}}{P_{\text{ISI}_i} + P_{PI_i} + P_{N_i}}, \quad i = 1, 2.$$

By using (1.4), the interference power at PR can be determined as

$$\begin{aligned} T &= E\{|\ u^H \Psi_{g_2} \widehat{F}_1 \widehat{x}_{s_1}(n)\ |^2\} + E\{|\ u^H \Psi_{g_2} \widehat{F}_2 \widehat{x}_{s_2}(n)\ |^2\} + E\{|\ u^H \Psi_{g_2} \widehat{G}_1 \widehat{x}_{s_1}(n)\ |^2\} \\ &+ E\{|\ u^H \Psi_{g_2} \widehat{F}_1 \widehat{I} \widehat{v}(n)\ |^2\} = u^H C u, \end{aligned}$$

where

$$C = p_1 \Psi_{g_2} \widehat{F}_1 \widehat{F}_1^H \Psi_{g_2}^H + p_2 \Psi_{g_2} \widehat{F}_2 \widehat{F}_2^H \Psi_{g_2}^H + p_p \Psi_{g_2} \widehat{G}_1 \widehat{G}_1^H \Psi_{g_2}^H + \sigma_v^2 \Psi_{g_2} \widehat{I} \widehat{I}^H \Psi_{g_2}^H.$$

Now, the problem formulation is given by

$$\begin{aligned} \min_u \max \quad & \{\text{SIPNR}_1, \text{SIPNR}_2\}, \\ \text{s.t} \quad & T \leq I_{th}, \end{aligned}$$

where I_{th} is the maximum allowable interference power level at PR.

1.2. **Contributions and structure.** This paper examines the nonconvex problem $(\mathcal{P})$ and finds its global optimum solution. A parametric approach reformulates the problem into a quadratic optimization format, allowing for the global optimum to be determined using semidefinite optimization (SDO) relaxation and the Dinkelbach–type schemes. SDO relaxation provides a powerful tool for approximating the solution space, significantly enhancing the efficiency of the search for the global optimum.

The paper is organized as follows: Section 2 introduces the classical Dinkelbach method to reformulate problem $(\mathcal{P})$ into a quadratic optimization problem, then presents an SDO relaxation scheme and a Dinkelbach–type scheme to attain a global optimum. At last, Section 3 reports numerical experiments on various randomized test problems.

2. Main results

First, we consider the following lemma to transmit problem $(\mathcal{P})$ into a nonfractional one.

Lemma 2.1. ([27]) *The following two statements are equivalent:*
1) $\min_x \frac{f_1(x)}{f_2(x)} = \lambda^*$.
2) $\mathcal{F}(\lambda^*) := \min_x \{f_1(x) - \lambda f_2(x)\} = 0$.

Notice that the function $\mathcal{F}(\lambda)$ is continuous, concave, and strictly decreasing with a unique root. Now, from Lemma 2.1, we can reformulate problem $(\mathcal{P})$ as follows:

$$\mathcal{F}(\lambda) := \min_{x \in \mathbf{C}^n} \max\{p_1(x) - \lambda p_2(x), p_3(x) - \lambda p_4(x)\}. \tag{2.1}$$

Meanwhile, problem (2.1) can be rewritten as

$$\begin{aligned} \min_{x \in \mathbf{C}^n, \theta} \quad & \theta, \\ \text{s.t.} \quad & \theta \geq p_1(x) - \lambda p_2(x), \\ & \theta \geq p_3(x) - \lambda p_4(x). \end{aligned} \tag{2.2}$$

In following, we propose two optimization algorithms to solve (2.2) globally.

