An Efficient Direct Solution of Cave-Filling Problems

Kalpana Naidu, *Student Member, IEEE*, Mohammed Zafar Ali Khan, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

AQ:1 ***Abstract*— Waterfilling problems subjected to peak power constraints are solved, which are known as cave-filling problems (CFP). The proposed algorithm finds both the optimum number of positive powers and the number of resources that are assigned the peak power before finding the specific powers to be assigned. The proposed solution is non-iterative and results in a computational complexity, which is of the order of M, $O(M)$, where M is the total number of resources, which is significantly lower than that of the existing algorithms given by an order of M^2, $O(M^2)$, under the same memory requirement and sorted parameters. The algorithm is then generalized both to weighted CFP (WCFP) and WCFP requiring the minimum power. These extensions also result in a computational complexity of the order of M, $O(M)$. Finally, simulation results corroborating the analysis are presented.**

***Index Terms*— Weighted waterfilling problem, Peak power constraint, less number of flops, sum-power constraint, cave waterfilling.**

I. Introduction

WATERFILLING Problems (WFP) are encountered in numerous communication systems, wherein specifically selected powers are allotted to the resources of the transmitter by maximizing the throughput under a total sum power constraint. Explicitly, the classic WFP can be visualized as filling a water tank with water, where the bottom of the tank has stairs whose levels are proportional to the channel quality, as exemplified by the Signal-to-Interference Ratio (SIR) of the Orthogonal Frequency Division Multiplexing (OFDM) sub-carriers [1], [2].

This paper deals with the WaterFilling Problem under Peak Power Constraints (WFPPPC) for the individual resources. In contrast to the classic WFP where the 'tank' has a 'flat lid', in WFPPPC the 'tank' has a 'staircase shaped lid', where the steps are proportional to the individual peak power constraint. This scenario is also metaphorically associated with a 'cave' where the stair-case shaped ceiling represents the peak power that can be assigned, thus fulfilling all the requirements of WFPPPC. Thus WFPPPC is often referred to as a 'Cave-Filling Problem' (CFP) [3], [4].

In what follows, we will use the 'cave-filling' metaphor to develop insights for solving the WFPPPC. Again, the user's resources can be the sub-carriers in OFDM or the tones in a Digital Subscriber Loop (DSL) system, or alternatively the same sub-carriers of distinct time slots [5].

More broadly, the CFP occurs in various disciplines of communication theory. A few instances of these are:

a) protecting the primary user (PU) in Cognitive Radio (CR) networks [6]–[9];
b) when reducing the Peak-to-Average-Power Ratio (PAPR) in Multi-Input-Multi-Output (MIMO)-OFDM systems [10], [11];
c) when limiting the crosstalk in Discrete Multi-Tone (DMT) based DSL systems [12]–[14];
d) in energy harvesting aided sensors; and
e) when reducing the interference imposed on nearby sensor nodes [15]–[17].

Hence the efficient solution of CFP has received some attention in the literature, which can be classified into iterative and exact direct computation based algorithms.

Iterative algorithms conceived for CFP have been considered in [18]–[20], which may exhibit poor accuracy, unless the initial values are carefully selected. Furthermore, they may require an extremely high number of iterations for their accurate convergence.

Exact direct computation based algorithms like the Fast WaterFilling (FWF) algorithm of [21], the Geometric WaterFilling with Peak Power (GWFPP) constraint based algorithm of [22] and the Cave-Filling Algorithm (CFA) obtained by minimizing Minimum Mean-Square Error (MMSE) of channel estimation in [3] solve CFPs within limited number of steps, but impose a complexity on the order of $O(M^2)$.

