Abstract

In this paper, based on the iterative water-filling (IWF) and primal-dual approaches, we proposed efficient algorithms to maximize the sum-rate and optimize the input distribution policy for the multi-user multiple input multiple output multiple access channels (MU-MIMO MAC) with concurrently accessing of the cognitive radio (CR) network. For the proposed IWF-CR algorithm, with more complicated problem structure of the mixed constraints, the conventional water-filling algorithm could not be able to complete the inner loop operation of the IWF-CR algorithm. We proposed a generalized water-filling algorithm to accomplish the inner loop operation. For the outer loop, the power distribution for the users is updated by using variable scale factors which efficiently maximize the objective function at each iteration. At the same time, distributed and parallel computation is employed to expedite convergence of the proposed algorithm. As a counter-part, we also proposed an algorithm based on the primal dual approach to solve the same target problem (PD-CR). Under this approach, to avoid introducing the new deeper inner loop that could stem from the structure of the system, another new water-filling is proposed during evaluation of the dual function. This new water-filling itself can extend the objective function to a combination of the sum-log function with a linear function. As a result, beside the original fixed point theory, the two new water-filling algorithms, as fundamental blocks
of the proposed algorithms respectively, also enrich the existing optimization theory and methods applied in communications. Numerical results verify that the proposed algorithms are efficient. Using the proposed approaches, for the simulated range, the required number of iterations for the convergence is at most seven for \textbf{IWF-CR} and eighteen for \textbf{PD-CR}. The required iteration number is not sensitive to the increase of the number of users for both algorithms. This feature is quite desirable for large scale systems with dense active users.

\textbf{Keywords}

Channel capacity, multi-user MIMO (MU-MIMO), multi-access channels (MAC), cognitive radio (CR), multiple-antenna broadcast systems, maximum sum-rate, optimal power distribution, optimization methods, water-filling algorithm with mixed constraints, iterative water-filling, primal-dual.

\footnote{1Subsection III.3 in this paper will be presented in part at the IEEE International Wireless Symposium, April, 2013, Beijing}
I Introduction

The radio spectrum is a precious resource that demands for efficient utilization as the currently licensed spectrum is severely underutilized [1]. Cognitive Radio (CR) [2]-[4], which adapts the radios operating characteristics to the real-time conditions, is the key technology that allows flexible, efficient and reliable spectrum utilization in wireless communications. This technology exploits the underutilized licensed spectrum of the primary user(s) (PU) and introduces the secondary user(s) (SU) to operate on the spectrum that is either opportunistically being available or concurrently being shared by the PU and the SU. Under this situation and according to the definition of the cognitive (radio) network [5], opportunistically utilizing the spectrum means that the SUs may fill the spectrum gaps or holes left by the PUs; whereas concurrently utilizing the spectrum means that the SUs transmit over the same spectrum as the PUs, in the way that the interference from the transmitting SUs does not violate the quality requirement from the PUs. This paper focuses on the latter case. Furthermore, the multiple-input multiple-output (MIMO) technology uses multiple antennas at either the transmitter or the receiver to significantly increase data throughput and link range without additional bandwidth or transmitted power. Thus it plays an important role in wireless communications today.

In infrastructure-supported networks, such as the widely used cellular network, the base stations are typical shared by a large number of users. Within the scope of this paper, it is therefore assumed that the base station under consideration is shared by multiple PUs and multiple SUs. In this paper, a MIMO-enhanced CR network is considered to fully ensure the quality of service (QoS) of the PUs as well as to maximize the sum-rate of the SUs. We consider multiple SUs accessing the base station, referred as multiple access channel (MAC). This paper investigates the effective algorithms to find the maximum sum-rate of SUs and the corresponding power allocation solutions.

For the non-CR systems, the sum-rate maximization problem has been intensively explored for both Gaussian broadcast channel (BC) [6, 7] and Gaussian MAC [8]. Typical approaches include iterative water-filling (IWF) algorithm [6, 8] and dual-decomposition [7]. In addition, the set up of the well known duality between the Gaussian BC and the sum-power constrained Gaussian dual MAC [9, 10, 11] facilities the transform of BC sum-rate problem into its dual MAC problem. For the CR-MAC systems, besides the individual power constraints to the SUs, the total interference power from the SUs needs
to be formed into the constraints of the target problem. Earlier works [12, 13] investigated sum-rate problem in CR-SIMO-MAC systems, *i.e.*, partial MIMO as single-input multiple-output (mobiles are equipped with single antenna). In addition, for the ergodic sum capacity of a single input single output (SISO) system, *i.e.*, every user and the BS have a single antenna, [14] studied the maximum sum-rate problem with simple forms of the objective function and the constraints.

In this paper, we extend the analysis to a multiple input multiple output multiple access channel (MIMO-MAC) in the CR network. Both iterative water-filling and dual-decomposition algorithms are investigated to solve the more complicated target problem. The IWF algorithm [6, 8] generally consists of two levels of the loops. The outer loop updates the covariance and inner loop solves the power allocation by fixing the powers of the others. The conventional water-filling algorithm [15] which is an efficient resource allocation approach needs to be used inside each of the iterations in the inner loop operation. However, for the cognitive radio networks, due to more complicated structure of its sum-rate maximization problem, the conventional water-filling [15] cannot be used in the inner loop to compute the solution. In this paper, by exploiting the structure of the proposed sum-rate optimization problem, we firstly propose an efficient iterative water-filling algorithm for the CR network (IWF-CR) to compute the optimal input policy and to maximize the sum-rate for the CR-MIMO-MAC systems via solving a generalized water-filling in each of the iterations.

The water-filling machinery is experiencing continuous development [15, 16, 17, 18, 19]. In this paper, we propose a generalized water-filling algorithm (GWFA) to form a fundamental step (the inner loop algorithm of IWF-CR) for the target problem. In the inner loop, the proposed sum-rate problem is decomposed into a series of generalized water-filling problems with the mixed constraints. Each of the mentioned water-filling problems can be solved by GWFA with a finite number of loops. For the outer loop of IWF-CR, variable scale factor is applied to update the covariance matrices of the users. The optimal scale factor is obtained by maximizing the target objective value (*i.e.*, the sum-rate) in the scalar variable $\beta$ to expedite convergence of the proposed algorithm. In order to achieve this purpose, we determine an optimal scale factor by searching in a range which consists of a few discrete values. As a result, not only does IWF-CR converge, but also it avoid introducing one more inner loop. Also, parallel operation can be used together to further expedite the search and to avoid the requirement of another nested loop. This parallel operation can be distributed to and
carried out by the multiple processors (for example, 4 processors).

Furthermore, we also developed a new iterative algorithm based on the primal-dual approach, referred as **PD-CR**. This algorithm consists of two levels of the loops. The purpose of the inner loop is to evaluate the value of the dual function. For the proposed problem with more constraints, the inner loop under the primal-dual approach only finds an approximate evaluation of the dual function to exit the inner loop and return to the outer loop. Thus, the primal-dual approach may have direct influence on the optimality of the solution. As a characteristic of the proposed **PD-CR** algorithm, to avoid introducing new deeper inner loop that stems from the system model, another new water-filling is proposed in the process to evaluate the dual function. This new water-filling itself can extend the objective function to a combination of the sum-log function with a linear function. As a result, this just mentioned new water-filling and the earlier mentioned **GWFA**, used in **IWF-CR**, extend water-filling method to solve more complicated problems with more general constraints and objective functions.

The advantage of the IWF algorithm is that it is a monotonic feasible operator to the iteration. As a result, the proposed **IWF-CR** algorithm generates a sequence composed of feasible points in its iterations. The objective function values, corresponding to this point sequence, are monotonically increasing. Hence, the stop criterion for computation might be easily set up. However, the primal-dual method is not a feasible point method. In addition, even if the PUs and SUs are served by different base stations, the proposed machinery can be used with some minor modifications.

In the remaining of this paper, system model for a CR-MIMO-MAC system and its sum-rate are described in Section II. Section III firstly proposes a generalized water-filling algorithm, as a fundamental block of the proposed algorithm to solve the maximal sum-rate problem. Then, the proposed algorithms based on the IWF (**IWF-CR**) and the primal-dual approach (**PD-CR**) are presented respectively. Section IV provides the convergence proof of the **IWF-CR**. Section V presents numerical results and some complexity analysis to show the effectiveness of the proposed algorithms.

Key notations that are used in this paper are as follows: $|A|$ and $\text{Tr}(A)$ give the determinant and the trace of a square matrix $A$, respectively; $E[X]$ is the expectation of the random variable $X$; the capital symbol $I$ for a matrix denotes the identity matrix with the corresponding size. A square matrix $B \succeq 0$ means that $B$ is a positive semi-definite matrix. Further, for arbitrary two positive semi-definite
matrices $B$ and $C$, the expression $B \succeq C$ means the difference of $B - C$ is a positive semi-definite matrix. In addition, for any complex matrix, its superscripts $†$ and $T$ denote the conjugate transpose and the transpose of the matrix, respectively.

