

Sum Power Iterative Water-filling for Multi-Antenna Gaussian Broadcast Channels

Nihar Jindal, *Member, IEEE*, Wonjong Rhee, *Member, IEEE*, Sriram Vishwanath, *Member, IEEE*, Syed Jafar, *Member, IEEE*, and Andrea Goldsmith *Fellow, IEEE*,

Abstract

In this paper we consider the problem of maximizing sum rate of a multiple-antenna Gaussian broadcast channel. It was recently found that dirty paper coding is capacity achieving for this channel. In order to achieve capacity, the optimal transmission policy (i.e. the optimal transmit covariance structure) given the channel conditions and power constraint must be found. However, obtaining the optimal transmission policy when employing dirty paper coding is a computationally complex non-convex problem. We use duality to transform this problem into a well-structured convex multiple-access channel problem. We exploit the structure of this problem and derive simple and fast iterative algorithms that provide the optimum transmission policies for the multiple-access channel, which can easily be mapped to the optimal broadcast channel policies.

Index Terms

Broadcast channel, Dirty paper coding, Multiple-access channel, Multiple-input multiple-output (MIMO) Systems, Duality

This manuscript will appear in *IEEE Transactions on Information Theory* in 2005. Manuscript submitted to *IEEE Transactions on Information Theory* on July 21, 2004 and revised on December 15, 2004. The material in this paper was presented in part at the *International Symposium on Information Theory*, July, 2003 in Yokohama, Japan [11] and at the *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2002 in Asilomar, CA [6].

This work was initiated while the authors were at Stanford University. The work of some authors was supported by the Stanford Networking Research Center. Nihar Jindal (nihar@ece.umn.edu) is now at Univ. of Minnesota, Wonjong Rhee (wonjong@dsl.stanford.edu) is at Assia Inc, Sriram Vishwanath (sriram@ece.utexas.edu) is at Univ. of Texas, Austin, Syed Jafar (syed@ece.uci.edu) is at Univ. of California, Irvine, and Andrea Goldsmith (andrea@systems.stanford.edu) is at Stanford University.

I. INTRODUCTION

There has been a great interest in characterizing and computing the capacity region of multiple-antenna broadcast (downlink) channels in recent years. An achievable region for the multiple-antenna downlink channel was found in [3], and this achievable region was shown to achieve the sum rate capacity in [3], [10], [12], [16], and was more recently shown to achieve the full capacity region in [14]. Though these results show that the general dirty paper coding strategy is optimal, one must still optimize over the transmit covariance structure (i.e. how transmissions over different antennas should be correlated) in order to determine the optimal transmission policy and the corresponding sum rate capacity. Unlike the single antenna broadcast channel, sum capacity is not in general achieved by transmitting to a single user. Thus, the problem cannot be reduced to a point-to-point MIMO problem, for which simple expressions are known. Furthermore, the direct optimization for sum rate capacity is a computationally complex non-convex problem. Therefore, obtaining the optimal rates and transmission policy is difficult¹.

A duality technique presented in [7], [10] transforms the non-convex downlink problem into a convex sum power *uplink* (multiple-access channel, or MAC) problem, which is much easier to solve, from which the optimal downlink covariance matrices can be found. Thus, in this paper we find efficient algorithms to find the sum capacity of the uplink channel, i.e. to solve the following convex optimization problem:

$$\max_{\{\mathbf{Q}_i\}_{i=1}^K: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \log \left| \mathbf{I} + \sum_{i=1}^K \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \right|. \quad (1)$$

In this sum power MAC problem, the users in the system have a joint power constraint instead of individual constraints as in the conventional MAC. As in the case of the conventional MAC, there exist standard interior point convex optimization algorithms [2] that solve (1). An interior point algorithm, however, is considerably more complex than our algorithms and does not scale well when there are large numbers of users. Recent work by Lan and Yu based on minimax optimization techniques appears to be promising but suffers from much higher complexity than our algorithms [8]. A steepest-descent method was proposed by Viswanathan, et. al. [13], and an alternative, dual decomposition based algorithm was proposed by Yu in [15]. The complexity of these two algorithms is on the same order as the complexity of the algorithms proposed here. However, we find our algorithm to converge more rapidly, and our algorithm is also considerably more intuitive than either of these approaches. In this paper, we exploit the structure

¹In the single transmit antenna broadcast channel, there is a similar non-convex optimization problem. However, it is easily seen that it is optimal to transmit with full power to only the user with the strongest channel. Such a policy is, however, not the optimal policy when the transmitter has multiple antennas.

of the sum capacity problem to obtain simple iterative algorithms for calculating sum capacity², i.e. for computing (1). This algorithm is inspired by and is very similar to the iterative waterfilling algorithm for the conventional individual power constraint MAC problem by Yu, Rhee, Boyd and Cioffi [17].

This paper is structured as follows. In the next section, the system model is presented. In Section III, expressions for the sum capacity of the downlink and dual uplink channels are stated. In Section IV, the basic iterative water-filling algorithm for the multiple-access channel is proposed and proven to converge when there are only two receivers. In Sections VI and VII two modified versions of this algorithm are proposed and shown to converge for any number of users. Complexity analyses of the algorithms are presented in Section VIII, followed by numerical results and conclusions in Sections IX and X, respectively.

II. SYSTEM MODEL

We consider a K user MIMO Gaussian broadcast channel (abbreviated as MIMO BC) where the transmitter has M antennas and each receiver has N antennas³. The downlink channel is shown in Figure 1 along with the *dual* uplink channel. The dual uplink channel is a K user multiple antenna uplink channel (abbreviated as MIMO MAC) where each of the dual uplink channels is the conjugate transpose of the corresponding downlink channel. The downlink and uplink channel are mathematically described as:

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x} + \mathbf{n}_i, \quad i = 1, \dots, K \quad \text{Downlink channel} \quad (2)$$

$$\mathbf{y}_{MAC} = \sum_{i=1}^K \mathbf{H}_i^\dagger \mathbf{x}_i + \mathbf{n} \quad \text{Dual uplink channel} \quad (3)$$

where $\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_K$ are the channel matrices (with $\mathbf{H}_i \in \mathbb{C}^{N \times M}$) of users 1 through K respectively on the downlink, the vector $\mathbf{x} \in \mathbb{C}^{M \times 1}$ is the downlink transmitted signal, and $\mathbf{x}_1, \dots, \mathbf{x}_K$ (with $\mathbf{x}_i \in \mathbb{C}^{N \times 1}$) are the transmitted signals in the uplink channel. This work applies only to the scenario where the channel matrices are fixed and are all known to the transmitter and to each receiver. In fact, this is the only scenario for which capacity results for the MIMO BC are known. The vectors $\mathbf{n}_1, \dots, \mathbf{n}_K$ and \mathbf{n} refer to independent additive Gaussian noise with unit variance on each vector component. We assume

²To compute other points on the boundary of the capacity region (i.e. non sum-capacity rate vectors), the algorithms in either [13] or [8] can be used

³We assume all receivers have the same number of antennas for simplicity. However, all algorithms easily generalize to the scenario where each receiver can have a different number of antennas.

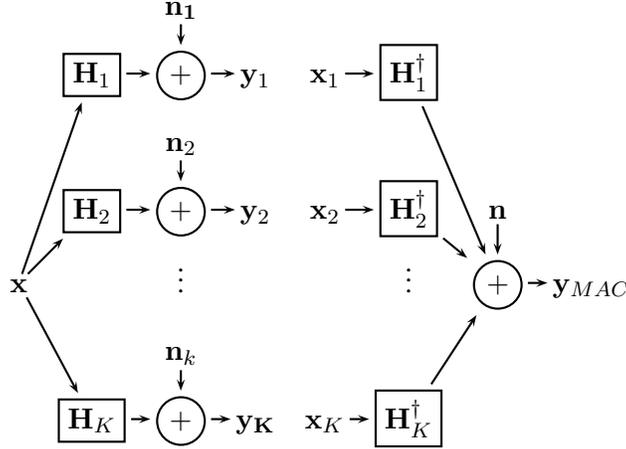


Fig. 1. System models of the MIMO BC(left) and the MIMO MAC (right) channels

there is a sum power constraint of P in the MIMO BC (i.e. $E[||\mathbf{x}||^2] \leq P$) and in the MIMO MAC (i.e. $\sum_{i=1}^K E[||\mathbf{x}_i||^2] \leq P$). Though the computation of the sum capacity of the MIMO BC is of interest, we work with the dual MAC, which is computationally much easier to solve, instead.

Notation: We use boldface to denote vectors and matrices. The function $[\cdot]_K$ is defined as $[x]_K \triangleq (x - 1 \bmod K) + 1$, i.e. $[0]_K = K$, $[1]_K = 1$, $[K]_K = K$, and so forth.

III. SUM RATE CAPACITY

In [3], [10], [12], [16], the sum rate capacity of the MIMO BC (denoted as $\mathcal{C}_{\text{BC}}(\mathbf{H}_1, \dots, \mathbf{H}_K, P)$) was shown to be achievable by dirty-paper coding [4]. From these results, the sum rate capacity can be written in terms of the following maximization:

$$\mathcal{C}_{\text{BC}}(\mathbf{H}_1, \dots, \mathbf{H}_K, P) = \max_{\{\boldsymbol{\Sigma}_i\}_{i=1}^K: \boldsymbol{\Sigma}_i \geq 0, \sum_{i=1}^K \text{Tr}(\boldsymbol{\Sigma}_i) \leq P} \log \left| \mathbf{I} + \mathbf{H}_1 \boldsymbol{\Sigma}_1 \mathbf{H}_1^\dagger \right| + \log \frac{\left| \mathbf{I} + \mathbf{H}_2 (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \mathbf{H}_2^\dagger \right|}{\left| \mathbf{I} + \mathbf{H}_2 \boldsymbol{\Sigma}_1 \mathbf{H}_2^\dagger \right|} + \dots + \log \frac{\left| \mathbf{I} + \mathbf{H}_K (\boldsymbol{\Sigma}_1 + \dots + \boldsymbol{\Sigma}_K) \mathbf{H}_K^\dagger \right|}{\left| \mathbf{I} + \mathbf{H}_K (\boldsymbol{\Sigma}_1 + \dots + \boldsymbol{\Sigma}_{K-1}) \mathbf{H}_K^\dagger \right|}. \quad (4)$$

The maximization is performed over downlink covariance matrices $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K$, each of which is a $M \times M$ positive semi-definite matrix. In this paper we are interested in finding the covariance matrices that achieve this maximum. It is easily seen that the objective (4) is not a concave function of $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K$. Thus, numerically finding the maximum is a non-trivial problem. However, in [10], a *duality* is shown to exist between the uplink and downlink which establishes that the dirty paper rate region for the MIMO

