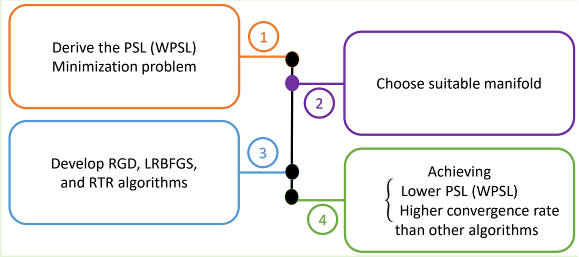
Minimum PSL Sequence Set Design for MIMO Radars via Manifold-Based Optimization

Hamideh Zebardast, Mahmoud Farhang and Abbas Sheikhi

***Abstract*—In this paper, we consider the problem of waveform design for multiple-input multiple-output (MIMO) radars via a peak sidelobe level (PSL) minimization approach. We derive the optimization problem as the PSL and weighted PSL (WPSL) minimization problem. To solve the optimization problem, we consider a suitable product manifold within the unimodular sequence space and develop numerical manifold-based algorithms such as Riemannian gradient descent (RGD), limited memory Riemannian Broyden, Fletcher, Goldfarb, and Shanno (LRBFGS), and Riemannian trust region (RTR) on this manifold. We investigate the performance of the proposed algorithms based on the achieved PSL and WPSL values and the convergence speed. Numerical results show that the proposed algorithms yield sequences with lower (weighted) peak sidelobe levels, compared to other state-of-the-art counterparts. Moreover, these algorithms converge with a high convergence rate, that is, they are so fast. Therefore, it can be said that these algorithms are very computationally efficient and can be used to design very long sequences.**



***Index Terms*—peak sidelobe level (PSL), manifold-based algorithms, MIMO radar, Riemannian manifold, signal design**

I. Introduction

SIGNAL design has a critical role in improving the performance of active sensing systems [1]. Specifically, target detection with high accuracy is of great importance in radars, and many researchers have considered signal design for better detection of targets and increased resistance to interference and clutter [2]–[4]. In this regard, the peak sidelobe level (PSL) of the transmitted sequences is a key design parameter [5]–[12]. In single-input single-output (SISO) and single-input multiple-output (SIMO) radars, the PSL is defined as the maximum sidelobe level of the autocorrelation function. Reducing the PSL improves the detection quality and prevents weak targets from being masked by side-lobes of a strong target [13], [14]. In multiple-input multiple-output (MIMO) radars, the PSL refers to the maximum sidelobe level of the autocorrelation and cross-correlation functions of the transmitted sequences. In these radars, reduction of the sidelobe level of the autocorrelation functions is very important, as in SISO and SIMO radars. In addition, small cross-correlation sidelobe values enable the MIMO radar receiver to differentiate the transmitted waveforms and exploit waveform diversity to construct the virtual array [5], [15], [16].

In some applications, minimization of the weighted peak sidelobe level (WPSL) is the relevant design approach, in which minimum sidelobe levels within a certain range are desired. For instance, in tracking mode, the radar's search window is smaller compared to the search mode [17], [18]. Therefore, it is enough to minimize the sidelobes close to the target in these radars, which will be possible by minimizing the WPSL.

Since constant-envelope signals are usually used in radars to achieve maximum power efficiency [19], the unimodularity constraint is imposed on most signal design problems [20], [21]. Several optimization-based algorithms are so far proposed to design unimodular sequences of desired length with minimum PSL/WPSL. Some methods use majorization-minimization (MM) to solve the optimization problem. In this context, the MM-based approach minimizing l_p-norm and monotonic minimizer for the l_p-norm are presented in [6] and [7], respectively. The majorizing function mainly determines the convergence speed of the MM method. Some acceleration techniques have been also presented to increase the convergence rate of these types of algorithms. However, they have only considered the sequence design for SISO radars. The gradient descent (GD) approach for l_p-norm minimization is proposed in [8]. Simplicity is one of the main features of this method. However, it uses a series of approximations to deal with the unimodularity constraint. Binary sequences sets (BiST) algorithm is based on the block coordinate descent (BCD) method to design a set of sequences with discrete phase constraint [9]. The weighted BSUM sequence set (WeBEST) method is used in [10] to minimize the l_p-norm of autocorrelation and cross-correlation sidelobes based on the MM method for a set of sequences. In each iteration of this algorithm, different local approximation functions are optimized concerning a block containing a code entry or a code vector. Thesee algorithms guarantee convergence to a stationary point for smooth objective functions. However, their

H. Zebardast, M. Farhang, and A. Sheikhi are with the School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran. (email: {hamideh.zebardast,mfarhang,sheikhi}@shirazu.ac.ir).

computational complexity is polynomial, which may make it difficult for large-scale problems. [11] also proposed an iterative monotonic algorithm based on MM technique called PSL minimizer, which generates sequence set by optimizing the exact PSL metric. The iterative steps of this algorithm are computationally not very demanding and they can be efficiently implemented via Fast Fourier Transform (FFT) operations. The performance of this algorithm is evaluated in the context of probing sequence set design for MIMO radar angle-range imaging application. [12] develops an algorithm based on MM framework, which transforms the complex p-order polynomial minimization problem into a series of low-order simple optimization problems. This algorithm minimizes PSL and integrated sidelobe level (ISL) simultaneously and strikes a good balance between the PSL and ISL. Compared with other state-of-the-art ISL minimization algorithms, this method obtains slightly higher ISL and much lower PSL.

In this paper, we consider the problem of designing constant-modulus sequence set by minimizing the l_p-norm of autocorrelation and cross-correlation sidelobes. Since optimization problems rarely have an analytical solution, numerical algorithms are usually needed to solve this problem. The unit-modulus condition prevents the direct use of conventional unconstrained numerical methods in solving this minimization problem. However, we show that the problem can be transformed into an unconstrained problem over a Riemannian manifold [22]. For this purpose, we choose three unconstrained optimization algorithms, namely, gradient descent (GD), Broyden-Fletcher-Goldfarb-Shanno (BFGS), and trust-region (TR), and apply them on a suitable Riemannian manifold. These algorithms ensure convergence with higher speed compared to other state-of-the-art algorithms. Also, compared to other algorithms, these algorithms reach a lower PSL, which shows their good performance.

The rest of the paper is organized as follows. In Section II, the MIMO PSL/WPSL minimization problem is formulated, and in Section III the optimization framework for solving this problem is introduced. Three manifold-based algorithms of RGD-MPSL, LRBFGS-MPSL, and RTR-MPSL are developed in Section IV. Some numerical simulations are given in Section V to evaluate the performance of the proposed algorithms. Finally, Section VI concludes the paper.

We adopt the notation of using lowercase letters to represent scalar variables, bold lowercase letters for vectors, and bold uppercase letters for matrices. We indicate the N-dimensional real space and the N-dimensional complex space by $\mathbb{R}^N$ and $\mathbb{C}^N$, respectively. The transpose, the conjugate, the inverse, and the derivative operators are shown by $(.)^T$, $(.)^*$, $(.)^{-1}$, and $(.)^{'}$, respectively. The absolute value of the variable x is denoted by $|x|$. $|\mathbf{x}|$ is a vector of element wise absolute values of the vector $\mathbf{x}$, and the norm of the vector $\mathbf{x}$ is represented by $\|\mathbf{x}\|$. For any vector $\mathbf{x}$, $\mathrm{Re}(\mathbf{x})$ denotes the real part of $\mathbf{x}$. The inner product is shown by $\langle .,. \rangle$, while the inner product on the tangent space of the manifold $\mathcal{N}$ is represented by $\langle .,. \rangle_{\mathbf{x}}$. For any matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M]$, where each $\mathbf{y}_m$, $m = 1, 2, \ldots, M$, is an $N \times 1$ vector, $\|\mathbf{Y}\|$ equals $\sqrt{\sum_{m=1}^{M} \langle \mathbf{y}_m, \mathbf{y}_m \rangle}$. Similarly, if $\mathbf{y}_m$ is on the tangent space of the manifold $\mathcal{N}$, $\|\mathbf{Y}\|$ equals $\sqrt{\sum_{m=1}^{M} \langle \mathbf{y}_m, \mathbf{y}_m \rangle_{\mathbf{x}_m}}$. The symbol $\odot$ denotes the Hadamard product, and the components of the matrix $\mathbf{Y} = \mathbf{X} \odot \frac{1}{|\mathbf{X}|}$ are obtained by dividing each member of the matrix $\mathbf{X}$ by its absolute value. The abbreviations "s.t." and "vs." stand for "subject to" and "versus", respectively, and $\mathbf{I}_N$ shows the N-dimensional unit matrix.