II MIMO-MAC in CR Network and Its Sum-Rate

For a MIMO-MAC in the CR network, as shown in Fig.1, assume that there are one base-station (BS) with $N_r$ antennas, and $K$ SUs and $N$ PUs, each of which is equipped with $N_t$ antennas. The received signal $y \in \mathbb{C}^{N_r \times 1}$ at the BS is described as

$$y = \sum_{i=1}^{K} H_i^\dagger x^i + \sum_{i=1}^{N} \hat{H}_i^\dagger \hat{x}^i + Z,$$

where $x^i \in \mathbb{C}^{N_t \times 1}$ is a complex input signal vector from the $i$-th SU and is assumed to be a Gaussian random vector having zero mean for any $i$, and $\{x^i\}_{i=1}^{K}$ are independent on the meaning of probability theory. The $\hat{x}^j$ is a complex input signal vector from the $j$-th PU and is assumed to be a Gaussian random vector having zero mean for any $j$, and $\{\hat{x}^j\}_{j=1}^{N}$ are independent on the meaning of probability theory. The noise term, $Z \in \mathbb{C}^{N_r \times 1}$ is an additive Gaussian noise random vector, i.e., $Z \sim \mathcal{N}(0, a^2 I)$, where, without loss of generality, the parameter of the covariance: $a$ is assumed to be a non-zero real number. The channel input, $\{\hat{x}^j\}_{j=1}^{N}$, $\{x^i\}_{i=1}^{K}$ and $Z$ are also assumed to be mutually independent. Furthermore, the $i$-th SU’s transmitted power can be expressed as

$$S_i \triangleq E[|x^i|^2]$$

Note that $S_i$, $\forall i$, is positively semi-definite, i.e., $S_i \succeq 0$.

The mathematical model of the sum-rate optimization problem for the MIMO-MAC in the CR network can be stated as follows (refer to (2.16) in [20] and therein):

Given a group of weights $\{w_k\}_{k=1}^{K}$ which must be in (non-strictly) decreasing order with the achiev-
able rate of the $k$th secondary user,

$$\log \frac{|C_0 + \sum_{j=1}^{k} H_j^H S_j H_j|}{|C_0 + \sum_{j=1}^{K} H_j^H S_j H_j|}, \forall k,$$

the sum-rate is organized by

$$C_{macn} \left( H_1^\dagger, \ldots, H_K^\dagger; P_1, \ldots, P_K; P_T \right) = \max_{\{S_k\}_{k=1}^K} w_1 \log \frac{|C_0 + H_1^H S_1 H_1|}{|C_0|} + \sum_{k=2}^{K} w_k \log \frac{|C_0 + \sum_{j=1}^{k-1} H_j^H S_j H_j|}{|C_0 + \sum_{j=1}^{K} H_j^H S_j H_j|}$$

Subject to: $S_k \succeq 0$; $\text{Tr}(S_k) \leq P_k, \forall k$; $\sum_{k=1}^{K} g_k \text{Tr}(S_k) \leq P_T,$

where, for the MAC cases, the peak power constraint on the $k$th SU exists and is denoted by a group of positive numbers: $P_k, k = 1, \ldots, K$; the power threshold to ensure the QoS of the PUs is denoted by a positive number $P_T$. Further, under no confusion, $C_{macn} + w_1 \log |C_0|$ is simply written as $f$. Also, the feasible set of the model (3) is denoted by $V$. Since $w_k - w_{k+1} = 0$ for $k = 1, \ldots, K - 1$; and $w_K = \frac{1}{K}$, (3) is simplified as follows:

$$C_{macn} \left( H_1^\dagger, \ldots, H_K^\dagger; P_1, \ldots, P_K; P_T \right) + \frac{1}{K} \log |C_0| = \max_{\{S_k\}_{k=1}^K} \log |C_0 + \sum_{j=1}^{K} H_j^H S_j H_j|$$

Subject to: $S_k \succeq 0$; $\text{Tr}(S_k) \leq P_k, \forall k$; $\sum_{k=1}^{K} g_k \text{Tr}(S_k) \leq P_T,$

Note that (3) and (4) have the same solution, i.e., they are equivalent. Since they only have a constant difference, they can be regarded as the same for simplicity. Further, the term $g_k = \sup_{x \neq 0} \frac{\|H_k x\|}{\|x\|}, \forall k,$ is the channel power gain of the $k$th SU to the BS with a proper norm. That is to say, $g_k$ is the norm
of a linear operator, and then it can be taken as the (non-negative) maximum eigenvalue of the linear operator $H_k^*H_k$. The norm of a linear operator is the maximum ratio of norm of the image for the operator to that of original image. Thus $g_k$ is a ratio from the powers from the two kinds of signals. Also, we denoted the covariance matrix of the random vector $\sum_{i=1}^{N} \hat{H}_i^*\hat{x}_i + Z$ by $C_0$ which is positive definite.

The constraint $\sum_{k=1}^{K} g_k \text{Tr}(S_k) \leq P_T$ is called the sum-power constraint with gains. The constraint is obtained in the following analysis. The received signal at BS in (1) can be written as

$$\mathbf{y} = \sum_{i=1}^{N} \hat{H}_i^*\hat{x}_i + \left( \sum_{i=1}^{K} H_i^*x_i + Z \right),$$

where $\sum_{i=1}^{K} H_i^*x_i + Z$ can be regarded as the additive interference and noise to the transmitted signal $\sum_{i=1}^{N} \hat{H}_i^*\hat{x}_i$ from the PUs. To guarantee the QoS for the PUs, the power of the interference and noise should be less than a threshold value, $P_{TH}$. This condition can be expressed as

$$\sum_{i=1}^{K} \text{Tr}(H_i^*E(xx^*)H_i + E(ZZ^*)) \leq P_{TH}.$$ (6)

It can be implied by the following condition:

$$\sum_{k=1}^{K} g_k \text{Tr}(S_k) \leq P_{TH} - N_r\sigma^2 = P_T,$$ (7)

where the power constraint value $P_T$ is the interference and noise threshold subtracted by the power of Gaussian noise.

As an alternative, to guarantee the QoS for each of the PUs individually, the power of the interference and noise should be less than a threshold value, $P_{TH}(i), \forall i$. Similarly, it is obtained that

$$\sum_{k=1}^{K} g_k \text{Tr}(S_k) \leq P_T(i), \forall i.$$ (8)
That is to say, the condition above is equivalent to

$$\sum_{k=1}^{K} g_k \text{Tr}(S_k) \leq \min_i \{ P_T(i) \}. \quad (9)$$

Name $\min_i \{ P_T(i) \}$ as $P_T$ and then the target model can still cover the case that the QoS for each of the PUs is considered. Note that at the base station with multiple antennas, the received signals can be regarded as a stochastic vector or point in a Hilbert space and its power is abstracted into the norm squared of the vector. The transmitted powers of the PUs have been taken into account by forming $C_0$ and $P_T$ mentioned above, which appear in (4).

A more strict sum-rate model can also be obtained that reflects the essence of the issue for the CR-MIMO-MAC. Along a similar way mentioned above, we may choose the power thresholds $\{ P_{T,i} \}$ to limit the impact from the SUs on each of the antennas of the BS. Thus the sum-power constraint with the gains is evolved into $\sum_{k=1}^{K} g_{k,i} \text{Tr}(S_k) \leq P_{T,i}, i = 1, 2, \ldots, N_r$. It is seen that such a sum-rate problem with more power constraints can be solved by solving a similar problem in (4). Therefore, the proposed paper aims at computing the solution to the problem (4). Note that if $\exists H_{i_0} = 0, 1 \leq i_0 \leq K$, for (4), we remove the user $i_0$ and then the number of the users is reduced to $K - 1$. Thus, on this way, we can assume that $H_i \neq 0, \forall i$. It implies $g_i > 0, \forall i$.

## III. Algorithms for MIMO-MAC CR Systems

In this section, we discuss the proposed algorithms to solve the maximal sum-rate problem in MIMO-MAC CR systems. The generalized water-filling problem with the mixed constraints is first investigated. The proposed **IWF-CR** and its implementation are presented in the second subsection. In the last subsection, we present an algorithm based on the primal-dual approach, **PD-CR**, to solve the target problem.

### III.1 Generalized Water-Filling Algorithm (GWFA)

Being a fundamental block of the optimum resource allocation problem for the CR-MU-MIMO systems, the generalized water-filling problem is abstracted as follows.
For a multiple receiving antenna system of parallel independent channels that are divided as $K$ groups, each group of which has $N_t$ channels, it is given that $P_T > 0$, as the total power or volume of the water; the propagation path (non-negative) gains $\{a_k\}_{k=1}^{KN_t}$ of the channels are partitioned as the $K$ groups, the index sets of which are labelled as $\{\Lambda_i\}, i = 1, \ldots, K$ such that $\{\Lambda_i\}_{i=1}^{K}$ is a partition of $\{1, 2, \ldots, KN_t\}$, and the cardinality of $\Lambda_i$ is $N_t, \forall i$; all the channels in the group $i$ share a power gain labelled as $g_i$, i.e., $\{a_j\}_{j \in \Lambda_i}$ corresponds to $g_i$; and the allocated powers of the $i$th group of channels are given as $\{s_{i,j}\}_{j \in \Lambda_i}, \forall i$.

From the partition mentioned above, a mapping $\sigma$ is defined, for clear statement, as follows: if for channel $j$, there exists a unique $i, 1 \leq i \leq K$ such that $j \in \Lambda_i$, then $\sigma(j) = i$. Without loss of generality, it is assumed that the sequence $\{a_k/g_{\sigma(k)}\}_{k=1}^{KN_t}$ is monotonically decreasing; else, we would have to tediously utilize permutation of the subscript sequence used by the summation operator. Thus, under the assumptions mentioned above, we can find that

$$\max_{\{s_k\}_{k=1}^{KN_t}} \sum_{k=1}^{KN_t} \log(1 + a_k s_k)$$
$$\text{subject to: } 0 \leq s_k, \forall k;$$
$$\sum_{k \in \Lambda_i} s_k \leq P_i, \forall i;$$
$$\sum_{i=1}^{K} \sum_{k \in \Lambda_i} g_i s_k \leq P_T.$$  

(10)

Note, as $\sum_{i=1}^{K} g_i P_i \leq P_T$, the solution to problem (4) is regressed into a trivial case. Hence, $\sum_{i=1}^{K} g_i P_i > P_T$ is assumed. When $P_i >> 0, \forall i$, then the problem (10) is reduced into the individual case that can be solved by the conventional water-filling problem [15]. In general case, the problem structure in (10) cannot be solved by the conventional water-filling. In the following, our proposed generalized water-filling algorithm (GWFA) is presented to solve this generalized radio resource management problem.