BC is equal to the capacity region of the dual MIMO MAC (described in (3)). This implies that the sum capacity of the MIMO BC is equal to the sum capacity of the dual MIMO MAC (denoted as $\mathcal{C}_{\text{MAC}}(\mathbf{H}_1, \dots, \mathbf{H}_K, P)$), i.e.

$$\mathcal{C}_{\text{BC}}(\mathbf{H}_1, \dots, \mathbf{H}_K, P) = \mathcal{C}_{\text{MAC}}(\mathbf{H}_1^\dagger, \dots, \mathbf{H}_K^\dagger, P). \quad (5)$$

The sum rate capacity of the MIMO MAC is given by the following expression [10]:

$$\mathcal{C}_{\text{MAC}}(\mathbf{H}_1^\dagger, \dots, \mathbf{H}_K^\dagger, P) = \max_{\{\mathbf{Q}_i\}_{i=1}^K: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \log \left| \mathbf{I} + \sum_{i=1}^K \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \right|, \quad (6)$$

where the maximization is performed over uplink covariance matrices $\mathbf{Q}_1, \dots, \mathbf{Q}_K$ (\mathbf{Q}_i is an $N \times N$ positive semi-definite matrix), subject to power constraint P . The objective in (6) is a concave function of the covariance matrices. Furthermore, in [10, Equations 8-10], a transformation is provided (this mapping is reproduced in Appendix I for convenience) that maps from uplink covariance matrices to downlink covariance matrices (i.e. from $\mathbf{Q}_1, \dots, \mathbf{Q}_K$ to $\Sigma_1, \dots, \Sigma_K$) that achieve the same rates and use the same sum power. Therefore, finding the optimal uplink covariance matrices leads directly to the optimal downlink covariance matrices.

In this paper, we develop specialized algorithms that efficiently compute (6). These algorithm converge, and utilize the water-filling structure of the optimal solution, first identified for the individual power constraint MAC in [17]. Note that the maximization in (6) is not guaranteed to have a unique solution, though it seems that uniqueness holds nearly all channel realizations. See [17] for a discussion of this same property for the individual power constraint MAC. Therefore, we are interested in finding any maximizing solution to the optimization.

IV. ITERATIVE WATER-FILLING WITH INDIVIDUAL POWER CONSTRAINTS

The iterative water-filling algorithm for the conventional MIMO MAC problem was obtained by Yu, Rhee, Boyd, and Cioffi in [17]. This algorithm finds the sum capacity of a MIMO MAC with *individual* power constraints P_1, \dots, P_K on each user, which is equal to:

$$\mathcal{C}_{\text{MAC}}(\mathbf{H}_1^\dagger, \dots, \mathbf{H}_K^\dagger, P_1, \dots, P_K) = \max_{\{\mathbf{Q}_i\}_{i=1}^K: \mathbf{Q}_i \geq 0, \text{Tr}(\mathbf{Q}_i) \leq P_i} \log \left| \mathbf{I} + \sum_{i=1}^K \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \right|. \quad (7)$$

This differs from (6) only in the power constraint structure. Notice that the objective is a concave function of the covariance matrices, and that the constraints in (7) are *separable* because there is an individual trace constraint on each covariance matrix. For such problems, it is generally sufficient to optimize with respect to the first variable while holding all other variables constant, then optimize with respect to the second

variable, etc., in order to reach a globally optimum point. This is referred to as the block-coordinate ascent algorithm and convergence can be shown under relatively general conditions [1, Section 2.7]. If we define the function $f(\cdot)$ as

$$f(\mathbf{Q}_1, \dots, \mathbf{Q}_K) \triangleq \log \left| \mathbf{I} + \sum_{i=1}^K \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \right|, \quad (8)$$

then in the $(n+1)$ -th iteration of the block-coordinate ascent algorithm,

$$\mathbf{Q}_i^{(n+1)} \triangleq \arg \max_{\mathbf{Q}_i: \mathbf{Q}_i \geq 0, \text{Tr}(\mathbf{Q}_i) \leq P_i} f(\mathbf{Q}_1^{(n)}, \dots, \mathbf{Q}_{i-1}^{(n)}, \mathbf{Q}_i, \mathbf{Q}_{i+1}^{(n)}, \dots, \mathbf{Q}_K^{(n)}) \quad (9)$$

for $i = [n]_K$ and $\mathbf{Q}_i^{(n+1)} = \mathbf{Q}_i^{(n)}$ for $i \neq [n]_K$. Notice that only one of the covariances is updated in each iteration.

The key to the iterative water-filling algorithm is noticing that $f(\mathbf{Q}_1, \dots, \mathbf{Q}_K)$ can be rewritten as:

$$\begin{aligned} f(\mathbf{Q}_1, \dots, \mathbf{Q}_K) &= \log \left| \mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j \mathbf{H}_j + \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \right| \\ &= \log \left| \mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j \mathbf{H}_j \right| + \\ &\quad \log \left| \mathbf{I} + \left(\mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j \mathbf{H}_j \right)^{-1/2} \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \left(\mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j \mathbf{H}_j \right)^{-1/2} \right| \end{aligned}$$

for any i , where we have used the property $|\mathbf{A}\mathbf{B}| = |\mathbf{A}||\mathbf{B}|$. Therefore, the maximization in (9) is equivalent to the calculation of the capacity of a point-to-point MIMO channel with channel $\mathbf{G}_i = \mathbf{H}_i \left(\mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j^{(n)} \mathbf{H}_j \right)^{-1/2}$:

$$\mathbf{Q}_i^{(n+1)} = \arg \max_{\mathbf{Q}_i: \mathbf{Q}_i \geq 0, \text{Tr}(\mathbf{Q}_i) \leq P_i} \log \left| \mathbf{I} + \mathbf{G}_i^\dagger \mathbf{Q}_i \mathbf{G}_i \right|. \quad (10)$$

It is well known that the capacity of a point-to-point MIMO channel is achieved by choosing the input covariance along the eigenvectors of the channel matrix and by water-filling on the eigenvalues of the channel matrix [9]. Thus, $\mathbf{Q}_i^{(n+1)}$ should be chosen as a *water-fill* of the channel \mathbf{G}_i , i.e. the eigenvectors of $\mathbf{Q}_i^{(n+1)}$ should equal the left eigenvectors of \mathbf{G}_i , with the eigenvalues chosen by the water-filling procedure.

At each step of the algorithm, exactly one user optimizes his covariance matrix while treating the signals from all other users as noise. In the next step, the next user (in numerical order) optimizes his covariance while treating all other signals, including the updated covariance of the previous user, as noise. This intuitively appealing algorithm can easily be shown to satisfy the conditions of [1, Section

2.7] and thus provably converges. Furthermore, the optimization in each step of the algorithm simplifies to water-filling over an effective channel, which is computationally efficient.

If we let $\mathbf{Q}_1^*, \dots, \mathbf{Q}_K^*$ denote the optimal covariances, then optimality implies:

$$f(\mathbf{Q}_1^*, \dots, \mathbf{Q}_K^*) = \max_{\mathbf{Q}_i: \mathbf{Q}_i \geq 0, \text{Tr}(\mathbf{Q}_i) \leq P_i} f(\mathbf{Q}_1^*, \dots, \mathbf{Q}_{i-1}^*, \mathbf{Q}_i, \mathbf{Q}_{i+1}^*, \dots, \mathbf{Q}_K^*). \quad (11)$$

for any i . Thus, \mathbf{Q}_1^* is a water-fill of the noise and the signals from all other users (i.e. is a waterfill of the channel $\mathbf{H}_1(\mathbf{I} + \sum_{j \neq 1} \mathbf{H}_j^\dagger \mathbf{Q}_j^* \mathbf{H}_j)^{-1/2}$), while \mathbf{Q}_2^* is simultaneously a water-fill of the noise and the signals from all other users, and so forth. Thus, the sum capacity achieving covariance matrices *simultaneously* water-fill each of their respective effective channels [17], with the water-filling levels (i.e. the eigenvalues) of each user determined by the power constraints P_j . In the next section, we will see that similar intuition describes the sum capacity achieving covariance matrices in the MIMO MAC when there is a sum power constraint instead of individual power constraints.

V. SUM POWER ITERATIVE WATER-FILLING

In the previous section we described the iterative water-filling algorithm that computes the sum capacity of the MIMO MAC subject to individual power constraints [17]. We are instead concerned with computing the sum capacity, along with the corresponding optimal covariance matrices, of a MIMO BC. As stated earlier, this is equivalent to computing the sum capacity of a MIMO MAC subject to a sum power constraint, i.e. computing:

$$\mathcal{C}_{\text{MAC}}(\mathbf{H}_1^\dagger, \dots, \mathbf{H}_K^\dagger, P) = \max_{\{\mathbf{Q}_i\}_{i=1}^K: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \log \left| \mathbf{I} + \sum_{i=1}^K \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i \right|. \quad (12)$$

If we let $\mathbf{Q}_1^*, \dots, \mathbf{Q}_K^*$ denote a set of covariance matrices that achieve the above maximum, it is easy to see that similar to the individual power constraint problem, each covariance must be a water-fill of the noise and signals from all other users. More precisely, this means that for every j , the eigenvectors of \mathbf{Q}_i^* are aligned with the left eigenvectors of $\mathbf{H}_i \left(\mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j^* \mathbf{H}_j \right)^{-1/2}$ and that the eigenvalues of \mathbf{Q}_i^* must satisfy the water-filling condition. However, since there is a *sum* power constraint on the covariances, the water level of all users must be equal. This is akin to saying that no advantage will be gained by transferring power from one user with a higher water-filling level to another user with a lower water-filling level. Note that this is different from the individual power constraint problem, where the water level of each user was determined individually and could differ from user to user. In the individual power constraint channel, since each user's water-filling level was determined by his own power constraint, the covariances of each user could be updated one at a time. With a sum power constraint, however, we must update all covariances *simultaneously* to maintain a constant water-level.

Motivated by the individual power algorithm, we propose the following algorithm in which all K covariances are simultaneously updated during each step, based on the covariance matrices from the previous step. This is a natural extension of the per-user sequential update described in Section IV. At each iteration step we generate an effective channel for *each* user based on the covariances (from the previous step) of all other users. In order to maintain a common water-level, we simultaneously water-fill across all K effective channels, i.e. we maximize the sum of rates on the K effective channels. The n -th iteration of the algorithm is described by the following:

- 1) Generate effective channels

$$\mathbf{G}_i^{(n)} = \mathbf{H}_i \left(\mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j^{(n-1)} \mathbf{H}_j \right)^{-1/2} \quad (13)$$

for $i = 1, \dots, K$.