II. Problem Statement

We consider a MIMO radar with M transmit antennas. Each antenna transmits a unimodular continuous phase sequence with length N. Let the code matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M) \in \mathbb{C}^{N \times M}$ be the set of transmitted sequences, where the mth antenna transmits $\mathbf{x}_m = [\mathbf{x}_m(0), \mathbf{x}_m(1), \ldots, \mathbf{x}_m(N-1)]^T$. The aperiodic autocorrelation function of the sequence $\mathbf{x}_m$ ($\mathbf{r}_{m,m}(k)$) and cross-correlation function of the sequences $\mathbf{x}_m$ and $\mathbf{x}_l$ ($\mathbf{r}_{m,l}(k)$) are expressed as [14],

$$\begin{cases} \mathbf{r}_{m,m}(k) = \sum_{n=0}^{N-k-1} \mathbf{x}_m(n)\mathbf{x}_m^*(n-k) = \mathbf{r}_{m,m}^*(-k) \\ \mathbf{r}_{m,l}(k) = \sum_{n=0}^{N-k-1} \mathbf{x}_m(n)\mathbf{x}_l^*(n-k) = \mathbf{r}_{l,m}^*(-k), \end{cases}$$

where $-(N-1) \leq k \leq N-1$ and $m, l \in \{1, 2, \ldots, M\}$. The WPSL for MIMO radars is defined as [15]

$$\mathrm{WPSL} = \max\{\max_{m} \max_{k \neq 0} |w_k \mathbf{r}_{m,m}(k)|, \max_{\substack{m,l \\ m \neq l}} \max_{k} |w_k \mathbf{r}_{m,l}(k)|\}.$$

where w_k are real and positive weights, and the PSL is attained by choosing $w_k = 1$. The maximum absolute value of the vector elements is obtained by its L-infinity norm, WPSL can be written as

$$\begin{aligned} \mathrm{WPSL} = \lim_{p \to \infty} \Big\{ \sum_{m=1}^{M} \Big(\sum_{\substack{k=-N+1 \\ k \neq 0}}^{N-1} |w_k \mathbf{r}_{m,m}(k)|^p \\ + \sum_{\substack{l=1 \\ l \neq m}}^{M} \sum_{k=-N+1}^{N-1} |w_k \mathbf{r}_{m,l}(k)|^p \Big) \Big\}^{\frac{1}{p}}, \end{aligned} \tag{1}$$

Since the L-infinity norm is computationally infeasible, the following cost function is commonly used [7], [8], [10]

$$\begin{aligned} f(\mathbf{X}) = & \Big\{ \sum_{m=1}^{M} \Big(\sum_{\substack{k=-N+1 \\ k \neq 0}}^{N-1} |w_k \mathbf{r}_{m,m}(k)|^p \\ & + \sum_{\substack{l=1 \\ l \neq m}}^{M} \sum_{k=-N+1}^{N-1} |w_k \mathbf{r}_{m,l}(k)|^p \Big) \Big\}^{\frac{1}{p}}, \end{aligned} \tag{2}$$

where $2 \leq p < +\infty$. By choosing a tractable large p, the l_p-norm minimization approaches the WPSL minimization problem. The function $\mathbf{r}_{m,l}$ can be written as $\mathbf{r}_{m,l}(k) = \mathbf{x}_m^H \mathbf{S}_k \mathbf{x}_l$, where $\mathbf{S}_k$ is an $N \times N$ matrix whose $\{i, j\}^{th}$ entry for $i, j \in \{1, 2, \ldots N\}$ is 1 for $j - i = k$, otherwise is 0. Using the Hermitian symmetry of the auto- and cross-correlation functions, the cost function can be simplified as

$$\begin{aligned} f(\mathbf{X}) = & \Big\{ \sum_{m=1}^{M} \Big(\sum_{k=1}^{N-1} |w_k (\mathbf{x}_m^H \mathbf{S}_k \mathbf{x}_m)|^p \\ & + \sum_{\substack{l=1 \\ l \neq m}}^{M} \sum_{k=0}^{N-1} |w_k (\mathbf{x}_m^H \mathbf{S}_k \mathbf{x}_l)|^p \Big) \Big\}^{\frac{1}{p}}. \end{aligned} \tag{3}$$

The WPSL minimization problem can be written as

$$\begin{cases}\min_{\mathbf{X}\in\mathbb{C}^{N\times M}} & f(\mathbf{X})\\ \text{s.t.} & |\mathbf{x}_m|=1,\quad m=1,2,\dots,M.\end{cases}\tag{4}$$

The PSL optimization problem is given by setting all coefficients w_k equal to 1.

III. Optimization Framework

For unconstrained optimization problems in which a smooth cost function f is to be minimized over the search space $\mathbb{R}^N$ or $\mathbb{C}^N$, conventional optimization methods, such as TR or GD are usually utilized. However, the optimization problem (4) has a constraint which limits the search space to the following product manifold

$$\begin{aligned}&\mathcal{M}=\mathcal{M}_1\times\mathcal{M}_2\times\ldots\times\mathcal{M}_M,\\ &\mathcal{M}_m\triangleq\{\mathbf{x}\in\mathbb{C}^N:|\mathbf{x}_m|=1,m=1,2,\dots,M\}.\end{aligned}\tag{5}$$

In the following, we briefly introduce three standard algorithms of GD, BFGS, and TR for unconstrained optimization problems, and then their extension for optimization on Riemannian manifolds are presented.

A. Unconstrained Optimization Algorithms

1) GD Algorithm: This first-order iterative optimization algorithm takes repeated steps in the opposite direction of the function's gradient at the current point and moves in this direction by the step size (α). For any function $f(\mathbf{x})$, the search direction is chosen as $\mathbf{v}_i=-\nabla_{\mathbf{x}}f(\mathbf{x}_i)$ in the ith iteration. The step size is set based on the backtracking approach [23]. In this method, α_i takes an arbitrary positive initial value. Then, this value is reduced as $\alpha_i\leftarrow\rho\alpha_i$ until the following condition is satisfied

$$f(\mathbf{x}_i)-f(\mathbf{x}_i+\alpha_i\mathbf{v}_i)\geq r\alpha_i\|\nabla_{\mathbf{x}}f(\mathbf{x}_i)\|^2,\tag{6}$$

where $\rho,r\in(0,1)$. The $\mathbf{v}_i$ and α_i are used to update $\mathbf{x}$ as

$$\mathbf{x}_{i+1}=\mathbf{x}_i+\alpha_i\mathbf{v}_i.\tag{7}$$

2) BFGS Algorithm : This algorithm is a second-order optimization algorithm. Same as GD, it computes a search direction, then finds an acceptable step size through a backtracking approach. The update formula in each iteration i is the same as (7), with $\mathbf{v}_i$ given by

$$\mathbf{v}_i=-(\nabla^2_{\mathbf{x}}f(\mathbf{x}_i))^{-1}\nabla_{\mathbf{x}}f(\mathbf{x}_i).\tag{8}$$

3) TR Algorithm : In this algorithm a trust region with a maximum radius of $\bar{\Delta}$ is first defined, within which the following model can approximate the original objective function:

$$g_i(\mathbf{v})=f(\mathbf{x}_i)+\langle\nabla_{\mathbf{x}}f(\mathbf{x}_i),\mathbf{v}\rangle+\frac{1}{2}\langle\mathcal{B}[\mathbf{v}],\mathbf{v}\rangle,\tag{9}$$

where $\mathcal{B}$ is the Hessian matrix approximation. Then, according to the model depicted within the trust region, this algorithm takes a step forward. The search direction $\mathbf{v}$ is given by

$$\mathbf{v}_i=\underset{\mathbf{v}\in\mathbb{V}}{\operatorname{argmin}}\,g_i(\mathbf{v}),\tag{10}$$

where $\mathbb{V}$ is the trust region. If the cost function is not decreased, then the maximum radius is decreased, and the search direction and step size values are calculated again [24]. The function g is defined in every iteration i, and the set of search directions is calculated. After determining the search direction, the code matrix is updated according to equation (7), taking $\alpha_i=1$.

