

# An Efficient Greedy Scheduler for Zero-Forcing Dirty-Paper Coding

Jisheng Dai, *Student Member, IEEE*, Chunqi Chang, *Member, IEEE*, Zhongfu Ye,  
and Yeung Sam Hung, *Senior Member, IEEE*

**Abstract**—In this paper, an efficient greedy scheduler for zero-forcing dirty-paper coding (ZF-DPC), which can be incorporated in complex Householder QR factorization of the channel matrix, is proposed. The ratio of the complexity of the proposed scheduler to the complexity of the channel matrix factorization required by ZF-DPC is  $\mathcal{O}(M^{-1})$ , while such ratio for the original greedy scheduler is  $\mathcal{O}(M)$ , where  $M$  is the number of transmitters. Therefore, the new scheduler reduces the overhead of scheduling from being the bottleneck of ZF-DPC to being negligible.

**Index Terms**—Efficient greedy scheduler, multiple-input multiple-output (MIMO), vector Gaussian broadcast channel (GBC), zero-forcing dirty-paper coding (ZF-DPC), QR factorization by Householder transformation.

## I. INTRODUCTION

COMPARED to traditional single-input single-output communication systems, multiple-input multiple-output (MIMO) systems support greater data rate and higher reliability over wireless links [1], [2]. Recently, there has been tremendous interest in MIMO multiuser systems with multiple-antenna (vector) Gaussian broadcast channel (GBC), for which the transmitters can cooperate in encoding their signals but the receivers (users) are constrained to decode their signals independently.

It is well known that dirty-paper coding (DPC) [3], which encodes signals at the transmitter with known interference and constrained transmitting power, achieves the capacity region of the multiple antenna GBC [4]–[6] for the case of flat-fading channels with perfect channel state information (CSI) at all transmitters. A practical coding strategy called zero-forcing dirty-paper coding (ZF-DPC) is proposed in [7] to approach the capacity. It is noticed in [7] that the sum rate (or throughput) of the channels with ZF-DPC can be affected by user ordering, and the maximal sum rate can be achieved when the number of ordered users is equal to the rank of the channel matrix. Hence, there is a need to find the optimal user ordering that approaches the maximal sum rate. A sub-optimal user ordering algorithm (scheduler) called greedy scheduling

is proposed in [8]. The performance of this scheduler and its slightly modified versions are analyzed in [9].

However, the complexity of greedy scheduler proposed in [8], [9] is very high if the number of transmit antennas is large. As will be shown in this paper, the computing overhead of the scheduler is much larger than the computation of the channel matrix factorization itself required by ZF-DPC, the ratio of the computational complexity of the overhead to that of the channel matrix factorization being linearly proportional to the number of selected users (which is chosen equal to the number of transmitters). Therefore, fast scheduling algorithms are desired to reduce the computational load of the scheduler. In this paper, an efficient algorithm for the greedy scheduler [8], [9] is proposed, which renders the overhead of scheduling in ZF-DPC to be negligible.

The rest of this paper is organized as follows. In Section II, ZF-DPC and greedy ZF-DPC algorithms are introduced. An efficient greedy scheduler for near-optimal user ordering is derived in Section III. Section IV evaluates the complexity of different algorithms. Conclusions are drawn in Section V.

The following notations are used in this paper. The superscripts  $(\cdot)^H$  and  $\|\cdot\|$  denote Hermitian operation and 2-norm of a vector, respectively.  $(\cdot)_{ij}$  denotes the  $(i, j)$ th element of a matrix, and  $\mathbf{I}$  denotes the identity matrix.  $\mathcal{A} \setminus \mathcal{B}$  denotes the set  $\mathcal{A}$  excluding set  $\mathcal{B}$ .  $\mathbb{C}^{n \times m}$  is the  $n \times m$  complex matrix space.