All the existing algorithms solve the CFPs by evaluating the required powers multiple times, whereas the proposed algorithm directly finds the required powers in a single step. Explicitly, the proposed algorithm reduces the number of Floating point operations (flops) by first finding the number of positive powers to be assigned, namely K, and the number of powers set to the maximum possible value, which is denoted by L. This is achieved in two (waterfilling) steps. First we use 'coarse' waterfilling to find the number of positive powers to

Manuscript received November 30, 2015; revised March 14, 2016; accepted
AQ:2 April 24, 2016. This work was supported in part by the Engineering and Physical Sciences Research Council EP/Noo4558/1 and EP/L018659/1, in part by the European Research Council advanced fellow grant under Beam-me-up, and in part by the Royal Society under Wolfson research merit award. The associate editor coordinating the review of this paper and approving it for publication was M. Tao.

K. Naidu and M. Z. Ali Khan are with the Department of Electrical Engineering, IIT Hyderabad, Hyderabad 502205, India (e-mail: ee10p002@iith.ac.in; zafar@iith.ac.in).

L. Hanzo is with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: lh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2016.2560813

be assigned and then we embark on step-by-step waterfilling to find the number of positive powers that have to be set to the affordable peak powers.

In this paper we present an algorithm designed for the efficient solution of CFPs. The proposed solution is then generalized for **conceiving** both a Weighted CFP (WCFP) and a WCFP having both a Minimum and a Maximum Power (WCFP-MMP) constraint. It is demonstrated that the maximum throughput is achieved at a complexity order of $O(M)$ by all the three algorithms proposed.

The outline of the paper is as follows. Section II outlines our system model and develops the algorithms for solving the CFP. In Section III we conceive the WCFP, while Section IV presents our WCFP-MMP. Our simulation results are provided in Section V, while Section VI concludes the paper.

II. The Cave-Filling Problem

In Subsection II-A, we introduce the CFP. The computation of the number of positive powers is presented in Subsection II-B, while that of the number of powers set to the maximum is presented in Subsection II-C. Finally, the computational complexity is evaluated in Subsection II-D.

A. The CFP

The CFP maximizes the attainable throughput, C, while satisfying the sum power constraint; Hence, the sum of powers allocated is within the prescribed power budget, P_t, while the power, $P_i, \forall i$ assigned for the i^{th} resource is less than the peak power, P_{it}, $\forall i$. Our optimization problem is then formulated as:

$$\begin{aligned} \max_{\{P_i\}_{i=1}^{M}} \; & C = \sum_{i=1}^{M} \log_2\left(1 + \frac{P_i}{N_i}\right) \\ \text{subject to}: \; & \sum_{i=1}^{M} P_i \leq P_t; \\ & P_i \leq P_{it}, \quad i \leq M, \\ & \text{and } P_i \geq 0, \quad i \leq M, \end{aligned} \tag{1}$$

where M is the total number of resources (such as OFDM sub-carriers) and $\{N_i\}_{i=1}^{M}$ is the sequence of interference plus noise samples. The above optimization problem occurs in the following scenarios:

(a) In the downlink of a wireless communication system, where the base station (BS) assigns a resource (e.g. frequency band) to a user and allocates a certain power, P_i, to the i^{th} resource while obeying the total power budget (P_t). The BS ensures that $P_i \leq P_{it}$ for avoiding the near-far problem [23].

(b) In an OFDM system, a transmitter assigns specific powers to the resources (e.g. sub-carriers) for satisfying the total power budget, P_t. Furthermore, to reduce the PAPR problem, the maximum powers assigned are limited to be within the peak powers [24], [25].

Theorem 1: The solution of the CFP (1) *is of the 'form'*

$$P_i = \begin{cases} \left(\frac{1}{\lambda} - N_i\right), & 0 < P_i < P_{it}; \\ P_{it}, & \frac{1}{\lambda} \geq H_i \triangleq (P_{it} + N_i); \\ 0, & \frac{1}{\lambda} \leq N_i \end{cases} \tag{2}$$

where "$\frac{1}{\lambda}$ is the water level of the CFP".