2.1. SDO **relaxation approach.** Note that problem (2.2) is equivalent to

$$\begin{aligned} \min_{X,\theta} \quad & \theta, \\ \text{s.t.} \quad & \theta \geq \widetilde{A}_1 \bullet X, \;\; i = 1, 2 \\ & X = xx^H, \end{aligned} \tag{2.3}$$

where $E \bullet G = \mathrm{Re}\left(\mathrm{tr}\left(E^H G\right)\right)$ and $\mathrm{tr}\,(\cdot)$ denotes the trace of a matrix and

$$\begin{aligned} \widetilde{A}_1 &= \begin{bmatrix} \overline{A}_1 & -\bar{a}_1 \\ -\bar{a}_1^H & \bar{\alpha}_1 \end{bmatrix}, \qquad \widetilde{A}_2 = \begin{bmatrix} \overline{A}_2 & -\bar{a}_2 \\ -\bar{a}_2^H & \bar{\alpha}_2 \end{bmatrix}. \\ \overline{A}_1 &= A_1 - \lambda A_2, \qquad \bar{a}_1 = a_1 - \lambda a_2, \qquad \bar{\alpha}_1 = \alpha_1 - \lambda \alpha_2, \\ \overline{A}_2 &= A_3 - \lambda A_4, \qquad \bar{a}_2 = a_3 - \lambda a_4, \qquad \bar{\alpha}_2 = \alpha_3 - \lambda \alpha_4. \end{aligned}$$

Define

$$\begin{aligned} \mathbf{S} &= \left\{ (x, X) \mid \theta \geq \widetilde{A}_i \bullet X, \;\; i = 1, 2, \;\; X = xx^H \right\}, \\ \widetilde{\mathbf{S}} &= \left\{ (x, X) \mid \theta \geq \widetilde{A}_i \bullet X, \;\; i = 1, 2, \;\; X \succeq xx^H \right\}. \end{aligned}$$

Moreover, we have

$$\begin{aligned} \mathrm{tr}(X) &= \mathrm{Re}\left(x^H x\right) \leq \sigma, \\ \sigma &:= \left(\max_{x \in \widetilde{\mathbf{S}}} \|x\|_\infty \right) \mathrm{Re}\left(x^H e\right), \end{aligned}$$

where e be column vector with all elements being 1, then problem (2.3) is convex. Now, since $\mathbf{S} \subseteq \widetilde{\mathbf{S}}$, (2.3) can be rewritten as

$$\begin{aligned} \min_{X,\theta} \quad & \theta, \\ \text{s.t.} \quad & \theta \geq \widetilde{A}_i \bullet X, \;\; i = 1, 2 \\ & \sigma \geq \mathrm{tr}\,(X), \\ & X = xx^H. \end{aligned} \tag{2.4}$$

(2.5)

The nonconvex constraint $X = xx^H$ in (2.4) can be relaxed to the convex constraint $X \succeq xx^H$, leading to the following convex SDO relaxation of (2.4):

$$\begin{aligned} \min_{X,\theta} \quad & \theta, \qquad\qquad (2.6) \\ \text{s.t.} \quad & \theta \geq \widetilde{A}_i \bullet X, \;\; i = 1, 2 \\ & \sigma \geq \mathrm{tr}\,(X), \qquad\qquad (2.7) \\ & X \succeq xx^H. \end{aligned}$$

It should be noted that incorporating constraint (2.7) improves the SDO relaxation of (2.3). Hence, we have the following theorem.

Theorem 2.2. *The optimal value of problem* (2.6) *is a lower bound for the optimal value of problem* (2.3).

Proof. Since the feasible set of (2.4) is a subset of that of (2.6), the optimal value of (2.6) serves as a lower bound for the optimal value of (2.3). □

We now need the following assumption and results to obtain the global optimum solution of (2.3).

Assumption 2.3. *There exist* $\eta_1, \eta_2 \geq 0$ *such that* $\eta_1 Q_1 + \eta_2 Q_2 \succ 0$.