- 2) Treating these effective channels as parallel, non-interfering channels, obtain the new covariance matrices $\{\mathbf{Q}_i^{(n)}\}_{i=1}^K$ by water-filling with total power P :

$$\{\mathbf{Q}_i^{(n)}\}_{i=1}^K = \arg \max_{\{\mathbf{Q}_i\}_{i=1}^K: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \sum_{i=1}^K \log \left| \mathbf{I} + \left(\mathbf{G}_i^{(n)} \right)^\dagger \mathbf{Q}_i \mathbf{G}_i^{(n)} \right|.$$

This maximization is equivalent to water-filling the block diagonal channel with diagonals equal to $\mathbf{G}_1^{(n)}, \dots, \mathbf{G}_K^{(n)}$. If the SVD of $\mathbf{G}_i^{(n)} \left(\mathbf{G}_i^{(n)} \right)^\dagger$ is written as $\mathbf{G}_i^{(n)} \left(\mathbf{G}_i^{(n)} \right)^\dagger = \mathbf{U}_i \mathbf{D}_i \mathbf{U}_i^\dagger$ with \mathbf{U}_i unitary and \mathbf{D}_i square and diagonal, then the updated covariance matrices are given by:

$$\mathbf{Q}_i^{(n)} = \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i^\dagger \quad (14)$$

where $\mathbf{\Lambda}_i = [\mu \mathbf{I} - (\mathbf{D}_i)^{-1}]^+$ and the operation $[\mathbf{A}]^+$ denotes a component-wise maximum with zero. Here the water-filling level μ is chosen such that $\sum_{i=1}^K \text{Tr}(\mathbf{\Lambda}_i) = P$.

We refer to this as the *original algorithm* [6]. This simple and highly intuitive algorithm does in fact converge to the sum rate capacity when $K = 2$, as we show next:

Theorem 1: The sum power iterative water-filling algorithm converges to the sum rate capacity of the MAC when $K = 2$.

Proof: In order to prove convergence of the algorithm for $K = 2$, consider the following related optimization problem:

$$\begin{aligned} \max_{\mathbf{A}_1, \mathbf{A}_2 \geq 0, \mathbf{B}_1, \mathbf{B}_2 \geq 0, \text{Tr}(\mathbf{A}_1 + \mathbf{A}_2) \leq P, \text{Tr}(\mathbf{B}_1 + \mathbf{B}_2) \leq P} & \frac{1}{2} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{A}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{B}_2 \mathbf{H}_2 \right| \\ & + \frac{1}{2} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{B}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{A}_2 \mathbf{H}_2 \right|. \end{aligned} \quad (15)$$

We first show that the solutions to the original sum rate maximization problem in (12) and (15) are the same. If we define $\mathbf{A}_1 = \mathbf{B}_1 = \mathbf{Q}_1$ and $\mathbf{A}_2 = \mathbf{B}_2 = \mathbf{Q}_2$, we see that any sum rate achievable in (12) is also achievable in the modified sum rate in (15). Furthermore, if we define $\mathbf{Q}_1 = \frac{1}{2}(\mathbf{A}_1 + \mathbf{B}_1)$ and $\mathbf{Q}_2 = \frac{1}{2}(\mathbf{A}_2 + \mathbf{B}_2)$, we have:

$$\log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{Q}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{Q}_2 \mathbf{H}_2 \right| \geq \frac{1}{2} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{A}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{B}_2 \mathbf{H}_2 \right| + \frac{1}{2} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{B}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{A}_2 \mathbf{H}_2 \right|$$

due to the concavity of $\log(\det(\cdot))$. Since $\text{Tr}(\mathbf{Q}_1) + \text{Tr}(\mathbf{Q}_2) = \frac{1}{2}\text{Tr}(\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{B}_1 + \mathbf{B}_2) \leq P$, any sum rate achievable in (15) is also achievable in the original (12). Thus, every set of maximizing covariances $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2)$ map directly to a set of maximizing $(\mathbf{Q}_1, \mathbf{Q}_2)$. Therefore, we can equivalently solve (15) to find the uplink covariances that maximize the sum-rate expression in (12).

Now notice that the maximization in (15) has separable constraints on $(\mathbf{A}_1, \mathbf{A}_2)$ and $(\mathbf{B}_1, \mathbf{B}_2)$. Thus, we can use the block coordinate ascent method in which we maximize with respect to $(\mathbf{A}_1, \mathbf{A}_2)$ while holding $(\mathbf{B}_1, \mathbf{B}_2)$ fixed, then with respect to $(\mathbf{B}_1, \mathbf{B}_2)$ while holding $(\mathbf{A}_1, \mathbf{A}_2)$ fixed, and so on. The maximization of (15) with respect to $(\mathbf{A}_1, \mathbf{A}_2)$ can be written as:

$$\max_{\mathbf{A}_1, \mathbf{A}_2 \geq 0, \text{Tr}(\mathbf{A}_1 + \mathbf{A}_2) \leq P} \log \left| \mathbf{I} + \mathbf{G}_1^\dagger \mathbf{A}_1 \mathbf{G}_1 \right| + \log \left| \mathbf{I} + \mathbf{G}_2^\dagger \mathbf{A}_2 \mathbf{G}_2 \right| \quad (16)$$

where $\mathbf{G}_1 = \mathbf{H}_1(\mathbf{I} + \mathbf{H}_2^\dagger \mathbf{B}_2 \mathbf{H}_2)^{-1/2}$ and $\mathbf{G}_2 = \mathbf{H}_2(\mathbf{I} + \mathbf{H}_1^\dagger \mathbf{B}_1 \mathbf{H}_1)^{-1/2}$. Clearly, this is equivalent to the iterative water-filling step described in the previous section where $(\mathbf{B}_1, \mathbf{B}_2)$ play the role of the covariance matrices from the previous step. Similarly, when maximizing with respect to $(\mathbf{B}_1, \mathbf{B}_2)$, the covariances $(\mathbf{A}_1, \mathbf{A}_2)$ are the covariance matrices from the previous step. Therefore, performing the cyclic coordinate ascent algorithm on (15) is equivalent to the sum power iterative water-filling algorithm described in Section V.

Furthermore, notice that each iteration is equal to the calculation of the capacity of a point-to-point (block-diagonal) MIMO channel. Water-filling is known to be optimal in this setting, and in Appendix II we show that the water-filling solution is the unique solution. Therefore, by [18, pg. 228] [1, Chapter 2.7], the block coordinate ascent algorithm converges because at each step of the algorithm there is a unique maximizing solution. Thus, the iterative water-filling algorithm given in Section V converges to the maximum sum rate when $K = 2$. ■

However, rather surprisingly, this algorithm does not always converge to the optimum when $K > 2$, and the algorithm can even lead to a strict decrease in the objective function. In the following sections, we

provide modified versions of this algorithm that do converge for all K .

VI. MODIFIED ALGORITHM

In this section we present a modified version of the sum power iterative water-filling algorithm and prove that it converges to the sum capacity for any number of users K . This modification is motivated by the proof of convergence of the original algorithm for $K = 2$. In the proof of Theorem 1, a sum of two log det functions, with four input covariances is considered instead of the original log *det* function. We then applied the provably convergent cyclic coordinate ascent algorithm, and saw that this algorithm is in fact identical to the sum power iterative algorithm. When there are more than 2 users (i.e. $K > 2$) we can consider a similar sum of K log *det* functions, and again perform the cyclic coordinate ascent algorithm to provably converge to the sum rate capacity. In this case, however, the cyclic coordinate ascent algorithm is not identical to the original sum power iterative water-filling algorithm. It can, however, be interpreted as the sum power iterative water-filling algorithm with a memory of the covariance matrices generated in the previous $K - 1$ iterations, instead of just in the previous iteration.

For simplicity, let us consider the $K = 3$ scenario. Similar to the proof of Theorem 1, consider the following maximization:

$$\begin{aligned} \max \quad & \frac{1}{3} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{A}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{B}_2 \mathbf{H}_2 + \mathbf{H}_3^\dagger \mathbf{C}_3 \mathbf{H}_3 \right| \\ & + \frac{1}{3} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{C}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{A}_2 \mathbf{H}_2 + \mathbf{H}_3^\dagger \mathbf{B}_3 \mathbf{H}_3 \right| \\ & + \frac{1}{3} \log \left| \mathbf{I} + \mathbf{H}_1^\dagger \mathbf{B}_1 \mathbf{H}_1 + \mathbf{H}_2^\dagger \mathbf{C}_2 \mathbf{H}_2 + \mathbf{H}_3^\dagger \mathbf{A}_3 \mathbf{H}_3 \right| \end{aligned} \quad (17)$$

subject to the constraints $\mathbf{A}_i \geq 0$, $\mathbf{B}_i \geq 0$, $\mathbf{C}_i \geq 0$ for $i = 1, 2, 3$, $\text{Tr}(\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3) \leq P$, $\text{Tr}(\mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3) \leq P$, and $\text{Tr}(\mathbf{C}_1 + \mathbf{C}_2 + \mathbf{C}_3) \leq P$. By the same argument used for the two user case, any solution to the above maximization corresponds to a solution to the original optimization problem in (12). In order to maximize (17), we can again use the cyclic coordinate ascent algorithm. We first maximize with respect to $\mathbf{A} \triangleq (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$, then with respect to $\mathbf{B} \triangleq (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)$, then with respect to $\mathbf{C} \triangleq (\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3)$, and so forth. As before, convergence is guaranteed due to the uniqueness of the maximizing solution in each step [1, Section 2.7]. In the two user case, the cyclic coordinate ascent method applied to the modified optimization problem yields the same iterative water-filling algorithm proposed in Section V where the effective user of each channel is based on the covariance matrices only from the previous step. In general, however, the effective channel of each user depends on covariances which are up to $K - 1$ steps old.

