

Article

A Rapid Convergent Low Complexity Interference Alignment Algorithm for Wireless Sensor Networks

Lihui Jiang ¹, Zhilu Wu ^{1,*}, Guanghui Ren ¹, Gangyi Wang ² and Nan Zhao ³

¹ School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China; E-Mails: jianglihui@hit.edu.cn (L.J.); rgh@hit.edu.cn (G.R.)

² School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China; E-Mail: WangGangyi@buaa.edu.cn

³ School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China; E-Mail: zhaonan@dlut.edu.cn

* Author to whom correspondence should be addressed; E-Mail: wuzhilu@hit.edu.cn; Tel./Fax: +86-451-8640-3135.

Academic Editor: Leonhard M. Reindl

Received: 9 April 2015 / Accepted: 23 July 2015 / Published: 29 July 2015

Abstract: Interference alignment (IA) is a novel technique that can effectively eliminate the interference and approach the sum capacity of wireless sensor networks (WSNs) when the signal-to-noise ratio (SNR) is high, by casting the desired signal and interference into different signal subspaces. The traditional alternating minimization interference leakage (AMIL) algorithm for IA shows good performance in high SNR regimes, however, the complexity of the AMIL algorithm increases dramatically as the number of users and antennas increases, posing limits to its applications in the practical systems. In this paper, a novel IA algorithm, called directional quartic optimal (DQO) algorithm, is proposed to minimize the interference leakage with rapid convergence and low complexity. The properties of the AMIL algorithm are investigated, and it is discovered that the difference between the two consecutive iteration results of the AMIL algorithm will approximately point to the convergence solution when the precoding and decoding matrices obtained from the intermediate iterations are sufficiently close to their convergence values. Based on this important property, the proposed DQO algorithm employs the line search procedure so that it can converge to the destination directly. In addition, the optimal step size can be determined analytically by optimizing a quartic function. Numerical results show that the proposed DQO algorithm can suppress the interference leakage more rapidly than the

traditional AMIL algorithm, and can achieve the same level of sum rate as that of AMIL algorithm with far less iterations and execution time.

Keywords: wireless sensor networks; multiple-input and multiple-output (MIMO); interference channel; interference alignment; line search; iterative algorithms

1. Introduction

Wireless sensor networks (WSNs) have received considerable attention, and have been applied to a large number of scenarios in recent years [1]. With the growth in network scale, interference management has become one of the major challenges in WSNs where multiple users often share some common resources simultaneously and high data rates are often demanded. A large amount of research has focused on interference management techniques in order to increase the sum rate of the system [2–5]. One recent interesting interference management scheme that can approach the channel capacity is interference alignment (IA), which was firstly studied by Maddah-Ali *et al.* [6] as well as Jafar and Shamai [7] for the multiple-input and multiple-output (MIMO) X channel. Subsequently, Cadambe and Jafar [8] proved that the throughput of the system varies linearly with the number of users in high signal-to-noise ratio (SNR) regimes by employing IA. In addition, the degrees of freedom (DoFs), which serve as one key measure of the channel capacity, were studied by Lee *et al.* [9] in the case of the MIMO Y channel and by Cadambe and Jafar [10] in the case of relay networks. Thanks to its bright prospects IA has been extended to use in many scenarios such as OFDM systems [11], femtocell networks [12], and cognitive networks [13,14]. In particular, Wu *et al.* [15] have leveraged the IA technique to mitigate the strong interference in large scale WSNs. In heterogeneous networks, Sharma *et al.* [16] have proposed a novel spectral coexistence mechanism which takes advantage of the IA technique.

Essentially, achieving IA is to design proper transmitting precoding matrices \mathbf{V} to align the desired signal and interference to different signal subspaces, and to design proper receiving decoding matrices \mathbf{U} to eliminate the interference at each receiver [17]. Tresch *et al.* [18] have developed closed-form solutions to IA for the MIMO interference channel with $M + 1$ users, one degree of freedom, M transmitting antennas, and M receiving antennas. However, the closed-form solutions to IA remain unknown except for some special situations [19]. Peters and Heath [20] have shown that it is difficult to formulate the closed-form solutions of \mathbf{V} and \mathbf{U} in the situation of more than three users. In addition, Razaviyayn *et al.* [21] have also shown that searching the optimal \mathbf{V} and \mathbf{U} is a nonconvex problem. Ma *et al.* [22] have performed a deep analysis on the computational complexity of interference alignment, and shown that the problem becomes NP-hard when the number of antennas at each node is larger than two. Therefore, many researchers [20–26] have focused on numerical methods to find the suboptimal \mathbf{V} and \mathbf{U} . Gomadam *et al.* [23] have leveraged the channel reciprocity and proposed the alternating minimization interference leakage (AMIL) and Max-SINR algorithms, which only require local channel state information (CSI). It has been shown by Xu *et al.* [27] that the AMIL and Max-SINR algorithms show close multiplexing gains in high SNR regimes and the former approach has lower complexity than the latter one. A similar alternating minimization IA algorithm has been proposed

in [20] without assuming the network reciprocity. Shen *et al.* [24] have developed the minimum mean square error (MMSE) IA algorithm for the case of imperfect CSI. A sequential antenna switch algorithm was proposed in [28], in which the quality of service of the IA system was considered.

Among these methods, the AMIL algorithm can suppress the interference leakage to a low level and thus can achieve high throughput in high SNR regimes where the sum rate of the system is determined by interference instead of noise. However, it might take a large number of iterations as well as long computational time in the case of large numbers of users and antennas. Furthermore, for the reason that CSI is time-variable, the practical IA system has limited computational time, which will certainly become the bottleneck of the implementation of AMIL algorithm when there are plenty of users and antennas. In this paper, we investigate the properties of the AMIL algorithm, and discover that the difference between the consecutive iteration results of the AMIL algorithm will approximately point to the convergence solution when the precoding and decoding matrices obtained from the intermediate iterations are sufficiently close to their convergence values. Based on this property, we propose a rapidly convergent low-complexity IA algorithm, *i.e.*, directional quartic optimal (DQO) algorithm. It leverages a line search (LS) optimization method, which iteratively generates searching directions and optimal step sizes. The searching direction is obtained by subtracting the consecutive results of AMIL algorithm, and the optimal step size is calculated by solving a quartic optimization problem.

The rest of the paper is organized as follows: in Section 2, we describe the system model. The properties of the AMIL algorithm are studied in Section 3, which will serve as the foundation of the algorithm proposed later. In Section 4, the DQO algorithm is proposed, and the corresponding procedure, optimal step size calculation, and complexity analysis are provided. Numerical results to evaluate the proposed algorithm are presented and discussed in Section 5. Finally, the paper is concluded in Section 6. As far as the notation used in the paper is concerned, we employ $(M \times N, d)^K$ to represent a K -user MIMO interference channel where each user wishes to transmit d data streams with M transmitting antennas and N receiving antennas. We use \mathbb{C} , \mathbb{R} , $\mathbb{C}^{M \times N}$, \mathbf{I} , and $\mathbf{CN}(\mu, \sigma^2)$ to represent the complex domain, the real domain, the $M \times N$ complex matrix, the identity matrix, and the complex Gaussian distribution with mean μ and variance σ^2 , respectively. $\text{Re}\{a\}$ denotes the real part of scalar a . \mathbf{A}^T , \mathbf{A}^* , \mathbf{A}^H , $\|\mathbf{A}\|$, $\text{Tr}[\mathbf{A}]$ and $\mathbf{A}^{*(l)}$ mean the transpose, the conjugate, the conjugate transpose, the Frobenius norm, the trace and the l -th column of matrix \mathbf{A} , respectively. $\text{eig}_l[\mathbf{A}]$ stands for the eigenvector associated with the l -th smallest eigenvalue of matrix \mathbf{A} . We employ \mathbf{A}^i to represent the value of \mathbf{A} at the i -th iterations. $\text{diag}(a_1, a_2, \dots, a_m)$ represents a diagonal matrix with its diagonal elements equal to a_1, a_2, \dots, a_m .

2. System Model

In this paper, the $(M \times N, d)^K$ MIMO interference channel is considered, and is depicted in Figure 1. The received signal of the k -th user can be represented as [23]:

$$\mathbf{Y}_k = \sum_{l=1}^K \mathbf{H}_{kl} \mathbf{V}_l \mathbf{s}_l + \mathbf{n}_k, k=1,2,\dots,K \quad (1)$$

where $\mathbf{s}_k \in \mathbb{C}^{d \times 1}$, $\mathbf{V}_k \in \mathbb{C}^{M \times d}$, and $\mathbf{Y}_k \in \mathbb{C}^{N \times 1}$ denote the data vector, the precoding matrix, and the received signal vector of the k -th user, respectively; $\mathbf{n}_k \in \mathbb{C}^{N \times 1}$ represents the noise vector with distribution of $\mathbf{CN}(0, \sigma^2)$ for each element; $\mathbf{H}_{kl} \in \mathbb{C}^{N \times M}$ represents the channel matrix from transmitter l

to receiver k . In the network we considered, CSI is obtained through CSI feedback, which serves as one critical technique in IA. A large number of researchers such as Ayach [29], Cho [30], and Zhang [31] have focused on the CSI feedback strategy. Since we mainly focus on designing the precoding and decoding matrices, further investigation on CSI feedback is beyond the scope of the paper and an accurate global CSI is assumed to be available at each node throughout this paper.

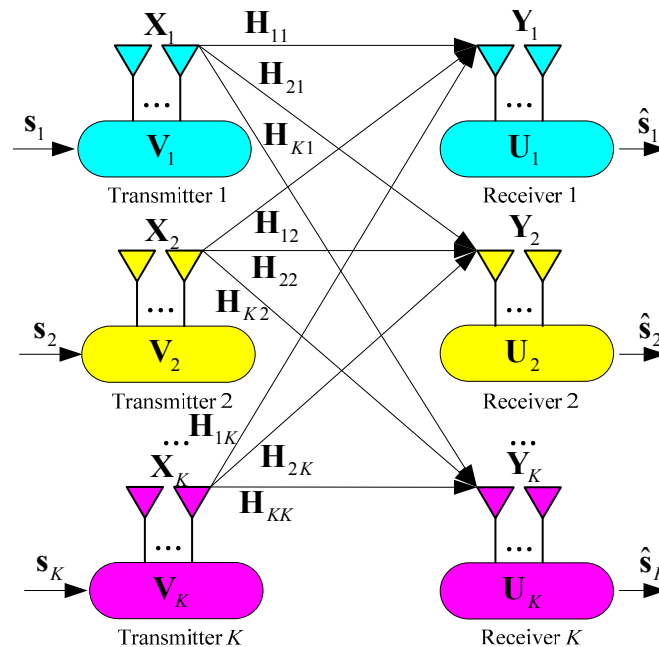


Figure 1. System model of the K -user MIMO interference channel.