Firstly let us introduce a vivid description of water-filling algorithm from a geometric point of view by pouring the water of volume $P_T$ into a tank with the bottom of $KN_t$ stairs as shown in Fig. 1 for four steps/stairs ($K = 4$) with unit width inside a water tank. For the conventional approach, the dashed horizontal line, which is the water level $\mu$, needs to be determined first and then the power allocated (water volume) above is solved.
In the following, we will introduce four variables used in our approach. The first variable, $P_2(k)$, is defined as the total water volume above the $k$th stair. The second variable is $s_i$, as the allocated power for the $i$th channel. The third variable is water level step, denoted as $k^*$. It denotes the highest step under water. The fourth variable, $s_{k^*}$ is defined as the optimal power allocated to the water level step. Fig. 1(a) illustrates the concept of $k^*$. Since the third level is the highest level under water, we have $k^* = 3$. The shaded area denotes the allocated power for the third step by $s_{3^*}$. Fig.1(b) and Fig.1(c) illustrate the concept of $P_2(k)$ when $k = 2$ and $k = 3$ respectively.

Let us use $g_{\sigma(i)}/a_i$ to denote the “step depth” of the $i$th stair which is the height of the $i$th step to the bottom of the tank, and is given as

$$d_i = \frac{g_{\sigma(i)}}{a_i}, \text{ for } i = 1, 2, \ldots, KN_t. \tag{11}$$

Since the sequence $\{a_i/g_{\sigma(i)}\}$ is sorted as monotonically decreasing, the step depth of the stairs indexed as $[1, \cdots, KN_t]$ is monotonically increasing. We further define $\delta_{i,j}$ as the “step depth difference” of the $i$th and the $j$th stairs, expressed as

$$\delta_{i,j} = d_i - d_j = \frac{g_{\sigma(i)}}{a_i} - \frac{g_{\sigma(j)}}{a_j}, \text{ as } i \geq j \text{ and } 1 \leq i, j \leq KN_t. \tag{12}$$

Instead of trying to determine the water level $\mu$ which is a real nonnegative number, as in the

![Figure 2: Illustration for the proposed Generalized Water-Filling (GWF) algorithm ($\sigma(i)$ is assumed to be a constant, for any $i$). (a) Illustration of water level step $k^* = 3$, allocated power for the third step $s_{3^*}^*$, and step/stair depth $d_i = g_{\sigma(i)}/a_i$. (b) Illustration of $P_2(k)$ (shadowed area, representing the total water/power above step $k$) when $k = 2$. (c) Illustration of $P_2(k)$ when $k = 3$.](https://example.com/figure2.png)
conventional WF algorithm, we aim to determine the water level step, \( k^* \), which is an integer number from 1 to \( K N_t \), as the highest step under water. Based on the result of \( k^* \), we can write out the solutions for power allocation instantly.

In the following, we explain how to find \( k^* \) without the knowledge of the water level \( \mu \). The value of \( P_2(k) \) can be solved by subtracting the volume of the water under step \( k \) from the total power \( P_T \), as

\[
P_2(k) = \left\{ P_T - \left[ \sum_{i=1}^{kN_t-1} \left( \frac{g_\sigma(k)}{a_k} - \frac{g_\sigma(i)}{a_i} \right) \right] \right\}^+ = \left\{ P_T - \left[ \sum_{i=1}^{kN_t-1} \delta_{k,i} \right] \right\}^+, \quad k = 1, \ldots, K N_t. \tag{13}
\]

As an example of Fig.1(c), the water volume under step 3 can be expressed as the sum of the two terms: (i) the step depth difference between the 3rd and the 1st step, \( \delta_{3,1} \), and (ii) the step depth difference between the 3rd and the 2nd step, \( \delta_{3,2} \). Thus, \( P_2(k = 3) \) can be written as

\[
P_2(k = 3) = [P_T - \delta_{3,1} - \delta_{3,2}]^+
\]

which is an expansion of the composite form of (13).

Generally, from Fig.1, we can have

\[
P_2(k) = \left\{ P_T - \sum_{i=1}^{E^\sharp-1} \left( \frac{g_\sigma(i_k)}{a_{i_k}} - \frac{g_\sigma(i_i)}{a_{i_i}} \right) \right\}^+, \quad \text{for } k = 1, \ldots, E^\sharp, \tag{14}
\]

where \( E \) is a subsequence of the sequence \( \{1, 2, \ldots, K N_t\} \), \( E^\sharp \) is the cardinality of the set \( E \), so \( E \) can be expressed as \( \{i_1, i_2, \ldots, i_{E^\sharp}\} \). Note a reminder of the definition of a special case for the summation is:

\[
\sum_{i=m}^{n} b_i = 0, \quad \text{as } m > n.
\]

Then the water level step \( k^* \) is given as

\[
k^* = \max \left\{ k \middle| P_2(k) > 0, \quad 1 \leq k \leq E^\sharp \right\} \tag{15}
\]
and the power level for this step is

\[ s_{i_k^*} = \frac{1}{k^* g_{\sigma(i_k^*)}} P_2(k^*). \] (16)

and the power levels for all other steps are given as

\[ s_i = \begin{cases} 
\frac{g_{\sigma(i_k^*)}}{g_{\sigma(i_t)}} \left( s_{i_{k^*}} + \frac{1}{a_{k^*}} \right) - \frac{1}{a_i}, & 1 \leq t \leq k^* \\
0, & k^* < t \leq E^#. 
\end{cases} \] (17)

Based on these results, the steps of \textbf{GWFA} can be described as below.

**Algorithm GWFA:**

\textbf{Input:} the channel gains \( \{a_i\}_{i=1}^{KN_t} \), the power gains \( \{g_i\}_{i=1}^{K} \), the individual power peak or upper limit \( \{P_i\}_{i=1}^{K} \), the index set \( E = (E_0 =) \{1, 2, \ldots, KN_t\} \), the partition \( \{\Lambda_i\}_{i=1}^{K} \) and the sum power constraint \( P_T \).

1) utilize (14)-(17) to compute \( \{s_i\} \).

2) The set \( \Lambda \) is defined by the set \( \{i| \sum_{j \in \Lambda_i} s_j > P_i, 1 \leq i \leq K\} \). If \( \Lambda \) is the empty set, output \( \{s_i\}_{i=1}^{KN_t} \), else, let \( \sum_{j \in \Lambda_i} s_j = P_i \), as \( i \in \Lambda \). Further, continuously utilize similar expressions to (14)-(17), these similar expressions only changing from \( P_T \) to \( g_i P_i \) and from \( E^# \) to \( \Lambda_i \) for any \( i \in \Lambda \), and then obtain \( s_j, j \in \cup_{i \in \Lambda} \Lambda_i \).

3) \( E <= E \setminus \cup_{i \in \Lambda} \Lambda_i \), where the symbol “<=” means the assignment operation forwarding the value of the RHS (right-hand side) to that of the LHS (left-hand side). \( P_T <= P_T - \sum_{i \in \Lambda} g_i P_i \). Then return to 1) of \textbf{GWFA}.

**Remark 3.1.** \textbf{GWFA} is a dynamic power distribution process. The state of this process is the difference between the individual peak power sequence \( \{P_i\} \) and the sums \( \sum_{k \in \Lambda_i} s_k \) from current power distribution sequence obtained by (14)-(17). The control of this process is to use the similar expressions (14)-(17) based on the state mentioned above. Thus, a new state for next time stage appears. Therefore, an optimal dynamic power distribution process, \textbf{GWFA}, with the state feedback is formed. Since the finite set \( E \) is getting smaller and smaller until the set \( \Lambda \) is empty, \textbf{GWFA} carries out \( K \) loops to compute the optimal solution, at most.
For the optimality of the proposed algorithm GWFA, we can obtain the following conclusion:

**Proposition 3.1:** GWFA can provide the exact optimal solution to the problem (10) via finite computation.

**Proof.** See Appendix A. □

From the computational details of GWFA, the computational complexity of GWFA is, at worst, \( \sum_{i=1}^{KN_t} (8i + 2) = 4K N_t^2 + 6KN_t \), which is a moderate computational complexity. In addition, for a weighted water-filling problem with sum and individual peak power constraints, as a generalized case, Proposition 3.1 is also applicable.