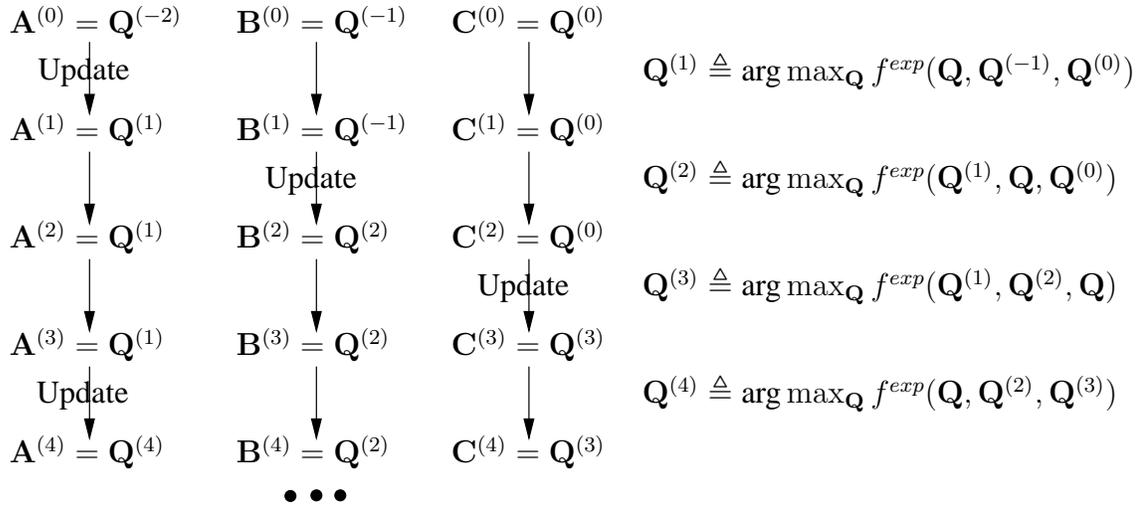


Fig. 2. Graphical representation of Algorithm 1

A graphical representation of the algorithm for three users is shown in Fig. 2. Here $\mathbf{A}^{(n)}$ refers to the triplet of matrices $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ after the n -th iterate. Furthermore, the function $f^{exp}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ refers to the objective function in (17). We begin by initializing all variables to some $\mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)}$. In order to develop a more general form that generalizes to arbitrary K , we also refer to these variables as $\mathbf{Q}^{(-2)}, \mathbf{Q}^{(-1)}, \mathbf{Q}^{(0)}$. Note that each of these variables refers to a triplet of covariance matrices. In step 1, \mathbf{A} is updated while holding variables \mathbf{B} and \mathbf{C} constant, and we define $\mathbf{Q}^{(1)}$ to be the updated variable $\mathbf{A}^{(1)}$:

$$\mathbf{Q}^{(1)} \triangleq \mathbf{A}^{(1)} = \arg \max_{\mathbf{Q}: \mathbf{Q}_i \geq 0, \sum_{i=1}^3 \text{Tr}(\mathbf{Q}_i) \leq P} f^{exp}(\mathbf{Q}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)}) \quad (18)$$

$$= \arg \max_{\mathbf{Q}: \mathbf{Q}_i \geq 0, \sum_{i=1}^3 \text{Tr}(\mathbf{Q}_i) \leq P} f^{exp}(\mathbf{Q}, \mathbf{Q}^{(-1)}, \mathbf{Q}^{(0)}) \quad (19)$$

In step 2, the matrices \mathbf{B} are updated with $\mathbf{Q}^{(2)} \triangleq \mathbf{B}^{(2)}$, and in step 3 the matrices \mathbf{C} are updated with $\mathbf{Q}^{(3)} \triangleq \mathbf{C}^{(3)}$. The algorithm continues cyclically, i.e. in step 4 \mathbf{A} is again updated, and so forth. Notice that $\mathbf{Q}^{(n)}$ is always defined to be the set of matrices updated in the n -th iteration.

In Appendix III we show that the following is a general formula for $\mathbf{Q}^{(n)}$:

$$\mathbf{Q}^{(n)} = \arg \max_{\mathbf{Q}: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} f^{exp}(\mathbf{Q}, \mathbf{Q}^{(n-K+1)}, \dots, \mathbf{Q}^{(n-1)}) \quad (20)$$

$$= \arg \max_{\mathbf{Q}: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \sum_{i=1}^K \log \left| \mathbf{I} + \left(\mathbf{G}_i^{(n)} \right)^\dagger \mathbf{Q}_i \mathbf{G}_i^{(n)} \right|. \quad (21)$$

where the effective channel of User i in the n -th step is:

$$\mathbf{G}_i^{(n)} = \mathbf{H}_i \left(\mathbf{I} + \sum_{j=1}^{K-1} \mathbf{H}_{[i+j]_K}^\dagger \mathbf{Q}_{[i+j]_K}^{(n-K+j)} \mathbf{H}_{[i+j]_K} \right)^{-1/2} \quad (22)$$

where $[x]_K = \text{mod}(x-1, K)+1$. Clearly, the previous $K-1$ states of the algorithm (i.e. $\mathbf{Q}^{(n-K+1)}, \dots, \mathbf{Q}^{(n-1)}$) must be kept in memory in order to generate these effective channels.

We now explicitly state the steps of Algorithm 1. The covariances are first initialized to scaled versions of the identity⁴, i.e. $\mathbf{Q}_j^{(n)} = \frac{P}{KN} \mathbf{I}$ for $j = 1, \dots, K$ and $n = -(K-2), \dots, 0$. The algorithm is almost identical to the original sum power iterative algorithm, with the exception that the expression for each effective channel now depends on covariance matrices generated in the previous $K-1$ steps, instead of just on the previous step:

- 1) Generate effective channels

$$\mathbf{G}_i^{(n)} = \mathbf{H}_i \left(\mathbf{I} + \sum_{j=1}^{K-1} \mathbf{H}_{[i+j]_K}^\dagger \mathbf{Q}_{[i+j]_K}^{(n-K+j)} \mathbf{H}_{[i+j]_K} \right)^{-1/2} \quad (23)$$

for $i = 1, \dots, K$.

- 2) Treating these effective channels as parallel, non-interfering channels, obtain the new covariance matrices $\{\mathbf{Q}_i^{(n)}\}_{i=1}^K$ by water-filling with total power P :

$$\{\mathbf{Q}_i^{(n)}\}_{i=1}^K = \arg \max_{\{\mathbf{Q}_i\}_{i=1}^K: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \sum_{i=1}^K \log \left| \mathbf{I} + (\mathbf{G}_i^{(n)})^\dagger \mathbf{Q}_i \mathbf{G}_i^{(n)} \right|.$$

We refer to this as *Algorithm 1*. Next we prove convergence to the sum rate capacity:

Theorem 2: Algorithm 1 converges to the sum rate capacity for any K .

Proof: Convergence is shown by noting that the algorithm is the cyclic coordinate ascent algorithm applied to the function $f^{exp}(\cdot)$. Since there is a unique (water-filling) solution to the maximization in step 2, the algorithm converges to the sum capacity of the channel for any number of users K .⁵ More precisely, convergence occurs in the objective of the expanded function:

$$\lim_{n \rightarrow \infty} f^{exp} \left(\mathbf{Q}^{(n-K+1)}, \dots, \mathbf{Q}^{(n)} \right) = \mathcal{C}_{\text{MAC}}(\mathbf{H}_1^\dagger, \dots, \mathbf{H}_K^\dagger, P) \quad (24)$$

⁴The algorithm converges from *any* starting point, but for simplicity we have chosen to initialize using the identity covariance. In Section IX we discuss the large advantage gained by using the original algorithm for a few iterations to generate a considerably better starting point.

⁵Notice that the modified algorithm and the original algorithm in V are equivalent only for $K = 2$.

A graphical representation of the new algorithm (referred to as Algorithm 2 herein) for $K = 3$ is provided in Fig. 3. Notice that the initialization matrices are chosen to be all equal. As in algorithm 1, in the first step \mathbf{A} is updated to give the temporary variable $\mathbf{S}^{(1)}$. In algorithm 1, we would assign $(\mathbf{A}^{(1)}, \mathbf{B}^{(1)}, \mathbf{C}^{(1)}) = (\mathbf{S}^{(1)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)})$, and then continue by updating \mathbf{B} , and so forth. In algorithm 2, however, before performing the next update (i.e. before updating \mathbf{B}), the three variables are *averaged* to give $\mathbf{Q}^{(1)} \triangleq \frac{1}{3}(\mathbf{S}^{(1)} + \mathbf{Q}^{(0)} + \mathbf{Q}^{(0)}) = \frac{1}{3}\mathbf{S}^{(1)} + \frac{2}{3}\mathbf{Q}^{(0)}$, and we set $(\mathbf{A}^{(1)}, \mathbf{B}^{(1)}, \mathbf{C}^{(1)}) = (\mathbf{Q}^{(1)}, \mathbf{Q}^{(1)}, \mathbf{Q}^{(1)})$. Notice that this averaging step does not decrease the objective, i.e. $f^{exp}(\mathbf{Q}^{(1)}, \mathbf{Q}^{(1)}, \mathbf{Q}^{(1)}) \geq f^{exp}(\mathbf{S}^{(1)}, \mathbf{Q}^{(0)}, \mathbf{Q}^{(0)})$, as we show later. This is in fact crucial in establishing convergence of the algorithm.

After the averaging step, the update is again performed, but this time on \mathbf{B} . The algorithm continues in this manner. It is easy to see that the averaging step essentially eliminates the need to retain the previous $K - 1$ states in memory, and instead only the previous state (i.e. $\mathbf{Q}^{(n-1)}$) needs to be stored. The general equations describing the algorithm are

$$\mathbf{S}^{(n)} = \arg \max_{\mathbf{Q}} f^{exp}(\mathbf{Q}, \mathbf{Q}^{(n-1)}, \dots, \mathbf{Q}^{(n-1)}) \quad (26)$$

$$\mathbf{Q}^{(n)} = \frac{1}{K}\mathbf{S}^{(n)} + \frac{K-1}{K}\mathbf{Q}^{(n-1)}. \quad (27)$$

The maximization in (26) that defines $\mathbf{S}^{(n)}$ is again solved by the waterfilling solution, but where the effective channel depends only on the covariance matrices from the previous state, i.e. $\mathbf{Q}^{(n-1)}$.

After initializing $\mathbf{Q}^{(0)}$, the algorithm proceeds as follows⁶:

- 1) Generate effective channels for each user:

$$\mathbf{G}_i^{(n)} = \mathbf{H}_i \left(\mathbf{I} + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j^{(n-1)} \mathbf{H}_j \right)^{-1/2} \quad i = 1, \dots, K. \quad (28)$$

- 2) Treating these effective channels as parallel, non-interfering channels, obtain covariance matrices $\{\mathbf{S}_i^{(n)}\}_{i=1}^K$ by water-filling with total power P :

$$\{\mathbf{S}_i^{(n)}\}_{i=1}^K = \arg \max_{\{\mathbf{S}_i\}_{i=1}^K: \mathbf{S}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{S}_i) \leq P} \sum_{i=1}^K \log \left| \mathbf{I} + \left(\mathbf{G}_i^{(n)} \right)^\dagger \mathbf{S}_i \mathbf{G}_i^{(n)} \right|.$$

- 3) Compute the updated covariance matrices $\mathbf{Q}_i^{(n)}$ as:

$$\mathbf{Q}_i^{(n)} = \frac{1}{K}\mathbf{S}_i^{(n)} + \frac{K-1}{K}\mathbf{Q}_i^{(n-1)} \quad i = 1, \dots, K. \quad (29)$$

Algorithm 2 (which first appeared in [11]) differs from the original algorithm only in the addition of the third step.