The k -th user’s covariance matrices of the forward and reciprocal networks are respectively given by:

$$\mathbf{Q}_k = \sum_{l=1, l \neq k}^K \frac{P_l}{d} \mathbf{H}_{kl} \mathbf{V}_l \mathbf{V}_l^H \mathbf{H}_{kl}^H \tag{2}$$

$$\mathbf{Q}_k^{\leftarrow} = \sum_{l=1, l \neq k}^K \frac{\bar{P}_l}{d} \mathbf{H}_{lk}^H \mathbf{U}_l \mathbf{U}_l^H \mathbf{H}_{lk} \tag{3}$$

where P_l and \bar{P} denote the transmitting power of the l -th user in the forward and the reciprocal networks, respectively. Then the interference leakage at receiver k can be calculated as:

$$L_k = \text{Tr}[\mathbf{U}_k^H \mathbf{Q}_k \mathbf{U}_k] \tag{4}$$

where $\mathbf{U}_k \in \mathbb{C}^{N \times d}$ is the decoding matrix at receiver k . The total interference leakage is defined as the sum of the leakage at each receiver, *i.e.*,

$$L = \sum_{k=1}^K L_k \tag{5}$$

Then the minimal interference leakage problem is to design proper transmitting precoding matrices \mathbf{V} and receiving decoding matrices \mathbf{U} to minimize Equation (5) under the following constraints:

$$\text{rank}(\mathbf{U}_k^H \mathbf{H}_{kk} \mathbf{V}_k) = d \tag{6}$$

$$\mathbf{V}_k^H \mathbf{V}_k = \mathbf{I}, \mathbf{U}_k^H \mathbf{U}_k = \mathbf{I} \tag{7}$$

The former constraint guarantees the feasibility of the desired signal reconstruction and the later one ensures the uniqueness of the IA solution [17].

3. Properties of AMIL Algorithm

The AMIL algorithm, whose procedure is summarized in Algorithm 1, was proposed by Gomadam *et al.* [23] to find the IA solution by exploiting the channel reciprocity. In this section, we investigate the properties of the AMIL algorithm, and show that the difference of the consecutive iteration results of AMIL algorithm will approximately point to the convergence solution when the precoding and decoding matrices obtained from the intermediate iterations are sufficiently close to their convergence values. This interesting property will serve as the foundation of the algorithm proposed later.

Algorithm 1 AMIL Algorithm

1. Initialize L_{th} , I_{max} , and \mathbf{V}_k^0 , with $(\mathbf{V}_k^0)^H \mathbf{V}_k^0 = \mathbf{I}$ ($k = 1, 2, \dots, K$).
 2. $i = 0$.
 3. **Repeat**
 4. $i = i + 1$
 5. (VU-step): Calculate \mathbf{Q}_k^i according to Equation (2). Update \mathbf{U}_k^i so that
 6. $\mathbf{U}_{k(l)}^i = \text{eig}_l[\mathbf{Q}_k^i], l = 1, 2, \dots, d$.
 7. (UV-step): Calculate $\overleftarrow{\mathbf{Q}}_k^i$ according to Equation (3). Update \mathbf{V}_k^i so that
 8. $\mathbf{V}_{k(l)}^i = \text{eig}_l[\overleftarrow{\mathbf{Q}}_k^i], l = 1, 2, \dots, d$.
 9. Calculate the interference leakage L according to Equation (5).
 10. **Until** $L < L_{th}$ or $i > I_{max}$.
-

As shown in Algorithm 1, the AMIL algorithm calculates the optimal decoding matrices \mathbf{U} by fixing \mathbf{V} in the forward channel, and computes the optimal \mathbf{V} by fixing \mathbf{U} in the reciprocal channel. It can monotonously suppress the interference leakage, and has been proved to converge to a local minimum. Although the convergence point is not guaranteed to be the optimal IA solution, it has been verified numerically that in many cases AMIL algorithm can achieve a low interference leakage level after sufficient iterations. Therefore, in high SNR regimes, where the sum rate is limited by interference rather than noise, the AMIL algorithm serves as a good method to increase the sum rate. In addition, the procedure of alternating minimization is very effective, and is often employed by many other algorithms. In spite of this, the iterations as well as time required by AMIL algorithm to reach a certain level of interference leakage increase dramatically with the number of users and antennas. Therefore, the properties of AMIL algorithm are studied in order to figure out a way to increase the convergence rate and thus to reduce its complexity as well as the required computational time.

Throughout this paper, $\mathbf{V}_k^i \in \mathbb{C}^{M \times d}$ and $\mathbf{U}_k^i \in \mathbb{C}^{N \times d}$ are employed to represent the k -th user's precoding and decoding matrices obtained from the i -th iteration, respectively. The convergence value of \mathbf{V}_k^i and \mathbf{U}_k^i are defined as $\tilde{\mathbf{V}}_k$ and $\tilde{\mathbf{U}}_k$, respectively. Define $\Delta \mathbf{V}_k^i \in \mathbb{C}^{M \times d}$ as the deviation from \mathbf{V}_k^i to $\tilde{\mathbf{V}}_k$, and $\Delta \mathbf{U}_k^i \in \mathbb{C}^{N \times d}$ as the deviation from \mathbf{U}_k^i to $\tilde{\mathbf{U}}_k$ so that:

$$\mathbf{V}_k^i = \tilde{\mathbf{V}}_k + \Delta \mathbf{V}_k^i \tag{8}$$

$$\mathbf{U}_k^i = \tilde{\mathbf{U}}_k + \Delta \mathbf{U}_k^i \tag{9}$$

Define $\bar{\mathbf{V}}^i, \tilde{\mathbf{V}}, \Delta \bar{\mathbf{V}}^i \in \mathbb{C}^{MdK \times 1}$ and $\bar{\mathbf{U}}^i, \tilde{\mathbf{U}}, \Delta \bar{\mathbf{U}}^i \in \mathbb{C}^{NdK \times 1}$ as:

$$\bar{\mathbf{V}}^i = \begin{bmatrix} \mathbf{V}_{1(1)}^i \\ \mathbf{V}_{1(2)}^i \\ \dots \\ \mathbf{V}_{1(d)}^i \\ \dots \\ \mathbf{V}_{K(1)}^i \\ \mathbf{V}_{K(2)}^i \\ \dots \\ \mathbf{V}_{K(d)}^i \end{bmatrix}, \tilde{\mathbf{V}} = \begin{bmatrix} \tilde{\mathbf{V}}_{1(1)} \\ \tilde{\mathbf{V}}_{1(2)} \\ \dots \\ \tilde{\mathbf{V}}_{1(d)} \\ \dots \\ \tilde{\mathbf{V}}_{K(1)} \\ \tilde{\mathbf{V}}_{K(2)} \\ \dots \\ \tilde{\mathbf{V}}_{K(d)} \end{bmatrix}, \Delta \bar{\mathbf{V}}^i = \begin{bmatrix} \Delta \mathbf{V}_{1(1)}^i \\ \Delta \mathbf{V}_{1(2)}^i \\ \dots \\ \Delta \mathbf{V}_{1(d)}^i \\ \dots \\ \Delta \mathbf{V}_{K(1)}^i \\ \Delta \mathbf{V}_{K(2)}^i \\ \dots \\ \Delta \mathbf{V}_{K(d)}^i \end{bmatrix}, \bar{\mathbf{U}}^i = \begin{bmatrix} \mathbf{U}_{1(1)}^i \\ \mathbf{U}_{1(2)}^i \\ \dots \\ \mathbf{U}_{1(d)}^i \\ \dots \\ \mathbf{U}_{K(1)}^i \\ \mathbf{U}_{K(2)}^i \\ \dots \\ \mathbf{U}_{K(d)}^i \end{bmatrix}, \tilde{\mathbf{U}} = \begin{bmatrix} \tilde{\mathbf{U}}_{1(1)} \\ \tilde{\mathbf{U}}_{1(2)} \\ \dots \\ \tilde{\mathbf{U}}_{1(d)} \\ \dots \\ \tilde{\mathbf{U}}_{K(1)} \\ \tilde{\mathbf{U}}_{K(2)} \\ \dots \\ \tilde{\mathbf{U}}_{K(d)} \end{bmatrix}, \Delta \bar{\mathbf{U}}^i = \begin{bmatrix} \Delta \mathbf{U}_{1(1)}^i \\ \Delta \mathbf{U}_{1(2)}^i \\ \dots \\ \Delta \mathbf{U}_{1(d)}^i \\ \dots \\ \Delta \mathbf{U}_{K(1)}^i \\ \Delta \mathbf{U}_{K(2)}^i \\ \dots \\ \Delta \mathbf{U}_{K(d)}^i \end{bmatrix} \tag{10}$$

From Equations (8) and (9), it can be obtained that:

$$\bar{\mathbf{V}}^i = \tilde{\mathbf{V}} + \Delta \bar{\mathbf{V}}^i \tag{11}$$

$$\bar{\mathbf{U}}^i = \tilde{\mathbf{U}} + \Delta \bar{\mathbf{U}}^i \tag{12}$$

The investigation begins with the linear transformation property of AMIL algorithm as follows:

Theorem 1. *If AMIL algorithm can converge to a point where the interference leakage is very small, there exist fixed linear transformations \mathbf{T}_V and \mathbf{T}_U , which do not change with iterations. When $\bar{\mathbf{V}}^i$ and $\bar{\mathbf{U}}^i$ are sufficiently close to $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{U}}$, respectively, the deviations of the $(i + 1)$ -th iteration can be approximated as follows:*

$$\Delta \bar{\mathbf{V}}^{i+1} \approx \mathbf{T}_V \Delta \bar{\mathbf{V}}^i \tag{13}$$

$$\Delta \bar{\mathbf{U}}^{i+1} \approx \mathbf{T}_U \Delta \bar{\mathbf{U}}^i \tag{14}$$

Proof of Theorem 1. In the situation when AMIL algorithm converges to a point where the interference leakage is very small, which is easy to achieve for AMIL algorithm, we have:

$$\tilde{\mathbf{U}}_j^H \mathbf{H}_{ji} \tilde{\mathbf{V}}_i \approx \mathbf{0}, j \neq i \tag{15}$$

Define the interference covariance matrices at the convergence point as:

$$\tilde{\mathbf{Q}}_k = \sum_{j=1, j \neq k}^K \mathbf{H}_{kj} \tilde{\mathbf{V}}_j \tilde{\mathbf{V}}_j^H \mathbf{H}_{kj}^H \tag{16}$$

$$\tilde{\mathbf{Q}}_k = \sum_{j=1, j \neq k}^K \mathbf{H}_{jk}^H \tilde{\mathbf{U}}_j \tilde{\mathbf{U}}_j^H \mathbf{H}_{jk} \tag{17}$$

Assume that the l -th smallest eigenvalue and the associated eigenvector of $\tilde{\mathbf{Q}}_k$ are $\bar{\lambda}_{k,l}$ and $\bar{\mathbf{v}}_{k,l}$, respectively. Similarly for $\tilde{\mathbf{Q}}_k$, we have $\lambda_{k,l}$ and $\mathbf{v}_{k,l}$. From the procedure of AMIL algorithm, $\tilde{\mathbf{V}}_k$ and