**Remark 3.2.** To decouple the variables in the objective function of the problem (4), a sum expression is acquired by adding the objective function \( K \) times. Then the sum expression is operated, by one matrix-valued variable being selected as an optimized variable with respect to the others being fixed. Through this way, the sum expression being obtained by adding the expression (4) \( K \) times, the problem (18)

\[
\max \left\{ X(i,j) \mid j=1, \ldots, N_t \sum_{j=1}^{N_t} X(i,j) \leq P_t \forall j; \sum_{i=1}^{K} g_i \sum_{j=1}^{N_t} X(i,j) \leq P_T \right\} 
\]

(18)
is obtained, where \( X(i,j) \) means the \( j \)th diagonal entry of the matrix \( S_i \) for \( j = 1, \ldots, N_t \) and any \( i \), \( a_j(G_i(G_i)\dagger) \) is the \( j \)th decreasing sorted eigenvalue of the matrix \( G_i(G_i)\dagger \), \( \forall j \) and \( i \), and can be obtained from the eigenvalue decomposition. The concise derivation from the sum expression to (18) is stated as follows:

Since

\[
\sum_{i=1}^{K} \log \left| C_0 + H_i^\dagger S_i H_i + \sum_{k \in \{1, \ldots, K\} \setminus \{i\}} H_k^\dagger S_k H_k \right| 
\]

= \[
\sum_{i=1}^{K} \log \left| I + G_i^\dagger S_i G_i \right| + \sum_{i=1}^{K} \left| C_0 + \sum_{k \in \{1, \ldots, K\} \setminus \{i\}} H_k^\dagger S_k H_k \right|, 
\]

(19)
where $S_k, \forall k$, is fixed and
\[
G_i = H_i \left( C_0 + \sum_{k \in \{1, \ldots, K\} \setminus \{i\}}^K H_k^\dagger S_k H_k \right)^{-\frac{1}{2}}, \forall i, \tag{20}
\]

the optimization problem
\[
\max_{\{S_k\}_{k=1}^K: S_k \succeq 0, \, \text{Tr}(S_k) \leq P_k, \forall k; \, \sum_{k=1}^K g_k \text{Tr}(S_k) \leq P_T} \sum_{i=1}^K \log \left| C_0 + H_i^\dagger S_i H_i + \sum_{k \in \{1, \ldots, K\} \setminus \{i\}}^K H_k^\dagger S_k H_k \right| \tag{21}
\]
is equivalent to the problem below:
\[
\max_{\{S_k\}_{k=1}^K: S_k \succeq 0, \, \text{Tr}(S_k) \leq P_k, \forall k; \, \sum_{k=1}^K g_k \text{Tr}(S_k) \leq P_T} \sum_{i=1}^K \log \left| I + G_i^\dagger S_i G_i \right|. \tag{22}
\]

From the well known Hadamard’s inequality on positive definite matrices and some matrix operations, (22) is equivalent to the problem (18). Note that, (10) earlier mentioned is the same as the problem (18).

Briefly in summary, the logical line mentioned above is: (4) leads to the sum expression, the sum expression implies (19) under the one variable being selected as an optimized variable with respect to the others being fixed, (19) leads to (22), and then (22) results in (18) or (10).

III.2 Algorithm IWF-CR and Its Implementation

The proposed IWF-CR is based on the combined problem of both the MIMO MAC and the CR network. The algorithm is listed below.

Algorithm IWF-CR:

**Input:** Matrices $H_i$, $S_i^{(0)} = 0$, $i = 1, \ldots, K$; $n = 1$.

1) Generate effective channels
\[
G_i^{(n)} = H_i \left( C_0 + \sum_{k \in \{1, \ldots, K\} \setminus \{i\}}^K H_k^\dagger S_k^{(n-1)} H_k \right)^{-\frac{1}{2}}, \forall i, \tag{23}
\]
where the superscript with a pair of bracket, \((n)\), represents the number of iterations.

2) Treating these effective channels as parallel, noninterfering channels, the new covariances \(\{\tilde{S}^{(n)}_i\}_{i=1}^K\) are generated by GWFA under the sum power constraint \(P_T\). That is to say, \(\{\tilde{S}^{(n)}_i\}_{i=1}^K\) is the optimal solution to (24):

\[
\max_{\{S_i\}_{i=1}^K} \quad \text{s.t.} \quad S_k \succeq 0, \quad \text{Tr}(S_k) \leq P_k, \quad \sum_{k=1}^K g_k \text{Tr}(S_k) \leq P_T \sum_{i=1}^K \log |I + (G^{(n)}_i \tilde{S}^{(n)}_j G^{(n)}_i)^\dagger| \leq P_T K \sum_{i=1}^K \log |I + (G^{(n)}_i \tilde{S}^{(n)}_j G^{(n)}_i)^\dagger|.
\]

(24)

Note that (24) is just the expression of (22) in this algorithm.

3) Update step: Let \(\gamma^{(n)}\) and \(p^{(n-1)}\) denote the newly obtained covariance set and the immediate past covariance set respectively,

\[
\gamma^{(n)} \triangleq \left(\tilde{S}^{(n)}_1, \tilde{S}^{(n)}_2, \ldots, \tilde{S}^{(n)}_K\right) \quad \text{and} \quad p^{(n-1)} \triangleq \left(S^{(n-1)}_1, S^{(n-1)}_2, \ldots, S^{(n-1)}_K\right).
\]

Let

\[
\beta^* = \max \left\{ \beta_1 \mid \beta_1 \in \arg \max_{\beta \in [1/K, 1]} f \left(\beta \gamma^{(n)} + (1 - \beta) p^{(n-1)}\right) \right\},
\]

(25)
as the innovation, where the function \(f\) or \(C_{macn} + w_1 \log |C_0|\) has been defined near (4). Then, the covariance update step is

\[
p^{(n)} = \left(S^{(n)}_1, S^{(n)}_2, \ldots, S^{(n)}_K\right) = \beta^* \gamma^{(n)} + (1 - \beta^*) p^{(n-1)}.
\]

(26)
The updated covariance is a convex combination of the newly obtained covariance and the immediate past covariance.

4) Let \(n <= n + 1\), where \(<=\) is the assignment operator. Go to 1) until convergence.

Note that the new algorithm employs variable weighting factors, which are obtained to maximize the objective function and to update the covariance.

In this section, the optimality of \(\{\tilde{S}^{(n)}_i\}_{i=1}^K\) has been proved, i.e., \(\{\tilde{S}^{(n)}_i\}_{i=1}^K\) is the solution to (18), by Proposition 3.1.
**Remark 3.3.** Due to the objective function $f\left(\beta \gamma^{(n)} + (1 - \beta)p^{(n-1)}\right)$ in Step 3) of Algorithm IWF-CR being (upper) convex, i.e., being concave, in the scalar variable $\beta$, for computing the maximum solution to the corresponding optimization problem, we can choose finite searching steps with even fewer evaluations of the objective function. Without loss of generality, the objective function in step 3) is evaluated at the four points $\{\beta = \frac{1}{K}, \frac{1}{K} + \frac{1}{3}(1 - \frac{1}{K}), \frac{1}{K} + \frac{2}{3}(1 - \frac{1}{K}) \text{ and } 1\}$ by parallel computation to determine $\beta^*$. That is to say, this parallel operation can be distributed to and carried out by the multiple processors (for example, 4 processors) of the base station in order to expedite convergence of the proposed algorithm. Finally, the obtained satisfied solution is then distributed or returned to the corresponding secondary users.

### III.3 Algorithm Based on Primal-Dual Approach

In the following, we discuss another algorithm, PD-CR, which uses the primal dual approach to solve the target problem as below.

Given $\lambda \geq 0$ and the optimization problem:

\[
\max_{\{S_k\}_{k=1}^K} \log |C_0 + \sum_{j=1}^K H_j S_j H_j^\dagger| - \lambda(\sum_{k=1}^K g_k \text{Tr}(S_k) - P_T)
\]

subject to $S_k \succeq 0, \text{Tr}(S_k) \leq P_k, \forall k$, \hspace{1cm} (27)

an efficient iterative algorithm is proposed here and the optimal objective function value, i.e., the dual function, for the problem (27) is denoted by $f_d(\lambda)$. It is seen that $f_d(\lambda)$ is a convex function over $\lambda \geq 0$, and $\lambda$ is a scalar. Thus, we may use the sub-gradient algorithm or a line search to obtain the optimal solution $\lambda^*$ to the minimum value problem, as the outer loop of the primal dual approach, of the dual function. Since the range to search is quite important, a pair of upper and lower bounds are proposed as follows.

**Proposition 3.2:** For the dual function $f_d(\lambda)$ minimization problem, as $\lambda \geq 0$ in (27), the optimal solution

$$\lambda^* \in \left[0, \frac{1}{a^2}\right].$$

Its proof comes from the observation of the KKT conditions of a derived model from the dual function $f_d(\lambda)$ minimization problem. It has been given that $a^2$ is of the covariance of the noise vector.
Tightness of the interval or pair of the upper and lower bounds means that there exists a set of channel gains such that its optimal Lagrange multiplier $\lambda^*$ touches either of the ends of the interval. For example, as $K = 1$, $g_1 = 1$, $P_T = 2$, $P_1 = 1$, $a = 1$ and $h_1 = 1$, it is seen that $\lambda^* = 0$.

Letting $\lambda \in (0, \frac{1}{a^2}]$, consider the evaluation of $f_d(\lambda)$. Note that the problem (27) has decoupled constraints. Therefore, the block coordinate ascend algorithm (BCAA) or the cyclic coordinate ascend algorithm (CCAA) (see [21]) can be used to solve the problem efficiently. The iterative algorithm works as follows. In each step, the objective function is maximized over a single matrix-valued variable $S_k$, while keeping all other $S_k$s fixed, $k = 1, \cdots, K$ and then repeating this process. Since the objective is nondecreasing with each iteration, the algorithm must converge to a fixed point. Using the fixed point theory, the fixed point is an optimal solution to the problem (27). Without loss of generality, let us consider an optimization problem below over $S_k$, $k = 1$, with respect to all other $S_k$s being fixed, as follows:

$$
\max_{\{S_1\}} \quad \log |C_0 + \sum_{j=1}^{K} H_j S_j H_j^\dagger| - \lambda(\sum_{k=1}^{K} g_k \text{Tr}(S_k) - P_T)
$$

subject to $S_k \succeq 0$, $\text{Tr}(S_k) \leq P_k$. \hfill (28)

It is known that $g_i > 0$ due to $H_i \neq 0$, $\forall i$. Note that the single user case above with respect to the fixed other users is different from the existing ones for the (dual) MIMO MAC cases since it has the more complicated objective function or feasible set. However, we can still exploit the new water-filling to solve the problem given in (28). The skeleton is stated as follows.