## II. ZF-DPC AND GREEDY ZF-DPC ALGORITHM

### A. ZF-DPC

Consider a  $N$ -user vector GBC with  $M (\leq N)$  transmitters. The relationship between the received signals  $\mathbf{y} = [y_1 \dots y_N]^T$  and transmitted signals  $\mathbf{x} = [x_1 \dots x_M]^T$  can be written as

$$\mathbf{y} = \begin{bmatrix} \mathbf{h}_1^H \\ \vdots \\ \mathbf{h}_N^H \end{bmatrix} \mathbf{x} + \mathbf{n} \triangleq \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (1)$$

where  $\mathbf{h}_i^H \in \mathbb{C}^{1 \times M}$  ( $i = 1, \dots, N$ ) is the channel vector denoting the path gain from the transmit antennas to the  $i$ th user, and  $\mathbf{n} \sim N(0, \sigma_n^2 \mathbf{I})$  is the circularly symmetric complex Gaussian noise. The power-constrained dirty-paper coding (DPC) can be implemented as follows. Assuming  $\text{rank}(\mathbf{H}) = K (\leq M)$ , we can perform a QR factorization of (strictly speaking, the transpose of)  $\mathbf{H}$ , so that

$$\mathbf{H} = \mathbf{G}\mathbf{Q}, \quad (2)$$

where  $\mathbf{G} \in \mathbb{C}^{N \times K}$  is lower triangular and  $\mathbf{Q} \in \mathbb{C}^{K \times M}$  has orthonormal rows.

Paper approved by A. Lozano, the Editor for Wireless Network Access and Performance of the IEEE Communications Society. Manuscript received December 17, 2007; revised July 18, 2008 and October 7, 2008.

J. Dai and Z. Ye are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui, 230027, P.R. China (e-mail: jisheng.dai@ieee.org; yezf@ustc.edu.cn). Z. Ye is also with the National Mobile Communications Research Laboratory, Southeast University, P. R. China.

C. Chang and Y. S. Hung are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong (e-mail: {cqchang, yshung}@eee.hku.hk).

This work is supported by the open issues of National Mobile Communications Research Laboratory, Southeast University (No. A200509).

Digital Object Identifier 10.1109/TCOMM.2009.07.070162

We wish to transmit information to a number of users with maximal sum rate (throughput) under the constraint of total transmitting power  $P$ . It is shown in [7] that to achieve the maximal sum rate we should transmit information to  $K$  users. Let the information to be transmitted to the users be  $\mathbf{u} = [u_1 \dots u_K]^T$ , then we can transform this information as  $\mathbf{x} = \mathbf{Q}^H \mathbf{u}$  and transmit  $\mathbf{x}$  through the channels of  $\mathbf{H}$ . From Eq (1) and (2) we have

$$y_k = G_{kk}u_k + \sum_{j < k} G_{kj}u_j + n_k, \quad k = 1, 2, \dots, K, \quad (3)$$

while there is no information sent to users  $K+1, \dots, N$ . This is called zero-forcing dirty-paper coding (ZF-DPC) because for the  $k$ -th user the signal of all other users with index greater than  $k$  are forced to zero in the  $k$ -th channel. In this scheme, if we encode the information in the order of  $u_1, u_2, \dots, u_K$ , then for the  $k$ -th user the interference  $\sum_{j < k} G_{kj}u_j$  is noncausally known at the transmitter, and according to the result of [6] codewords can be designed in a way that the capacity of the channel is the same as if  $\sum_{j < k} G_{kj}u_j, k = 1, 2, \dots, K$ , were not present.

The sum rate for this ZF-DPC program can be maximized when the power is optimally allocated across the first  $K$  channels. This maximal sum rate is determined in [7], [8] as

$$R = \sum_{k=1}^K [\log(\xi d_k)]_+ \quad (4)$$

where  $d_k = |G_{kk}|^2$ ,  $[x]_+ \triangleq \max(x, 0)$ , and  $\xi$  is decided by the following total power constraint