$\tilde{\mathbf{U}}_k$ should be the eigenvectors corresponding to the d smallest eigenvalues of $\tilde{\mathbf{Q}}_k$ and $\tilde{\mathbf{Q}}_k$, respectively. Therefore, $\tilde{\mathbf{V}}_k$ and $\tilde{\mathbf{U}}_k$ can be respectively given by:

$$\tilde{\mathbf{V}}_k = [\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, \dots, \mathbf{v}_{k,d}] \quad (18)$$

$$\tilde{\mathbf{U}}_k = [\tilde{\mathbf{v}}_{k,1}, \tilde{\mathbf{v}}_{k,2}, \dots, \tilde{\mathbf{v}}_{k,d}] \quad (19)$$

At the $(i + 1)$ -th iteration, the VU-step of AMIL algorithm is performed on the precoding matrices $\{\mathbf{V}_k^i\}$ and covariance matrix at the $(i + 1)$ -th iteration can be expressed as:

$$\begin{aligned} \mathbf{Q}_k^{i+1} &= \sum_{j=1, j \neq k}^K \mathbf{H}_{kj} \mathbf{V}_j^i \mathbf{V}_j^{iH} \mathbf{H}_{kj}^H \\ &= \sum_{j=1, j \neq k}^K \mathbf{H}_{kj} [\tilde{\mathbf{V}}_j + \Delta \mathbf{V}_j^i] [\tilde{\mathbf{V}}_j + \Delta \mathbf{V}_j^i]^H \mathbf{H}_{kj}^H \\ &= \tilde{\mathbf{Q}}_k + \Delta \mathbf{Q}_k^{i+1} \end{aligned} \quad (20)$$

where $\Delta \mathbf{Q}_k^{i+1}$ is given by:

$$\Delta \mathbf{Q}_k^{i+1} = \sum_{j=1, j \neq k}^K \mathbf{H}_{kj} [\Delta \mathbf{V}_j^i \tilde{\mathbf{V}}_j^H + \tilde{\mathbf{V}}_j \Delta \mathbf{V}_j^{iH} + \Delta \mathbf{V}_j^i \Delta \mathbf{V}_j^{iH}] \mathbf{H}_{kj}^H \quad (21)$$

When $\bar{\mathbf{V}}^i$ and $\bar{\mathbf{U}}^i$ are sufficiently close to $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{U}}$ respectively, the deviation $\Delta \mathbf{V}_j^i$ is very small. Therefore, the high order term $\Delta \mathbf{V}_j^i \Delta \mathbf{V}_j^{iH}$ can be ignored, and Equation (21) can be simplified as:

$$\Delta \mathbf{Q}_k^{i+1} \approx \sum_{j=1, j \neq k}^K \mathbf{H}_{kj} [\Delta \mathbf{V}_j^i \tilde{\mathbf{V}}_j^H + \tilde{\mathbf{V}}_j \Delta \mathbf{V}_j^{iH}] \mathbf{H}_{kj}^H \quad (22)$$

According to the AMIL algorithm, \mathbf{U}_k^{i+1} should be the eigenvector corresponding to the d smallest eigenvalues of \mathbf{Q}_k^{i+1} . From Equations (19) and (20), using the Eigenvalue Perturbation Theory, the l -th column of \mathbf{U}_k^{i+1} can be approximated by:

$$\begin{aligned} \mathbf{U}_{k(l)}^{i+1} &\approx \tilde{\mathbf{v}}_{k,l} + \sum_{j=1, j \neq l}^N \frac{\tilde{\mathbf{v}}_{k,j}^H \Delta \mathbf{Q}_k^{i+1} \tilde{\mathbf{v}}_{k,l}}{\lambda_{k,l} - \lambda_{k,j}} \tilde{\mathbf{v}}_{k,j} \\ &= \tilde{\mathbf{U}}_{k(l)} + \sum_{j=1, j \neq l}^N \frac{\tilde{\mathbf{v}}_{k,j}^H \Delta \mathbf{Q}_k^{i+1} \tilde{\mathbf{U}}_{k(l)}}{\lambda_{k,l} - \lambda_{k,j}} \tilde{\mathbf{v}}_{k,j} \\ &= \tilde{\mathbf{U}}_{k(l)} + \Delta \mathbf{U}_{k(l)}^{i+1} \end{aligned} \quad (23)$$

where $\Delta \mathbf{U}_{k(l)}^{i+1}$ is given by:

$$\Delta \mathbf{U}_{k(l)}^{i+1} = \sum_{j=1, j \neq l}^N \frac{\tilde{\mathbf{v}}_{k,j}^H \Delta \mathbf{Q}_k^{i+1} \tilde{\mathbf{U}}_{k(l)}}{\lambda_{k,l} - \lambda_{k,j}} \tilde{\mathbf{v}}_{k,j} \quad (24)$$

Substitute Equation (22) into (24), and $\Delta \mathbf{U}_{k(l)}^{i+1}$ can be rewritten as:

$$\Delta \mathbf{U}_{k(l)}^{i+1} \approx \sum_{j=1, j \neq l}^N \frac{1}{\lambda_{k,l} - \lambda_{k,j}} \tilde{\mathbf{v}}_{k,j}^H \left\{ \sum_{r=1, r \neq k}^K \mathbf{H}_{kr} [\Delta \mathbf{V}_r^i \tilde{\mathbf{V}}_r^H + \tilde{\mathbf{V}}_r \Delta \mathbf{V}_r^{iH}] \mathbf{H}_{kr}^H \right\} \tilde{\mathbf{U}}_{k(l)} \tilde{\mathbf{v}}_{k,j} \quad (25)$$

From Equation (15), the terms $\tilde{\mathbf{V}}_r^H \mathbf{H}_{kr}^H \tilde{\mathbf{U}}_{k(l)}$ in Equation (25) is approximately zero and thus can be omitted. Therefore, Equation (25) can be simplified as:

$$\Delta \mathbf{U}_{k(l)}^{i+1} \approx \sum_{j=1, j \neq l}^N \frac{\tilde{\mathbf{v}}_{k,j}^H \sum_{r=1, r \neq k}^K \mathbf{H}_{kr} \tilde{\mathbf{V}}_r \Delta \mathbf{V}_r^{iH} \mathbf{H}_{kr}^H \tilde{\mathbf{U}}_{k(l)}}{\overline{\lambda_{k,l}} - \overline{\lambda_{k,j}}} \tilde{\mathbf{v}}_{k,j} \quad (26)$$

From Equation (26), it can be seen that $\Delta \mathbf{U}_{k(l)}^{i+1}$ is approximately the linear combination of the elements of $(\Delta \mathbf{V}_r^i)^*$, and the coefficients of the combination are corresponding to $\tilde{\mathbf{v}}_{k,j}^H$, \mathbf{H}_{kr} , $\tilde{\mathbf{V}}_r$, $\tilde{\mathbf{U}}_k$ and $\overline{\lambda_{k,j}}$, which are fixed and do not vary with iterations i . Hence there exists a fixed matrix $\mathbf{T}_{VU} \in \mathbb{C}^{NKd \times MKd}$ which does not change with iterations, so that:

$$\Delta \bar{\mathbf{U}}^{i+1} \approx \mathbf{T}_{VU} \cdot (\Delta \bar{\mathbf{V}}^i)^* \quad (27)$$

Similarly for the UV-step, there exists a fixed matrix $\mathbf{T}_{UV} \in \mathbb{C}^{MKd \times NKd}$ which does not change with iterations, so that:

$$\Delta \bar{\mathbf{V}}^{i+1} \approx \mathbf{T}_{UV} \cdot (\Delta \bar{\mathbf{U}}^{i+1})^* \quad (28)$$

From Equations (27) and (28), we have:

$$\Delta \bar{\mathbf{V}}^{i+1} \approx \mathbf{T}_{UV} \mathbf{T}_{VU}^* \cdot \Delta \bar{\mathbf{V}}^i = \mathbf{T}_V \Delta \bar{\mathbf{V}}^i \quad (29)$$

$$\Delta \bar{\mathbf{U}}^{i+1} \approx \mathbf{T}_{VU} \mathbf{T}_{UV}^* \cdot \Delta \bar{\mathbf{U}}^i = \mathbf{T}_U \Delta \bar{\mathbf{U}}^i \quad (30)$$

where:

$$\mathbf{T}_V = \mathbf{T}_{UV} \mathbf{T}_{VU}^* \in \mathbb{C}^{MKd \times MKd} \quad (31)$$

$$\mathbf{T}_U = \mathbf{T}_{VU} \mathbf{T}_{UV}^* \in \mathbb{C}^{NKd \times NKd} \quad (32)$$

It can be seen that \mathbf{T}_V and \mathbf{T}_U determine the properties of AMIL algorithm when the current point is sufficiently close to its convergence value. As the analytical forms of \mathbf{T}_V and \mathbf{T}_U are extremely complex, the features of these two transformations are investigated by simulation and one interesting property observed is given as follows:

Property 1. In the case of $(M \times M, d)^K$ channel, there exist two nonsingular matrices \mathbf{P}_V and \mathbf{P}_U , so that \mathbf{T}_V and \mathbf{T}_U can be expressed as:

$$\mathbf{T}_V = \mathbf{P}_V^{-1} \mathbf{A} \mathbf{P}_V \quad (33)$$

$$\mathbf{T}_U = \mathbf{P}_U^{-1} \mathbf{A} \mathbf{P}_U \quad (34)$$

where $\mathbf{A} = \text{diag}(\kappa_1, \kappa_2, \dots, \kappa_{MKd})$ with $\kappa_1 > \kappa_2 > \dots > \kappa_r$ and $\kappa_{r+1} = \kappa_{r+2} = \dots = \kappa_{MKd} = 0$ ($1 < r < MKd$).

Based on the properties above, we further study the situation when consecutive applications of \mathbf{T}_V and \mathbf{T}_U are exerted on the deviation, and come to the following theorem.