Problem (28) is equivalent to the following problem:

$$
\max_{\{S_1\}} \quad \log |I + GS_1 G^\dagger| - \lambda(g_1 \text{Tr}(S_1) - P_T)
$$

subject to $S_1 \succeq 0$, $\text{Tr}(S_1) \leq P_1$, \hfill (29)

where $G \triangleq (C_0 + \sum_{j \neq 1} H_j S_j H_j^\dagger)^{-\frac{1}{2}} H_1$. Problem (29) can further be proven to be equivalent to the problem (30) given as,

$$
\max_{\{x_i\}^{N_1}} \quad \sum_{i=1}^{N_1} \log(1 + \lambda_i x_i) - \lambda(g_1 \sum_{i=1}^{N_1} x_i - P_T)
$$

subject to $x_i \geq 0, \forall i, \sum_{i=1}^{N_1} x_i \leq P_1$, \hfill (30)
where the matrix \( \text{diag} (\lambda_1, \cdots, \lambda_N) \), with \( \{\lambda_i\} \) being decreasing ordered, is unitarily equivalent to the matrix \( \mathbf{G}^\dagger \mathbf{G} \) by the unitary matrix \( \mathbf{U} \) that is a matrix expression of the unitary similarity transformation. It is seen that we can compute the optimal solution \( \{x_i^*\} \) to the problem (30) and then obtain the optimal solution \( \mathbf{U} \text{diag} (x_1^*, \cdots, x_N^*) \mathbf{U}^\dagger \) to the problem (28).

How to compute the optimal solution \( \{x_i^*\} \) to the problem (30), as the new water-filling, is concisely proposed as follows: If \( \lambda_i = 0 \), then \( x_i^* = 0 \); if \( 0 < \lambda_i \leq \lambda g_1 \), then \( x_i^* = 0 \). Further, if \( \exists k_0 \) such that \( \lambda g_1 < \lambda k_0 \triangleq \max \{k | \lambda g_1 < \lambda k, 1 \leq k \leq N_t\} \). Thus, under the condition above, if \( \frac{k^*}{\lambda g_1} - \sum_{k=1}^{k^*} \frac{1}{\lambda_k} < P_1 \), then \( x_i^* = \frac{1}{\lambda g_1} - \frac{1}{\lambda_i}, \forall i : 1 \leq i \leq k^* \), and \( x_i^* = 0, \forall i : k^* < i \leq N_t \); else if \( \frac{k^*}{\lambda g_1} - \sum_{k=1}^{k^*} \frac{1}{\lambda_k} \geq P_1 \), then \( x_i^* = \frac{P_1 + \sum_{k=1}^{k^*} \frac{1}{\lambda_k}}{\lambda_i} - \frac{1}{\lambda_i}, \forall i = 1, \ldots, k^* \), and \( x_i^* = 0, \forall i = k^* + 1, \ldots, N_t \).

Therefore, for \( \lambda \), given \( \{\mathbf{S}_1^{(n)}, \cdots, \mathbf{S}_K^{(n)}\} \), the BCAA is used from the first matrix-valued variable to the \( K \)-th variable, and we obtain

\[
\{\mathbf{S}_1^{(n+1)}, \cdots, \mathbf{S}_K^{(n+1)}\}.
\]

Thus there is a mapping which projects

\[
\{\mathbf{S}_1^{(n)}, \cdots, \mathbf{S}_K^{(n)}\} \rightarrow \{\mathbf{S}_1^{(n+1)}, \cdots, \mathbf{S}_K^{(n+1)}\}, \forall n.
\]

This mapping is denoted by \( f_1 \).

With the assumptions and the concepts introduced, the proposed PD-CR algorithm, which is based on the primal-dual approach and the new water-filling algorithm, is concisely described as follows.

**Algorithm PD-CR:**

1) Given \( \varepsilon > 0 \), initialize

\[
\{\mathbf{S}_1^{(0)} = 0, \cdots, \mathbf{S}_K^{(0)} = 0\}, \lambda_{\text{min}} \text{ and } \lambda_{\text{max}}.
\]

2) Set \( \lambda = (\lambda_{\text{min}} + \lambda_{\text{max}})/2 \).

3) Compute

\[
\{\mathbf{S}_k^{(n+1)}\}^K_{k=1} = f_1 \left( \{\mathbf{S}_k^{(n)}\}^K_{k=1} \right).
\]
Then $n <= n + 1$. Repeat the above process until the optimal solution $\{S_k^*\}_{k=1}^K$ to the problem (27) is reached, where the BCAA mentioned before is used.

4) If $\sum_{k=1}^K g_k \text{Tr}(S_k^*) - P_T > 0$, then $\lambda_{\min}$ is assigned by $\lambda$;
   
   if $\sum_{k=1}^K g_k \text{Tr}(S_k^*) - P_T < 0$, then $\lambda_{\max}$ is assigned by $\lambda$;
   
   If $\sum_{k=1}^K g_k \text{Tr}(S_k^*) - P_T = 0$, stop.

5) If $|\lambda_{\min} - \lambda_{\max}| \leq \varepsilon$, stop. Otherwise, goto step 2).

Note that according to Proposition 3.2, the initial $\lambda_{\min}$ is chosen as 0, and $\lambda_{\max}$ is chosen as $\frac{1}{\sigma^2}$, respectively. In the item 4) of PD-CR, it is not difficult to see that $P_T - \sum_{k=1}^K g_k \text{Tr}(S_k^*)$ is a subgradient of the function $f_d(\lambda)$. This is because

$$f_d(\lambda_1) = \max_{\{S_k\}_{k=1}^K} \{ \log |C_0 + \sum_{j=1}^K \mathbf{H}_j \mathbf{S}_j \mathbf{H}_j^\dagger| - \lambda_1 (\sum_{k=1}^K g_k \text{Tr}(S_k) - P_T) \}$$

$$\geq \log |C_0 + \sum_{j=1}^K \mathbf{H}_j S_j^* \mathbf{H}_j^\dagger| - \lambda_1 (\sum_{k=1}^K g_k \text{Tr}(S_k^*) - P_T)$$

$$\geq f_d(\lambda) + (\lambda_1 - \lambda)(P_T - \sum_{k=1}^K g_k \text{Tr}(S_k^*)),$$

where $\{S_k^*\}$ is an optimal solution to (27), to evaluate $f_d(\lambda)$.

According to the definition of the subgradient,

$$P_T - \sum_{k=1}^K g_k \text{Tr}(S_k^*)$$

is a subgradient of the function $f_d(\lambda)$, although $f_d(\lambda)$ cannot be guaranteed to be differential. Thus, we can follow the subgradient algorithm to search better $\lambda$ in the item 4).

### IV Convergence of Algorithm IWF-CR

For convergence of the proposed algorithm, according to [22], if Algorithm IWF-CR took $\beta^* = \frac{1}{K}$ and the target problem were not a cognitive network, it would have been proven that the proposed algorithm is convergent. Thus, we will strictly prove the proposed algorithm to be convergent, for theoretic integrity. Since our convergence proof is based on more general functions including an
objective function and constraint functions, it will also enrich the optimization theory and methods. We assume that a mapping projects a point to a set. First, two concepts are introduced. The first concept is of an image of a mapping (or algorithm) that projects a point to a set; the second one is of a fixed point under the mapping (algorithm). Then, three lemmas are proposed, followed by the convergence proof of the proposed Algorithm.

Definition 4.1. (Image under Mapping or Algorithm A) (see e.g. [23], page 84) Assume that $X$ and $Y$ are two sets. Let $A$ be a mapping or an algorithm from $X$ to $Y$, which projects from a point in $X$ to a set of points in $Y$. If the point in $X$ is denoted by $x$ and the set of the points in $Y$ is denoted by $A(x)$, then $A(x)$ is called the image of $x$ under $A$.

Definition 4.2. (Fixed Point under Mapping or Algorithm A) Let $A$ be a mapping or an algorithm from $X$ to $Y$. Assume $x \in X$. If $x \in A(x)$, $x$ is said to be a fixed point under $A$.

Note that (18) can be changed into a general form:

$$\left\{ S_i^{(n)} \right\}_{i=1}^{K} \in \arg\max_{\left\{ S_i \right\}_{i=1}^{K}, \; S_i \succeq 0, \; \text{Tr}(S_i) \leq P_i, \; \forall i, \; \sum_{i=1}^{K} \text{Tr}(S_i) \leq P_T, \; \sum_{i=1}^{K} \log |I + \left( G_i^{(n)} \right)^\dagger S_i G_i^{(n)}|, \; \forall n, \right\}$$

Due to the condition of the optimal solution uniqueness being removed. Further, corresponding to this change, step 2) of Algorithm IWF-CR will be carried out in this way: given a feasible point $\left\{ S_i^{(n)} \right\}_{i=1}^{K}$, its image under step 2) of Algorithm IWF-CR is a set of points. A point in this set is chosen arbitrarily as the next point $\left\{ S_i^{(n+1)} \right\}_{i=1}^{K}$ generated by Algorithm IWF-CR. Thus, Algorithm IWF-CR can generate a point sequence under this change. In the following, we will still call this algorithm IWF-CR despite the changes.