$$\sum_{k=1}^K [\xi - 1/d_k]_+ = P. \quad (5)$$

### B. Greedy ZF-DPC Algorithm

Notice that the sum rate determined by Eq (4) depends on  $d_k, k = 1, 2, \dots, K$ , which in turn are determined by the QR factorization (2). If we permute the rows of the channel matrix  $\mathbf{H}$  by a permutation matrix  $\mathbf{\Pi}$  to  $\mathbf{\Pi H}$ , which is equivalent to re-ordering the receivers (users), and perform ZF-DPC on the permuted channel matrix  $\mathbf{\Pi H}$ , then we get a different sum rate. This has been noted in [7]. Hence there is a need to find an optimal user order, i.e. an optimal permutation matrix  $\mathbf{\Pi}_o$ , so that ZF-DPC based on  $\mathbf{\Pi}_o \mathbf{H}$  will give the maximal sum rate.

Obviously it is not easy to solve this combinatorial optimization problem especially when  $N$  and  $K$  are large. Therefore, in [8] a sub-optimal approach to this optimal user ordering (scheduling) problem called greedy ZF-DPC scheduler is proposed. This algorithm finds the sub-optimal permutation vector  $\pi = [\pi(1), \dots, \pi(K)]$  for the first  $K$  rows of  $\mathbf{\Pi}_o$ , outlined as follows.

#### Greedy ZF-DPC Scheduler Algorithm

- 1) Initialization: Let  $K = 0$ .
- 2) For  $k = 1, \dots, M$

Let  $\tilde{\pi}_k = \{\pi(i) | i < k\}$ . ( $\tilde{\pi}_1 = \emptyset$ )

Project all the  $\mathbf{h}_j^H, j \in \{1, \dots, N\} \setminus \tilde{\pi}_k$ , onto the orthogonal complement of the subspace spanned

by  $\{\mathbf{h}_{\tilde{\pi}(i)}^H | i < k\}$ , and denote the 2-norm of these projected vectors as  $\gamma_j, j \in \{1, \dots, N\} \setminus \tilde{\pi}_k$ .

Let  $k^* = \arg \max_j \gamma_j$ .

If  $\gamma_{k^*} = 0$

go to step 3

else

Let  $\pi(k) = k^*$  and  $K = K + 1$ .

End

- 3) Output the scheduling represented by the permutation vector  $\pi = [\pi(1), \dots, \pi(K)]$ .

This algorithm also serves as the basis of the scheduler employed in [9]–[11]. As we will see later, the computational complexity of this scheduler is of the order  $\mathcal{O}(NM^3)$  flops (floating operations), while the computational complexity of the factorization of the channel matrix  $\mathbf{H}$  in Eq (2) is of the order  $\mathcal{O}(NM^2)$ . Therefore this scheduling algorithm is the bottleneck in the computation of the ZF-DPC. In the following, we propose an efficient scheduling algorithm to substantially reduce its computational complexity.

## III. EFFICIENT GREEDY SCHEDULER FOR ZF-DPC

### A. QR factorization of the channel matrix by Householder transformation

Given two vectors  $\mathbf{a}^H, \mathbf{b}^H \in \mathbb{C}^{1 \times m}$  with  $\|\mathbf{a}\| = \|\mathbf{b}\|$ , the Householder transformation  $\mathcal{H}(\mathbf{a}^H, \mathbf{b}^H) \in \mathbb{C}^{m \times m}$  reflects  $\mathbf{a}^H$  to  $\mathbf{b}^H$  and is given by [12]

$$\mathcal{H}(\mathbf{a}^H, \mathbf{b}^H) = \mathbf{I} - \frac{\mathbf{z}\mathbf{z}^H}{\mathbf{a}^H \mathbf{z}} = \mathbf{I} + \frac{\mathbf{z}\mathbf{z}^H}{\mathbf{z}^H \mathbf{b}}, \quad (6)$$

where  $\mathbf{z} = \mathbf{a} - \mathbf{b}$ . It can be readily checked that  $\mathbf{a}^H \mathcal{H}(\mathbf{a}^H, \mathbf{b}^H) = \mathbf{b}^H$  and  $\mathcal{H}(\mathbf{a}^H, \mathbf{b}^H)$  is unitary.