Proof: The proof is in Appendix VI-A. □

Remark 1: Note that as in the case of conventional water-filling, the solution of CFP is of the form (2). *The actual solution is obtained by solving the solution form along with the primal feasibility constraints. Furthermore, for the set of primal feasibility constraints of our CFP, the Peak Power Constraint of $P_i \leq P_{it}, \forall i$ is incorporated in the solution form. By contrast, the sum power constraint is considered along with* (2) *to obtain the solution in Propositions 1 and 2.*

Remark 2: Observe from (2) *that for $0 < P_i < P_{it}$, $P_i = (\frac{1}{\lambda} - N_i)$ which allows $\frac{1}{\lambda}$ to be interpreted as the 'water level'. However, in contrast to conventional water-filling, the 'water level' is upper bounded by* $\max_i P_{it}$. *Beyond this value, no 'extra' power can be allocated and the 'water level' cannot increase. The solution of this case is considered in Proposition 1.*

It follows that (2) *has a nice physical interpretation, namely that if the 'water level' is below the noise level N_i, no power is allocated. When the 'water level' is between N_i and P_{it}, the difference of the 'water level' and the noise level is allocated. Finally, when the 'water level' is higher than the 'peak level', H_i; the peak power P_{it} is allocated.*

The above solution 'form' can be rewritten as

$$P_i = \left(\frac{1}{\lambda} - N_i\right)^{+}, \quad i = 1, \cdots, M; \quad \textit{and} \tag{3}$$

$$P_i \leq P_{it}, \quad i = 1, \cdots, M \tag{4}$$

where we have $A^{+} \triangleq \max(A, 0)$. The solution for (1) has a simple form for the case the 'implied' power budget, P_{It} as defined as $P_{It} = \sum_{i=1}^{M} P_{it}$ is less than or equal to P_t and is given in Proposition 1.

Proposition 1: If the 'implied' power budget is less than or equal to the power budget ($\sum_{i=1}^{M} P_{it} \leq P_t$), then peak power allocation to all the M resources gives optimal capacity.

Proof: Taking summation on both sides of $P_i \leq P_{it}, \forall i$, we obtain the 'implied' power constraint

$$\sum_{i=1}^{M} P_i \leq \underbrace{\sum_{i=1}^{M} P_{it}}_{P_{IT}}. \tag{5}$$

However from (1) we have

$$\sum_{i=1}^{M} P_i \leq P_t. \tag{6}$$

Consequently, if $P_{It} \leq P_t$, then peak power allocation to all the M resources (i.e. $P_i = P_{it}$, $\forall i$) fulfils all the constraints of (1). Consequently, the total power allocated to M resources $\sum_{i=1}^{M} P_{it}$. Since the maximum power that can be allocated to any resource is it's peak power, peak power allocation to all the M resources produces optimal capacity. □

Note that in this case the total power allocated is less than (or equal to) P_t. However, if $P_t < \sum_{i=1}^{M} P_{it}$, then all the M resources cannot be allocated peak powers since it violates the total sum power constraint in (1).

In what follows, we pursue the solution of (1) for the case

$$P_t < \sum_{i=1}^{M} P_{it}. \tag{7}$$

We have,

Proposition 2: The optimal powers and hence optimal capacities are achieved in (1) *(under the assumption* (7)*) only if*

$$\sum_{i=1}^{M} P_i = P_t. \tag{8}$$

Proof: The proof is in Appendix VI-B. □

Since finding both the number of positive powers and the number of powers that are set to the maximum is crucial for solving the CFP, we formally introduce the following definitions.

Definition 1 (The Number of Positive Powers, K): Let $\mathcal{I} = \{i; \text{such that } P_i > 0\}$ be the set of resource indices, where P_i is positive. Then the number of positive powers, $K = |\mathcal{I}|$, is given by the cardinality, $|\mathcal{I}|$, of the set.