The mathematical expression (4) may be regarded as an optimization problem over the field of complex numbers. However, this optimization problem is also equivalent to an optimization problem over the field of real numbers based on the following lemma. This lemma is an indispensable tool to prove the necessary condition in Lemma 4.3 below, and it could avoid discussing the differentiability of the objective function in (4).

Lemma 4.2. The optimization problem in (4) is equivalent to a convex optimization problem over the field of real numbers.

Proof of Lemma 4.2 is easy to show due to the existence of isomorphism between $S_i$ and $(\Re (S_i),$
Only note that the constraints of the Hermitian matrix \( S_i \) and \( \sum_{i=1}^{K} g_i \text{Tr} (S_i) \leq P_T \) are changed into the constraints on \((\Re(e(S_i)), \Im(S_i))\) that define a feasible set \( V_r \) with respect to \( x_i \), where \( x_i \) is defined as \((\text{vec}(\Re(e(S_i))), \text{vec}(\Im(S_i)))\), for \( i = 1, 2, \cdots, K \). It is easily shown that \( V_r \) is convex and closed. That is to say, the objective function of the problem in (4) is transformed into a function, denoted by \( f_r \), of \( \{x_i\}_{i=1}^{K} \).

For any convergent subsequence, whose limit is denoted by \((S_1, \cdots, S_K)\), generated by Algorithm IWF-CR, we may use the following lemma to prove that the limit is a fixed point under Algorithm IWF-CR, when Algorithm IWF-CR is regarded as a mapping.

**Lemma 4.3.** A point is the limit of a convergent subsequence of the point sequence generated by Algorithm IWF-CR if and only if this point is a fixed point under Algorithm IWF-CR.

**Proof.** See Appendix B.

**Lemma 4.4.** \((S_1, \cdots, S_K) \in V\) is a fixed point under Algorithm IWF-CR if and only if \((S_1, \cdots, S_K) \in V\) is one of the optimal solutions to the problem in (4).

**Proof.** See Appendix C.

Based on the lemmas above, we obtain the conclusion that Algorithm IWF-CR is convergent. At the same time, step 3) of Algorithm IWF-CR is then regarded as a computation for a point. With these lines of proofs, Algorithm IWF-CR generates a point sequence and every point of the point sequence consists of the \( K \) matrices, e.g. \((S_1^{(n)}, \cdots, S_K^{(n)})\). The details are described below.

**Theorem IV.1.** Algorithm IWF-CR is convergent. At the same time, the sequence of objective values, obtained by evaluating the objective function at the point sequence, monotonically increases to the optimal objective value.

**Proof.** Due to compactness of the set of feasible solutions for the problem in (4), the point sequence generated by Algorithm IWF-CR already includes a convergent subsequence. For every convergent subsequence, according to **Lemma 4.3**, the convergent subsequence must converge to a fixed point under Algorithm IWF-CR. Then, according to **Lemma 4.4**, it converges to one of the optimal solutions to the problem in (4).
In addition, conversely, as stated by the sufficient and necessary conditions of Lemma 4.3 and Lemma 4.4, for any optimal solution to the problem in (4), there is a point sequence generated by Algorithm IWF-CR such that the point sequence converges to that optimal solution.

With Algorithm IWF-CR generating the point sequence, the definition of Algorithm IWF-CR and Eq. (34) in Appendix B imply that the sequence of the objective values, obtained by evaluating the objective function at the point sequence, monotonically increases to the optimal objective value. This is due to Eq. (34) and any convergent subsequence of the point sequence converging to one of the optimal solutions to the optimization problem in (4).

Therefore, Algorithm IWF-CR is convergent.

To reduce the cost of computation, Eq. (25) in Section III may also utilize the Fibonacci search. In this paper, to improve the performance of the algorithm and reduce the cost of the computation, the objective function in step 3) of IWF-CR can be evaluated at the four points, as an instant of the proposed machinery that was mentioned in Remark 3.3, without loss of any convergence of the proposed IWF-CR. As further improvement, parallel computation can be utilized to accelerate computing the estimate of $\beta^*$. 

V Numerical Results

In this section, numerical examples are provided to illustrate the effectiveness of the proposed algorithms. In the simulation, the numbers of the antennas at base station ($N_r$) and at each mobile station ($N_t$) are set to be 10 and 5 respectively. Channel gain matrices are generated randomly using random $N_r \times N_t$ matrices with each entry drawn independently from the standard Gaussian distribution. $\{P_k\}$ is the set of randomly chosen positive numbers. The sum power constraint is $P_T = 10$ dB.

Fig. 3 shows the sum-rate values as a function of the number of iterations when the number of the user $K$ is 10 for the two proposed algorithms, IWF-CR and PD-CR, represented by the cross marked curves and the circle marked curves respectively. It is shown that at the fifth iteration, sum-rate for both algorithms looks close to the optimal value. When we take a close look at these two algorithms as shown in Fig. 4, through IWF-CR, the objective function values are monotonically increasing; whereas through PD-CR, they are not and also show sub-optimality of PD-CR. Fig. 6 - Fig. 7 are
Figure 3: Sum-rates (Unit: bits) of IWF-CR and PD-CR, as K=10;

the corresponding results for $K = 15$. Differently, when $K = 15$, both algorithms obtain satisfying solutions with a few iterations.

Let $f^*$ be the maximum sum-rate, $f^{(n)}$ the sum-rate at the $n$-th iteration and $|f^{(n)} - f^*|$ the error in the sum-rate. Fig. 5 and Fig. 8 show the corresponding error in the sum-rate versus the number of iterations. Note that using the fixed-point theory from the convergence proof of the proposed algorithm can determine the maximum sum-rate $f^*$ mentioned. Fig. 4, as an example, shows that, the proposed PD-CR under the primal-dual approach cannot achieve the optimal solution and then it is a suboptimal non-feasible algorithm when compared with IWF-CR. Note that for those evaluated sum-rate values in PD-CR over $f^*$, they are not feasible. As shown in Fig. 5 and Fig. 8, IWF-CR converges linearly. The suboptimality feature of PD-CR is also depicted by the error floor shown in these figures.

We can further observe that the convergence rate of the proposed algorithm is not sensitive with the increase of the number of users. For clearly understanding, we define

$$N_{IWF/PD-CR} \triangleq \min\{n| |f^{(j)} - f^*| < \epsilon f^*, \text{ as } j \geq n\}$$
Figure 4: Different levels between performances of IWF-CR and PD-CR, as $K=10$;

Figure 5: Error functions of IWF-CR and PD-CR, as $K=10$;
Figure 6: Sum-rates (Unit: bits) of IWF-CR and PD-CR, as K=15;

Figure 7: Different levels between performances of IWF-CR and PD-CR, as K=15;
where the point \( \{ (j, f^{(j)}) \} \) is generated by the corresponding algorithm and \( \epsilon = 10^{-3} \) without loss of generality. We simulate different selection of \( K \), and list the corresponding number of iterations required for convergence in Table 1. We can observe that in the simulated range, using the proposed algorithm, the required number of iterations for convergence is about 6 For OWF-CR. At the same time, for PD-CR, the required number of iterations is more.

Since PD-CR and IWF-CR algorithms use the same matrix inverse operations, which consist of the most significant part of the computation, to compute \( G_i^{(n)} \) in (23) and \( G \) below (29), both algorithms have similar computational complexity \( O(m^3) \) in each of the iterations (refer to [24]). This is because for an \( m \times m \) square matrix, its inverse needs \( m(m^2 - 1) + m(m - 1)^2 \), i.e., \( O(m^3) \), arithmetic operations; its determinant needs \( \frac{2}{3}m^3 + m \), i.e., \( O(m^3) \), operations (the Cholesky decomposition approach is used for efficiency). Thus, since these operations are used with finite times for each iteration, computational complexities for both PD-CR and IWF-CR are \( O(m^3) \).
Table 1: Convergence rate ($N_t = 5; N_r = 10$)

<table>
<thead>
<tr>
<th>$K$</th>
<th>$\hat{N}_{IWF-CR}$</th>
<th>$\hat{N}_{PD-CR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>50</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>60</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>80</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>100</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
<td>18</td>
</tr>
</tbody>
</table>

VI Conclusion

In this paper, two algorithms (IWF-CR and PD-CR) are proposed based on the iterative water-filling and the primal-dual approaches respectively to solve the sum-rate maximization problem in MIMO-MAC CR networks.

By exploiting the concept of variable weighting factor for covariance update, together with the machinery of distributed and parallel computation, IWF-CR owns fast convergence of the sum-rate maximization computation. Furthermore, a novel GWFA, as a fundamental block of the proposed algorithm, is proposed. Convergence of the proposed algorithm is strictly proved in this paper. It implies an equivalent optimality condition, i.e., a point is one of the optimal solutions to the problem of maximum sum-rate for the MIMO-MAC in the CR network if and only if the point is a fixed point of IWF-CR. This criterion makes simulations of the convergent speed meaningful. In the derivation, for more general problems, the assumption in the block coordinate ascent algorithm [25]: the solution to a block of variables should be unique, could be eliminated to prove convergence of the proposed algorithm.

As a counter-part, we also proposed an algorithm based on the primal dual approach to solve the same problem (PD-CR). In the process to develop the algorithm, another new water-filling was proposed. It extends the objective function, a sum-log function, of conventional water-filling to the combined form of a sum-log function with a linear function. If the coefficient(s) of the linear function part is regressed into zero(s), this extended water-filling is regressed into a conventional water-
filling. As a result, the water-filling used in PD-CR and GWFA in IWF-CR are extensions of the conventional water-filling to handle the more complicated constraints and objective function.