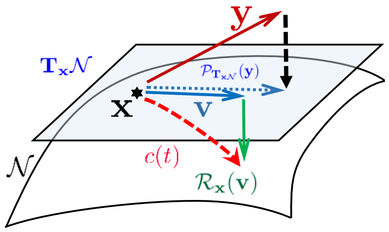

$\mathbf{T}_{\mathbf{x}}\mathcal{N}$
$\mathbf{y}$
$\mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\mathbf{y})$
$\mathbf{x}$
$\mathbf{v}$
$\mathcal{N}$
$c(t)$
$\mathcal{R}_{\mathbf{x}}(\mathbf{v})$

Fig. 1: Tangent space, Projection and Retraction operators

Global convergence is guaranteed by all the above methods. Global convergence means the local optimal point is achieved for each arbitrary initial condition [25]. GD has straightforward calculations, but its convergence rate is slow. BFGS requires various calculations, but its convergence rate is high. The BFGS method with limited memory (LBFGS) is proposed to simplify these calculations. This method reduces the computation required for large-scale problems by allocating a limited amount of memory [26]. The TR-based method has the advantages of the above methods. This method has high-speed convergence with quite simple calculations.

B. Optimization on Manifold

In this subsection, extensions of the above unconstrained optimization algorithms for optimization on Riemannian manifolds are described. At first, some required concepts of optimization on manifold are introduced.

Tangent Space: For any smooth manifold $\mathcal{N}$ in $\mathbb{C}^N$, the tangent space at $\mathbf{x}\in\mathcal{N}$ is defined as

$$\mathbf{T}_{\mathbf{x}}\mathcal{N}=\{\mathbf{v}\in\mathbb{C}^N:\mathbf{v}=c'(0)\},\tag{11}$$

where the smooth curve $c(t)$ can be considered as $c(t)=\mathbf{x}+t\mathbf{v},\forall\mathbf{x},\mathbf{v}\in\mathbb{C}^N$. The concept of tangent space is shown in Fig. 1. For smooth manifold $\mathcal{M}$ in (5), the tangent space at $(\mathbf{x}_1\in\mathcal{M}_1,\mathbf{x}_2\in\mathcal{M}_2,\dots,\mathbf{x}_M\in\mathcal{M}_M)$ is defined as

$$\mathbf{T}_{(\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_M)}\mathcal{M}=\mathbf{T}_{\mathbf{x}_1}\mathcal{M}_1\times\ldots\times\mathbf{T}_{\mathbf{x}_M}\mathcal{M}_M.\tag{12}$$

By endowing the tangent space with its inner product that varies smoothly with $\mathbf{X}$, we construct a Riemannian structure on the manifold, turning it into a Riemannian manifold. It is noteworthy that this structure suffices to define the gradient and Hessian on the manifold.

Projection: The operator $\mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\mathbf{y})$ whose concept we show in Fig. 1, projects any vector $\mathbf{y}$ in the smooth manifold $\mathcal{N}$ onto $\mathbf{T}_{\mathbf{x}}\mathcal{N}$ as

$$\mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\mathbf{y})=(\mathbf{I}_N-\mathbf{x}\mathbf{x}^T)\mathbf{y}.\tag{13}$$

This operator for any vector $\mathbf{x}\in\mathcal{M}_m$, and any vector $\mathbf{y}\in\mathbb{C}^N$ can be expressed as [27]

$$\mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{M}_m}(\mathbf{y})=\mathbf{y}-\text{Re}\{\mathbf{y}^*\odot\mathbf{x}\}\odot\mathbf{x}.\tag{14}$$

Retraction: This operator whose concept we show in Fig. 1, retracts any displacement along the direction of $\mathbf{V}$ to the smooth manifold $\mathcal{M}$ as [27]

$$\mathcal{R}_{\mathbf{X}}(\mathbf{V}) = (\mathbf{X}+\mathbf{V}) \odot \frac{1}{|\mathbf{X}+\mathbf{V}|}. \tag{15}$$

Riemannian Gradient: The Riemannian gradient of any function $f(\mathbf{x})$ at point $\mathbf{x} \in \mathcal{N}$ is the projection of Euclidean gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ onto $\mathbf{T}_{\mathbf{x}}\mathcal{N}$, i.e.,

$$\mathrm{grad} f(\mathbf{x}) = \mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\nabla_{\mathbf{x}} f(\mathbf{x})). \tag{16}$$

The Riemannian gradient of $f(\mathbf{X})$ can be written as

$$\mathrm{grad} f(\mathbf{X}) = \Big(\mathrm{grad}_1 f(\mathbf{X}), \dots, \mathrm{grad}_M f(\mathbf{X})\Big), \tag{17}$$

where $\mathrm{grad}_m f(\mathbf{X})$ with $\mathbf{x}_m \in \mathcal{M}_m$ for $m = 1, 2, \dots, M$ is attained by

$$\mathrm{grad}_m f(\mathbf{X}) = \mathcal{P}_{\mathbf{T}_{\mathbf{x}_m}\mathcal{M}_m}(\nabla_{\mathbf{x}_m} f(\mathbf{X})). \tag{18}$$

Riemannian Hessian: We assume $\mathcal{N}$ is a smooth manifold. The Riemannian Hessian of $f(\mathbf{x})$ at point $\mathbf{x} \in \mathcal{N}$ along the direction of $\mathbf{v} \in \mathbf{T}_{\mathbf{x}}\mathcal{N}$ is given by [22]

$$\begin{aligned} &\mathrm{Hess} f(\mathbf{x})[\mathbf{v}] = \mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\nabla_{\mathbf{x}}\mathrm{grad} f(\mathbf{x})[\mathbf{v}]) = \mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\nabla^2_{\mathbf{x}} f(\mathbf{x})[\mathbf{v}] \\ &- \langle \mathbf{v}, \nabla_{\mathbf{x}} f(\mathbf{x})\rangle_{\mathbf{x}}\mathbf{x} + \langle \mathbf{x}, \nabla^2_{\mathbf{x}} f(\mathbf{x})[\mathbf{v}]\rangle_{\mathbf{x}}\mathbf{x} - \langle \mathbf{x}, \nabla_{\mathbf{x}} f(\mathbf{x})\rangle_{\mathbf{x}}\mathbf{v}) \\ &= \mathcal{P}_{\mathbf{T}_{\mathbf{x}}\mathcal{N}}(\nabla^2_{\mathbf{x}} f(\mathbf{x})[\mathbf{v}]) - \langle \mathbf{x}, \nabla_{\mathbf{x}} f(\mathbf{x})\rangle_{\mathbf{x}}\mathbf{v}. \end{aligned} \tag{19}$$

The Riemannian Hessian of $f(\mathbf{X})$, with $\mathbf{X} \in \mathcal{M}$ for $m = 1, 2, \dots, M$ is obtained as

$$\begin{cases} \mathrm{Hess} f(\mathbf{X})[\mathbf{v}_1, \dots, \mathbf{v}_M] = (H_1, \dots, H_M), \\ H_m = \mathrm{Hess}_{\mathbf{x}_m} f(\mathbf{X})[\mathbf{v}_m] + \sum_{\substack{l=1 \\ l \neq m}}^{M} \nabla_{\mathbf{x}_m}\mathrm{grad}_{\mathbf{x}_l} f(\mathbf{X})[\mathbf{v}_l] \end{cases} \tag{20}$$