Theorem 2. In the $(M \times M, d)^K$ channel, if AMIL algorithm can converge to a point where the interference leakage is very small, the convergence solutions $\tilde{\mathbf{V}}_k$ and $\tilde{\mathbf{U}}_k$ ($k = 1, 2, \dots, K$) can be

approximated by the following Equations (35) and (36), when V_k^i and U_k^i are sufficiently close to \tilde{V}_k and \tilde{U}_k , respectively:

$$\tilde{V}_k \approx \mathbf{V}_k^i + t(\mathbf{V}_k^i - \mathbf{V}_k^{i-1}), t \in \mathbb{R} \quad (35)$$

$$\tilde{U}_k \approx \mathbf{U}_k^i + t(\mathbf{U}_k^i - \mathbf{U}_k^{i-1}) \quad (36)$$

Proof of Theorem 2. At the $(i + i_1)$ -th iteration, from Equations (29), (30), (33) and (34), we have:

$$\begin{aligned} \Delta \bar{\mathbf{V}}^{i+i_1} &\approx \mathbf{T}_V^{i_1} \Delta \bar{\mathbf{V}}^i \\ &= \mathbf{P}_V^{-1} \Lambda^{i_1} \mathbf{P}_V \Delta \bar{\mathbf{V}}^i \\ &= \mathbf{P}_V^{-1} \text{diag}(\kappa_1^{i_1}, \kappa_2^{i_1}, \dots, \kappa_{MKd}^{i_1}) \mathbf{P}_V \Delta \bar{\mathbf{V}}^i \end{aligned} \quad (37)$$

$$\begin{aligned} \Delta \bar{\mathbf{U}}^{i+i_1} &\approx \mathbf{T}_U^{i_1} \Delta \bar{\mathbf{U}}^i \\ &= \mathbf{P}_U^{-1} \Lambda^{i_1} \mathbf{P}_U \Delta \bar{\mathbf{U}}^i \\ &= \mathbf{P}_U^{-1} \text{diag}(\kappa_1^{i_1}, \kappa_2^{i_1}, \dots, \kappa_{MKd}^{i_1}) \mathbf{P}_U \Delta \bar{\mathbf{U}}^i \end{aligned} \quad (38)$$

Define $\mathbf{e}_k, \mathbf{g}_k \in \mathbb{C}^{MKd \times 1}$ and $f_k, s_k \in \mathbb{C}$ so that:

$$\mathbf{P}_V^{-1} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{MKd}] \quad (39)$$

$$\mathbf{P}_U^{-1} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{MKd}] \quad (40)$$

$$\mathbf{P}_V \Delta \bar{\mathbf{V}}^i = [f_1, f_2, \dots, f_{MKd}]^T \quad (41)$$

$$\mathbf{P}_U \Delta \bar{\mathbf{U}}^i = [h_1, h_2, \dots, h_{MKd}]^T \quad (42)$$

Thus Equations (37) and (38) can be written as:

$$\Delta \bar{\mathbf{V}}^{i+i_1} \approx \sum_{j=1}^{MKd} \kappa_j^{i_1} f_j \mathbf{e}_j \quad (43)$$

$$\Delta \bar{\mathbf{U}}^{i+i_1} \approx \sum_{j=1}^{MKd} \kappa_j^{i_1} h_j \mathbf{g}_j \quad (44)$$

According to Property 1, when i_1 is large enough, it can be obtained that:

$$\kappa_1^{i_1} \gg \kappa_2^{i_1} \gg \dots \gg \kappa_r^{i_1}, \kappa_{r+1}^{i_1} = \kappa_{r+2}^{i_1} = \dots = \kappa_{MKd}^{i_1} = 0 \quad (45)$$

Therefore, Equations (43) and (44) can be approximated by:

$$\Delta \bar{\mathbf{V}}^{i+i_1} \approx \kappa_1^{i_1} f_1 \mathbf{e}_1 \quad (46)$$

$$\Delta \bar{\mathbf{U}}^{i+i_1} \approx \kappa_1^{i_1} h_1 \mathbf{g}_1 \quad (47)$$

At the $(i + i_1 + 1)$ -th iteration, it can be obtained that:

$$\Delta \bar{\mathbf{V}}^{i+i_1+1} \approx \kappa_1^{i_1+1} f_1 \mathbf{e}_1 \approx \kappa_1 \Delta \bar{\mathbf{V}}^{i+i_1} \quad (48)$$

$$\Delta \bar{\mathbf{U}}^{i+i_1+1} \approx \kappa_1^{i_1+1} h_1 \mathbf{g}_1 \approx \kappa_1 \Delta \bar{\mathbf{U}}^{i+i_1} \quad (49)$$

From Equations (11), (12), (48) and (49), we have:

$$\tilde{\mathbf{V}} = \bar{\mathbf{V}}^{i+i_1+1} - \Delta \bar{\mathbf{V}}^{i+i_1+1} \approx \bar{\mathbf{V}}^{i+i_1+1} + \frac{\kappa_1}{1-\kappa_1} (\Delta \bar{\mathbf{V}}^{i+i_1+1} - \Delta \bar{\mathbf{V}}^{i+i_1}) \quad (50)$$

$$\tilde{\mathbf{U}} = \bar{\mathbf{U}}^{i+i_1+1} - \Delta \bar{\mathbf{U}}^{i+i_1+1} \approx \bar{\mathbf{U}}^{i+i_1+1} + \frac{\kappa_1}{1-\kappa_1} (\Delta \bar{\mathbf{U}}^{i+i_1+1} - \Delta \bar{\mathbf{U}}^{i+i_1}) \quad (51)$$

Define $t = \kappa_1/(1 - \kappa_1) \in \mathbb{R}$ and replace $i + i_1 + 1$ with i , it can be obtained that:

$$\begin{aligned} \tilde{\mathbf{V}} &\approx \bar{\mathbf{V}}^i + t(\Delta \bar{\mathbf{V}}^i - \Delta \bar{\mathbf{V}}^{i-1}) \\ &= \bar{\mathbf{V}}^i + t(\bar{\mathbf{V}}^i - \bar{\mathbf{V}}^{i-1}) \end{aligned} \quad (52)$$

$$\begin{aligned} \tilde{\mathbf{U}} &\approx \bar{\mathbf{U}}^i + t(\Delta \bar{\mathbf{U}}^i - \Delta \bar{\mathbf{U}}^{i-1}) \\ &= \bar{\mathbf{U}}^i + t(\bar{\mathbf{U}}^i - \bar{\mathbf{U}}^{i-1}) \end{aligned} \quad (53)$$

Substituting Equations (10) into Equations (52) and (53), then Equations (35) and (36) can be obtained. Therefore the difference between the consecutive iteration results of AMIL algorithm, *i.e.*, $\mathbf{V}_k^i - \mathbf{V}_k^{i-1}$ and $\mathbf{U}_k^i - \mathbf{U}_k^{i-1}$ will approximately point to their convergence values $\tilde{\mathbf{V}}_k$ and $\tilde{\mathbf{U}}_k$, when \mathbf{V}_k^i and \mathbf{U}_k^i are sufficiently close to $\tilde{\mathbf{V}}_k$ and $\tilde{\mathbf{U}}_k$, respectively.

Theorem 2 provides an enlightenment to reach the convergence point more rapidly. Instead of going along the circuitous route of VU-step and UV-step of AMIL algorithm, we can approach the destination more directly by searching along the new direction, *i.e.*, $\mathbf{V}_k^i - \mathbf{V}_k^{i-1}$ and $\mathbf{U}_k^i - \mathbf{U}_k^{i-1}$. This interesting discovery inspires us to propose the rapid convergent IA algorithm which will be shown in the next section.

4. Directional Quartic Optimal Algorithm

The traditional AMIL algorithm suffers from high complexity when there are plenty of users and antennas. The large number of iterations and long computational time required by the AMIL algorithm limit its application to practical IA systems where the CSI is always changing. Therefore, a high efficiency algorithm is needed to reduce the computational cost. In this section, we focus on a rapid convergent low-complexity IA approach, and propose the DQO algorithm. As shown in Section 3, the direction attained from the difference of the consecutive iteration results of the AMIL algorithm serves as a good searching direction, through which we can go to the convergence point almost directly. In addition, the optimal step size can be calculated by solving a quartic optimization problem. In this section, the details of the proposed algorithm are provided, and the associated computational complexity is analyzed.

4.1. The Procedure of DQO Algorithm

The procedure of DQO algorithm is summarized in Algorithm 2. The framework of DQO algorithm is based on the AMIL algorithm, and the LS procedure is added. In the initial phase of DQO algorithm, it operates just like the AMIL algorithm. The VU and UV steps are executed, and the interference leakage L is evaluated in each iteration. When L is smaller than the preset threshold th_1 , as shown in the 21-th line of Algorithm 2, the variable *flag* is set and the current number of iterations is recorded as i_b , indicating that the current point is close enough to the convergence value and the LS procedure can be

implemented. Notice that $flag$ and i_b are updated only once and will remain unchanged afterward. After $flag$ is set, the algorithm will execute the LS procedure every $interval$ iterations, as shown in the 10-th line. The LS procedure cannot be carried out in every iteration for the reason that the approximations Equations (46) and (47) can be attained only when i_1 is large enough.

Algorithm 2 DQO Algorithm

1. Initialize \mathbf{V}_k^0 , with $(\mathbf{V}_k^0)^H \mathbf{V}_k^0 = \mathbf{I}$ ($k = 1, 2, \dots, K$).
 2. Initialize parameters: th_1, th_2 ($th_2 < th_1$), I_{max} , and $interval$.
 3. Set $flag = 0$, $i = 0$, and $i_b = 0$.
 4. **Repeat**
 5. $i = i + 1$.
 6. (VU-step): Calculate \mathbf{Q}_k^i according to Equation (2). Update \mathbf{U}_k^i so that
 7. $\mathbf{U}_{k(l)}^i = \text{eig}_l[\mathbf{Q}_k^i], l = 1, 2, \dots, d$.
 8. (UV-step): Calculate $\overleftarrow{\mathbf{Q}}_k^i$ according to Equation (3). Update \mathbf{V}_k^i so that
 9. $\mathbf{V}_{k(l)}^i = \text{eig}_l[\overleftarrow{\mathbf{Q}}_k^i], l = 1, 2, \dots, d$.
 10. **If** $flag = 1$ and $(i - i_b) \bmod interval = 0$, execute the LS procedure as:
 11. (a) Calculate the new iteration direction as:
 12. $\Delta \mathbf{U}_k = \mathbf{U}_k^i - \mathbf{U}_k^{i-1}$
 13. $\Delta \mathbf{V}_k = \mathbf{V}_k^i - \mathbf{V}_k^{i-1}$
 14. (b) Calculate the optimal step size t^* .
 15. (c) Update \mathbf{V}_k^i and \mathbf{U}_k^i as:
 16. $\mathbf{U}_k^i \leftarrow \mathbf{U}_k^i + t^* \Delta \mathbf{U}_k$
 17. $\mathbf{V}_k^i \leftarrow \mathbf{V}_k^i + t^* \Delta \mathbf{V}_k$
 18. (d) Normalize each column of \mathbf{V}_k^i and \mathbf{U}_k^i .
 19. **else**
 20. Calculate the interference leakage L according to Equation (5)
 21. **If** $th_2 < L < th_1$ and $flag = 0$
 22. Set $flag = 1$ and $i_b = i$.
 23. **end if.**
 24. **end if.**
 25. **Until** $L < th_2$ or $i > I_{max}$.
-

Therefore, it has to take several UV and VU steps before κ_1^i takes the dominant proposition so that Equations (35) and (36) hold. As a result, the parameter $interval$ cannot be set to be one, and it will be further validated by simulation in Section 5.1. During each LS procedure, the new direction is calculated by subtracting the consecutive iteration results. The optimal step size t^* can be obtained by solving a quartic optimization problem, and the associated details will be provided in Section 4.2. The new precoders and decoders are updated as the 16-th and 17-th lines. To guarantee the unit norm of the precoders and decoders, we normalize each column of \mathbf{U} and \mathbf{V} in the 18-th line. The algorithm will stop

when the interference leakage is smaller than the objective threshold th_2 or the maximal iterations number I_{\max} is reached.