The QR factorization of the channel matrix  $\mathbf{H}$  in Eq (2) can be implemented by means of the Householder transformation. First we reflect  $\mathbf{h}_1^H$  to  $\|\mathbf{h}_1\| \mathbf{e}_1^H$ , where  $\mathbf{e}_1^H = [1 \ 0 \ \dots \ 0]$ , by  $\mathcal{H}(\mathbf{h}_1^H, \|\mathbf{h}_1\| \mathbf{e}_1^H)$ , so that

$$\mathbf{H} \mathcal{H}(\mathbf{h}_1^H, \|\mathbf{h}_1\| \mathbf{e}_1^H) = \begin{bmatrix} \|\mathbf{h}_1\| & \mathbf{0} \\ \times & \mathbf{H}_{N-1} \end{bmatrix}. \quad (7)$$

Then we reflect  $\hat{\mathbf{h}}_1^H$ , the first row of  $\mathbf{H}_{N-1}$ , to  $\|\hat{\mathbf{h}}_1\| \mathbf{e}_1^H$  by  $\mathcal{H}(\hat{\mathbf{h}}_1^H, \|\hat{\mathbf{h}}_1\| \mathbf{e}_1^H)$ , so that

$$\begin{aligned} & \begin{bmatrix} \|\mathbf{h}_1\| & \mathbf{0} \\ \times & \mathbf{H}_{N-1} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathcal{H}(\hat{\mathbf{h}}_1^H, \|\hat{\mathbf{h}}_1\| \mathbf{e}_1^H) \end{bmatrix} \\ &= \begin{bmatrix} \|\mathbf{h}_1\| & \mathbf{0} & \mathbf{0} \\ \times & \|\hat{\mathbf{h}}_1\| & \mathbf{0} \\ \times & \times & \mathbf{H}_{N-2} \end{bmatrix}. \end{aligned} \quad (8)$$

Applying this procedure recursively to  $\mathbf{H}_{N-2}$ , we get after at most  $M$  steps the QR factorization of  $\mathbf{H}$  as in Eq (2). Details of the Householder QR can be found in [13].

### B. Efficient Greedy Scheduler

The greedy ZF-DPC as proposed in [8] consists of two independent steps, namely the greedy scheduling and the QR factorization. We note that if the QR factorization is performed recursively by the Householder transformation described above, then the scheduler can be incorporated very

efficiently within the Householder QR factorization. The key observations in our proposed method are:

- The subspace projection of step (2) of the greedy scheduler is automatically performed as part of the Householder QR and therefore to perform this scheduling separately is a duplication of computational effort.
- The size of the subspace projection problem gets progressively smaller in the recursive procedure of the Householder QR factorization, and since the 2-norm of the original projection is equal to the 2-norm of the projection onto the space of progressively reduced dimension, the 2-norm in turn can be inferred very efficiently in a recursive manner.

According to the Greedy Scheduler Algorithm described in the previous section, the first user selected is the one whose channel has the maximal 2-norm, so we permute  $\mathbf{H}$  accordingly to  $\mathbf{\Pi}_1\mathbf{H}$  with the first row associated to the selected user, and we denote this channel as  $\mathbf{h}^H$  to simplify the presentation. Applying Householder transformation to the first row of  $\mathbf{\Pi}_1\mathbf{H}$  gives

$$\mathbf{\Pi}_1\mathbf{H} = \begin{bmatrix} \|\mathbf{h}\| & \mathbf{0} \\ \times & \mathbf{H}_{N-1} \end{bmatrix} \mathbf{Q}_1, \quad (9)$$

where  $\mathbf{Q}_1^H$  is the unitary Householder transformation matrix for the first row of  $\mathbf{\Pi}_1\mathbf{H}$ .