Numerical examples are presented to demonstrate the effectiveness of the proposed algorithms. In the simulated range, the required numbers of iteration for convergence are shown to be fixed around six for IWF-CR and slightly higher for PD-CR.

Appendices

A Proof of Proposition 3.1

If the final set $E$ in Algorithm GWFA is empty, it implies that $\sum_{i=1}^{K} g_i P_i \leq P_1$. Then it is easy to see the optimal solutions $\{s_j\}_{j \in \Lambda_i}$ for any $i$, which only require to satisfy $\sum_{j \in \Lambda_i} s_j \leq P_i$, naturally satisfy the entire sum power constraint. Thus, appending all the groups of the solutions from GWFA, we can obtain the solution to the problem (10).

If it is non-empty, it implies that

$$\frac{1}{g_\sigma(k^*) \left( \frac{1}{a_{k^*}} + s_{k^*} \right)} = \frac{1}{g_\sigma(j) \left( \frac{1}{a_j} + s_j \right)}$$

as $\{j, k^*\} \subset E$ and $s_j > 0$. Let

$$\lambda_E = \frac{1}{g_\sigma(k^*) \left( \frac{1}{a_{k^*}} + s_{k^*} \right)}.$$

According to the definitions of $k^*$ and $s_{k^*}$, for $j \in E$ and $s_j = 0$,

$$\frac{1}{g_\sigma(k^*) \left( \frac{1}{a_{k^*}} + s_{k^*} \right)} > \frac{1}{g_\sigma(j) \left( \frac{1}{a_j} + s_j \right)},$$

let

$$\sigma_j = \frac{1}{g_\sigma(k^*) \left( \frac{1}{a_{k^*}} + s_{k^*} \right)} - \frac{1}{g_\sigma(j) \left( \frac{1}{a_j} + s_j \right)} > 0.$$
and it is easy to see $\overline{\sigma}_E = 0$; for $j \notin E$, let $j \in \Lambda_i$. Then

$$\sum_{j \in \Lambda_i} s_j = P_i, \quad \lambda_{\Lambda_i} = \frac{1}{g_\sigma(k^*(\Lambda_i))\left(\frac{1}{\sigma_{k^*(\Lambda_i)}} + s_{k^*(\Lambda_i)}\right)} = \frac{1}{g_\sigma(j)\left(\frac{1}{\sigma_j} + s_j\right)},$$

as $s_j > 0$. If $s_j = 0$, then

$$\sigma_j = \frac{1}{g_\sigma(k^*(\Lambda_i))\left(\frac{1}{\sigma_{k^*(\Lambda_i)}} + s_{k^*(\Lambda_i)}\right)} - \frac{1}{g_\sigma(j)\left(\frac{1}{\sigma_j} + s_j\right)} > 0$$

and $\overline{\sigma}_{\Lambda_i} = 0$. Similarly, (14)-(16) in the initial utilization can obtain $s_{k^*}$, which can be written as $s_{k^*(E_0)}$. Thus we can obtain

$$\lambda_{E_0} = \frac{1}{g_\sigma(k^*(E_0))\left(\frac{1}{\sigma_{k^*(E_0)}} + s_{k^*(E_0)}\right)},$$

$\sigma_j$ and $\overline{\sigma}(E_0)$. (14)-(16) lead to that $\lambda_{E_0} \leq \lambda_{A_i}$, where $\forall \Lambda_i \cap E = \emptyset$, and $\lambda_E \leq \lambda_{A_{E_0}}$. Hence, we have obtained $\{\lambda_E, \overline{\sigma}_E, \{\sigma_j|j \in E\}\}$, $\{\lambda_{A_i}, \overline{\sigma}_{A_i}, \{\sigma_j|j \in A_i\}\}$ and $\{\lambda_{E_0}, \overline{\sigma}_{E_0}, \{\sigma_j|j \in E_0\}\}$.

Therefore, there exist the Lagrange multipliers $\lambda, \{\sigma_j\}_{j=1}^{K N_i}$ and $\{\overline{\sigma}_i\}_{i=1}^K$, the Lagrange function of which, for the problem (10), is:

$$L(\{s_i\}, \lambda, \{\overline{\sigma}_i\}, \{\sigma_j\}) = \sum_{k=1}^{K N_i} \log (1 + a_k s_k) - \lambda \left(\sum_{j=1}^{K N_i} g_\sigma(j)s_j - P_i\right) - \sum_{i=1}^K \overline{\sigma}_i (\sum_{j \in A_i} s_j - P_i) + \sum_{k=1}^{K N_i} \sigma_k s_k,$$

where $\lambda = \lambda_E$; $\overline{\sigma}_j = \overline{\sigma}_E$, as $j \in E$; $\overline{\sigma}_j = \overline{\sigma}_{A_i} + (\lambda_{A_i} - \lambda_E)g_{j}$, as $j \in A_i$ and $\forall i$; and the other Lagrange multipliers have been assigned above. By observation, they satisfy the KKT conditions. Since the problem (10) is a differentiable convex optimization problem with linear constraints, not only are the KKT conditions mentioned above sufficient, but they are also necessary for optimality. Note that it is easily seen that the constraint qualification of the problem (10) holds. Proposition 3.1 hence is proved.

B Proof of Lemma 4.3

Note that in the following proof, we use the notation $n$ to stand for number of the iterations for convenience.
The necessity is proved first. For the limit $(\overline{Q}_1, \ldots, \overline{Q}_K)$ of any convergent subsequence, there is a convergent subsequence $\left\{(Q^{(n_k)}_1, \ldots, Q^{(n_k)}_K)\right\}_{k=0}^{\infty} \subseteq \left\{(Q^{(n)}_1, \ldots, Q^{(n)}_K)\right\}_{n=0}^{\infty}$ such that

$$\overline{Q} = \lim_{k \to \infty}(Q^{(n_k)}_1, \ldots, Q^{(n_k)}_K),$$

is the point sequence generated by Algorithm IWF-CR.

It is seen that

$$(S^{(n_k+1)}_1, \ldots, S^{(n_k+1)}_K) \in \arg \max_{(S_1, \ldots, S_K) \in V} \sum_{i=1}^{K} f(Q^{(n_k)}_i, \ldots, Q^{(n_k)}_i, S_i, Q^{(n_k)}_{i+1}, \ldots, Q^{(n_k)}_K)$$

from the definition of Algorithm IWF-CR. The definition of Algorithm IWF-CR implies that

$$\sum_{i=1}^{K} f(Q^{(n)}_i, \ldots, Q^{(n)}_i, S^{(n+1)}_i, Q^{(n)}_{i+1}, \ldots, Q^{(n)}_K) \geq \sum_{i=1}^{K} f(Q^{(n)}_i, \ldots, Q^{(n)}_i, S_i, Q^{(n)}_{i+1}, \ldots, Q^{(n)}_K), \quad (32)$$

for any $n$ and $(S_1, \ldots, S_K) \in V$. Replacing $n$ with $n_k$, we obtain:

$$\sum_{i=1}^{K} f(Q^{(n_k)}_i, \ldots, Q^{(n_k)}_i, S^{(n_k+1)}_i, Q^{(n_k)}_{i+1}, \ldots, Q^{(n_k)}_K) \geq \sum_{i=1}^{K} f(Q^{(n_k)}_i, \ldots, Q^{(n_k)}_i, S_i, Q^{(n_k)}_{i+1}, \ldots, Q^{(n_k)}_K). \quad (33)$$

We have the following relationships:

$$f(Q^{(n+1)}_1, \ldots, Q^{(n+1)}_i, S^{(n+1)}_i, Q^{(n+1)}_{i+1}, \ldots, Q^{(n+1)}_K)$$

$$\geq f\left(\frac{K-1}{K}(Q^{(n)}_1, \ldots, Q^{(n)}_K) + \frac{1}{K}(S^{(n+1)}_1, \ldots, S^{(n+1)}_K)\right)$$

$$= f\left(\sum_{i=1}^{K} \frac{1}{K}(Q^{(n)}_1, \ldots, Q^{(n)}_i, S^{(n+1)}_i, Q^{(n)}_{i+1}, \ldots, Q^{(n)}_K)\right)$$

$$\geq \frac{1}{K} \sum_{i=1}^{K} f(Q^{(n)}_1, \ldots, Q^{(n)}_i, S^{(n+1)}_i, Q^{(n)}_{i+1}, \ldots, Q^{(n)}_K)$$

$$\geq \frac{1}{K} \sum_{i=1}^{K} f(Q^{(n)}_1, \ldots, Q^{(n)}_i, Q^{(n)}_{i+1}, \ldots, Q^{(n)}_K)$$

$$= f(Q^{(n)}_1, \ldots, Q^{(n)}_K). \quad (34)$$

Between relationships mentioned above, the first inequality and the first equality hold due to step 3) of Algorithm IWF-CR; the second inequality results from the function $f$ being concave; the third inequality and the second equality are true because of step 2) of Algorithm IWF-CR, i.e., the definition of $(S^{(n+1)}_1, \ldots, S^{(n+1)}_K)$.  