Definition 2 (The Number of Powers Set to the Peak Power, L): Let $\mathcal{I}_{\mathcal{P}} = \{i; \text{such that } P_i = P_{it}\}$ be the set of resource indices, where P_i has the maximum affordable value of P_{it}. Then the number of powers set to the peak power, $L = |\mathcal{I}_{\mathcal{P}}|$, is the cardinality, $|\mathcal{I}_{\mathcal{P}}|$ of the set.

Without loss of generality, we assume that the interference plus noise samples N_i are sorted in ascending order, so that the first K powers are positive, while the remaining ones are set to zero. Then, (8) becomes

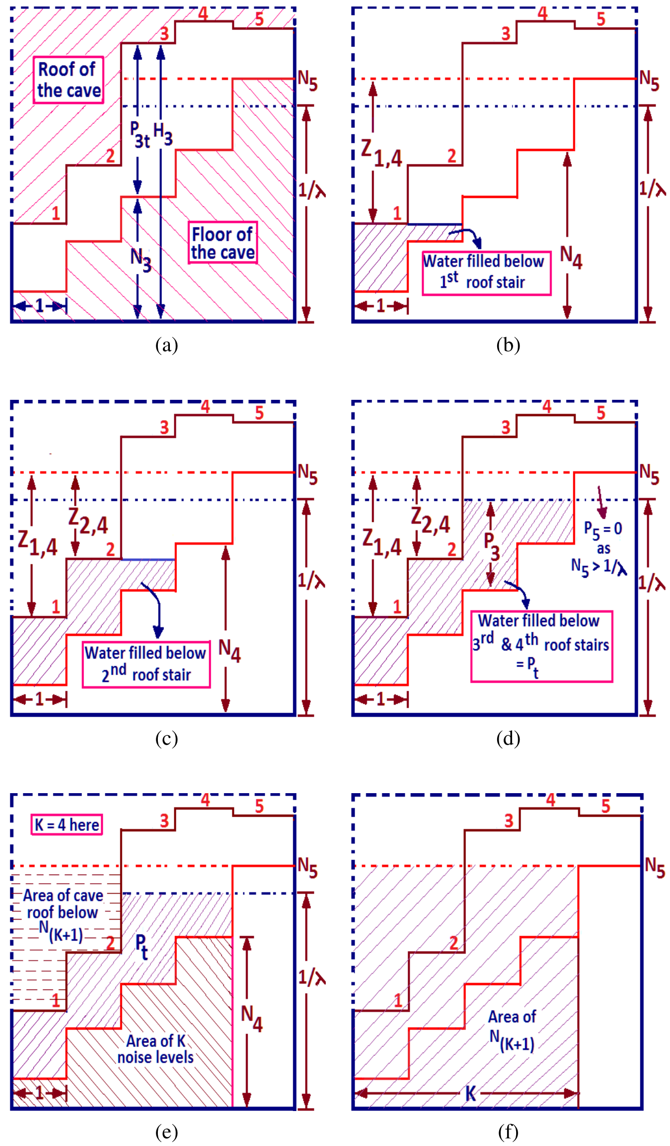
$$\sum_{i=1}^{K} P_i = P_t. \tag{9}$$

Note that H_i and P_{it} are also arranged in the ascending order of N_i, in order to preserve the original relationship between H_i and N_i.

B. Computation of the Number of Positive Powers

The CFP can be visualized as shown in Fig. 1a. In a cave, the water is filled i.e. the power is apportioned between the floor of the cave and the ceiling of the cave. The levels of the i^{th} 'stair' of the floor staircase and of the ceiling staircase are N_i and $H_i \triangleq (P_{it} + N_i)$, respectively. The widths of all stairs are assumed to be 1. Since the power gap between the floor stair and the ceiling stair is P_{it}, the allocated power has to satisfy $P_i \leq P_{it}$.