Lemma 2.4. *([4]) Let $G \in \mathbb{R}^{n\times n}$ be symmetric, $g \in \mathbb{R}^n$ and $\varrho \in \mathbb{R}$, then*

$$x^T Gx - 2g^T x + \varrho \geq 0 \Leftrightarrow \begin{bmatrix} G & -g \\ -g^T & \varrho \end{bmatrix} \succeq 0.$$

Theorem 2.5. *For any well defined problem $(\mathcal{P})$ satisfying feasibility condition*

$$\lambda^* = \max_{\gamma_1,\gamma_2\geq 0,\lambda} \left\{ \lambda : \begin{bmatrix} 0_{n\times n} & 0_{n\times 1} \\ 0_{1\times n} & \theta \end{bmatrix} - \gamma_1 \begin{bmatrix} \bar{A}_1 & -\bar{a}_1 \\ -\bar{a}_1^H & \bar{\alpha}_1 - \theta \end{bmatrix} - \gamma_2 \begin{bmatrix} \bar{A}_2 & -\bar{a}_2 \\ -\bar{a}_2^H & \bar{\alpha}_2 - \theta \end{bmatrix} \succeq 0_{(n+1)\times(n+1)} \right\}. \tag{2.8}$$

Proof. It is enough that we consider the Lagrangian dual problem of (2.2) as follows:

$$\lambda^* = \max_{\lambda} \left\{ \lambda : \exists \gamma_1, \gamma_2 \geq 0,\ \theta - \gamma_1 \left(p_1(x) - \lambda p_2(x) - \theta\right) - \gamma_2 \left(p_3(x) - \lambda p_4(x) - \theta\right) \geq 0 \right\}.$$

Then, according to the Lemma 2.4 and Assumption 2.3, the proof is complete. □

Therefore, as summarized in the following, the SDO relaxation scheme in two–step is employed to solve $(\mathcal{P})$.

Algorithm 1 SDO relaxation scheme

Input: Parameters $A_1, A_2, A_3, A_4 \in \mathbb{H}^n$, $a_1, a_2, a_3, a_4 \in \mathbb{C}^n$ and $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$
Output: λ^*

```
Solve problem (2.8) to find λ*.
Solve problem (2.6) with λ = λ*
```

Problem (2.6) can be solved by using the Dinkelbanch–type algorithm [7, 8]. Hence, the algorithm can be described in the following way.

Algorithm 2 Dinkelbanch–type scheme

Input: Parameters $A_1, A_2, A_3, A_4 \in \mathbb{H}^n$, $a_1, a_2, a_3, a_4 \in \mathbb{C}^n$, $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$, $k_{\max}$ and $x_0 \in \mathbb{C}^n$
Output: λ^*

Initialize: $k := 1$, an accuracy parameter $\epsilon := 10^{-8}$ and $\lambda_1 := \max\left\{\frac{p_1(x_0)}{p_2(x_0)}, \frac{p_3(x_0)}{p_4(x_0)}\right\}$
while $k \leq k_{\max}$ **do**
  Solve SDO problem (2.6) with $\lambda = \lambda_k$ and obtain the solution x_k
  if $\mid \mathcal{F}(\lambda_k) \mid \leq \epsilon$ **then**
    break
  else
    $\lambda_{k+1} := \max\left\{\frac{p_1(x_k)}{p_2(x_k)}, \frac{p_3(x_k)}{p_4(x_k)}\right\}$
  end if
  $k := k + 1$
end while
Return λ^*

It must be pointed out that the Dinkelbach–type algorithm converges superlinearly to the unique zero of $\mathcal{F}(\lambda)$ [12].

3. Numerical experiments

This section discusses preliminary numerical experiments that compare the proposed schemes for solving $(\mathcal{P})$ on random test problems with dimensions from 50 to 1000 and varying densities. The average of numerical experiments for five times running is reported in Tables 1 and 2, which include λ^* and computational time in seconds (time(s)). Moreover, in these tables, means the algorithm fails to solve the problem. The implementation is done in MATLAB R2023a on a Core i7 with 8 GB of RAM, utilizing CVX [14] to solve SDO problems.