We now proceed to select the  $(k+1)$ -th user by an induction argument. Suppose the previous  $k$  users have already been selected by the greedy scheduler, whereby  $\mathbf{H}$  is permuted to  $\mathbf{\Pi}_k\mathbf{H}$  whose first  $k$  rows are associated with the selected  $k$  users in an ordered way. Also assume that we have applied Householder transformation to the first  $k$  rows of  $\mathbf{\Pi}_k\mathbf{H}$  so that

$$\mathbf{\Pi}_k\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{X} & \mathbf{H}_{N-k} \end{bmatrix} \mathbf{Q}_k, \quad (10)$$

where  $\mathbf{R}$  is a lower triangular matrix, and  $\mathbf{Q}_k$  is unitary since  $\mathbf{Q}_k^H$  is a product of  $k$  Householder transformation matrices. Partitioning the matrices in Eq (10) accordingly, we have

$$\begin{bmatrix} \hat{\mathbf{H}}_1 \\ \hat{\mathbf{H}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{X} & \mathbf{H}_{N-k} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{k,1} \\ \mathbf{Q}_{k,2} \end{bmatrix}. \quad (11)$$

To select the  $(k+1)$ -th user in the greedy scheduling algorithm, we need to project rows of  $\hat{\mathbf{H}}_2$  onto the orthogonal complement of the subspace spanned by rows of  $\hat{\mathbf{H}}_1$ , and order the users by the 2-norm of these projections. To show that the greedy scheduling scheme can be incorporated into the QR factorization, we need the following result.

*Theorem 1:* Let rows of  $\hat{\mathbf{H}}_2^p$  be the projection of corresponding rows of  $\hat{\mathbf{H}}_2$  onto the orthogonal complement of the subspace spanned by rows of  $\hat{\mathbf{H}}_1$ , where  $\hat{\mathbf{H}}_1$  and  $\hat{\mathbf{H}}_2$  are defined by Eq (11). Then the 2-norm of the rows of  $\hat{\mathbf{H}}_2^p$  are equal to the 2-norm of the corresponding rows of  $\mathbf{H}_{N-k}$ , i.e.,  $\text{diag}\{\hat{\mathbf{H}}_2^p(\hat{\mathbf{H}}_2^p)^H\} = \text{diag}\{\mathbf{H}_{N-k}\mathbf{H}_{N-k}^H\}$ .

*Proof:* Since  $\hat{\mathbf{H}}_1 = \mathbf{R}\mathbf{Q}_{k,1}$  and  $\mathbf{R}$  is nonsingular, the row space of  $\hat{\mathbf{H}}_1$  is the same as the row space of  $\mathbf{Q}_{k,1}$ . As  $\mathbf{Q}_{k,2}\mathbf{Q}_{k,1}^H = \mathbf{0}$ , the row space of  $\mathbf{Q}_{k,2}$  is just the orthogonal

complement of the row space of  $\hat{\mathbf{H}}_1$ . Then we have

$$\begin{aligned} \hat{\mathbf{H}}_2^p &= \hat{\mathbf{H}}_2\mathbf{Q}_{k,2}^H\mathbf{Q}_{k,2} \\ &= (\mathbf{X}\mathbf{Q}_{k,1} + \mathbf{H}_{N-k}\mathbf{Q}_{k,2})\mathbf{Q}_{k,2}^H\mathbf{Q}_{k,2} \\ &= \mathbf{H}_{N-k}\mathbf{Q}_{k,2}, \end{aligned} \quad (12)$$

$$\begin{aligned} \hat{\mathbf{H}}_2^p(\hat{\mathbf{H}}_2^p)^H &= \mathbf{H}_{N-k}\mathbf{Q}_{k,2}\mathbf{Q}_{k,2}^H\mathbf{H}_{N-k}^H \\ &= \mathbf{H}_{N-k}\mathbf{H}_{N-k}^H. \end{aligned} \quad (13)$$

This proves that  $\text{diag}\{\hat{\mathbf{H}}_2^p(\hat{\mathbf{H}}_2^p)^H\} = \text{diag}\{\mathbf{H}_{N-k}\mathbf{H}_{N-k}^H\}$ . ■

By Theorem 1, we can select the  $(k+1)$ -th user as the one whose corresponding row in  $\mathbf{H}_{N-k}$  has the maximal 2-norm among all the rows of  $\mathbf{H}_{N-k}$ , which is equivalent to Step 2 of the greedy ZF-DPC scheduler given in Section II-B. Then we have the following algorithm of efficient ZF-DPC with greedy scheduling.