$\mathrm{Hess}_{\mathbf{x}_m} f(\mathbf{X})$ along the direction of $\mathbf{v}_m$ is attained as

$$\begin{aligned} \mathrm{Hess}_{\mathbf{x}_m} f(\mathbf{X})[\mathbf{v}_m] = \mathcal{P}_{\mathbf{T}_{\mathbf{x}_m}\mathcal{M}_m}(\nabla^2_{\mathbf{x}_m} f(\mathbf{X})[\mathbf{v}_m]) \\ - \langle \mathbf{x}_m, \nabla_{\mathbf{x}_m} f(\mathbf{X})\rangle_{\mathbf{x}_m}\mathbf{v}_m, \end{aligned} \tag{21}$$

and $\nabla_{\mathbf{x}_m}\mathrm{grad}_{\mathbf{x}_l} f(\mathbf{X})[\mathbf{v}_l]$ for any $m, l \in \{1, 2, \dots, M\}, l \neq m$ is given by

$$\begin{aligned} &\nabla_{\mathbf{x}_m}\mathrm{grad}_{\mathbf{x}_l} f(\mathbf{X})[\mathbf{v}_l] = \nabla_{\mathbf{x}_m}\{\mathcal{P}_{\mathbf{T}_{\mathbf{x}_l}\mathcal{M}_l}(\nabla_{\mathbf{x}_l} f(\mathbf{X}))[\mathbf{v}_l]\} \\ &= \nabla_{\mathbf{x}_m}\{\nabla_{\mathbf{x}_l} f(\mathbf{X}) - \langle \mathbf{x}_l, \nabla_{\mathbf{x}_l} f(\mathbf{X})\rangle_{\mathbf{x}_l}\mathbf{x}_l\}[\mathbf{v}_l] \\ &= \nabla_{\mathbf{x}_m}\nabla_{\mathbf{x}_l} f(\mathbf{X})[\mathbf{v}_l] - \langle \mathbf{x}_l, \nabla_{\mathbf{x}_m}\nabla_{\mathbf{x}_l} f(\mathbf{X})[\mathbf{v}_l]\rangle_{\mathbf{x}_l}\mathbf{x}_l \\ &= \mathcal{P}_{\mathbf{T}_{\mathbf{x}_l}\mathcal{M}_l}(\nabla_{\mathbf{x}_m}\nabla_{\mathbf{x}_l} f(\mathbf{X})[\mathbf{v}_l]), \end{aligned}$$

Optimization algorithms on Riemannian manifolds can be summarized in the following three steps, which should be performed in each iteration i.

1) Search direction and projection: Get the set of search directions $\mathbf{V}_i = [\mathbf{v}_{1,i}, \mathbf{v}_{2,i}, \dots, \mathbf{v}_{M,i}]$. Project any search direction $\mathbf{v}_m$ onto $\mathbf{T}_{\mathbf{x}_m}\mathcal{M}_m$ for $m = 1, 2, \dots, M$. The projected set of search directions is denoted by $\mathbf{PV}_i$.

2) Step size: Determine the step size α_i.

3) Update and retraction: Update the code matrix $\mathbf{X}_i$ as

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \alpha_i\mathbf{PV}_i. \tag{22}$$

The updated $\mathbf{X}_{i+1}$ in (22) is retracted to $\mathcal{M}$, i.e., $\mathbf{X}_{i+1} = \mathcal{R}_{\mathbf{X}_i}(\alpha_i\mathbf{PV}_i)$.

These steps are repeated until the algorithm converges. Point $\bar{\mathbf{X}}$ will be a local minimum for GD and BFGS algorithms if the condition $\mathrm{grad} f(\bar{\mathbf{X}}) = 0$ holds. In the TR algorithm, the condition $\mathrm{Hess} f(\bar{\mathbf{X}}) \succeq 0$ must also be satisfied [22].

Algorithm 1 RGD-MPSL Algorithm

```
Input: Riemannian manifold $\mathcal{M}$ (5), function $f(\mathbf{X})$ (3), $\mathbf{X}_0 \in \mathcal{M}$, $i = 0$, $\epsilon \ll 1$
Output: Optimum code matrix $\bar{\mathbf{X}}$ on $\mathcal{M}$ (5)
while $\|\mathrm{grad} f(\mathbf{X}_i)\| \geq \epsilon$ (17) do
    Search Direction and Projection: Calculate the search directions using equation (18)
        $\mathbf{v}_{m,i} = -\mathrm{grad}_m f(\mathbf{X}_i), m = 1, 2, \dots, M,$
    to form the set of search directions $\mathbf{V}_i$. $\mathbf{v}_{m,i}$ is on the tangent space $\mathbf{T}_{\mathbf{x}_m}\mathcal{M}_m$, so $\mathbf{PV}_i = \mathbf{V}_i$
    Step Size: Set $\alpha_i$ based on backtracking approach satisfying condition (23).
    Update and Retraction: Update the code matrix and retract it using the retraction equation (15)
        $\mathbf{X}_{i+1} = \mathcal{R}_{\mathbf{X}_i}(\alpha_i\mathbf{PV}_i)$.
    $i = i + 1$.
end while
```

IV. Proposed Method

In this section, we solve problem (4) with Riemannian manifold-based GD, LBFGS, and TR methods. All these algorithms follow three steps in Subsection III-B.

A. RGD Algorithm to Minimize PSL (RGD-MPSL)

In this subsection, we extend the RGD method in [28] to the product manifold of (5) to solve problem (4). This algorithm is presented in Algorithms 1. In each iteration i, the step size α_i is chosen based on the backtracking method. In this case, the condition (6) becomes

$$f(\mathbf{X}_i) - f(\mathcal{R}_{\mathbf{X}_i}(\alpha_i\mathbf{PV}_i)) \geq r\alpha_i\|\mathrm{grad} f(\mathbf{X}_i)\|^2. \tag{23}$$

The inner product on the product manifold $\mathcal{M}$ equals the sum of the inner products on each manifold $\mathcal{M}_m$, so $\|\mathrm{grad} f(\mathbf{X})\|^2$ can be written as [22]

$$\|\mathrm{grad} f(\mathbf{X})\|^2 = \sum_{m=1}^{M} \langle \mathrm{grad}_m f(\mathbf{X}), \mathrm{grad}_m f(\mathbf{X})\rangle_{\mathbf{X}}$$

B. LRBFGS Algorithm to Minimize PSL (LRBFGS-MPSL)

We extend the LRBFGS algorithm given in [29] to the product manifold of (5) to solve problem (4), and summarize it in Algorithms 2. The step size α_i is chosen similar to the RGD method. The function $\mathcal{T}_{\mathbf{v}}(\mathbf{u})$ in Algorithms 2 is defined as [30] $\mathcal{T}_{\mathbf{v}}(\mathbf{u}) = (\mathbf{I}_N - (\mathbf{x}+\mathbf{v})(\mathbf{x}+\mathbf{v})^T/\|\mathbf{x}+\mathbf{v}\|^2)\mathbf{u}$, where $\mathcal{N}$ is a smooth manifold, $\mathbf{x} \in \mathcal{N}$, and $\mathbf{v}, \mathbf{u} \in \mathbf{T}_{\mathbf{x}}\mathcal{N}$.