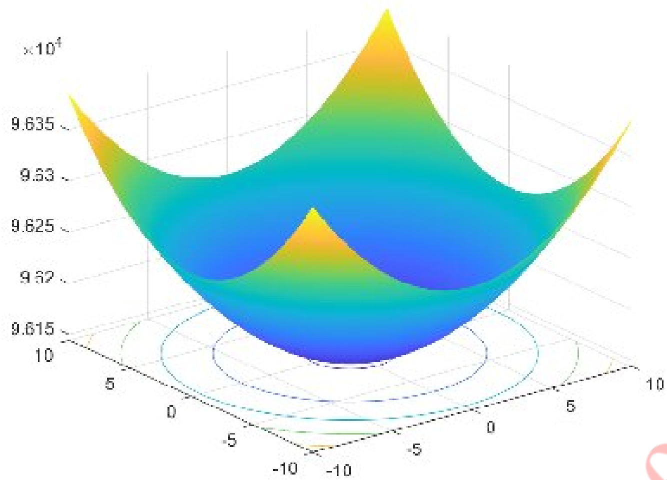

FIGURE 2. A two-dimensional instance and its contour.

Example 3.1. Let $n = 2$ and

$$A = \begin{bmatrix} 3 & 1-2i \\ 1+2i & 7 \end{bmatrix}, \ b = \begin{bmatrix} -10 \\ 4+i \end{bmatrix}, \ L = \begin{bmatrix} 0.1 \\ 0.8 \end{bmatrix}, \ \rho = 1.2.$$

Figure 2 shows the objective function (1.1) for this setting. The Dinkelbach–type scheme found the optimal value $\lambda^* = 1.8727$ in 16.9640 CPU seconds, while the SDO relaxation scheme achieved the same result in only 0.6221 seconds.

Example 3.2. Consider the following problem

$$\min_{x\in\mathbb{C}^n} \max \left\{ \frac{\|A\mathrm{Re}(x) - b\|^2}{\|x\|^2+1}, \rho\|\mathrm{Im}(Lx)\|^2 \right\},$$

where $A \in \mathbb{H}^n$, $b \in \mathbb{C}^n$, L is the first–derivative operator implemented in the function get_$l(n,1)$ [15] and $\rho = 0.9$ and 1.2. Matrices and vectors are generated using the following MATLAB code:

1. n =input(enter the size of the matrix=);
2. density= input(enter the density of the matrix=);
3. Q =sprandn(n,n,density)+i*sprandn(n,n,density);
4. $A = (Q + Q')/2$;
5. b =complex(randn(n,1),randn(n,1));

The numerical experiments indicate that while the Dinkelbach–type scheme can solve some problems, it requires more time for instances compared to the SDO relaxation scheme, which successfully resolves all instances in a reasonable timeframe. Overall, the comparison shows that the SDO relaxation scheme uses less CPU time.

4. CONCLUSIONS

This paper has presented a version of SDO relaxation approach for attaining a global optimum in the complex quadratic double–ratio minimax optimization problem. Furthermore, it has introduced an exact two–step SDO relaxation and Dinkelbach-type optimization schemes. Experimental results have shown that the SDO relaxation scheme surpasses the other.