#### *Efficient ZF-DPC with Greedy Scheduling*

- 1) Let  $K = 0$ ,  $\mathbf{H}_N = \mathbf{H}$ , and  $\mathbf{G}$  be a zero matrix with the same size as  $\mathbf{H}$ .
- 2) For  $k = 1, \dots, M$

Compute the 2-norm of the rows of  $\mathbf{H}_{N-(k-1)}$ . Permute the rows of  $\mathbf{H}_{N-(k-1)}$  to  $\mathbf{H}_*$  so that its first row, denoted as  $\mathbf{h}^H$ , has the maximal 2-norm among all its rows.

If  $\|\mathbf{h}\| = 0$

go to step 3

else

Let  $\mathcal{H}(\mathbf{h}^H, \|\mathbf{h}\|\mathbf{e}_1^H) = \mathbf{I} - \frac{\mathbf{h} - \|\mathbf{h}\|\mathbf{e}_1}{\mathbf{h}^H(\mathbf{h} - \|\mathbf{h}\|\mathbf{e}_1)}(\mathbf{h} - \|\mathbf{h}\|\mathbf{e}_1)^H \triangleq \mathbf{I} - \mathbf{v}_k\mathbf{w}_k^H$ .

Perform the Householder transformation

$$\mathbf{H}_*\mathcal{H}(\mathbf{h}^H, \|\mathbf{h}\|\mathbf{e}_1^H) = \begin{bmatrix} \|\mathbf{h}\| & \mathbf{0} \\ \mathbf{g} & \mathbf{H}_{N-k} \end{bmatrix}. \quad (14)$$

Define the  $k$ th column of  $\mathbf{G}$  to be  $\begin{bmatrix} \mathbf{0}_{(k-1) \times 1} \\ \|\mathbf{h}\| \\ \mathbf{g} \end{bmatrix}$ .

Let  $K = K + 1$ .

End

- 3) Output the lower triangular matrix  $\mathbf{G}$ , and  $\mathbf{Q} = \prod_{k=1}^K \mathbf{Q}_k$ , where  $\mathbf{Q}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathbf{v}_k\mathbf{w}_k^H \end{bmatrix}^H$ . We note that  $\mathbf{Q}$  needs not to be obtained explicitly since it is used only for coding  $\mathbf{x} = \mathbf{Q}^H\mathbf{u}$  and this can be calculated by using  $\mathbf{v}_k$  and  $\mathbf{w}_k$ ,  $k = 1, \dots, K$ , with complexity of the same order as that of calculating  $\mathbf{Q}^H\mathbf{u}$  directly.

#### IV. COMPLEXITY EVALUATION

In practice, the number of users (receivers)  $N$  is always much greater than the number of transmitting antennas  $M$ , i.e.  $N \geq M$ . If the channels are not degenerate, which is assumed in this section, the rank of the channel matrix  $\mathbf{H}$  is  $M$  so that the number of selected users  $K = M$ . In the following, we calculate the number of real multiplications and the number of real additions separately, which are required by different algorithms. Note that one complex multiplication takes 4 real

multiplications and 2 real additions, and one complex addition needs 2 real additions. Unless otherwise specified, multiplications and additions refer to complex operations throughout this section.

#### A. Complexity of Householder QR for channel matrix factorization

For each iteration of the QR factorization, we need to calculate  $\mathbf{H}_* \mathcal{H} = \mathbf{H}_* (\mathbf{I} - \mathbf{v}_k \mathbf{w}_k^H) = \mathbf{H}_* - (\mathbf{H}_* \mathbf{v}_k) \mathbf{w}_k^H$ , where the size of  $\mathbf{H}_*$  is  $(N - k + 1) \times (M - k + 1)$ . To calculate  $\mathbf{v}_k$  and  $\mathbf{w}_k$  we need  $3(M - k + 1)$  multiplications and  $2(M - k) + 1$  additions, and we need  $2(N - k + 1)(M - k + 1)$  multiplications and  $(2M - 2k + 1)(N - k + 1)$  additions for the remaining calculation. Therefore the number of real multiplications needed in total is