4.2. Optimal Step Size Calculation

The step size in DQO algorithm is an essential parameter which has a significant impact on the efficiency. Aiming at minimizing the interference leakage along the line search direction, the optimal step size can be calculated analytically by solving a quartic optimization problem. As is shown in Equations (35) and (36), the step size t should be a real value. From Equations (2), (4) and (5), the total interference leakage in the case of $\mathbf{U}_k + t\Delta\mathbf{U}_k$, $\mathbf{V}_k + t\Delta\mathbf{V}_k$ ($k = 1, 2, \dots, K$) can be formulated as:

$$L(t) = \sum_{k=1}^K \sum_{l=1, l \neq k}^K \frac{P_l}{d} \text{Tr}[(\mathbf{U}_k^H + t\Delta\mathbf{U}_k^H)\mathbf{H}_{kl}(\mathbf{V}_l + t\Delta\mathbf{V}_l)(\mathbf{V}_l^H + t\Delta\mathbf{V}_l^H)\mathbf{H}_{kl}^H(\mathbf{U}_k + t\Delta\mathbf{U}_k)] \quad (54)$$

Without loss of generality, we assume $P_l/d = 1$ and define:

$$\boldsymbol{\alpha}_{kl} = \Delta\mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l \quad (55)$$

$$\boldsymbol{\beta}_{kl} = \Delta\mathbf{U}_k^H \mathbf{H}_{kl} \Delta\mathbf{V}_l \quad (56)$$

$$\boldsymbol{\gamma}_{kl} = \mathbf{U}_k^H \mathbf{H}_{kl} \Delta\mathbf{V}_l \quad (57)$$

$$\boldsymbol{\delta}_{kl} = \mathbf{U}_k^H \mathbf{H}_{kl} \mathbf{V}_l \quad (58)$$

Then Equation (54) can be formulated as a real quartic function of t :

$$L(t) = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (59)$$

where:

$$a_4 = \sum_{k=1}^K \sum_{l=1, l \neq k}^K \text{Tr}[\boldsymbol{\beta}_{kl} \boldsymbol{\beta}_{kl}^H] \quad (60)$$

$$a_3 = \sum_{k=1}^K \sum_{l=1, l \neq k}^K 2\text{Re}\{\text{Tr}[\boldsymbol{\alpha}_{kl} \boldsymbol{\beta}_{kl}^H + \boldsymbol{\gamma}_{kl} \boldsymbol{\beta}_{kl}^H]\} \quad (61)$$

$$a_2 = \sum_{k=1}^K \sum_{l=1, l \neq k}^K \{\text{Tr}[\boldsymbol{\alpha}_{kl} \boldsymbol{\alpha}_{kl}^H + \boldsymbol{\gamma}_{kl} \boldsymbol{\gamma}_{kl}^H] + 2\text{Re}\{\text{Tr}[\boldsymbol{\alpha}_{kl} \boldsymbol{\gamma}_{kl}^H + \boldsymbol{\beta}_{kl} \boldsymbol{\delta}_{kl}^H]\}\} \quad (62)$$

$$a_1 = \sum_{k=1}^K \sum_{l=1, l \neq k}^K 2\text{Re}\{\text{Tr}[\boldsymbol{\delta}_{kl} \boldsymbol{\alpha}_{kl}^H + \boldsymbol{\delta}_{kl} \boldsymbol{\gamma}_{kl}^H]\} \quad (63)$$

$$a_0 = \sum_{k=1}^K \sum_{l=1, l \neq k}^K \text{Tr}[\boldsymbol{\delta}_{kl} \boldsymbol{\delta}_{kl}^H] \quad (64)$$

In order to obtain the global minimum value of Equation (59), the following lemma is firstly introduced:

Lemma 1. For a real quartic function $f(t) = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$, $t \in \mathbb{R}$, with $a_4 > 0$, there exists an optimal $t^* \in \mathbb{R}$ that satisfies $f'(t^*) = 0$, so that $f(t^*)$ is the global minimal value of $f(t)$.

Proof of Lemma 1. As $a_4 > 0$ and $f'(t) = 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1$, it can be obtained that $\lim_{t \rightarrow +\infty} f'(t) = +\infty$ and $\lim_{t \rightarrow -\infty} f'(t) = -\infty$. Therefore, there exist $a, b \in \mathbb{R}$, so that $f'(t) > 0, t \in [b, \infty)$ and $f'(t) < 0, t \in (-\infty, a]$. Thus, $f(t)$ increases monotonously in $t \in [b, \infty)$ and has the minimal value $f(b)$ at $t = b$. Similarly, $f(t)$ decreases monotonously in $t \in (-\infty, a]$ and has the minimal value $f(a)$ at $t = a$. When $t \in [a, b]$, due to continuity, $f(t)$ must have the minimal value $f(c)$ at $t = c$ where $c \in [a, b]$. Hence, $\min\{f(a), f(b), f(c)\}$ is the global minimum of $f(t)$. In conclusion, there exists $t^* \in \mathbb{R}$ (or $t^* \in \{a, b, c\}$) so that $f(t^*)$ is the global minimal value. For $f(t) \geq f(t^*)$, the left and right derivatives at t^* can be formulated as:

$$f'_-(t^*) = \lim_{\Delta t \rightarrow 0^-} \frac{f(t^* + \Delta t) - f(t^*)}{\Delta t} \leq 0$$

$$f'_+(t^*) = \lim_{\Delta t \rightarrow 0^+} \frac{f(t^* + \Delta t) - f(t^*)}{\Delta t} \geq 0$$

For $f'(t^*) = f'_-(t^*) = f'_+(t^*)$, we must have $f'(t^*) = f'_-(t^*) = f'_+(t^*) = 0$. Therefore, the global optimal value t^* must be chosen from the solutions that satisfy $f'(t^*) = 0$. When $f'(t^*) = 0$ has only one real solution, it is the optimal t^* . When $f'(t^*) = 0$ has 2 or 3 real solutions, t^* is the one that has the smallest function value.

From Equation (60) we have $a_4 > 0$. According to Lemma 1, it can be deduced that there exists $t^* \in \mathbb{R}$ with zero derivative, which makes $L(t^*)$ as the global minimal value of Equation (59). Let:

$$dL/dt = 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 = 0 \quad (65)$$

The discriminant of Equation (65) is: $\Delta = \Delta_1^2 + \Delta_2^3$, where $\Delta_1 = (bc)/(6a^2) - b^3/(27a^3) - d/(2a)$, $\Delta_2 = c/(3a) - b^2/(9a^2)$, $a = 4a_4$, $b = 3a_3$, $c = 2a_2$, and $d = a_1$. And the 3 solutions of Equation (65) are given by:

$$t_1 = x_1 + x_2 + x_3 \quad (66)$$

$$t_2 = x_1 + \frac{(-1 + \sqrt{3}i)x_2}{2} + \frac{(-1 - \sqrt{3}i)x_3}{2} \quad (67)$$

$$t_3 = x_1 + \frac{(-1 - \sqrt{3}i)x_2}{2} + \frac{(-1 + \sqrt{3}i)x_3}{2} \quad (68)$$

where $x_1 = -b/(3a)$, $x_2 = \sqrt[3]{\Delta_1 + \sqrt{\Delta}}$, and $x_3 = \sqrt[3]{\Delta_1 - \sqrt{\Delta}}$. When $\Delta > 0$ or $\Delta = \Delta_1 = \Delta_2 = 0$, there is only one real root t_1 which is the optimal t^* ; when $\Delta = 0$ and $\Delta_1 = \Delta_2 \neq 0$, there are 2 real roots; when $\Delta < 0$, there are 3 real roots. If there are more than one real root, t^* is the one that has the smallest function value.

4.3. Computational Complexity Analysis

The computational complexity of the AMIL and DQO algorithms is analyzed according to the number of complex multiplications (NoCM). For the AMIL algorithm, the complexity comes from the covariance matrices calculation, eigenvalue decomposition, and interference leakage evaluation with the complexity of $K(K-1)(MNd + Md^2 + Nd^2)$, $9K(M^3 + N^3)$, and $K(K-1)(d^2M + d^3)$, respectively [17]. Therefore the NoCM per iteration of the AMIL algorithm is summarized as:

$$C_{\text{AMIL}} = K(K-1)(2MNd + Nd^2 + M^2d) + 9K(N^3 + M^3) + K(K-1)(d^2M + d^3) \quad (69)$$

For the DQO algorithm, the complexity of the VU-step, UV-step, and interference leakage calculations are the same as those of the AMIL algorithm, and the extra complexity comes from the LS procedure. The complexity of LS comes from the coefficients calculation of Equations (55)–(58) and Equations (60)–(63), solving Equation (65), and normalization. The complexity of each LS procedure is analyzed as follows:

(1) As shown in Equation (55), \mathbf{a}_{kl} can be rewritten as $\mathbf{a}_{kl} = [(\mathbf{U}_k^i)^H \mathbf{H}_{kl} - (\mathbf{U}_k^{i-1})^H \mathbf{H}_{kl}] \mathbf{V}_l$. Since $(\mathbf{U}_k^i)^H \mathbf{H}_{kl}$ and $(\mathbf{U}_k^{i-1})^H \mathbf{H}_{kl}$ have been calculated in $\bar{\mathbf{Q}}_k$ as shown in Equation (3), the complexity of computing \mathbf{a}_{kl} only comes from the multiplication of $[(\mathbf{U}_k^i)^H \mathbf{H}_{kl} - (\mathbf{U}_k^{i-1})^H \mathbf{H}_{kl}]$ and \mathbf{V}_l , with the NoCM of d^2M . As k and l traverse from 1 to K with $l \neq k$, the total complexity of calculating all \mathbf{a}_{kl} is $K(K-1)d^2M$. Similarly, $\mathbf{\beta}_{kl}$, $\mathbf{\gamma}_{kl}$, and $\mathbf{\delta}_{kl}$ have the same complexity as \mathbf{a}_{kl} .

(2) The complexity of calculating a_1 – a_4 mainly comes from the multiplications among \mathbf{a}_{kl} , $\mathbf{\beta}_{kl}$, $\mathbf{\gamma}_{kl}$, and $\mathbf{\delta}_{kl}$. Notice that only the traces of the products are required, and we don't have to compute all the elements of the products. Therefore, the complexity of calculating a_1 – a_4 are $2K(K-1)d^2$, $4K(K-1)d^2$, $2K(K-1)d^2$, and $K(K-1)d^2$, respectively. Notice that there is no need to calculate a_0 .

(3) The number of real solutions of Equation (65) depends on Δ_1 , Δ_2 as well as Δ , and we consider the most complex case of 3 real solutions. The details of the complexity of solving the cubic equation are listed in Table 1. NoRM, NoRD, NoSRC, and NoCRC are employed to represent the number of real multiplications, real divisions, square root calculations, and cubic root calculations, respectively.