Table 1. Numerical Results for Example (3.2) with $\rho = 0.9$

n	density	Dinkelbach−type scheme		SDO relaxation scheme	
		λ^*	time(s)	λ^*	time(s)
50	1	0.2375	3.4490	0.2375	1.2155
100	1	0.7316	11.9146	0.7316	3.2837
200	1	0.4213	67.9731	0.4213	14.6468
300	1	—	—	0.4425	91.2869
400	1	—	—	0.6207	164.5892
50	0.1	0.0952	3.3023	0.0952	1.1732
100	0.1	0.0878	11.5993	0.0878	3.0242
200	0.1	0.3665	56.7443	0.3665	13.3697
300	0.1	—	—	0.1734	43.6905
400	0.1	—	—	0.5216	154.2084
500	0.1	—	—	0.4923	172.1121
50	0.01	0.0527	3.0943	0.0527	1.0197
100	0.01	0.0671	10.3875	0.0671	2.8308
200	0.01	0.9342	46.4289	0.9342	12.1792
300	0.01	0.9116	132.4162	0.9116	35.2379
400	0.01	—	—	0.5312	148.9371
500	0.01	—	—	0.7038	169.4605
600	0.01	—	—	0.1557	201.2367
50	0.001	0.0068	2.8325	0.0068	0.7863
100	0.001	0.2657	9.6852	0.2657	2.2429
200	0.001	0.0650	40.1715	0.0650	14.6725
300	0.001	0.2435	104.5415	0.2435	26.5485
400	0.001	—	—	0.3020	125.8908
500	0.001	—	—	0.1894	153.2877
1000	0.001	—	—	0.9311	263.0845

TABLE 2. Numerical Results for Example (3.2) with $\rho = 1.2$

		Dinkelbach−type scheme		SDO relaxation scheme	
n	**density**	λ^*	**time(s)**	λ^*	**time(s)**
50	1	0.7825	24.0469	0.4825	1.8687
100	1	0.8967	74.3189	0.8967	3.3701
200	1	0.7272	132.1270	0.7272	16.7399
300	1	—	—	0.3849	94.0207
400	1	—	—	0.2651	172.3496
50	0.1	0.6240	23.8999	0.6240	1.3266
100	0.1	0.0428	72.5110	0.0428	3.1848
200	0.1	0.1643	125.8348	0.1643	12.2455
300	0.1	0.7273	219.2478	0.7273	91.3708
400	0.1	—	—	0.2418	167.2565
500	0.1	—	—	0.2835	194.2019
50	0.01	0.3045	20.1145	0.3045	1.1534
100	0.01	0.5960	70.9392	0.5960	2.6375
200	0.01	0.9004	116.2118	0.9004	11.6407
300	0.01	0.1583	194.3775	0.1583	87.9016
400	0.01	—	—	0.4371	165.3372
500	0.01	—	—	0.2532	191.0046
600	0.01	—	—	0.0063	203.5967
50	0.001	0.2297	18.4506	0.2297	0.8564
100	0.001	0.3241	69.7169	0.3241	2.1937
200	0.001	0.5604	121.6146	0.5604	8.2216
300	0.001	0.0752	193.2874	0.0752	83.1193
400	0.001	—	—	0.2468	159.3746
500	0.001	—	—	0.1730	186.3492
1000	0.001	—	—	0.2078	198.0107

Acknowledgment

This research has been facilitated by the Data Science Lab of the Faculty of Mathematical Sciences at Alzahra University.

References

[1] A. Beck and A. Ben–Tal, *On the solution of the Tikhonov regularization of the total least squares problem*, SIAM J. Optim., *17* (1) (2006), 98–118.

[2] A. Beck and M. Teboulle, *A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid*, Math. Program., *118*(1) (2009), 13–35.

[3] A. Beck, A. Ben–Tal, and M. Teboulle, *Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares*, SIAM J. Matrix Anal. Appl., *28*(2) (2006), 425–445.

[4] A. Ben–Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis*, Algorithms and Engineering Applications, SIAM, Philadelphia, 2001.

[5] Y. Bresler and A. Macovski, *Exact maximum likelihood parameter estimation of superimposed exponential signals in noise*, IEEE Trans. Acust., Speech, Signal Process., *34*(5) (1986), 1081–1089.

[6] L. Consolini, M. Locatelli, J. Wang, and Y. Xia, *Efficient local search procedures for quadratic fractional programming problems*, Comput. Optim. Appl., *76* (2020), 201–232.

[7] J. P. Crouzeix, J. A. Ferland, and S. Schaible, *An algorithm for generalized fractional programs*, J. Optim. Theory Appl., *47*(1) (1985), 35–49.