⁶As discussed in Section IX, the original algorithm can be used to generate an excellent starting point for Algorithm 2.

Theorem 3: Algorithm 2 converges to the sum rate capacity for any K .

Proof: Convergence of the algorithm is proven by showing that algorithm 1 is equivalent to algorithm 2 with the insertion of a non-decreasing (in the objective) operation in between every iteration. The spacer step theorem of [18, Chapter 7.11] asserts that if an algorithm satisfying the conditions of the global convergence theorem [18, Chapter 6.6] is combined with *any* series of steps that do not decrease the objective, then the combination of these two will still converge to the optimal. The cyclic coordinate ascent algorithm does indeed satisfy the conditions of the global convergence theorem, and below we prove that the averaging step does not decrease the objective. Thus, algorithm 2 converges⁷.

Consider the n -iteration of the algorithm, i.e.

$$(\mathbf{Q}^{(n-1)}, \dots, \mathbf{Q}^{(n-1)}) \rightarrow (\mathbf{S}^{(n)}, \mathbf{Q}^{(n-1)}, \dots, \mathbf{Q}^{(n-1)}) \quad (30)$$

$$\rightarrow \left(\frac{1}{K} \mathbf{S}^{(n)} + \frac{K-1}{K} \mathbf{Q}^{(n-1)}, \dots, \frac{1}{K} \mathbf{S}^{(n)} + \frac{K-1}{K} \mathbf{Q}^{(n-1)} \right) \quad (31)$$

where the mapping in (30) is the cyclic coordinate ascent algorithm performed on the first set of matrices, and the mapping in (31) is the averaging step. The first step is clearly identical to algorithm 1, while the second step (i.e. the averaging step) has been added. We need only show that the averaging step is non-decreasing, i.e.

$$f^{exp} \left(\mathbf{S}^{(n)}, \mathbf{Q}^{(n-1)}, \dots, \mathbf{Q}^{(n-1)} \right) \leq f^{exp} \left(\frac{1}{K} \mathbf{S}^{(n)} + \frac{K-1}{K} \mathbf{Q}^{(n-1)}, \dots, \frac{1}{K} \mathbf{S}^{(n)} + \frac{K-1}{K} \mathbf{Q}^{(n-1)} \right). \quad (32)$$

Notice that we can rewrite the left side as:

$$\begin{aligned} f^{exp} \left(\mathbf{S}^{(n)}, \mathbf{Q}^{(n-1)}, \dots, \mathbf{Q}^{(n-1)} \right) &= \frac{1}{K} \sum_{i=1}^K \log \left| \mathbf{I} + \mathbf{H}_i^\dagger \mathbf{S}_i^{(n)} \mathbf{H}_i + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j^{(n-1)} \mathbf{H}_j \right| \\ &\leq \log \left| \frac{1}{K} \sum_{i=1}^K \left(\mathbf{I} + \mathbf{H}_i^\dagger \mathbf{S}_i^{(n)} \mathbf{H}_i + \sum_{j \neq i} \mathbf{H}_j^\dagger \mathbf{Q}_j^{(n-1)} \mathbf{H}_j \right) \right| \\ &= \log \left| \mathbf{I} + \sum_{j=1}^K \mathbf{H}_j^\dagger \left(\frac{1}{K} \mathbf{S}_j^{(n)} + \frac{K-1}{K} \mathbf{Q}_j^{(n-1)} \right) \mathbf{H}_j \right| \\ &= f^{exp} \left(\frac{1}{K} \mathbf{S}^{(n)} + \frac{K-1}{K} \mathbf{Q}^{(n-1)}, \dots, \frac{1}{K} \mathbf{S}^{(n)} + \frac{K-1}{K} \mathbf{Q}^{(n-1)} \right) \end{aligned}$$

where the inequality follows from the concavity of the $\log |\cdot|$ function. Since the averaging step is non-decreasing, the algorithm converges. More precisely, this means $f^{exp}(\mathbf{Q}^{(n)}, \dots, \mathbf{Q}^{(n)})$ converges to

⁷There is also a technical condition regarding compactness of the set with larger objective than the objective evaluated for the initialization matrices that is trivially satisfied due to the properties of Euclidean space.

the sum capacity. Since this quantity is equal to $f(\mathbf{Q}^{(n)})$, we have

$$\lim_{n \rightarrow \infty} f(\mathbf{Q}^{(n)}) = \mathcal{C}_{\text{MAC}}(\mathbf{H}_1^\dagger, \dots, \mathbf{H}_K^\dagger, P) \quad (33)$$

■

VIII. COMPLEXITY ANALYSIS

In this section we provide complexity analyses of the three proposed algorithms and other algorithms in literature. Each of the three proposed algorithms here have complexity that increases *linearly* with K , the number of users. This is an extremely desirable property when considering systems with large numbers of users (i.e. 50 or 100 users). The linear complexity of our algorithm is quite easy to see if one goes through the basic steps of the algorithm. For simplicity, we consider Algorithm 1, which is the most complex of the algorithms. Calculating the effective channels in step 1 requires calculating the total interference seen by each user (i.e. a term of the form of $|\mathbf{I} + \sum_{j \neq i} \mathbf{H}_i^\dagger \mathbf{Q}_j \mathbf{H}_i|$). A running sum of such a term can be maintained, such that calculating the effective channel of each user requires only a finite number of subtractions and additions. The waterfilling operation in step 2 can also be performed in linear time by taking the SVD of each of the effective channels and then waterfilling. It is important not to perform a standard waterfilling operation on the block diagonal channel, because the size of the involved matrices grow with K . In general, the key idea behind the linear complexity of our algorithm is that the entire input space is never considered (i.e. only $N \times N$ and $M \times M$ matrices, and never matrices whose size is a function of K , are considered). This, however, is not true of general optimization methods which do not take advantage of the structure of the sum capacity problem.

Standard interior point methods have complexity that is cubic with respect to the dimensionality of the input space (i.e. with respect to K , the number of users), due to the complexity of the inner Newton iterations [2]. The minimax based approach in [8] also has complexity that is cubic in K because matrices whose size is a function of K are inverted in each step. For very small problems, this is not significant, but for even reasonable values of K (i.e. $K = 10$ or $K = 20$) this increase in complexity makes such methods computationally prohibitive.

The other proposed specialized algorithms [13] [15] are also linear in complexity (in K). However, the steepest descent algorithm proposed in [13] requires a line search in each step, which does not increase the complexity order but does significantly increase run time. The dual decomposition algorithm proposed in [15] requires an inner optimization to be performed within each iteration (i.e. user-by-user iterative waterfilling [17] with a fixed water level, instead of individual power constraints, must be performed

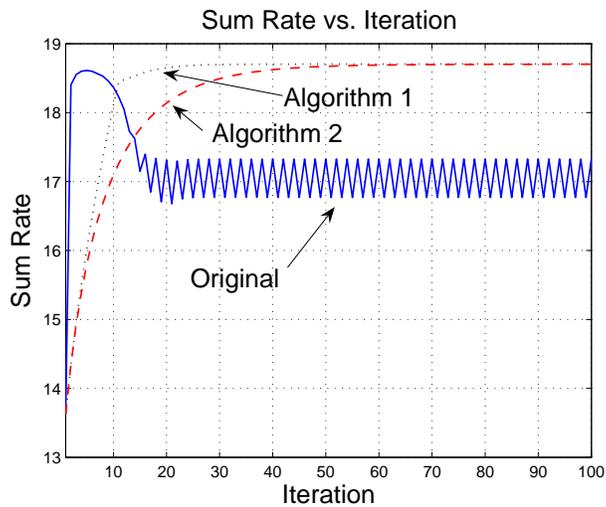


Fig. 4. Algorithm comparison for a divergent scenario

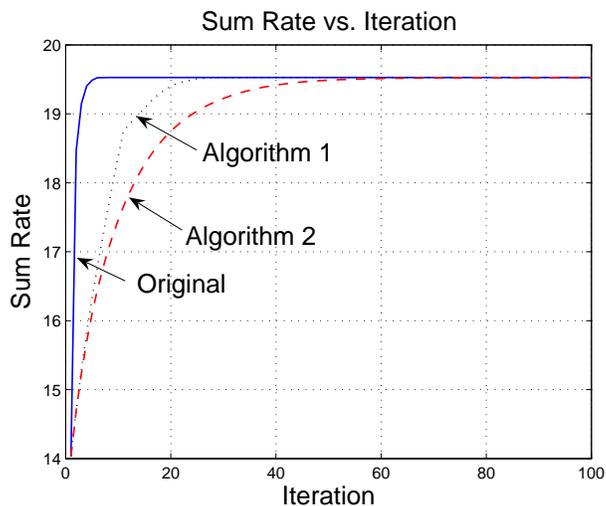


Fig. 5. Algorithm comparison for a convergent scenario

repeatedly), which significantly increases run time. Our sum power iterative waterfilling algorithms, on the other hand, do not require a line search or an inner optimization within each iteration, thus leading to a faster run time. In addition, we find the iterative waterfilling algorithms to converge faster than the other linear complexity algorithms for almost all channel realizations. Some numerical results and discussion of this is presented in the next section.

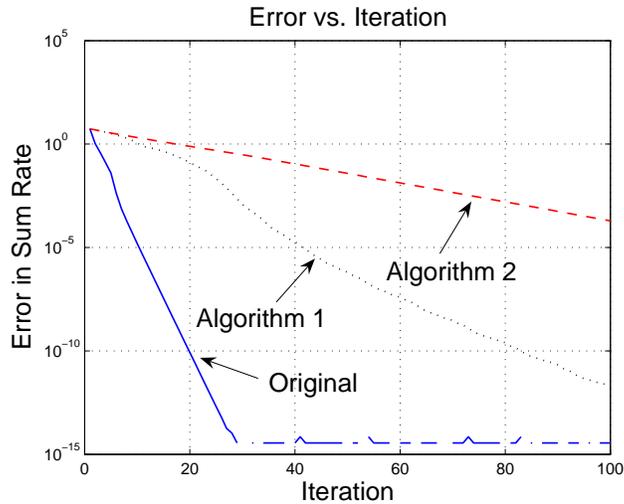


Fig. 6. Error comparison for a convergent scenario

IX. NUMERICAL RESULTS

In this section we provide some numerical results to show the behavior of the three algorithms. In Figure 4, a plot of sum rate vs. iteration number is provided for a 10 user channel with 4 transmit and 4 receive antennas. In this example, the original algorithm does not converge and can be seen to oscillate between two sub-optimal points. Algorithms 1 and 2 do converge, however, as guaranteed by Theorems 2 and 3. In general, it is not difficult to randomly generate channels for which the original algorithm does not converge and instead oscillates between sub-optimal points. This divergence occurs because not only can the original algorithm lead to a decrease in the sum rate, but additionally there appear to exist sub-optimal points between which the original algorithm can oscillate, i.e. point 1 is generated by iteratively waterfilling from point 2, and vice versa.