As the water is poured into the cave, observe from Fig. 1b that it obeys the classic waterfilling upto the point where the 'waterlevel' ($\frac{1}{\lambda}$) reaches the ceiling stair of the 1^{st} resource. From this point onwards, water can only be stored above the second stair, as depicted in Fig. 1c upto a point where



AQ:3

Fig. 1. Geometric Interpretation of CFP for $K = 4$. (a) Heights of i^{th} stair in cave floor staircase and cave roof staircase are N_i and $H_i(= P_{it} + N_i)$. (b) Water is filled (Power is allotted) in between the cave roof stair and cave floor stair, at this waterlevel the peak power constraint for P_1 constraints further allocation to P_1. (c) A similar issue occurs to P_2 also.Observe that the variable $Z_{m,4}$ represents the height of m^{th} cave roof stair below the $(4+1)^{th}$ cave floor stair. (d) Power allotted for i^{th} resource is $P_i = min\{\frac{1}{\lambda}, H_i\} - N_i$. Observe the waterlevel between 4^{th} and 5^{th} resource. (e) The area $\frac{1}{\lambda}K$, shown in this figure, is smaller than the area $N_{K+1}K$ shown in (f).

the water has filled the gap between the floor stair and the ceiling stair of both the first and the second stairs. In terms of power, we have $P_i = P_{it}$ for the resources $i = 1$ and 2. Mathematically, we have $P_i = P_{it}$ if $H_i \leq \frac{1}{\lambda}$.

As more water is poured, observe from Fig. 1d that for the third and the fourth stairs, we have $H_i > \frac{1}{\lambda}$. It is clear from the above observations (also from (2)) that the power assigned to the i^{th} resource becomes:

$$P_i = min\left\{\frac{1}{\lambda}, H_i\right\} - N_i, \quad i \leq K. \tag{10}$$

In Fig. 1d, the height of the fifth floor stair exceeds $\frac{1}{\lambda}$. As water can only be filled below the level $\frac{1}{\lambda}$, no water is

Algorithm 1 ACF Algorithm for Obtaining K

Require: Inputs required are M, P_t, N_i & H_i (in ascending order of N_i).

Ensure: Output is K, $I_{R_{K-1}}$,I_{R_K}, d_K.

1: $i = 1$. Denote $d_0 = P_t$, $U_0 = 0$ and $I_{R_0} = \emptyset$
2: Calculate $d_i = d_{i-1} + N_i$.
3: $\triangleright$ Calculate the area $U_i = \sum_{m=1}^{i} Z_{m,i}^{+}$ as follows:
4: $I_{R_i} = I_{R_{i-1}} \cup \{m;$ *such that* $N_{i+1} > H_m$ & $m \not\in I_{R_{i-1}}\}$;
$Z_{m,i} = N_{(i+1)} - H_m, m \in (I_{R_i} - I_{R_{i-1}})$
5: $U_i = U_{i-1} + |I_{R_{i-1}}|(N_{i+1} - N_i) + \sum_{m\in(I_{R_i} - I_{R_{i-1}})} Z_{m,i}^{+}$
6: Calculate the area $Q_i = i\,N_{(i+1)}$
7: **if** $Q_i \geq (d_i + U_i)$ **then**
8: $K \leftarrow i$. Exit the algorithm.
9: **else**
10: $i \leftarrow$ i+1, Go to 2
11: **end if**

Algorithm 2 'Step-Based' Waterfilling Algorithm for Obtaining L

Require: Inputs required are K, d_K, $I_{R_{K-1}}$, I_{R_K}, N_i & H_i (in ascending order of N_i)

Ensure: Output is L, I_S.

1: Calculate $P_R = d_K - KN_K + |I_{R_{K-1}}|N_K - \sum_{m\in I_{R_{K-1}}} H_m$
2: Calculate $I_B = I_{R_K} - I_{R_{K-1}}$ & $D_1 = K - |I_{R_{K-1}}|$.
3: If $|I_B| = 0$, set $L = 0$, $I_S = \emptyset$. Exit the algorithm.
4: Sort $\{H_m\}_{m\in I_B}$ in ascending order and denote it as $\{H_{mB}\}$ and the sorting index as I_S.
5: Initialize $m = 1$, $F_m = (H_{mB} - N_K)D_m$.
6: **while** $F_m < P_R$ **do**
7: $m = m + 1$.
8: $D_m = D_{m-1} - 1$
9: $F_m = F_{m-1} + (H_{mB} - H_{(m-1)B})D_m$
10: **end while**
11: $L = m - 1$.

filled above the fifth bottom stair. This results in $K = 4$, as shown in Fig. 1d. The area of the water-filled cave cross-section becomes equal to P_t.