C. RTR Algorithm to Minimize PSL (RTR-MPSL)

In this section, we extend the RTR algorithm of [31] to the product manifold of (5) to solve problem (4). This algorithm is presented in Algorithms 3. The function g_i in iteration i is

Algorithm 2 LRBFGS-MPSL Algorithm

```
Input: Riemannian manifold $\mathcal{M}$ (5), function $f(\mathbf{X})$ (3), $\mathbf{X}_0 \in \mathcal{M}$, $i = 0$, $\epsilon \ll 1$, $s > 0$, , $l = 0$, $\gamma_0 = 1$ ,$0 < c_1 < \frac{1}{2} < c_2 < 1$
Output: Optimum code matrix $\bar{\mathbf{X}}$ on $\mathcal{M}$ (5)
while $\|\mathrm{grad} f(\mathbf{X}_i)\| \geq \epsilon$ (17) do
    Search Direction and Projection: Calculate the search directions $\mathbf{v}_{m,i}$ by following these steps to form the set of search directions $\mathbf{V}_i$:
        $\mathbf{q}_m \leftarrow \mathrm{grad}_m f(\mathbf{X}_i)$(18)
        for $j = i-1, i-2, \dots, l$ do
            $\lambda_j \leftarrow \rho_j \sum_{m=1}^{M} \langle \mathbf{s}_{m,j}^{(i)}, \mathbf{q}_m \rangle_{\mathbf{x}_{m,i}}$
            $\mathbf{q}_m \leftarrow \mathbf{q}_m - \lambda_j \mathbf{y}_{m,j}^{(i)}$
        end for
        $\mathbf{r}_m = \gamma_i \mathbf{q}_m$
        for $j = l, l+1, \dots, i-1$ do
            $\omega \leftarrow \rho_j \sum_{m=1}^{M} \langle \mathbf{y}_{m,j}^{(i)}, \mathbf{r}_m \rangle_{\mathbf{x}_{m,i}}$
            $\mathbf{r}_m \leftarrow \mathbf{r}_m + \mathbf{s}_{m,j}^{(i)}(\lambda_j - \omega)$
        end for
        $\mathbf{v}_{m,i} = -\mathbf{r}_m$
    Project any search direction $\mathbf{v}_m$ onto $\mathbf{T}_{\mathbf{x}_m}\mathcal{M}_m$ to form $\mathbf{PV}_i$.
    Step Size: Set $\alpha_i$ based on backtracking approach satisfying condition (23).
    Update and Retraction: Update the code matrix and retract it using the retraction equation (15) as
        $\mathbf{X}_{i+1} = \mathcal{R}_{\mathbf{X}_i}(\alpha_i \mathbf{PV}_i)$.
        $\mathbf{s}_{m,i}^{(i+1)} = \mathcal{T}_{\alpha_i \mathbf{v}_{m,i}}(\alpha_i \mathbf{v}_{m,i})$
        $\mathbf{y}_{m,i}^{(i+1)} = \mathrm{grad}_m f(\mathbf{X}_{i+1}) - \mathcal{T}_{\alpha_i \mathbf{v}_{m,i}} \mathrm{grad}_m f(\mathbf{X}_{i+1})$
        $\rho_i = \frac{1}{\sum_{m=1}^{M} \langle \mathbf{s}_{m,i}^{(i+1)}, \mathbf{y}_{m,i}^{(i+1)} \rangle_{\mathbf{x}_{m,i}}}$
        $\gamma_{i+1} = \frac{\sum_{m=1}^{M} \langle \mathbf{s}_{m,i}^{(i+1)}, \mathbf{y}_{m,i}^{(i+1)} \rangle_{\mathbf{x}_{m,i}}}{\sum_{m=1}^{M} \|\mathbf{y}_{m,i}^{(i+1)}\|^2}$
        $l = \max\{i - s, 0\}$
        for $m = 1, 2, \dots, M$ do
            add $\mathbf{s}_{m,i}^{(i+1)}, \mathbf{y}_{m,i}^{(i+1)}, \rho_i$ into storage
            if $i > s$ then
                discard $\mathbf{s}_{m,l-1}^{(i)}, y_{m,l-1}^{(i)}, \rho_{l-1}$ from storage
            end if
        end for
        for $j = l, l+1, \dots, i-1$ do
            transport $\mathbf{s}_{m,j}^{(i)}$ to get $\mathbf{s}_{m,j}^{(i+1)}$
            transport $\mathbf{y}_{m,j}^{(i)}$ to get $\mathbf{y}_{m,j}^{(i+1)}$
        end for
    $i = i + 1$;
end while
```

Algorithm 3 RTR-MPSL Algorithm

```
Input: Riemannian manifold $\mathcal{M}$ (5), function $f(\mathbf{X})$ (3), maximum radius $\bar{\Delta} > 0$, $\mathbf{X}_0 \in \mathcal{M}$, $i = 0$, $\epsilon \ll 1$, $\Delta_0 \in (0, \bar{\Delta})$, $\rho' \in (0, \frac{1}{4})$
Output: Optimum code matrix $\bar{\mathbf{X}}$ on $\mathcal{M}$ (5)
while $\|\mathrm{grad} f(\mathbf{X}_i)\| \geq \epsilon$ (17) do
    Search Direction and Projection: Define the function $g_i$ (24) using $\mathcal{B}_i$ equal to the Riemannian Hessian matrix (20). Solve subproblem (10) to get $\mathbf{V}_i$ using the tCG algorithm [31], then project any search direction $\mathbf{v}_m$ onto $\mathbf{T}_{\mathbf{x}_m}\mathcal{M}_m$ to form $\mathbf{PV}_i$.
    Step Size: Set $\alpha_i = 1$
    Update and Retraction:
        $\rho_i = \frac{f(\mathbf{X}_i) - f(\mathbf{X}_{i+1})}{g_i(0) - g_i(\mathbf{V}_i)}$
        $\mathbf{X}_{i+1} = \begin{cases} \mathcal{R}_{\mathbf{X}_i}(\mathbf{PV}_i)(15) & \text{if} \quad \rho_i > \rho' \quad \text{(accept)} \\ \mathbf{X}_i & \text{otherwise} \quad \text{(reject)} \end{cases}$
        $\Delta_{i+1} = \begin{cases} \frac{1}{4}\Delta_i & \text{if} \quad \rho_i < \frac{1}{4} \\ \min(2\Delta_i, \bar{\Delta}) & \text{if} \quad \rho_i > \frac{3}{4} \\ \Delta_i & \text{otherwise} \end{cases}$
    $i = i + 1$
end while
```

TABLE I: Computational complexity of the basic terms and the proposed algorithms

Term/Algorithm	Complexity
$f(\mathbf{X})$ (3)	$\mathcal{O}(\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$
$\mathrm{grad} f(\mathbf{X})$ (17)	$\mathcal{O}(\max\{MN^2, MN\log_2(\frac{p}{2})\})$
$\mathrm{Hess} f(\mathbf{X})[\mathbf{v}_1, \dots, \mathbf{v}_M]$ (20)	$\mathcal{O}(\max\{MN^2, MN\log_2(\frac{p}{2})\})$
Projection (14)	$\mathcal{O}(N)$
Projection (15)	$\mathcal{O}(N)$
RGD-MPSL	$\mathcal{O}(\mathcal{I}\mathcal{L}\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$
LRBFGS-MPSL	$\mathcal{O}(\mathcal{I}\mathcal{L}\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$
RTR-MPSL	$\mathcal{O}(\mathcal{I}\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$

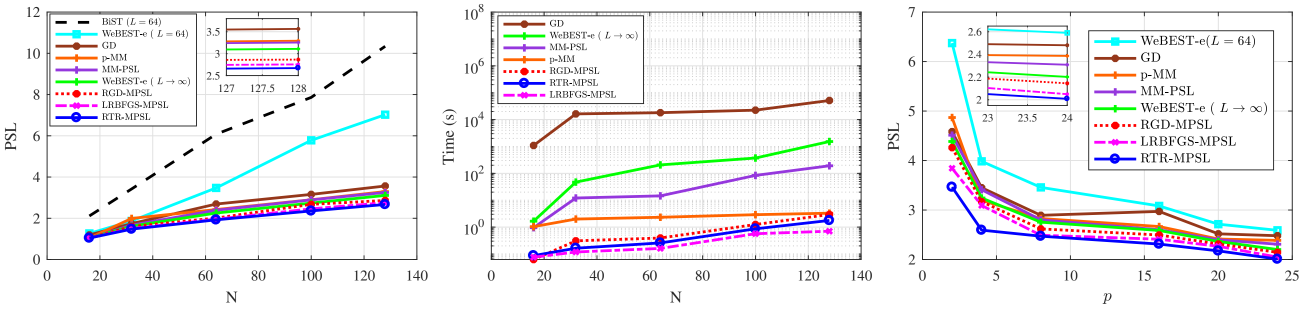
defined as

$$g_i(\mathbf{V}) = f(\mathbf{X}_i) + \sum_{m=1}^{M} \langle \mathrm{grad}_m f(\mathbf{X}_i), \mathbf{v}_m \rangle_{\mathbf{x}_{m,i}} + \frac{1}{2} \sum_{m=1}^{M} \langle \mathcal{B}_{m,i}[\mathbf{v}_m], \mathbf{v}_m \rangle_{\mathbf{x}_{m,i}}. \tag{24}$$

where $\mathcal{B}_{m,i}[\mathbf{v}_m]$ is the approximation of the Hessian matrix H_m in iteration i. The optimal set for search direction in iteration i is obtained by solving subproblem (10) and replacing $\mathbf{v}$ by $\mathbf{V}$.