Table 1. Complexity of solving the cubic Equation (65).

Types	NoRM	NoRD	NoSRC	NoCRC
a, b, c, d	3	0	0	0
Δ_1	9	3	0	0
Δ_2	4	2	0	0
Δ	3	0	0	0
x_1	1	1	0	0
x_2	0	0	1	1
x_3	0	0	1	1
t_1	0	0	0	0
t_2	8	0	0	0
t_3	8	0	0	0
$L(t)$	21	0	0	0
Total	57	6	2	0

(4) \mathbf{V}_k^i and \mathbf{U}_k^i are normalized so that the norm of each column is one. The normalization of one column of \mathbf{V}_k^i takes M complex multiplications, one square root calculation and two real divisions. Therefore the normalization of all the precoding matrices takes dMK complex multiplications, dK square root calculations, and $2dK$ real divisions. Similarly, the normalization of all the decoding matrices takes dNK complex multiplications, dK square root calculations, and $2dK$ real divisions.

(5) As the complexity of one complex multiplication equals four real multiplications, the number of real multiplications will be replaced with the equivalent number of complex multiplications. And the complexity of one line search is summarized in Table 2.

Therefore the total number of equivalent complex multiplications of one line search is:

$$C_{LS} = 4K(K-1)d^2M + 9K(K-1)d^2 + dK(M+N) + 14.25 \quad (70)$$

And the number of real divisions, square root calculations and cubic root calculations in one LS are $4dK + 6$, $2dK + 2$, and 2, respectively. As LS is implemented every *interval* iteration, the average NoCM per iteration of DQO algorithm is:

$$C_{DQO} = C_{AMIL} + C_{LS} / interval \quad (71)$$

Table 2. Complexity of one line search.

Types	NoRM	NoRD	NoSRC	NoCRC
α_{kl}	$K(K-1)d^2M$	0	0	0
β_{kl}	$K(K-1)d^2M$	0	0	0
γ_{kl}	$K(K-1)d^2M$	0	0	0
δ_{kl}	$K(K-1)d^2M$	0	0	0
a_1	$2K(K-1)d^2$	0	0	0
a_2	$4K(K-1)d^2$	0	0	0
a_3	$2K(K-1)d^2$	0	0	0
a_4	$K(K-1)d^2$	0	0	0
Solve (65)	14.25	6	2	2
Normalization	$dK(M+N)$	$4dK$	$2dK$	0

The complexity of the AMIL and DQO algorithms with the parameter *interval* = 20 (the mechanism for determining the parameter will be provided in Section 5.1) is compared in Table 3. From the comparison, it can be seen that the average NoCM of the DQO algorithm per iteration is only slightly higher than that of the AMIL algorithm. Although LS will bring extra complexity per iteration, it can reduce the number of iterations significantly and thus lower the overall computational complexity, which will be shown in the numerical results.

Table 3. Complexity of AMIL and DQO algorithms in $(M \times M, d)^K$ with *interval* = 20.

$(M \times N, d)^K$	C_{AMIL}	C_{DQO}	C_{DQO}/C_{AMIL}
$(6 \times 6, 1)^{11}$	5.61×10^4	5.63×10^4	1.003
$(6 \times 6, 2)^5$	2.49×10^4	2.50×10^4	1.006
$(7 \times 7, 1)^{13}$	1.06×10^5	1.06×10^5	1.003
$(7 \times 7, 2)^6$	4.78×10^4	4.80×10^4	1.005
$(8 \times 8, 1)^{15}$	1.82×10^5	1.83×10^5	1.002
$(8 \times 8, 2)^7$	8.37×10^4	8.40×10^4	1.004
$(9 \times 9, 1)^{17}$	2.94×10^5	2.95×10^5	1.002
$(9 \times 9, 2)^8$	1.37×10^5	1.37×10^5	1.004
$(10 \times 10, 1)^{19}$	4.52×10^5	4.53×10^5	1.002
$(10 \times 10, 2)^9$	2.12×10^5	2.12×10^5	1.003

5. Numerical Results

In this section, the performance of the proposed DQO algorithm is evaluated and compared with the traditional AMIL algorithm by simulation. We will employ the interference leakage and sum rate as the performance metrics. The definition of interference leakage is shown in Equation (5), and that of sum

rate can be found in reference [23]. Unless specially specified, 250 realizations of different channel coefficients with distribution $\mathbf{CN}(0, 1)$ and the corresponding initial precoding matrices are randomly generated to evaluate the average interference leakage and sum rate of the algorithms in each simulation.

5.1. Parameter Analysis

There are four parameters in the DQO algorithm: the interference leakage threshold for starting LS th_1 , the desired interference leakage th_2 , the interval for executing the LS *interval*, and the maximum iteration I_m . The parameters th_2 and I_m are set according to the practical requirements. The other two have impacts on the efficiency of DQO algorithm. It is difficult to provide the exact optimal values of the two parameters analytically. Therefore, simulations are employed to determine the parameters th_1 and *interval* in this section.

The parameter th_1 decides the interference leakage threshold for starting line search. It is difficult to provide the exact optimal value of th_1 , and the performance of DQO algorithm is evaluated with different th_1 for the $(5 \times 5, 2)^4$ and $(10 \times 10, 1)^{19}$ channels in Figures 2 and 3, respectively. The parameter *interval* is preset as 20, and th_1 is chosen to be 1, 0.1, 0.01, and 0.001.

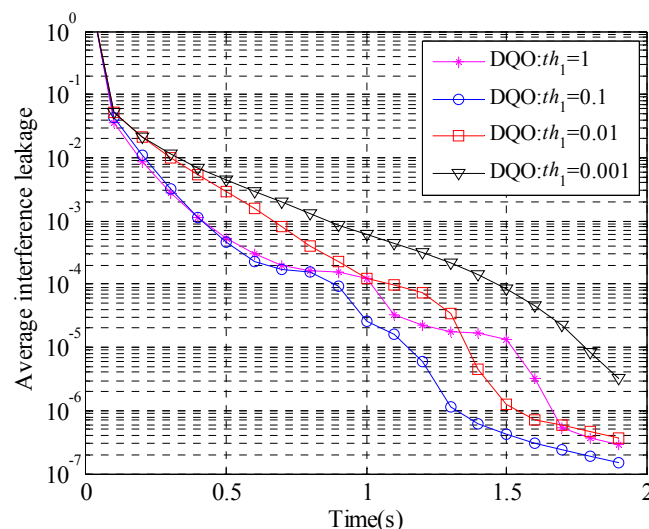


Figure 2. Convergence curves of the average interference leakage of DQO algorithm with $th_1 = 1, 0.1, 0.01,$ and 0.001 in $(5 \times 5, 2)^4$.

As is shown in Theorem 2, the difference between the two consecutive iteration results of AMIL algorithm will approximately point to the convergence solution when the current precoding and decoding matrices are sufficiently close to their convergence values. In DQO algorithm, the interference leakage is employed to be a measure of the distance between the current point and the convergence point, and line search is started when the interference leakage is smaller than a preset threshold th_1 . If th_1 is set to be too large, the generated direction does not point to the convergence value. If th_1 is set to be too small, the line search procedure cannot accelerate the convergence rate in time, and the effectiveness of DQO algorithm is degraded.

As is shown in Figures 2 and 3, $th_1 = 0.1$ has better performance than the too large value 1 and the too small value 0.001, which accords with the analysis above. Therefore, the parameter th_1 is set to be 0.1 in the following simulations.

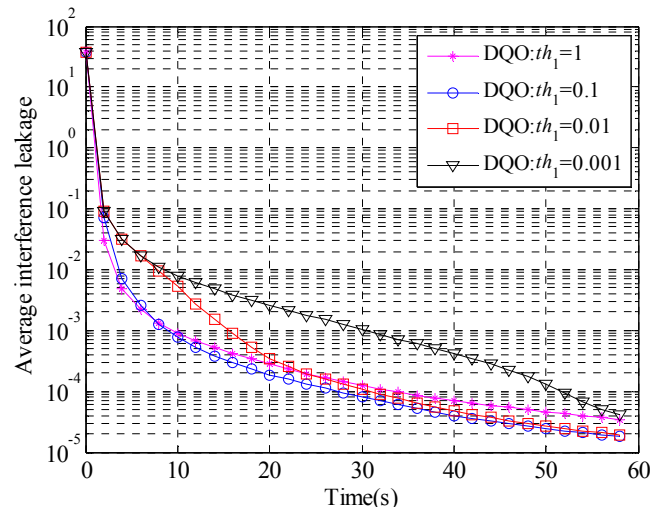


Figure 3. Convergence curves of the average interference leakage of DQO algorithm with $th_1 = 1, 0.1, 0.01, \text{ and } 0.001$ in $(10 \times 10, 1)^{19}$.

The parameter *interval* controls the frequency of executing line search. The parameter cannot be set to be too small. This can be explained by the fact that Equations (46) and (47) can be attained only when i_1 in Equation (45) is large enough. Therefore, it has to take several UV and VU steps before κ_1^i in Equation (45) takes the dominant proposition so that Equations (35) and (36) hold.

The performance of DQO algorithm with different *interval* for the $(5 \times 5, 2)^4$ and $(10 \times 10, 1)^{19}$ channels is evaluated in Figures 4 and 5, respectively. The parameter th_1 is presetted as 0.1, and *interval* is chosen to be 1, 2, 10, 15, and 20. The curves with *interval* = 1 converge slowly, which verifies the analysis above. On the other hand, when *interval* is large enough, the convergence curves are very close to each other. As *interval* = 20 has better performance than the others in Figure 4, *interval* is chosen to be 20 in the following simulations.

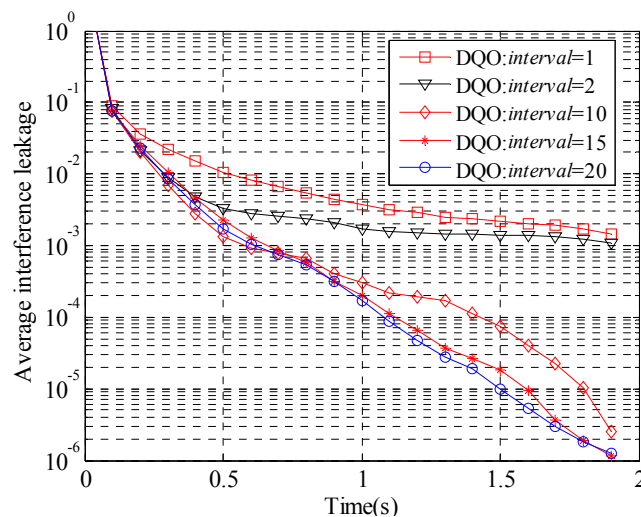


Figure 4. Convergence curves of the average interference leakage of DQO algorithm with *interval* = 1, 2, 10, 15, and 20 in $(5 \times 5, 2)^4$.