Fig. 1c also introduces the variable $Z_{i,k}$ as the depth of the i^{th} ceiling stair below the $(k+1)^{st}$ bottom stair; that is, we have:

$$Z_{i,k} = N_{(k+1)} - H_i, \quad i \leq k. \tag{11}$$

The variable $Z_{i,k}$ allows us to have a reference, namely a constant roof ceiling of N_{i+1}, while verifying whether $K = i$. Figure 1c depicts this dynamic for $i = 4$. The constant roof reference is given at N_{i+1}. Observe that we have $Z_{i,k}^{+} > 0$ for $i = 1, 2$ and $Z_{i,k}^{+} = 0$ for $i = 3, 4$ with $k = 4$. This allows us to quantify the total cave cross-section area in Fig 1e, upto the i^{th} step in three parts:

- the area occupied by roof stairs below the constant roof reference, given by $\sum_{k=1}^{i} Z_{k,i}^{+}$;
- the area occupied by the 'water', given by P_t;
- the area occupied by the floor stairs, $\sum_{k=1}^{i} N_k$.

This is depicted in Fig. 1e. Observe from Fig. 1e that if the waterlevel of $\frac{1}{\lambda}$ is less than the $(i+1)^{st}$ water level ($i + 1 = 5$ in this case), then the cave cross-section area given by $\sum_{k=1}^{i} Z_{k,i}^{+} + P_t + \sum_{k=1}^{i} N_k$ (shown in Fig. 1e) would be less than the total area of $i\,N_{i+1}$, as shown in Fig. 1f. Furthermore, if the waterlevel $\frac{1}{\lambda}$ is higher than the $(i+1)^{st}$ water level ($i + 1 = 2, 3, 4$ in this case), then the area given by $\sum_{k=1}^{i} Z_{k,i}^{+} + P_t + \sum_{k=1}^{i} N_k$ would be higher than the total area of $i\,N_{i+1}$, as shown in Fig. 1f.

Based on the insight gained from the above geometric interpretation of the CFP, we develop an algorithm for finding K for any arbitrary CFP, which we refer to as the **Area based Cave-Filling (ACF)** of Algorithm 1.

Note that d_0 in Algorithm 1 represents an initialization step that eliminates the need for the addition of P_t at every resource-index i and the set I_{R_i} contains the indices of the ceiling steps, whose 'height' is below N_{i+1}. Furthermore, the additional outputs of Algorithm 1 are required for finding the number of roof stairs that are below the waterlevel in Algorithm 2. We now prove that Algorithm 1 indeed finds the optimal value of K.

Theorem 2: The Algorithm 1 delivers the optimal value of the number of positive powers, K, as defined in Definition 1.

Proof: We prove Theorem 2 by first proving that $\phi(i) = d_i + U_i$, is a monotonically increasing function of the resource-index i. It then follows that $Q_i \geq (d_i + U_i)$ gives the first i, for which the waterlevel is below the next step. Consider

$$\begin{aligned}
&\phi(i) - \phi(i-1) \\
&\quad = d_i - d_{i-1} + U_i - U_{i-1} && (12) \\
&\quad = N_i + |I_{R_{i-1}}|\,(N_{i+1} - N_i) + \sum_{m\in(I_{R_i} - I_{R_{i-1}})}^{i} Z_{m,i}^{+} && (13) \\
&\quad > 0, && (14)
\end{aligned}$$

where (13) follows from (12) by using the definitions of d_i and U_i in Algorithm 1. Since the interference plus noise levels N_i are positive, we have $(N_{i+1} - N_i) \geq 0$, and since the N_i's are in ascending order, (14) follows from (13).