31
Thus, \( f(Q_1^{(n)}, \ldots, Q_K^{(n)}) \) is monotonically increasing with respect to \( n \) increasing, and

\[
f(Q_1^{(n)}, \ldots, Q_K^{(n)}) \leq \frac{1}{K} \sum_{i=1}^{K} f(Q_i^{(n)}, \ldots, Q_{i-1}^{(n)}, S_i^{(n+1)}, Q_{i+1}^{(n)}, \ldots, Q_K^{(n)}) \leq f(Q_1^{(n+1)}, \ldots, Q_K^{(n+1)}).
\]

(35)

From (35), we obtain:

\[
\sum_{i=1}^{K} f(Q_i^{(n_k)} : \ldots, Q_{i-1}^{(n_k)}, S_i^{(n_k+1)}, Q_{i+1}^{(n_k)}, \ldots, Q_K^{(n_k)}) \leq K f(Q_1^{(n_k+1)}, \ldots, Q_K^{(n_k+1)}).
\]

From (33), we acquire:

\[
\sum_{i=1}^{K} f(Q_i^{(n_k)} : \ldots, Q_{i-1}^{(n_k)}, S_i^{(n_k+1)}, Q_{i+1}^{(n_k)}, Q_K^{(n_k)}) \geq \sum_{i=1}^{K} f(Q_i^{(n_k)} : \ldots, Q_{i-1}^{(n_k)}, S_i, Q_{i+1}^{(n_k)}, \ldots, Q_K^{(n_k)}).
\]

Hence, it is true that \( K f(Q_1^{(n_k+1)}, \ldots, Q_K^{(n_k+1)}) \geq \sum_{i=1}^{K} f(Q_i^{(n_k)} : \ldots, Q_{i-1}^{(n_k)}, S_i, Q_{i+1}^{(n_k)}, \ldots, Q_K^{(n_k)}). \) Letting \( k \) approach to the infinity, we may acquire that

\[
\sum_{i=1}^{K} f(Q_i^{(n_k)} : \ldots, Q_{i-1}^{(n_k)}, S_i, Q_{i+1}^{(n_k)}, \ldots, Q_K^{(n_k)}) = K f(Q_1, \ldots, Q_K) \geq \sum_{i=1}^{K} f(Q_i^{(n_k)} : \ldots, Q_{i-1}^{(n_k)}, S_i, Q_{i+1}^{(n_k)}, \ldots, Q_K^{(n_k)}),
\]

where \( \forall (S_1, \ldots, S_K) \in V \). Thus,

\[
(Q_1, \ldots, Q_K) \in \arg\max_{(S_1, \ldots, S_K) \in V} \sum_{i=1}^{K} f(Q_i, \ldots, Q_{i-1}, S_i, Q_{i+1}, \ldots, Q_K).
\]

Note that the set \( \arg\max_{(S_1, \ldots, S_K) \in V} \sum_{i=1}^{K} f(Q_i, \ldots, Q_{i-1}, S_i, Q_{i+1}, \ldots, Q_K) \) is not guaranteed to be a single-point set. However, we may choose \((\overline{Q}_1, \ldots, \overline{Q}_K)\) is an optimal solution to the problem \( \max_{(S_1, \ldots, S_K) \in V} \sum_{i=1}^{K} f(\overline{Q}_1, \ldots, \overline{Q}_{i-1}, S_i, \overline{Q}_{i+1}, \ldots, \overline{Q}_K) \). This corresponds to step 2) of Algorithm IWF-CR. Further, \((\overline{Q}_1, \ldots, \overline{Q}_K) = \beta^*(\overline{Q}_1, \ldots, \overline{Q}_K) + (1 - \beta^*)(\overline{Q}_1, \ldots, \overline{Q}_K)\), based on the choice of the optimal solution mentioned above. This corresponds to step 3) of Algorithm IWF-CR.

Therefore, resulting from the two correspondences mentioned above and the definition of Algorithm IWF-CR, it is true that \((\overline{Q}_1, \ldots, \overline{Q}_K)\) is a fixed point under Algorithm IWF-CR, which is viewed as a mapping.

The sufficiency will be proved as follows:
If \((\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\) is a fixed point under Algorithm **IWF-CR**, it is seen that
if \((\mathbf{Q}_1(0), \cdots, \mathbf{Q}_K(0))\) is denoted by \((\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\), then \((\mathbf{Q}_1(1), \cdots, \mathbf{Q}_K(1)) = (\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\), i.e., the former is assigned by the latter, due to \((\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\) being a fixed point under Algorithm **IWF-CR**. If it is assumed that \((\mathbf{Q}_1(n), \cdots, \mathbf{Q}_K(n)) = (\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\), then \((\mathbf{Q}_1(n+1), \cdots, \mathbf{Q}_K(n+1)) = (\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\) due to \((\mathbf{Q}_1, \cdots, \mathbf{Q}_K)\) being a fixed point under Algorithm **IWF-CR**. According to the principle of mathematical induction, \((\mathbf{Q}_1(n), \cdots, \mathbf{Q}_K(n)) = (\mathbf{Q}_1, \cdots, \mathbf{Q}_K) \in \mathcal{V}, \forall n\). Furthermore, \(\lim_{n \to \infty} (\mathbf{Q}_1(n), \cdots, \mathbf{Q}_K(n)) = (\mathbf{Q}_1, \cdots, \mathbf{Q}_K) \in \mathcal{V}\). Therefore, the sufficiency is true.

Note that in the proving process above, we do not have the following assumption:

\[(\mathbf{Q}_1, \cdots, \mathbf{Q}_K) = \lim_{k \to \infty} (\mathbf{Q}_1^{(n_k+1)}, \cdots, \mathbf{Q}_K^{(n_k+1)}).\]

### C  Proof of Lemma 4.4

The necessity is proved first.

According to the correspondences between (4) and its real form mentioned in **Lemma 4.2**, it is easily known, for the fixed point \((\mathbf{Q}_1, \cdots, \mathbf{Q}_K) \in \mathcal{V}\), that

\[\mathbf{x}(1, \cdots, \mathbf{x}_K) \in \max_{(x_1, \cdots, x_K) \in \mathcal{V}_r} \sum_{i=1}^{K} f_r(\mathbf{x}_1, \cdots, x_i-1, x_i, x_i+1, \cdots, \mathbf{x}_K), \text{ where } (\mathbf{x}_1, \cdots, \mathbf{x}_K) \in \mathcal{V}_r.\]

(36)

Since (36) is a convex optimization problem with a concave objective function, noting the optimality condition, which is necessary and sufficient for (36), of the convex optimization problems (refer to Proposition 3.1 in [21]), formula (36) implies that

\[(f_{rx_1}(\mathbf{x}_1, \cdots, \mathbf{x}_K), \cdots, f_{rx_K}(\mathbf{x}_1, \cdots, \mathbf{x}_K)) \cdot ((x_1 - \mathbf{x}_1)^T, \cdots, (x_K - \mathbf{x}_K)^T)^T \leq 0, \quad (37)\]

where, \(\forall (x_1, x_2, \cdots, x_K) \in \mathcal{V}_r\), we denote a transpose of the gradient with respect to the variables \(x_i\) of \(f_r\) by the row vector \(f_{rx_i}\).

It is seen that formula (37) is just the optimal condition of the real form of the optimization problem (4). Due to the real form itself being a convex optimization problem with a concave objective func-

33
tion, \((\overline{x}_1, \ldots, \overline{x}_K)\) is an optimal solution to the real form, as an optimization problem. Furthermore, according to the correspondence between (4) and the real form, the fixed point \((\overline{Q}_1, \ldots, \overline{Q}_K) \in V\) is an optimal solution to the problem in (4).

The sufficiency will be proved as follows:

\[
\sum_{i=1}^{K} f(\overline{Q}_1, \ldots, \overline{Q}_{i-1}, S_i, \overline{Q}_{i+1}, \ldots, \overline{Q}_K) = K \sum_{i=1}^{K} \frac{1}{K} f(\overline{Q}_1, \ldots, \overline{Q}_{i-1}, S_i, \overline{Q}_{i+1}, \ldots, \overline{Q}_K) \leq K f(\frac{1}{K} (S_1, \ldots, S_K) + \frac{K-1}{K} (\overline{Q}_1, \ldots, \overline{Q}_K)) \leq K f(\overline{Q}_1, \ldots, \overline{Q}_K) = \sum_{i=1}^{K} f(\overline{Q}_1, \ldots, \overline{Q}_K).
\]

Between relationships mentioned above, due to \((S_1, \ldots, S_K) \in V\), the first equality holds; because the function \(f\) is concave and the set of feasible solutions \(V\) is convex, the first inequality holds; since \((\overline{Q}_1, \ldots, \overline{Q}_K) \in V\) is the optimal solution to the problem in (4), the second inequality is true. Hence, \(\sum_{i=1}^{K} f(\overline{Q}_1, \ldots, \overline{Q}_{i-1}, S_i, \overline{Q}_{i+1}, \ldots, \overline{Q}_K) \leq \sum_{i=1}^{K} f(\overline{Q}_1, \ldots, \overline{Q}_K), \forall (S_1, \ldots, S_K) \in V\).

According to definition of the optimal solution to (31) mentioned above,

\[(\overline{Q}_1, \ldots, \overline{Q}_K) \in \text{arg max}_{(S_1, \ldots, S_K) \in V} \sum_{i=1}^{K} f(\overline{Q}_1, \ldots, \overline{Q}_{i-1}, S_i, \overline{Q}_{i+1}, \ldots, \overline{Q}_K).\]

According to steps 2) and 3) of Algorithm IWF-CR, \((\overline{Q}_1, \ldots, \overline{Q}_K) \in V\) is a fixed point under Algorithm IWF-CR.

**Acknowledgement**

The authors sincerely acknowledge the support from Natural Sciences and Engineering Research Council (NSERC) of Canada under grant number RGPIN/293237-2009.

**References**