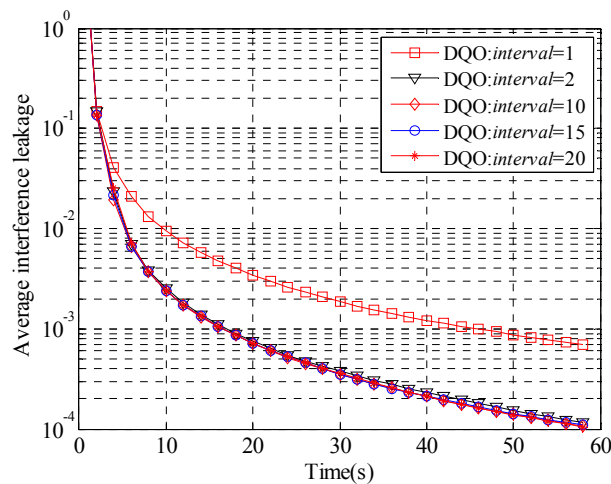


Figure 5. Convergence curves of the average interference leakage of DQO algorithm with $interval = 1, 2, 10, 15,$ and 20 in $(10 \times 10, 1)^{19}$.

5.2. Comparison of Different Algorithms

In this subsection, the performance of the DQO and AMIL algorithms is compared and analyzed. The parameters for the DQO algorithm are set as $th_1 = 0.1$ and $interval = 20$. The interference leakage with respect to iterations for one realization of randomly generated channel coefficients in the $(5 \times 5, 2)^4$ channel is shown in Figure 6, which provides us the overall effectiveness of DQO algorithm. As reflected in the figure, the interference leakage of the two algorithms remains the same before 168 iterations since they employ the same VU and UV steps. When the interference leakage is smaller than threshold th_1 , DQO algorithm begins to perform the LS procedure, and a step-like decrease in interference leakage can be observed. After 350 iterations, DQO algorithm reaches the interference leakage of 1.2×10^{-4} , which is one order lower than that of AMIL algorithm. The difference of the interference leakage becomes larger as the iterations increase, and DQO algorithm can reach the level which is 3 orders lower than that of AMIL algorithm. Therefore, the DQO algorithm can reach the same level of interference leakage as the traditional AMIL method with much fewer iterations.

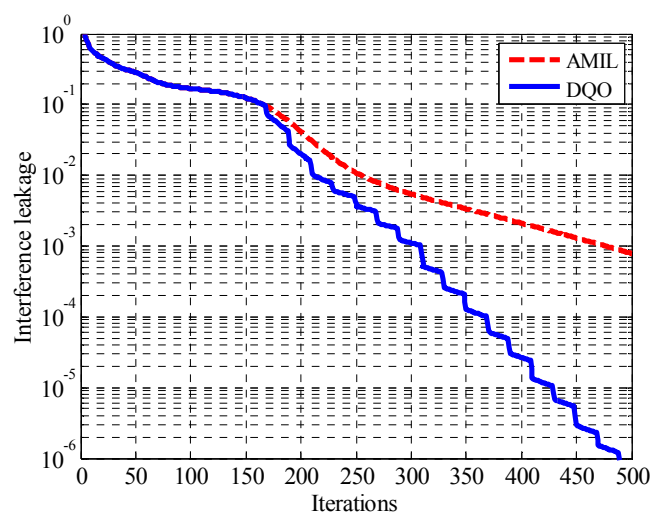


Figure 6. Convergence curves of the interference leakage for one realization of randomly generated channel coefficients in $(5 \times 5, 2)^4$.

As iterations cannot represent the complexity of the two algorithms sufficiently, the average performance of the two algorithms is evaluated in terms of execution time in the following simulations. Average interference leakage as well as sum rate in $(5 \times 5, 2)^4$ and $(10 \times 10, 1)^{19}$ channels are considered. The convergence of average interference leakage for $(5 \times 5, 2)^4$ channel is illustrated in Figure 7.

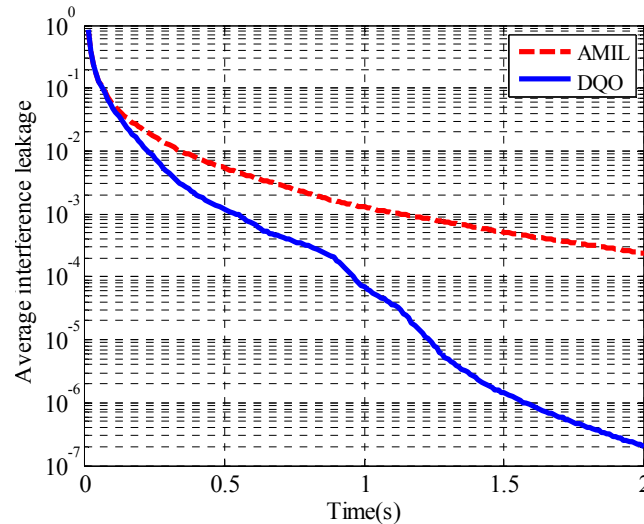


Figure 7. Convergence curves of the average interference leakage in $(5 \times 5, 2)^4$.

From the results, it can be seen that the DQO algorithm can suppress the interference leakage to 6.8×10^{-5} while the AMIL algorithm can only reach the level of 1.2×10^{-3} at 1 s. The difference of the average interference leakage between them increases with time, and the DQO algorithm outperforms the AMIL algorithm with three orders lower at 2 s. The convergence of the average sum rate with SNR = 40 dB for $(5 \times 5, 2)^4$ channel is depicted in Figure 8.

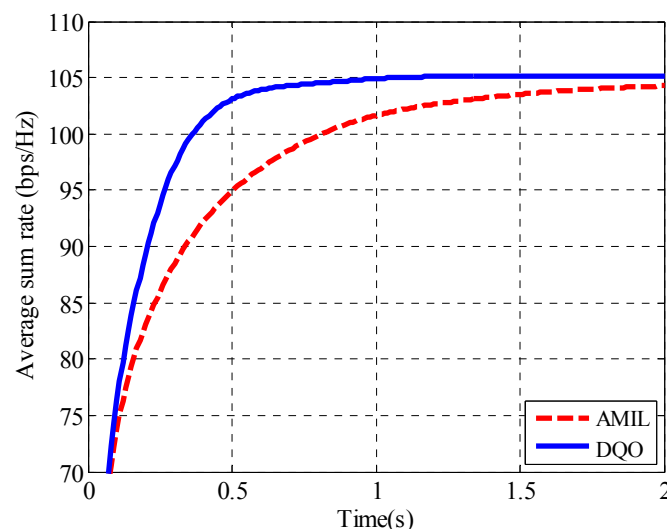


Figure 8. Convergence curves of the average sum rate in $(5 \times 5, 2)^4$.

As reflected in the figure, the average sum rate of the DQO algorithm increases more rapidly than that of the AMIL algorithm during the first 0.5 s and the DQO algorithm converges approximately at 0.61 s with the average sum rate of 104 bps/Hz while the AMIL algorithm takes about 1.78 s to reach

the same level. Like the $(5 \times 5, 2)^4$ case, the convergence curves of average interference leakage as well as sum rate for $(10 \times 10, 1)^{19}$ channel are plotted in Figures 9 and 10, respectively. As the number of users and antennas has become larger, it requires more iterations as well as computational time to suppress the interference leakage and to increase the sum rate. In Figure 9, the DQO algorithm can suppress the interference leakage more rapidly than the AMIL algorithm and achieves the interference leakage of 5.6×10^{-5} at 60 s, which is one order lower than that of the AMIL algorithm. In particular, the AMIL algorithm seems to converge at approximately 60 s while the DQO algorithm has the tendency to reach an even lower level of interference leakage afterward. In Figure 10, the average sum rate of the DQO algorithm converges much more rapidly than that of the AMIL algorithm during the first 10 s and the DQO algorithm achieves a higher average sum rate than the AMIL algorithm. To further compare their efficiency, average iterations N_{AMIL} and N_{DQO} as well as executing time T_{AMIL} and T_{DQO} with $th_2 = 10^{-4}$ are listed in the cases of different numbers of users and antennas in Table 4. Yetis *et al.* [32] have provided the feasible condition for IA in the $(M \times N, d)^K$ channel as $d \leq (M + N)/(K + 1)$, therefore we mainly provide the results of the most complex situation when $d = (M + N)/(K + 1)$.

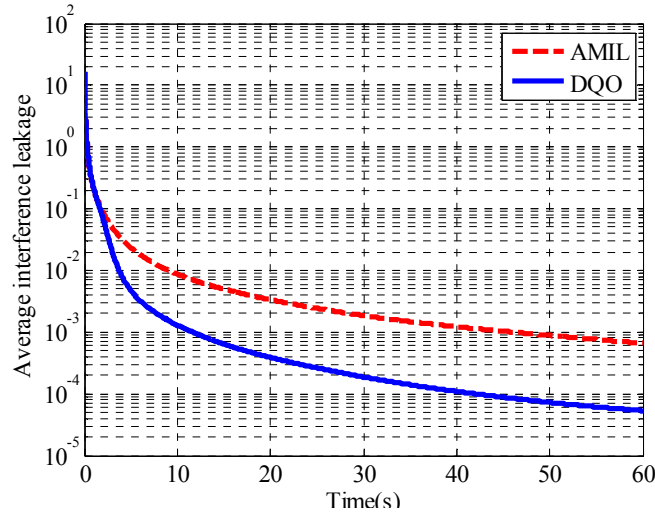


Figure 9. Convergence curves of the average interference leakage in $(10 \times 10, 1)^{19}$.

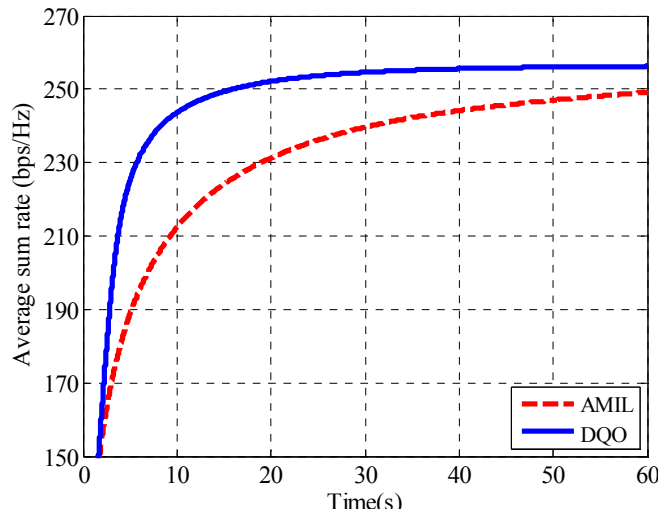


Figure 10. Convergence curves of the average sum rate in $(10 \times 10, 1)^{19}$.