$$C_{QR}^\times = 4NM^2 - \frac{4}{3}M^3 + \mathcal{O}(NM), \quad (15)$$

and the number of real additions is

$$C_{QR}^+ = 4NM^2 - \frac{4}{3}M^3 + \mathcal{O}(NM). \quad (16)$$

#### B. Complexity of Greedy ZF-DPC Scheduler

For each iteration, we need  $2kM$  multiplications and  $2kM - M - k$  additions for Gram-Schmidt orthogonalization,  $kM(N - k + 1)$  multiplications and  $kM(N - k) + M + k - N$  additions for projection, and  $2M(N - k)$  real multiplications and  $(M - 1)(N - k) + M(N - k)$  real additions for 2-norm calculation. Thus, the total number of real multiplications is

$$C_S^\times = 2NM^3 - \frac{4}{3}M^4 + 5M^3 + 2NM^2 + \mathcal{O}(NM), \quad (17)$$

and the total number of real additions is

$$C_S^+ = 2NM^3 - \frac{4}{3}M^4 + 4M^3 + 2NM^2 + \mathcal{O}(NM). \quad (18)$$

#### C. Complexity of Efficient Greedy Scheduler

In our efficient greedy scheduler, scheduling is performed inside the QR factorization loop. The overhead of scheduling is to compute the 2-norm of rows of  $\mathbf{H}_{N-k}$  at each iteration  $k + 1$ . From Eq (14) we have

$$\begin{aligned} \mathbf{H}_* \mathbf{H}_*^H &= \begin{bmatrix} \|\mathbf{h}\| & \mathbf{0} \\ \mathbf{g} & \mathbf{H}_{N-k} \end{bmatrix} \begin{bmatrix} \|\mathbf{h}\| & \mathbf{g}^H \\ \mathbf{0} & \mathbf{H}_{N-k}^H \end{bmatrix} \\ &= \begin{bmatrix} \|\mathbf{h}\|^2 & \|\mathbf{h}\| \mathbf{g}^H \\ \|\mathbf{h}\| \mathbf{g} & \mathbf{g} \mathbf{g}^H + \mathbf{H}_{N-k} \mathbf{H}_{N-k}^H \end{bmatrix}. \end{aligned} \quad (19)$$

The diagonal of  $\mathbf{H}_* \mathbf{H}_*^H$  and diagonal of  $\mathbf{H}_{N-k} \mathbf{H}_{N-k}^H$  are the squared 2-norm of rows of  $\mathbf{H}_*$  and rows of  $\mathbf{H}_{N-k}$ , respectively. So the squared 2-norm of  $i$ -th row of  $\mathbf{H}_{N-k}$  can be calculated as

$$(\mathbf{H}_{N-k} \mathbf{H}_{N-k}^H)_{ii} = (\mathbf{H}_* \mathbf{H}_*^H)_{ii} - |g_i|^2, \quad (20)$$

where  $g_i$  is the  $i$ -th entry of  $\mathbf{g}$ . Since  $(\mathbf{H}_* \mathbf{H}_*^H)_{ii}$  has been calculated in the previous step, only 2 real multiplications and 2 real additions are needed for the calculation of  $(\mathbf{H}_{N-k} \mathbf{H}_{N-k}^H)_{ii}$ . The initial calculation of the squared 2-norm of rows of  $\mathbf{H}$  requires  $2NM$  real multiplications and  $N(M - 1) + NM$  real additions, so the additional real

multiplications and real additions needed in our proposed scheduling are

$$\begin{aligned} C_{ES}^\times &= 2NM + \sum_{k=1}^{M-1} 2(N - k) + \mathcal{O}(N) \\ &= 4MN - M^2 + \mathcal{O}(N), \end{aligned} \quad (21)$$

$$\begin{aligned} C_{ES}^+ &= N(M - 1) + NM + \sum_{k=1}^{M-1} 2(N - k) + \mathcal{O}(N) \\ &= 4MN - M^2 + \mathcal{O}(N). \end{aligned} \quad (22)$$