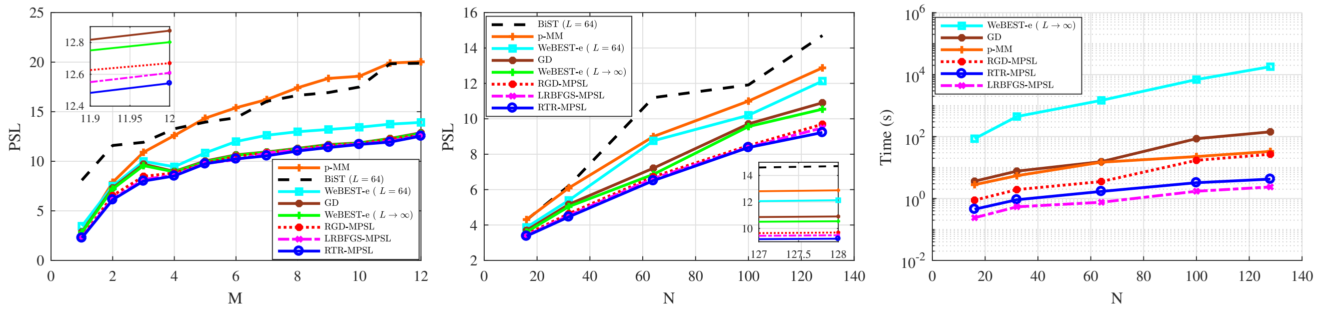
D. Computational Complexity

In this subsection, we analyze the computational complexity of the proposed algorithms. All the proposed algorithms go through three steps in each iteration: determining the search direction, adjusting the step size, and updating. The proposed algorithms should calculate the cost function, Riemannian gradient, Riemannian Hessian in the desired direction (for the RTR-MPSL algorithm), and projection and retraction operators in each step. The computational complexity of these terms is presented in Table I. The computational complexity of the RGD-MPSL algorithm is dominated by the backtracking method, which is used to determine the optimum step size. Condition (6) is checked in each step of this method, which requires the cost function in terms of the new α. If the backtracking method



(a) PSL vs. N $(M = 1, p = 8)$ (b) Time vs. N $(M = 1, p = 8)$ (c) PSL vs p $(M = 1,\ N = 100)$

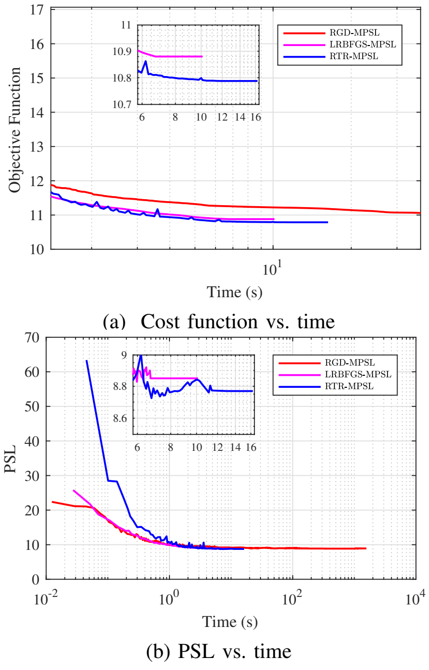
Fig. 2: PSL and Execution Time



(a) PSL vs. M $(N = 100, p = 8)$ (b) PSL vs. N $(M = 3, p = 8)$ (c) Time vs. N $(M = 3, p = 8)$

Fig. 3: PSL and Execution Time for different values of sequence length and antenna number

proceeds $\mathcal{L}$ steps to find the optimum α, then the per iteration computational complexity of the RGD-MPSL algorithm will be $\mathcal{O}(\mathcal{L}\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$. The LRBFGS-MPSL algorithm also uses the backtracking method in each iteration, whose computational complexity dominates the other steps of this algorithm. Therefore, each iteration of the LRBFGS-MPSL algorithm takes $\mathcal{O}(\mathcal{L}\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$ operations. RTR-MPSL algorithm uses the tCG method to find search directions, which takes $\mathcal{O}(N)$ operations. The computational complexity of the other steps of this algorithm are of the order of N. So, the computational complexity of the RTR-MPSL algorithm in each iteration is $\mathcal{O}(\max\{M^2N^2, M^2N\log_2(\frac{p}{2})\})$. Assuming each algorithm converges after $\mathcal{I}$ iterations, the computational complexity of each algorithm is given in Table I.



(a) Cost function vs. time

(b) PSL vs. time

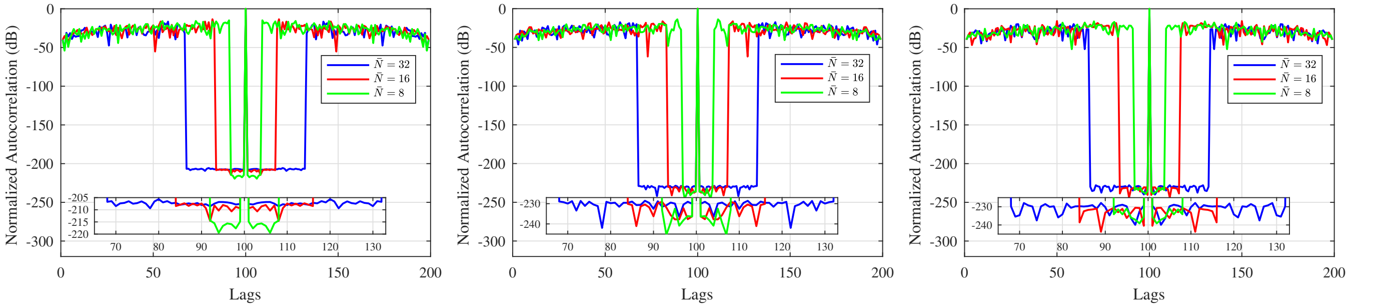
Fig. 4: Cost funtion and PSL versus time $(N = 128, M = 3, p = 24)$

V. Simulation Results

In this section, we will evaluate the performance of the proposed algorithms through some numerical simulations. All simulations were performed in MATLAB on a PC with Intel (R) Core (TM) i9-9900K CPU @ 3.60GHz and 64.00 GB of RAM. Each algorithm was repeated 100 times initializing with random sequences the average results are reported. For all algorithm, each entry of the random initial matrix $\mathbf{X}_0$ is chosen as $e^{j2\pi\theta}$, where θ is drawn randomly from a uniform distribution on $[0, 1]$, and we use the following convergence criterion:

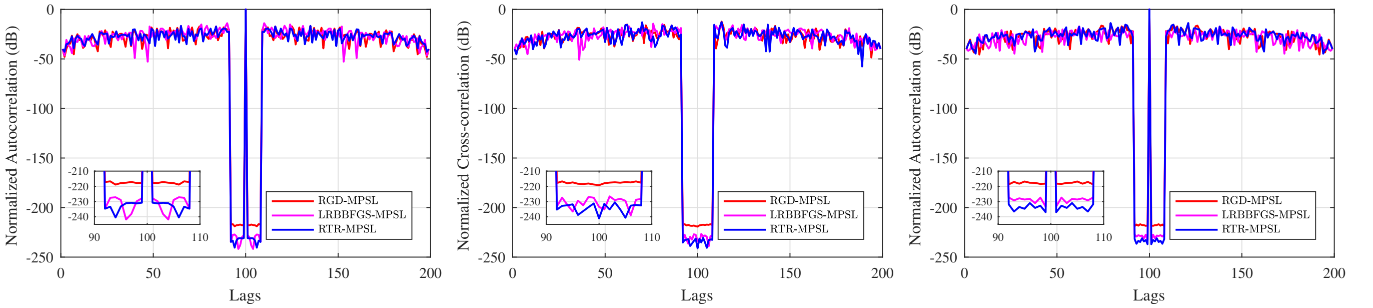
$$\frac{|f(\mathbf{X}_i) - f(\mathbf{X}_{i+1})|}{f(\mathbf{X}_i)} \leq \epsilon, \tag{25}$$

where $f(\mathbf{X}_i)$ is the cost function at iteration i and ϵ is set to 10^{-10}. In the LRBFGS-MPSL algorithm, the initial values of the parameters are chosen as $s = 30$, $c_1 = 10^{-4}$, and $c_2 = 0.999$. The initial parameter values of the RTR-MPSL algorithm are chosen as $\bar{\Delta} = \pi\sqrt{N}$, $\Delta_0 = \frac{\bar{\Delta}}{8}$, and $\rho' = 0.1$ [32], [33]. First, we consider the SISO case $(M = 1)$, and compare the performance of the proposed algorithms with state-of-the-art algorithms such as WeBEST (for alphabet size $L = 64$ and $L \to \infty$) [10], BiST (in PSL minimization mode, $L = 64$) [9], GD [8], MM-PSL [7], and p-MM [12]. The performance of the proposed algorithms is examined in terms of the achieved PSL value and the execution time. It should be noted that all algorithms are implemented for the continuous phase mode, except BiST $(L = 64)$ and WeBEST $(L = 64)$, in



(a) RGD-MPSL (b) LRBFGS-MPSL (c) RTR-MPSL

Fig. 5: The impact of weighting in the proposed algorithms ($M = 1, N = 100, p = 8$)



(a) The Autocorrelation of the $\mathbf{x}_1$ (b) The cross-correlation of $\mathbf{x}_1$ and $\mathbf{x}_2$ (c) The Autocorrelation of $\mathbf{x}_2$

Fig. 6: WPSL of the proposed algorithms ($M = 2, N = 100, p = 8, \bar{N} = 8$)

which a finite alphabet size is assumed. Therefore, comparing their convergence speed with these two algorithms is not fair. Fig. 2a shows the PSL vs. sequence length. It can be seen that the proposed algorithms outperform other algorithms, and RTR-MPSL yields the lowest PSL value. The execution time of the algorithms is given in Fig. 2b. This figure shows as the length of the sequence increases, the speed of convergence of all algorithms decreases. It can be seen here that the proposed algorithms outperform other algorithms and LRBFGS-MPSL has the lowest execution time (i.e., it is the fastest algorithm).

We also evaluate the performance of the proposed algorithms by increasing p. We use different values of p, to design a sequence of length $N = 100$. Fig. 2c shows the performance of the proposed algorithms compared to other methods. It should be noted that BiST uses the coordinate descent method with a finite alphabet size to directly minimize the PSL (not its l_p norm approximation), and thus is not included in this figure. It can be seen from this figure that the PSL decreases with increasing p for all algorithms, while the proposed algorithms perform best and among them RTR-MPSL gets the lowest PSL.

Now, we consider the MIMO case and compare the performance of the proposed algorithms with WeBEST (for alphabet size $L = 64$ and $L \to \infty$) [10], BiST (in PSL minimization mode, $L = 64$) [9], GD [8], and p-MM [12] in optimizing cost function (3). Figs. 3a and 3b show the PSL for different number of transmitting antennas and sequence lengths, respectively. Both figures show that PSL increases with increasing number of antennas and sequence length. In both cases, the proposed algorithms outperform other algorithms. In addition, RTR-MPSL has the best performance among the proposed algorithms. The execution time for the proposed algorithms, WeBEST ($L \to \infty$), and GD is given in Fig. 3c. This figure shows that the proposed algorithms are faster and, among them, LRBFGS-MPSL has the least execution time.

TABLE II: peak auto- and cross-correlation values for $M = 3, N = 64, p = 8$

	BiST ($L = 64$)	p-MM	WeBEST ($L = 64$)	GD	WeBEST ($L \to \infty$)	RGD	LRBFGS	RTR
$\mathbf{r}_{1,1}$	11.96	8.8012	8.6343	7.0411	6.1780	6.2780	6.4580	6.3080
$\mathbf{r}_{2,2}$	10.9475	8.6609	8.8600	7.2918	6.2130	6.7240	6.3640	6.1990
$\mathbf{r}_{3,3}$	10.1717	8.6302	8.4729	7.9076	6.2990	6.3900	6.5060	6.2160
$\mathbf{r}_{1,2}$	11.0105	8.8820	8.6940	7.1923	7.4300	6.3440	6.1910	6.4760
$\mathbf{r}_{1,3}$	10.3788	8.9023	8.6746	7.8723	7.7560	6.2910	6.4280	6.3050
$\mathbf{r}_{2,3}$	10.4823	8.6423	8.4414	7.0654	6.9960	6.2900	6.3280	6.4150

The convergence curve of the proposed algorithms for the cost function (3) and obtained PSL versus time is shown in Fig. 4. As can be seen, all three algorithms converge to a stationary point. The value of the cost function is always greater than the PSL, which is consistent with what we have stated.

The peak correlation values for autocorrelation and cross-correlation functions are presented in Tables II and III for different sequence sets. These tables show that the proposed algorithms outperform the state-of-the-art algorithms at the maximum sidelobe level. In addition, the maximum value of the sidelobe achieved by RTR-MPSL is lower than other algorithms.

Finally, we evaluate the performance of the proposed algorithms using the WPSL metric. We set the weights w_k to put nulls within the range $[-\bar{N}, \bar{N}]$ in the autocorrelation and

TABLE III: peak auto- and cross-correlation values for $M = 4, N = 128, p = 8$

	BiST $(L = 64)$	p-MM	WeBEST $(L = 64)$	GD	WeBEST $(L \to \infty)$	RGD	LRBFGS	RTR
$\mathbf{r}_{1,1}$	16.1009	13.3635	13.3635	12.9007	9.8440	10.0539	9.9533	9.9338
$\mathbf{r}_{2,2}$	17.6669	14.2406	13.7343	12.1023	10.0707	9.8713	9.7408	10.1987
$\mathbf{r}_{3,3}$	16.8498	14.3201	13.3043	12.6723	10.0840	10.8726	9.7865	9.8234
$\mathbf{r}_{4,4}$	17.2610	13.9834	13.3828	12.7012	10.2385	9.9048	9.7843	9.8295
$\mathbf{r}_{1,2}$	16.6916	13.8739	13.1791	12.6623	12.6396	9.6897	9.9026	10.4362
$\mathbf{r}_{1,3}$	16.7867	14.0239	13.7175	12.5098	11.5255	12.1684	11.7651	9.8893
$\mathbf{r}_{1,4}$	17.3235	13.9203	13.2978	12.9702	12.7536	10.0771	10.1509	9.8827
$\mathbf{r}_{2,3}$	16.9027	14.2212	13.2371	12.3049	12.2655	12.0634	10.5941	10.1970
$\mathbf{r}_{2,4}$	17.3537	13.8920	13.6826	12.8374	12.8339	9.9491	9.9807	10.2022
$\mathbf{r}_{3,4}$	16.9789	14.1123	13.3372	12.8734	10.7014	11.9665	11.8514	9.8202

cross-correlation functions as

$$\left\{ \begin{array}{ll} w_k = 1 & k = 0, 1, \ldots, \bar{N} \\ w_k = 0 & k = \bar{N} + 1, \ldots, N. \end{array} \right. \tag{26}$$

In Fig. 5, the impact of weighting with $M = 1, N = 100$ and different region of desired lags is depicted. As it can be seen from this figure, we get a deeper null for all algorithms by reducing the region of desired lags. For $M = 2$, we show the impact of weighting on the autocorrelation and cross-correlation functions of the proposed algorithms in Fig. 6. In this figure, we show normalized autocorrelation and cross-correlation values of two sequence in dB (i.e., as $20\log_{10}|\frac{w_k \mathbf{r}_{m,l}(k)}{N}|$) with weights in (26), $-N + 1 \leq k \leq N - 1$, $m, l \in \{1, 2\}$, and $N = 100, \bar{N} = 8$. It is noteworthy that RTR-MPSL reaches the lowest value of PSL in the desired lags and performs best among the proposed methods.