In Fig. 5 the same plot is shown for a different channel (with the same system parameters as in Fig. 4: $K = 10$, $M = N = 4$) in which the original algorithm does in fact converge. Notice that the original algorithm performs best, followed by Algorithm 1, and then Algorithm 2. The same trend is seen in Figure 6, which plots the error in capacity. Additionally, notice that all three algorithms converge linearly, as expected for this class of algorithms. Though these plots are only for a single instantiation of channels, the same ordering has always occurred, i.e. the original algorithm performs best (in situations where it converges) followed by Algorithm 1 and then Algorithm 2.

The fact that the original algorithm converges faster than the modified algorithms is intuitively not surprising, because the original algorithm updates matrices at a much faster rate than either of the modified

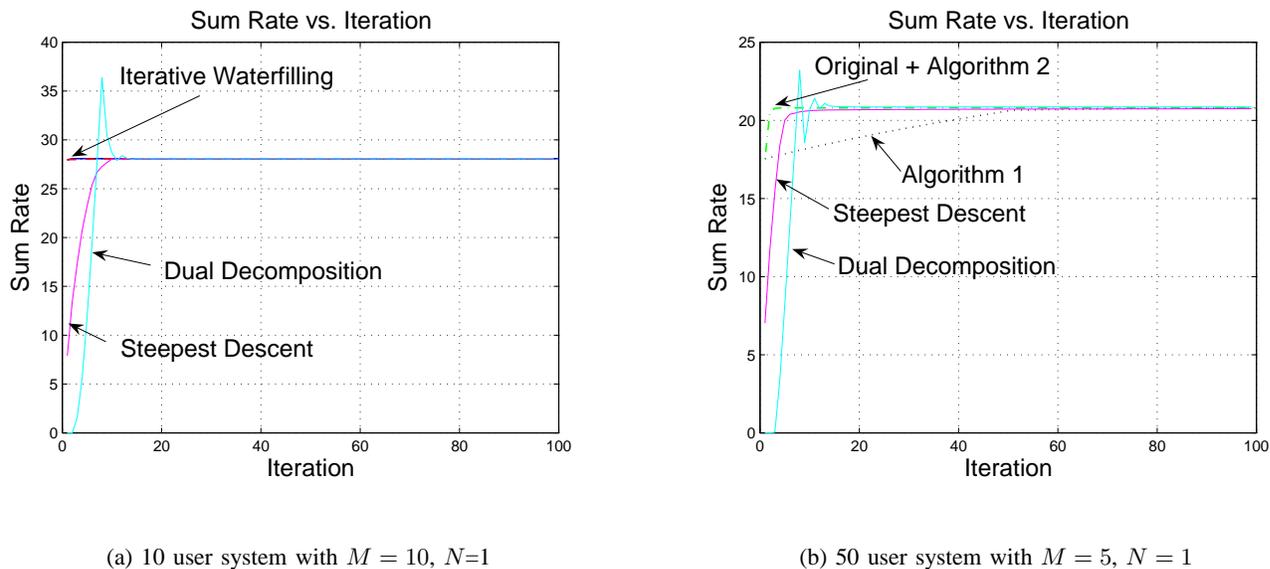


Fig. 7. Comparison of linear complexity algorithms

versions of the algorithm. In Algorithm 1, there are K covariances for each user (corresponding to the K previous states) that are averaged to yield the set of covariances that converge to the optimal. The most recently updated covariances therefore make up only a fraction $1/K$ of the average, and thus the algorithm moves relatively slowly. In Algorithm 2, the updated covariances are very similar to the covariances from the previous state, as the updated covariances are equal to $(K-1)/K$ times the previous state's covariances plus only a factor of $1/K$ times the covariances generated by the iterative waterfilling step. Thus, it should be intuitively clear that in situations where the original algorithm actually converges, convergence is much faster for the original algorithm than for either of the modified algorithms. From the plot it is clear that the performance difference between the original algorithm and Algorithms 1 and 2 is quite significant. At the end of this section, however, we discuss how the original algorithm can be combined with either Algorithm 1 or 2 to improve performance considerably while still maintaining guaranteed convergence. Of the two modified algorithms, Algorithm 1 is almost always seen to outperform Algorithm 2. However, there does not appear to be an intuitive explanation for this behavior.

In Figure 7(a) sum rate is plotted for the three iterative waterfilling algorithms (original, Algorithm 1, and Algorithm 2), the steepest descent method [13], and the dual decomposition method [15], for a channel with $K=10$, $M=10$, and $N=1$. The three iterative waterfilling algorithms perform nearly identically for this channel, and three curves are in fact superimposed on one and other in the figure.

Furthermore, the iterative waterfilling algorithms converge more rapidly than either of the alternative methods. The iterative waterfilling algorithms outperform the other algorithms in many scenarios, and the gap is particularly large when the number of transmit antennas (M) and users (K) are large. It should be noted that there are certain situations where the steepest descent and dual decomposition algorithms outperform the iterative waterfilling algorithm, in particular when the number of users is much larger than the number of antennas. Figure 7(b) contains a convergence plot of a 50 user system with $M = 5$ and $N = 1$. Algorithm 1 converges rather slowly precisely because of the large number of users (i.e. because the covariances can only change at approximately a rate of $1/K$ in each iteration, as discussed earlier). Notice that both the steepest descent and dual decomposition algorithms converge faster. However, the results for a *hybrid* algorithm are also plotted here (referred to as “Original + Algorithm 2”). In this hybrid algorithm, the original iterative waterfilling algorithm is performed for the first five iterations, and then Algorithm 2 is used for all subsequent iterations. The original algorithm is essentially used to generate a good starting point for Algorithm 2. This hybrid algorithm converges, because the original algorithm is only used a finite number of times, and is seen to outperform any of the other alternatives. In fact, we find that the combination of the original algorithm with either Algorithm 1 or 2 converges extremely rapidly to the optimum and outperforms the alternative linear complexity approaches in the very large majority of scenarios, i.e. for any number of users and antennas. This is true even for channels for which the original algorithm itself does not converge, because running the original algorithm for a few iterations still provides an excellent starting point.

X. CONCLUSIONS

In this paper we proposed two algorithms that find the sum capacity achieving transmission strategies for the multiple antenna broadcast channel. We use the fact that the Gaussian broadcast and multiple-access channels are duals in the sense that their capacity regions, and therefore their sum capacities, are equal. These algorithms compute the sum capacity achieving strategy for the dual multiple-access channel, which can easily be converted to the equivalent optimal strategies for the broadcast channel. The algorithms exploit the inherent structure of the multiple-access channel and employ a simple iterative water-filling procedure that provably converges to the optimum. The two algorithms are extremely similar, as both are based on the cyclic coordinate ascent and use the single-user waterfilling procedure in each iteration, but they offer a simple tradeoff between performance and required memory. The convergence speed, low complexity, and simplicity make the iterative waterfilling algorithms extremely attractive methods to find the sum capacity of the multiple antenna broadcast channel.

APPENDIX I

MAC-BC TRANSFORMATION

In this appendix, we restate the mapping from uplink covariance matrices to downlink matrices. Given uplink covariances $\mathbf{Q}_1, \dots, \mathbf{Q}_K$, the transformation in [10, Equations 8-10] outputs downlink covariance matrices $\mathbf{\Sigma}_1, \dots, \mathbf{\Sigma}_K$ that achieve the same rates (on a user-by-user basis, and thus also in terms of sum rate) using the same sum power, i.e. with $\sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) = \sum_{i=1}^K \text{Tr}(\mathbf{\Sigma}_i)$. For convenience, we first define the following two quantities:

$$\mathbf{A}_i \triangleq \mathbf{I} + \mathbf{H}_i \left(\sum_{l=1}^{i-1} \mathbf{\Sigma}_l \right) \mathbf{H}_i^\dagger, \quad \mathbf{B}_i \triangleq \mathbf{I} + \sum_{l=i+1}^K \mathbf{H}_l^\dagger \mathbf{Q}_l \mathbf{H}_l \quad (34)$$

for $i = 1, \dots, K$. Furthermore, we write the SVD decomposition of $\mathbf{B}_i^{-1/2} \mathbf{H}_i^\dagger \mathbf{A}_i^{-1/2}$ as $\mathbf{B}_i^{-1/2} \mathbf{H}_i^\dagger \mathbf{A}_i^{-1/2} = \mathbf{F}_i \mathbf{D}_i \mathbf{G}_i^\dagger$, where \mathbf{D}_i is a square and diagonal matrix⁸. Then, the equivalent downlink covariance matrices can be computed via the following transformation:

$$\mathbf{\Sigma}_i = \mathbf{B}_i^{-1/2} \mathbf{F}_i \mathbf{G}_i^\dagger \mathbf{A}_i^{1/2} \mathbf{Q}_i \mathbf{A}_i^{1/2} \mathbf{G}_i \mathbf{F}_i^\dagger \mathbf{B}_i^{-1/2}, \quad (35)$$

beginning with $i = 1$. See [10] for a derivation and more detail.

APPENDIX II

UNIQUENESS OF WATER-FILLING SOLUTION

In this appendix we show there is a unique solution to the following maximization:

$$\max_{\mathbf{Q} \geq 0, \text{Tr}(\mathbf{Q}) \leq P} \log \left| \mathbf{I} + \mathbf{H} \mathbf{Q} \mathbf{H}^\dagger \right| \quad (36)$$

for any non-zero $\mathbf{H} \in \mathbb{C}^{N \times M}$ for arbitrary M, N . This proof is identical to the proof of optimality of water-filling in [9, Section 3.2], with the addition of a simple proof of uniqueness.