Let  $\alpha = M/N$ , which tends to 0 when  $N \gg M$ , then we have

$$\frac{C_{ES}}{C_{QR}} = \frac{1 - \alpha/4}{1 - \alpha/3} M^{-1} + \mathcal{O}(M^{-2}), \quad (23)$$

$$\frac{C_S}{C_{QR}} = \frac{1 - 2\alpha/3}{2 - 2\alpha/3} M + \mathcal{O}(M^0). \quad (24)$$

Note that the superscript (“ $\times$ ” or “ $+$ ”) is dropped because of the approximately equal number of multiplications and additions in each algorithm.

By using our proposed efficient greedy scheduling algorithm, we can thus reduce the overhead of scheduling from being the bottle neck of ZF-DPC to being negligible.

## V. CONCLUSIONS

Zero-forcing dirty paper coding (ZF-DPC) is used in vector Gaussian broadcast channel. To maximize the sum rate, a greedy scheduling is proposed in [8] to get a sub-optimal user ordering. In this paper we propose an efficient algorithm for this greedy scheduler by doing the scheduling within the QR factorization of the channel matrix, which is a necessary step in ZF-DPC. Thus, for a  $N$  receiver  $M$  transmitter system the computational complexity of scheduling is reduced from  $\mathcal{O}(NM^3)$  to  $\mathcal{O}(NM)$ , i.e., from being the bottleneck of ZF-DPC to being negligible. Our algorithm can also be employed by related greedy schedulers described in [9]–[11], but the computational complexity should be recalculated if the number of selected users is  $1 \leq K < M$ .

## REFERENCES

- [1] E. Telatar, “Capacity of multi-antenna Gaussian channels,” *European Trans. Telecommun.*, vol. 10, no. 6, pp. 585–595, 1999.
- [2] J. Yang and S. Roy, “On joint transmitter and receiver optimization for multiple-input-multiple-output (MIMO) transmission systems,” *IEEE Trans. Commun.*, vol. 42, no. 12, pp. 3221–3231, 1994.
- [3] M. H. M. Costa, “Writing on dirty paper,” *IEEE Trans. Inform. Theory*, vol. 29, no. 3, pp. 439–441, 1983.
- [4] H. Weingarten, Y. Steinberg, and S. S. Shamai, “The capacity region of the Gaussian multiple-input multiple-output broadcast channel,” *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 3936–3964, 2006.
- [5] P. Viswanath and D. N. Tse, “Sum capacity of the vector Gaussian broadcast channel and downlink-uplink duality,” *IEEE Trans. Inform. Theory*, vol. 49, no. 8, pp. 1912–1921, 2003.
- [6] S. Vishwanath, N. Jindal, and A. Goldsmith, “Duality, achievable rates, and sum-rate capacity of Gaussian MIMO broadcast channels,” *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2658–2668, 2003.
- [7] G. Caire and S. Shamai, “On the achievable throughput of a multi-antenna Gaussian broadcast channel,” *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1691–1706, 2003.
- [8] Z. Tu and R. S. Blum, “Multiuser diversity for a dirty paper approach,” *IEEE Commun. Lett.*, vol. 7, no. 8, pp. 370–372, 2003.

- [9] J. Jiang, R. M. Buehrer, and W. H. Tranter, "Greedy scheduling performance for a zero-forcing dirty-paper coded system," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 789–793, 2006.
- [10] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 528–541, 2006.
- [11] G. Dimic and N. D. Sidiropoulos, "On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm," *IEEE Trans. Signal Processing*, vol. 53, no. 10, pp. 3857–3868, 2005.
- [12] K. L. Chung and W. M. Yan, "The complex Householder transform," *IEEE Trans. Signal Processing*, vol. 45, no. 9, pp. 2374–2376, 1997.
- [13] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.