VI. Conclusion

We designed unimodular sequences with minimum PSL/WPSL in MIMO radar. The optimization problem of multi-sequence design within a continuous phase space was derived and to solve this optimization problem, numerical manifold-based algorithms were used. Three unconstrained optimization methods of GD, BFGS, and TR were extended to manifold-based ones, and the RGD-MPSL, LRBFGS-MPSL, and RTR-MPSL algorithms were presented. A suitable product manifold was chosen in the continuous phase space to implement the above algorithms. Simulation results were presented to evaluate the performance of the proposed algorithms and compare them with state-of-the-art algorithms. The results showed that the proposed algorithms achieve lower (weighted) side lobe levels at a higher convergence speed.

References

[1] S.W. Golomb, G. Gong, "Signal design for good correlation: for wireless communication, cryptography, and radar," in *Cambridge Univ. Press*, 2005 Jul 11.

[2] T. Long, Z. Liang, Q. Liu, "Advanced technology of high-resolution radar: target detection, tracking, imaging, and recognition," *Sci. China Inf. Sci.* 2019 Apr;62:1-26.

[3] J. Ruoskanen, P. Eskelinen, H. Heikkila, , "Millimeter wave radar with clutter measurements," *IEEE AES Magazine*, 2003 Oct;18(10):19-23.

[4] J. Fuchs, A. Dubey, M. Lübke, R. Weigel , F. Lurz, "Automotive radar interference mitigation using a convolutional autoencoder," in *IEEE Int. Radar Conf.*, 2020 Apr 28 (pp. 315-320). IEEE.

[5] H. Esmaeili-Najafabadi, M. Ataei , M.F. Sabahi, "Designing sequence with minimum PSL using Chebyshev distance and its application for chaotic MIMO radar waveform design," *IEEE Trans. Signal Process.*, 2016 Oct 26;65(3):690-704.

[6] L. Zhao, J. Song, P. Babu, D.P. Palomar, "A unified framework for low autocorrelation sequence design via majorizationminimization," *IEEE Trans. Signal Process.*, 2016 Oct 21;65(2):438-53.

[7] J. Song, P. Babu, D.P. Palomar, "Sequence design to minimize the weighted integrated and peak sidelobe levels," *IEEE Trans. Signal Process.*, 2015 Dec 22;64(8):2051-64.

[8] J.M. Baden, B. O'Donnell, L. Schmieder, "Multiobjective sequence design via gradient descent methods," *IEEE Trans. Aerosp. Electron. Syst.*, 2017 Dec 6;54(3):1237-52.

[9] M. Alaee-Kerahroodi, M. Modarres-Hashemi, M.M. Naghsh, "Designing sets of binary sequences for MIMO radar systems," *IEEE Trans. Signal Process.*, 2019 May 6;67(13):3347-60.

[10] E. Raei ,M. Alaee-Kerahroodi, P. Babu, MB. Shankar, "Generalized waveform design for sidelobe reduction in MIMO radar systems," *Signal Process.*, 2023 206:108914.

[11] S.P. Sankuru, and R. Jyothi, and P. Babu, and M. Alaee-Kerahroodi, "Designing sequence set with minimal peak side-lobe level for applications in high resolution RADAR imaging," *IEEE Open J. Signal Process.*, 2020 Dec 10;2:17-32.

[12] T. Liu, J. Sun, G. Wang, X. Du, W. Hu, "Designing Low Side-lobe Level Phase Coded Waveforms for MIMO Radar Using p-Norm Optimization," *IEEE Trans. Aerosp. Electron. Syst.*, 2023 Mar 30.

[13] S.B. Hanbali, "A review of radar signals in terms of Doppler tolerance, time-sidelobe level, and immunity against jamming," *Int. J. Microw. Wireless Technol.*, 2018 Dec;10(10):1134-42.

[14] S.D. Blunt, E.L. Mokole, "Overview of radar waveform diversity," *IEEE Trans. Aerosp. Electron. Syst.*, 2016 Nov;31(11):2-42.

[15] W. Huang, M.M. Naghsh, R. Lin, J. Li, "Doppler sensitive discrete-phase sequence set design for MIMO radar," *IEEE Trans. Aerosp. Electron. Syst.* 2020 May 13;56(6):4209-23.

[16] M. Alaee-Kerahroodi, P. Babu, M.Soltanalian, MB. Shankar, "Signal Design for Modern Radar Systems," Artech House, 2022.

[17] J. Xie, W. Zhou, G. Zhou, Y. Yuan, S. Li, "Tracking of range and azimuth for continuous imaging of marine target in monopulse ISAR with wideband echoes," *Int. J. Antennas Propag.*, 2016 Jan 1;2016.

[18] J. Scheer, W.A. Holm, "Principles of modern radar," USA:SciTech Publishing, 2010.

[19] H. Griffiths, L. Cohen, S. Watts, E. Mokole, C. Baker, M. Wicks, S. Blunt, "Radar spectrum engineering and management: Technical and regulatory issues," *Proc. IEEE*, 2014 Nov 26;103(1):85-102.

[20] J. J. Benedetto, I. Konstantinidis, M. Rangaswamy, "Phase-coded waveforms and their design," *IEEE Signal Process. Mag.*, 26(1), 22-31.

[21] P. Stoica, J. Li, M. Xue, "Transmit codes and receive filters for radar," *IEEE Signal Process. Mag.*, 25(6), 94-109.

[22] N. Boumal, "An introduction to optimization on smooth manifolds," Cambridge Univ. Press, 2023 Mar 16.

[23] T.T. Truong, H.T. Nguyen, "Backtracking gradient descent method and some applications in large scale optimisation. Part 2: Algorithms and experiments," *Appl. Math. Optim.*, 2021 Dec;84(3):2557-86.

[24] J.J. Moré, "Recent developments in algorithms and software for trust region methods," *Mathematical Programming*, The State of the Art: Bonn 1982. 1983:258-87.

[25] J. Nocedal, S.J. Wright, "Numerical optimization," New York, NY: Springer, New York; 1999 Aug 27.

[26] J.L. Morales, "A numerical study of limited memory BFGS methods," *Appl. Math. Lett.*, 2002 May 1;15(4):481-7.

[27] K. Alhujaili, V. Monga, M. Rangaswamy, "Transmit MIMO radar beampattern design via optimization on the complex circle manifold," *IEEE Trans. Signal Process.*, 2019 May 6;67(13):3561-75.

[28] S. Hosseini, W. Huang, R. Yousefpour, "Line search algorithms for locally Lipschitz functions on Riemannian manifolds," *SIAM J. Optim.*, 2018;28(1):596-619.

[29] X. Yuan, W. Huang, P.A. Absil, K.A. Gallivan, "A Riemannian limited-memory BFGS algorithm for computing the matrix geometric mean," *Procedia Comput. Sci.*, 2016 Jan 1;80:2147-57.

[30] P.A. Absil, R. Mahony, R. Sepulchre, "Optimization algorithms on matrix manifolds," Princeton University Press, Princeton, NJ, 2008.

[31] P.A. Absil, C.G. Baker, K.A. Gallivan, "Trust-region methods on Riemannian manifolds," *Found. Comput. Math.*, 2007 Jul;7:303-30.

[32] N. Boumal, P.A. Absil, "Low-rank matrix completion via preconditioned optimization on the Grassmann manifold," *Linear Algebra Appl.*, 2015 Jun 15;475:200-39.

[33] A.R. Conn, N.I Gould, P.L. Toint, "Trust region methods," SIAM, Philadelphia, PA, USA, 2000