The convergence rate and complexity of the algorithms can be compared by the iterations and computational time which are required to reach a certain level of interference leakage. As is shown in Table 4, the iterations and executing time of the DQO algorithm are only 22%–33% and 23%–33% of those of the AMIL algorithm, respectively. Therefore, simulations above confirm that DQO algorithm can converge faster than the AMIL algorithm and save nearly 2/3 of the computational complexity.

Table 4. Comparison between AMIL and DQO algorithms in $(M \times N, d)^K$.

$(M \times N, d)^K$	N_{AMIL}	N_{DQO}	N_{DQO}/N_{AMIL}	$T_{AMIL}(s)$	$T_{DQO}(s)$	T_{AMIL}/T_{DQO}
$(6 \times 6, 1)^{11}$	3.38×10^3	7.76×10^2	0.22	17.12	3.89	0.23
$(6 \times 6, 2)^5$	2.63×10^3	8.13×10^2	0.31	4.09	1.27	0.31
$(7 \times 7, 1)^{13}$	4.93×10^3	1.27×10^3	0.25	35.75	9.03	0.25
$(7 \times 7, 2)^6$	4.60×10^3	1.47×10^3	0.32	9.94	3.18	0.32
$(8 \times 8, 1)^{15}$	7.23×10^3	1.74×10^3	0.24	64.26	15.35	0.24
$(8 \times 8, 2)^7$	6.16×10^3	2.06×10^3	0.33	17.75	5.91	0.33
$(9 \times 9, 1)^{17}$	8.35×10^3	2.04×10^3	0.24	99.33	24.16	0.24
$(9 \times 9, 2)^8$	8.12×10^3	2.24×10^3	0.27	30.61	8.41	0.27
$(10 \times 10, 1)^{19}$	9.44×10^3	2.52×10^3	0.27	140.12	36.71	0.27
$(10 \times 10, 2)^9$	8.94×10^3	2.35×10^3	0.26	42.49	11.10	0.26

6. Conclusions and Future Work

In this paper, the properties of the traditional AMIL algorithm for IA have been studied. It has been found that if the AMIL algorithm can converge to a point where the interference leakage is very small, there exist fixed linear transformations \mathbf{T}_V and \mathbf{T}_U that exert on the deviations when the current point is sufficiently close to the convergence value. Particularly, successive exertions of the transformations on the deviations will lead to an interesting property that the difference of the consecutive iterations of AMIL algorithm will approximately point to the convergence value. Based on this discovery, a rapid convergent low-complexity minimization interference leakage algorithm, namely the DQO algorithm, has been proposed to speed up the convergence rate by employing optimal line search procedure. The direction for LS is generated by subtracting the consecutive iteration results of the AMIL algorithm, and the optimal step size can be determined analytically by optimizing a quartic function. Complexity analysis of the DQO and AMIL algorithms has been provided, and simulation results have shown that the DQO algorithm can reduce the number of iterations and execution time significantly under the same interference leakage and sum rate conditions. In the future, we will focus on extending the proposed algorithm to many more scenarios, such as systems with imperfect CSI [33,34], cooperative communications [35], and so on.

Acknowledgments

We would like to express our appreciation to the editors and reviewers for their comments and suggestions, which have helped us improve the quality of the article. This research was supported partly by the National Natural Science Foundation of China under Grant No. 61102085 and No. 61201224, Natural Scientific Research Innovation Foundation in Harbin Institute of Technology under

No. HIT.NSRIF 201151, China Postdoctoral Science Foundation Special Funded Project under 2013T60282 and the Fundamental Research Funds for the Central Universities under DUT14QY44.

Author Contributions

Z.W. and L.J. conceived the scheme; G.R. and G.W. performed the experiments; N.Z. analyzed the data; L.J. wrote the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Rafique, Z.; Seet, B.C.; Al-Anbuky, A. Performance analysis of cooperative virtual MIMO systems for wireless sensor networks. *Sensors* **2013**, *13*, 7033–7052.
2. Mhiri, F.; Sethom, K.; Bouallegue, R. A survey on interference management techniques in femtocell self-organizing networks. *J. Netw. Comput. Appl.* **2013**, *36*, 58–65.
3. Jeon, S.-W.; Gastpar, M. A survey on interference networks: Interference alignment and neutralization. *Entropy* **2012**, *14*, 1842–1863.
4. Zhang, H.; Chu, X.; Ma, W.; Zheng, W.; Wen, X. Resource allocation with interference mitigation in OFDMA femtocells for co-channel deployment. *EURASIP J. Wirel. Commun. Netw.* **2012**, *2012*, doi:10.1186/1687-1499-2012-289.
5. Zhao, N.; Yu, F.R.; Leung, V.C.M. Opportunistic communications in interference alignment networks with wireless power transfer. *IEEE Wirel. Commun.* **2015**, *22*, 88–95.
6. Maddah-Ali, M.A.; Motahari, A.S.; Khandani, A.K. Communication over MIMO X Channels: Interference alignment, decomposition, and performance analysis. *IEEE Trans. Inf. Theory* **2008**, *54*, 3457–3470.
7. Jafar, S.A.; Shamai, S. Degrees of freedom region of the MIMO X channel. *IEEE Trans. Inf. Theory* **2008**, *54*, 151–170.
8. Cadambe, V.R.; Jafar, S.A. Interference alignment and degrees of freedom of the K -user interference channel. *IEEE Trans. Inf. Theory* **2008**, *54*, 3425–3441.
9. Lee, N.; Lim, J.-B.; Chun, J. Degrees of freedom of the MIMO Y channel: Signal space alignment for network coding. *IEEE Trans. Inf. Theory* **2010**, *56*, 3332–3342.
10. Cadambe, V.R.; Jafar, S.A. Degrees of freedom of wireless networks with relays, feedback, cooperation, and full duplex operation. *IEEE Trans. Inf. Theory* **2009**, *55*, 2334–2344.
11. Maso, M.; Debbah, M.; Vangelista, L. A distributed approach to interference alignment in OFDM-based two-tiered networks. *IEEE Trans. Veh. Technol.* **2013**, *62*, 1935–1949.
12. Lertwiram, N.; Popovski, P.; Sakaguchi, K. A study of trade-off between opportunistic resource allocation and interference alignment in femtocell scenarios. *IEEE Wirel. Commun. Lett.* **2012**, *1*, 356–359.
13. Xu, Y.; Mao, S.W. Stackelberg game for cognitive radio networks with MIMO and distributed interference alignment. *IEEE Trans. Veh. Technol.* **2014**, *63*, 879–892.

14. Tsinos, C.G.; Berberidis, K. Blind opportunistic interference alignment in MIMO cognitive radio systems. *IEEE Trans. Emerg. Sel. Top. Circuits Syst.* **2013**, *3*, 626–639.
15. Wu, Z.; Jiang, L.; Ren, G.; Zhao, N.; Zhao, Y. A novel joint spatial-code clustered interference alignment scheme for large-scale wireless sensor networks. *Sensors* **2015**, *15*, 1964–1997.
16. Sharma, S.K.; Chatzinotas, S.; Ottersten, B. Interference alignment for spectral coexistence of heterogeneous networks. *EURASIP J. Wirel. Commun. Netw.* **2013**, *2013*, doi:10.1186/1687-1499-2013-46.
17. Sun, C.; Yang, Y.C.; Yuan, Y.X. Low complexity interference alignment algorithms for desired signal power maximization problem of MIMO channels. *EURASIP J. Adv. Signal Process.* **2012**, *2012*, 1–13.
18. Tresch, R.; Guillaud, M.; Riegler, E. On the achievability of interference alignment in the K-user constant MIMO interference channel. In Proceedings of the IEEE Workshop on Statistical Signal Processing, Cardiff, Wales, UK, 31 August–3 September 2009; pp. 277–280.
19. Wilson, C.; Veeravalli, V. A convergent version of the max SINR algorithm for the MIMO Interference channel. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 2952–2961.
20. Peters, S.W.; Heath, R.W. Interference alignment via alternating minimization. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 19–24 April 2009; pp. 2445–2448.
21. Razaviyayn, M.; Sanjabi, M.; Luo, Z.-Q. Linear transceiver design for interference alignment: Complexity and computation. *IEEE Trans. Inf. Theory* **2012**, *58*, 2896–2910.
22. Ma, L.; Xu, T.; Sternberg, G. Computational complexity of interference alignment for symmetric MIMO networks. *IEEE Commun. Lett.* **2013**, *17*, 2308–2311.
23. Gomadam, K.; Cadambe, V.R.; Jafar, S.A. A distributed numerical approach to interference alignment and applications to wireless interference networks. *IEEE Trans. Inf. Theory* **2011**, *57*, 3309–3322.
24. Shen, H.; Li, B.; Tao, M.; Luo, Y. The new interference alignment scheme for the MIMO interference channel. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Sydney, Australia, 18–21 April 2010; pp. 1–6.
25. Santamaria, I.; Gonzalez, O.; Heath, R.W.; Peters, S.W. Maximum sum-rate interference alignment algorithms for MIMO channels. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Miami, FL, USA, 6–10 December 2010; pp. 1–6.
26. Ram, B.H.; Wei, L.; Ayyar, A.; Lilleberg, J.; Giridhar, K. Precoder design for K-user interference channels with finite alphabet signals. *IEEE Commun. Lett.* **2013**, *17*, 681–684.
27. Xu, T.; Xia, X.G. A diversity analysis for distributed interference alignment using the max-SINR algorithm. *IEEE Trans. Inf. Theory* **2014**, *60*, 1857–1868.
28. Zhao, N.; Yu, F.R.; Sun, H.; Nallanathan, A.; Yin, H. A novel interference alignment scheme based on sequential antenna switching in wireless networks. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 5008–5021.
29. Ayach, O.E.; Heath, R.W. Interference alignment with analog channel state feedback. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 626–636.

30. Cho, S.; Huang, K.; Kim, D.K.; Lau, V.K.N.; Chae, H.; Seo, H.; Kim, B.-H. Feedback-topology designs for interference alignment in MIMO interference channels. *IEEE Trans. Signal Process.* **2012**, *60*, 6561–6575.
31. Zhang, L.; Song, L.; Ma, M.; Zhang, Z.; Lei, M.; Jiao, B. Interference alignment with differential feedback. *IEEE Trans. Veh. Technol.* **2012**, *61*, 2878–2883.
32. Yetis, C.M.; Gou, T.; Jafar, S.A.; Kayran, A.H. On feasibility of interference alignment in MIMO interference networks. *IEEE Trans. Signal Process.* **2010**, *58*, 4771–4782.
33. Taki, M.; Rezaee, M.; Guillaud, M. Adaptive modulation and coding for interference alignment with imperfect CSIT. *IEEE Trans. Wireless Commun.* **2014**, *13*, 5264–5273.
34. Razavi, S.M.; Ratnarajah, T. Performance Analysis of interference alignment under CSI mismatch. *IEEE Trans. Veh. Technol.* **2014**, *63*, 4740–4748.
35. Wang, H.; Xia, X. Asynchronous cooperative communication systems: A survey on signal designs. *Sci. China-Inf. Sci.* **2011**, *54*, 1547–1561.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).