Let us now consider the reference area, $Q_i = i\,N_{i+1}$. Within this reference area; certain parts are occupied by the floor stairs, others by the projections of the ceiling stairs and finally by the space in between the floor and the ceiling; filled by 'water'. This is given by $W_i = Q_i - \sum_m^i N_m - U_i$. Recall that the total amount of water that can be stored is P_t. If we have $P_t > W_i$, then there is more water than the space available, hence the water will overflow to the next stair(s). Otherwise, if we have $P_t \leq W_i$, all the water can be contained within the space above this stair and the lower stairs. Substituting the value of W_i in this inequality, we have

$$P_t \leq Q_i - \sum_m^i N_m - U_i \tag{15}$$

$$\Rightarrow P_t + \sum_m^i N_m + U_i \leq Q_i \tag{16}$$

$$d_i + U_i \leq Q_i \tag{17}$$

where (16) is obtained from (15) by rearranging. Then using the definition of d_i in Algorithm 1, we arrive at (17).

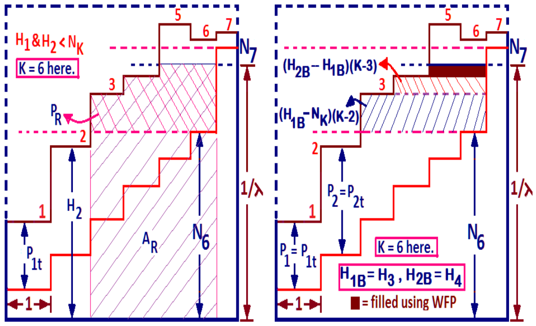


Fig. 2. Peak power allocation for resources having their H_i's in between N_K and $N_{(K+1)}$.

Since Algorithm 1 outputs the (first) smallest value of the resource-index i for which (17) is satisfied, it represents the optimal value of K.

This completes the proof. □

Once K is obtained, it might appear straightforward to obtain the values of P_i, $i \in [1, K]^{\ddagger}$ as in [26] and [27]; but in reality it is not. This is because of the need to find the specific part of the cave roof, which is below the 'current' waterlevel.

Note that $I_{R_{K-1}} \subset I_P \subset I_{R_K}$ where I_P is the set of roof stairs below the current waterlevel and I_{R_K} is the set of roof stairs below N_{K+1}. This is because the waterlevel of $\frac{1}{\lambda}$ is between N_K and N_{K+1}.

C. Waterfilling for Finding the Number of Powers Having the Peak Allocation

In order to develop an algorithm for finding L, we first consider the geometric interpretation of an example shown in Fig. 2. Note that the H_m's below N_K, $(N_K - H_m) > 0$, belong to $I_{R_{K-1}}$ and the H_m values above N_{K+1} belong to I_{U_K}. This is clearly depicted in Fig. 2 for $K = 6$, where $I_{R_{K-1}} = \{1, 2\}$ and $I_{U_K} = \{5, 6\}$.

The contentious H_m's are those whose heights lie between N_K and N_{K+1}. The indices of these H_m's are denoted by I_B (in Fig. 2, $I_B = \{3, 4\}$). Without loss of generality, we assume that B roof stairs, H_m's, lie between N_K and N_{K+1}. We now have to find among these B stairs, those particular ones whose heights lie below the water level, $\frac{1}{\lambda}$ (for which peak powers are allotted). Note that $B = |I_{R_K}| - |I_{R_{K-1}}|$ and $I_B = [1, K] - I_{R_{K-1}} - I_{U_K} = I_{R_K} - I_{R_{K-1}}$.

This is achieved by a 'second' waterfilling style technique as detailed below.