Since $\mathbf{H}^\dagger \mathbf{H} \in \mathbb{C}^{M \times M}$ is Hermitian and positive semi-definite, we can diagonalize it and write $\mathbf{H}^\dagger \mathbf{H} = \mathbf{U} \mathbf{D} \mathbf{U}^\dagger$ where $\mathbf{U} \in \mathbb{C}^{M \times M}$ is unitary and $\mathbf{D} \in \mathbb{R}^{M \times M}$ is diagonal with non-negative entries. Since the ordering of the columns of \mathbf{U} and the entries of \mathbf{D} are arbitrary and because \mathbf{D} must have at least one strictly positive entry (because \mathbf{H} is not the zero matrix), for simplicity we assume $\mathbf{D}_{ii} > 0$ for $i = 1, \dots, L$ and $\mathbf{D}_{ii} = 0$ for $i = L + 1, \dots, M$ for some $1 \leq L \leq M$. Using the identity $|\mathbf{I} + \mathbf{A} \mathbf{B}| = |\mathbf{I} + \mathbf{B} \mathbf{A}|$, we can rewrite the objective function in (36) as:

$$\log \left| \mathbf{I} + \mathbf{H} \mathbf{Q} \mathbf{H}^\dagger \right| = \log \left| \mathbf{I} + \mathbf{Q} \mathbf{H}^\dagger \mathbf{H} \right| = \log \left| \mathbf{I} + \mathbf{Q} \mathbf{U} \mathbf{D} \mathbf{U}^\dagger \right| = \log \left| \mathbf{I} + \mathbf{U}^\dagger \mathbf{Q} \mathbf{U} \mathbf{D} \right|. \quad (37)$$

⁸Note that the standard SVD command in MATLAB does not return a square and diagonal \mathbf{D}_i . This is accomplished by using the “0” option in the SVD command in MATLAB, and is referred to as the “economy size” decomposition.

If we define $\mathbf{S} \triangleq \mathbf{U}^\dagger \mathbf{Q} \mathbf{U}$, then $\mathbf{Q} = \mathbf{U} \mathbf{S} \mathbf{U}^\dagger$. Since $\text{Tr}(\mathbf{A}\mathbf{B}) = \text{Tr}(\mathbf{B}\mathbf{A})$ and \mathbf{U} is unitary, we have $\text{Tr}(\mathbf{S}) = \text{Tr}(\mathbf{U}^\dagger \mathbf{Q} \mathbf{U}) = \text{Tr}(\mathbf{Q} \mathbf{U} \mathbf{U}^\dagger) = \text{Tr}(\mathbf{Q})$. Furthermore, $\mathbf{S} \geq 0$ if and only if $\mathbf{Q} \geq 0$. Therefore, the maximization can equivalently be carried out over \mathbf{S} , i.e.:

$$\max_{\mathbf{S} \geq 0, \text{Tr}(\mathbf{S}) \leq P} \log |\mathbf{I} + \mathbf{S} \mathbf{D}|. \quad (38)$$

In addition, each solution to (36) corresponds to a different solution of (38) via the invertible mapping $\mathbf{S} = \mathbf{U}^\dagger \mathbf{Q} \mathbf{U}$. Thus, if the maximization in (36) has multiple solutions, the maximization in (38) must also have multiple solutions. Therefore, it is sufficient to show that (38) has a unique solution, which we prove next.

First we show by contradiction that any optimal \mathbf{S} must satisfy $\mathbf{S}_{ij} = 0$ for all $i, j > L$. Consider an $\mathbf{S} \geq 0$ with $\mathbf{S}_{ij} \neq 0$ for some $i > L$ and $j > L$. Since $|\mathbf{S}_{ij}| \leq \sqrt{\mathbf{S}_{ii} \mathbf{S}_{jj}}$ for any $\mathbf{S} \geq 0$, this implies $\mathbf{S}_{ii} > 0$ and $\mathbf{S}_{jj} > 0$, i.e. at least one diagonal entry of \mathbf{S} is strictly positive below the L -th row/column. Using Hadamard's inequality [5] and the fact that $\mathbf{D}_{ii} = 0$ for $i > L$, we have $|\mathbf{I} + \mathbf{S} \mathbf{D}| \leq \prod_{i=1}^M (1 + \mathbf{S}_{ii} \mathbf{D}_{ii}) = \prod_{i=1}^L (1 + \mathbf{S}_{ii} \mathbf{D}_{ii})$. We now construct another matrix \mathbf{S}' that achieves a strictly larger objective than \mathbf{S} . We define \mathbf{S}' to be diagonal with

$$\mathbf{S}'_{ii} = \begin{cases} \mathbf{S}_{11} + \sum_{i=L+1}^M \mathbf{S}_{ii} & i = 1 \\ \mathbf{S}_{ii} & i = 2, \dots, L \\ 0 & i = L + 1, \dots, M \end{cases} \quad (39)$$

Clearly $\mathbf{S}' \geq 0$ and $\text{Tr}(\mathbf{S}') = \sum_{i=1}^L \mathbf{S}'_{ii} = \mathbf{S}_{11} + \sum_{i=L+1}^M \mathbf{S}_{ii} + \sum_{i=2}^L \mathbf{S}_{ii} = \text{Tr}(\mathbf{S})$. Since \mathbf{S}' is diagonal, the matrix $\mathbf{S}' \mathbf{D}$ is diagonal and we have $\log |\mathbf{I} + \mathbf{S}' \mathbf{D}| = \log \prod_{i=1}^L (1 + \mathbf{S}'_{ii} \mathbf{D}_{ii}) > \log \prod_{i=1}^L (1 + \mathbf{S}_{ii} \mathbf{D}_{ii}) \geq \log |\mathbf{I} + \mathbf{S} \mathbf{D}|$, where the strict inequality is due to the fact that $\mathbf{S}'_{11} > \mathbf{S}_{11}$ and $\mathbf{D}_{11} > 0$. Therefore, the optimal \mathbf{S} must satisfy $\mathbf{S}_{ij} = 0$ for all $i, j > L$.

Next we show by contradiction that any optimal \mathbf{S} must also be diagonal. Consider any $\mathbf{S} \geq 0$ that satisfies the above condition ($\mathbf{S}_{ij} = 0$ for all $i, j > L$) but is not diagonal, i.e. $\mathbf{S}_{kj} \neq 0$ for some $k \neq j$ and $k, j \leq L$. Since \mathbf{D} is diagonal and $\mathbf{D}_{ii} > 0$ for $i = 1, \dots, L$, the matrix $\mathbf{S} \mathbf{D}$ is not diagonal because $(\mathbf{S} \mathbf{D})_{kj} = \mathbf{S}_{kj} \mathbf{D}_{jj} \neq 0$. Since Hadamard's inequality holds with equality only for diagonal matrices, we have $\log |\mathbf{I} + \mathbf{S} \mathbf{D}| < \log \prod_{i=1}^L (1 + \mathbf{S}_{ii} \mathbf{D}_{ii})$. Let us define a diagonal matrix \mathbf{S}' with $\mathbf{S}'_{ii} = \mathbf{S}_{ii}$ for $i = 1, \dots, M$. Clearly $\text{Tr}(\mathbf{S}') = \text{Tr}(\mathbf{S})$ and $\mathbf{S}' \geq 0$. Since \mathbf{S}' is diagonal, the matrix $\mathbf{S}' \mathbf{D}$ is diagonal and thus $\log |\mathbf{I} + \mathbf{S}' \mathbf{D}| = \log \prod_{i=1}^L (1 + \mathbf{S}_{ii} \mathbf{D}_{ii}) > \log |\mathbf{I} + \mathbf{S} \mathbf{D}|$. Therefore, the optimal \mathbf{S} must be diagonal, as well as satisfy $\mathbf{S}_{ij} = 0$ for $i, j > L$.

Therefore, in order to find *all solutions* to (38), it is sufficient to only consider the class of diagonal,

positive semi-definite matrices \mathbf{S} that satisfy $\mathbf{S}_{ij} = 0$ for all $i, j > L$ and $\text{Tr}(\mathbf{S}) \leq P$. The positive semi-definite constraint is equivalent to $\mathbf{S}_{ii} \geq 0$ for $i = 1, \dots, L$, and the trace constraint gives $\sum_{i=1}^L \mathbf{S}_{ii} \leq P$. Since $\log |\mathbf{I} + \mathbf{S}'\mathbf{D}| = \log \prod_{i=1}^L (1 + \mathbf{S}'_{ii}\mathbf{D}_{ii})$ for this class of matrices, we need only consider the following maximization:

$$\max_{\{\mathbf{S}_{ii}\}_{i=1}^L: \mathbf{S}_{ii} \geq 0, \sum_{i=1}^L \mathbf{S}_{ii} \leq P} \sum_{i=1}^L \log(1 + \mathbf{S}_{ii}\mathbf{D}_{ii}). \quad (40)$$

Since $\mathbf{D}_{ii} > 0$ for $i = 1, \dots, L$, the objective in (40) is a strictly concave function, and thus has a unique maximum. Thus, (38) has a unique maximum, which implies that (36) also has a unique maximum.

APPENDIX III

DERIVATION OF ALGORITHM 1

In this section we derive the general form of Algorithm 1 for an arbitrary number of users. In order to solve the original sum rate capacity maximization in (12), we consider an alternative maximization:

$$\max_{\mathbf{S}(1), \dots, \mathbf{S}(K)} f^{exp}(\mathbf{S}(1), \dots, \mathbf{S}(K)) \quad (41)$$

where we define $\mathbf{S}(i) \triangleq (\mathbf{S}(i)_1, \dots, \mathbf{S}(i)_K)$ for $i = 1, \dots, K$ with $\mathbf{S}(i)_j \in \mathbb{C}^{N \times N}$, and the maximization is performed subject to the constraints $\mathbf{S}(i)_j \geq 0$ for all i, j and $\sum_{j=1}^K \text{Tr}(\mathbf{S}(i)_j) \leq P$ for $i = 1, \dots, K$.