[8] J. P. Crouzeix, J. A. Ferland, and S. Schaible, *A note on an algorithm for generalized fractional programs*, J. Optim. Theory Appl., *50*(1) (1986), 183–187.

[9] M. Cui, G. Zhang, and R. Zhang, *Secure wireless communication via intelligent reflecting surface*, IEEE Wirel. Commun., *8*(5) (2019), 1410–1414.

[10] N. Datta and D. Bhatia, *Duality for a class of nondifferentiable mathematical programming problems in complex space* , J. Math. Anal. Appl., *101*(1) (1984), 1–11.

[11] G. Dartmann, X. Gong, W. Afzal, and G. Ascheid, *On the duality of the max–min beamforming problem with per-antenna and per-antenna-array power constraints*, IEEE Trans. Veh. Technol., *62*(2) (2012), 606–619.

[12] J. Flachs, *Generalized Cheney–Loeb–Dinkelbach–type algorithms*, Math. Oper. Res., *10*(4) (1985), 674–687.

[13] A. Gharanjik, M. R. B. Shankar, M. Soltanalian, and B. Ottersten, *Max–min transmit beamforming via iterative regularization*, in 50th Asilomar Conf. Signals, Systems & Computers, 2016, 1437–1441.

[14] M. Grant and S. Boyd, *Cvx: Matlab software for disciplined convex programming*, Version 2.1, 2014.

[15] P. C. Hansen, *Regularization tools: a matlab package for analysis and solution of discrete ill–posed problems*, Numer. Algorithms, *6*(1) (1994), 1–35.

[16] T. Y. Huang, *Second-order parametric free dualities for complex minimax fractional programming*, Mathematics, *8*(1) (2020), 67-79.

[17] E. Karipidis, N. D. Sidiropoulos, and Z. Q. Luo, *Quality of service and max–min fair transmit beamforming to multiple cochannel multicast groups*, IEEE Trans. Signal Process., *56*(3) (2008), 1268–1279.

[18] H. C. Lai and T. Y. Huang, *Complex analysis methods related an optimization problem with complex variables*, Eur. j. pure appl., *3*(6) (2010), 989–1005.

[19] H. C. Lai and J. C. Liu, *Complex fractional programming involving generalized quasi/pseudo convex functions* , J. Appl. Comput. Mech., *82*(3) (2002), 159–166.

[20] B. Tausiesakul and K. Asavaskulkiet, *TDoA Localization in Wireless Sensor Networks Using Constrained Total Least Squares and Newtons Methods*, IEEE Access., *12* (2024), 39238–39260.

[21] M. Schubert and H. Boche, *Solution of the multiuser downlink beamforming problem with individual SINR constraints*, IEEE Trans. Veh. Technol., *53*(1) (2004), 18–28.

[22] M. Shamailzadeh and A. H. Bastami, *Cooperative beamforming in cognitive radio network with network–coded bidirectional filter–and–forward relaying*, in Proc. ICEE, Shiraz, Iran, May, 2016, 1454–1459.

[23] N. D. Sidiropoulos, T. N. Davidson, and Z.Q. Luo, *Transmit beamforming for physical-layer multicasting*, IEEE Trans. Signal Process., *54*(6) (2006), 2239–2251.
[24] V. P. Singh and A. K. Chaturvedi, *Max–min fairness based linear transceiver–relay design for MIMO interference relay channel*, IET Commun., *11*(9) (2017), 1485–1496.
[25] A. Zare, *Solving the complex quadratic double-ratio minimax optimization under a quadratic constraint*, J. Appl. Math., *69*(1) (2023), 589–602.
[26] A. Zare, A. Ashrafi, and Y. Xia, *Quadratic double-ratio minimax optimization*, Oper. Res. Lett., *49*(4) (2021), 543–547.
[27] A. Zhang and S. Hayashi, *Celis–Dennis–Tapia based approach to quadratic fractional programming problems with two quadratic constraints*, Numer. Algebra, Control and Optim., (1) (2011), 83–98.