The function $f^{exp}(\cdot)$ is defined as:

$$f^{exp}(\mathbf{S}(1), \dots, \mathbf{S}(K)) = \frac{1}{K} \sum_{i=1}^K \log \left| \mathbf{I} + \sum_{j=1}^K \mathbf{H}_j^\dagger \mathbf{S}([j-i+1]_K)_j \mathbf{H}_j \right|. \quad (42)$$

In the notation used in Section VI, we would have $\mathbf{A} = \mathbf{S}(1)$, $\mathbf{B} = \mathbf{S}(2)$, $\mathbf{C} = \mathbf{S}(3)$. As discussed earlier, every solution to the original sum rate maximization problem in (12) corresponds to a solution to (41), and vice versa. Furthermore, the cyclic coordinate ascent algorithm can be used to maximize (41) due to the separability of the constraints on $\mathbf{S}(1), \dots, \mathbf{S}(K)$. If we let $\{\mathbf{S}(i)^{(n)}\}_{i=1}^K$ denote the n -th iteration of the cyclic coordinate ascent algorithm, then the following holds:

$$\mathbf{S}(l)^{(n)} = \begin{cases} \arg \max_{\mathbf{S}} f^{exp}(\mathbf{S}(1)^{(n-1)}, \dots, \mathbf{S}(m-1)^{(n-1)}, \mathbf{S}, \mathbf{S}(m+1)^{(n-1)}, \dots, \mathbf{S}(K)^{(n-1)}) & l = m \\ \mathbf{S}(l)^{(n-1)} & l \neq m \end{cases} \quad (43)$$

for $l = 1, \dots, K$, where $m = [n]_K$. For each n , we define $\mathbf{Q}^{(n)}$ to be the updated matrices in that iteration:

$$\mathbf{Q}^{(n)} \triangleq \mathbf{S}(m)^{(n)} \quad (44)$$

$$= \arg \max_{\mathbf{S}} f^{exp}(\mathbf{S}(1)^{(n-1)}, \dots, \mathbf{S}(m-1)^{(n-1)}, \mathbf{S}, \mathbf{S}(m+1)^{(n-1)}, \dots, \mathbf{S}(K)^{(n-1)}) \quad (45)$$

$$= \arg \max_{\mathbf{S}} f^{exp}(\mathbf{S}, \mathbf{S}(m+1)^{(n-1)}, \dots, \mathbf{S}(K)^{(n-1)}, \mathbf{S}(1)^{(n-1)}, \dots, \mathbf{S}(m-1)^{(n-1)}) \quad (46)$$

where in the final step we used the fact that

$$f^{exp}(\mathbf{S}(1), \dots, \mathbf{S}(K)) = f^{exp}(\mathbf{S}(l), \dots, \mathbf{S}(K), \mathbf{S}(1), \dots, \mathbf{S}(l-1)) \quad (47)$$

for any l due to the circular structure of f^{exp} and the uniqueness of the waterfilling solution to (46).

Plugging in recursively for $\mathbf{Q}^{(n)}$ for all n , we get:

$$\mathbf{Q}^{(n)} = \arg \max_{\mathbf{Q}} f^{exp}(\mathbf{Q}, \mathbf{Q}^{(n-K+1)}, \dots, \mathbf{Q}^{(n-1)}) \quad (48)$$

$$= \arg \max_{\mathbf{Q}: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \sum_{i=1}^K \log \left| \mathbf{I} + \mathbf{H}_i^\dagger \mathbf{Q}_i \mathbf{H}_i + \sum_{j=1}^{K-1} \mathbf{H}_{[i+j]_K}^\dagger \mathbf{Q}_{[i+j]_K}^{(n-K+j)} \mathbf{H}_{[i+j]_K} \right| \quad (49)$$

$$= \arg \max_{\mathbf{Q}: \mathbf{Q}_i \geq 0, \sum_{i=1}^K \text{Tr}(\mathbf{Q}_i) \leq P} \sum_{i=1}^K \log \left| \mathbf{I} + \left(\mathbf{G}_i^{(n)} \right)^\dagger \mathbf{Q}_i \mathbf{G}_i^{(n)} \right|. \quad (50)$$

The final maximization is equivalent to waterfilling over effective channels \mathbf{G}_j , given by:

$$\mathbf{G}_i^{(n)} = \mathbf{H}_i \left(\mathbf{I} + \sum_{j=1}^{K-1} \mathbf{H}_{[i+j]_K}^\dagger \mathbf{Q}_{[i+j]_K}^{(n-K+j)} \mathbf{H}_{[i+j]_K} \right)^{-1/2} \quad (51)$$

for $i = 1, \dots, K$.

ACKNOWLEDGMENT

The authors wish to thank Daniel Palomar and Tom Luo for helpful discussions regarding convergence issues.

REFERENCES

- [1] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.
- [2] S. Boyd and L. Vandenberghe. *Introduction to convex optimization with engineering applications*. Course Reader, 2001.
- [3] G. Caire and S. Shamai. On the achievable throughput of a multiantenna Gaussian broadcast channel. *IEEE Trans. Inform. Theory*, 49(7):1691–1706, July 2003.
- [4] M. Costa. Writing on dirty paper. *IEEE Trans. Inform. Theory*, 29(3):439–441, May 1983.
- [5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.
- [6] N. Jindal, S. Jafar, S. Vishwanath, and A. Goldsmith. Sum power iterative water-filling for multi-antenna Gaussian broadcast channels. In *Proceedings of Asilomar Conf. on Signals, Systems, & Comp.*, 2002.
- [7] N. Jindal, S. Vishwanath, and A. Goldsmith. On the duality of Gaussian multiple-access and broadcast channels. *IEEE Trans. Inform. Theory*, 50(5):768–783, May 2004.
- [8] T. Lan and W. Yu. Input optimization for multi-antenna broadcast channels and per-antenna power constraints. In *Proceedings of IEEE Globecom*, 2004.
- [9] E. Telatar. Capacity of multi-antenna Gaussian channels. *European Trans. on Telecomm. ETT*, 10(6):585–596, November 1999.

- [10] S. Vishwanath, N. Jindal, and A. Goldsmith. Duality, achievable rates, and sum-rate capacity of MIMO broadcast channels. *IEEE Trans. Inform. Theory*, 49(10):2658–2668, Oct. 2003.
- [11] S. Vishwanath, W. Rhee, N. Jindal, S.A. Jafar, and A. Goldsmith. Sum power iterative water-filling for Gaussian vector broadcast channels. In *Proceedings of Int. Symp. Inform. Theory*, page 467, July 2003.
- [12] P. Viswanath and D. N. Tse. Sum capacity of the vector Gaussian broadcast channel and uplink-downlink duality. *IEEE Trans. Inform. Theory*, 49(8):1912–1921, Aug. 2003.
- [13] H. Viswanathan, S. Venkatesan, and H. C. Huang. Downlink capacity evaluation of cellular networks with known interference cancellation. *IEEE J. Sel. Areas Commun.*, 21:802–811, June 2003.
- [14] H. Weingarten, Y. Steinberg, and S. Shamai. The capacity region of the Gaussian MIMO broadcast channel. In *Proceedings of Conference on Information Sciences and Systems*, March 2004.
- [15] W. Yu. A dual decomposition approach to the sum power Gaussian vector multiple-access channel sum capacity problem. In *Proceedings of Conference on Information Sciences and Systems (CISS)*, 2003.
- [16] W. Yu and J. M. Cioffi. Sum capacity of a Gaussian vector broadcast channel. In *Proc. of Int. Symp. Inform. Theory*, page 498, June 2002.
- [17] W. Yu, W. Rhee, S. Boyd, and J. Cioffi. Iterative water-filling for Gaussian vector multiple-access channels. *IEEE Trans. Inform. Theory*, 50(1):145–152, Jan. 2004.
- [18] W. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice Hall, 1969.

Nihar Jindal received the B.S. degree in Electrical Engineering and Computer Science from U.C. Berkeley in 1999, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 2001 and 2004. He is currently an assistant professor at the University of Minnesota. His industry experience includes summer internships at Intel Corporation, Santa Clara, CA in 2000 and at Lucent Bell Labs, Holmdel, NJ in 2002. His research interests include multiple-antenna/multi-user channels, dynamic resource allocation, and sensor and ad-hoc networks.

Wonjong Rhee received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1996 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1998 and 2002, respectively. He was with ArrayComm, Inc., San Jose, CA, from 2001 to 2003 where he performed various research on CDMA (Code Division Multiple Access) and SDMA (Space Division Multiple Access) systems. Since 2004, he has been with ASSIA, Inc., Redwood City, CA. His research interests include multiuser wireless communications with multiple antennas and dynamic spectrum optimization.

Sriram Vishwanath (S'99) received his B.Tech. degree from the Indian Institute of Technology (IIT) Madras in 1998, his M.S. degree, from the California Institute of Technology (Caltech) in 1999, and his Ph.D. from Stanford University in 2003, all in electrical engineering. He is currently an assistant professor in the department of electrical and computer engineering at the University of Texas at Austin. His industry experience includes work at the National Semiconductor Corporation in 2000 and at the Lucent Bell Laboratories in 2002. His research interests includes network information theory, coding theory and information hiding.

Syed Ali Jafar (S' 99- M'04) received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology (IIT), Delhi, India in 1997, the M.S. degree in Electrical Engineering from California Institute of Technology (Caltech), Pasadena USA in 1999, and the Ph.D. degree in Electrical Engineering from Stanford University, Stanford, CA USA in 2003 .

He was a summer intern in the Wireless Communications Group of Lucent Bell Laboratories, Holmdel, NJ, in 2001. He was an engineer in the Satellite Networks Division of Hughes Software Systems from 1997-1998 and a senior engineer at Qualcomm Inc., San Diego, CA in 2003. He is currently an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of California Irvine, Irvine, CA USA. His research interests include multiuser information theory and wireless communications, with a focus on multiple-antenna systems.

Dr. Jafar serves as the Editor for Wireless Communication Theory and CDMA for the IEEE Transactions on Communications.

Andrea Goldsmith received the B.S., M.S., and Ph.D. degrees in electrical engineering from U.C. Berkeley in 1986, 1991, and 1994, respectively. From 1986-1990 she was affiliated with Maxim Technologies, where she worked on packet radio and satellite communication systems, and from 1991-1992 she was affiliated with AT&T Bell Laboratories, where she worked on propagation models and channel estimation for cellular systems. She was an assistant professor of Electrical Engineering at the California Institute of Technology from 1994-1998, and then joined the Electrical Engineering department at Stanford University where she is now an associate professor. Her research includes work in the capacity of wireless channels and networks, wireless communication and information theory, adaptive resource allocation in wireless networks, multiantenna wireless systems, energy-constrained wireless communications, wireless communications for distributed control, and cross-layer design for cellular systems, ad-hoc wireless networks, and sensor networks.

Dr. Goldsmith holds the Bredt Faculty Development Scholar Chair at Stanford and is a recipient of the National Academy of Engineering Gilbreth Lectureship, the Alfred P. Sloan Fellowship, the National Science Foundation CAREER Development Award, the Office of Naval Research Young Investigator Award, the National Semiconductor Faculty Development Award, the Okawa Foundation Award, and the David Griep Memorial Prize from U.C. Berkeley. She is a Fellow of the IEEE and was an editor for the IEEE Transactions on Communications from 1995 to 2002. She has also been an editor for the IEEE Wireless Communications Magazine since 1995 and for the Journal on Foundations and Trends in Communications and Information Theory since 2003. She is an elected member of Stanford's Faculty Senate and Senate Steering Committee and the Board of Governors for the IEEE Information Theory Society. She is also currently serving as Vice-Chair for the Communications Theory Committee of the IEEE Communications Society.