Clearly, the resources that belong to the set $I_{R_{K-1}}$ are allotted with peak powers as $(H_m - \frac{1}{\lambda}) < 0, m \in I_{R_{K-1}}$. The remaining ceiling stairs in I_B will submerge one by one as the waterlevel increases from N_K. For this reason; the heights $\{H_m\}_{m \in I_B}$ are sorted in ascending order to obtain H_{mB} and I_S is the sort index for H_{mB}.

After allotting $I_{R_{K-1}}$ resources with peak powers, whose sum is equal to $\sum_{m \in I_{R_{K-1}}} P_{mt}$, we can allocate $(N_K - N_m)^+, m \in I^c_{R_{K-1}}$ power to the remaining resources indexed by $I^c_{R_{K-1}}$, where for a set A, $A^c = [1, K] - A$ represents its complement. That is we allot power to remaining resources with the 'present' waterlevel being N_K. The power that remains to be allocated for $I^c_{R_{K-1}}$ resources is given by

$$P_R = P_t - \sum_{m \in I_{R_{K-1}}} P_{mt} - \sum_{m \in I^c_{R_{K-1}}} (N_K - N_m)^+ \qquad (18)$$

$$= P_t + \sum_{m=1}^{K} N_m - K N_K + |I_{R_{K-1}}| N_K - \sum_{m \in I_{R_{K-1}}} H_m. \qquad (19)$$

Equation (19) is obtained using a geometric interpretation as follows; the term $d_K = P_t + \sum_{m=1}^{K} N_m$ is the sum of total water and K floor stairs. Subtracting from it the reference area of $K N_K$ gives the excess water that is in excess amount; without considering the ceiling stairs. Further subtracting the specific part of the ceiling stairs that are below N_K namely $\sum_{m \in I_{R_{K-1}}} H_m - |I_{R_{K-1}}| N_K$ gives the residual 'water' amount, P_R.

Note from Fig. 2 that once P_R amount of 'water' has been poured, and provided that $P_R < (K - |I_{R_{K-1}}|)(H_{1B} - N_K)$ is satisfied, then we have $L = |I_{R_{K-1}}|$ and hence no more 'water' is left to be poured. Otherwise, $F_1 = (K - |I_{R_{K-1}}|)$ $(H_{1B} - N_K)$ amount of 'water' is used for completely submerging the 1^{st} ceiling stair (H_{1B}) and the 'present' waterlevel increases to H_{1B}. Similarly, $F_2 = (K - |I_{R_{K-1}}| - 1)$ $(H_{2B} - H_{1B})$ amount of water is used for submerging the second ceiling stair and hence the waterlevel increases to H_{2B}. This process continues until all the 'water' has been poured. We refer to this process as 'step-based' waterfilling since the waterlevel is changed in steps given by the size of the roof stairs.

The formal algorithm, which follows the above geometric interpretation but it aims for a low complexity, is given in Algorithm 2. Let us now prove that Algorithm 2 delivers the optimal value of L.

Theorem 3: Algorithm 2 finds the optimal value L of the number of powers that are assigned peak powers, where L is defined in Definition 2 .

Proof: First observe that the F_m values are monotonically increasing functions of the index m. Since the H_{mB} values are sorted in ascending order, the water filling commences from $m = 1$. The condition $F_m < P_R$ is true, as long as the total amount of water required to submerge the m^{th} roof stair, F_m, is less than the available water. It follows then that the algorithm outputs the largest m, for which the inequality is satisfied which hence represents the optimal value of L. □

The resources for which peak powers are allotted are indexed by $I_P = I_{R_{K-1}} \cup I_S(1 : L)$, where $I_S(1 : L)$ stands for the first 'L' resources of I_S. The remaining resources, indexed by $I^c_P = [1, K] - I_P$, are allotted specific powers using waterfilling.

In Fig. 2, the I^c_P resources are 5 and 6 with associated 'L' $= 2$ while $P_R - F_L$ represents the darkened area in Fig. 2. The waterlevel for I^c_P resources is equal to the height, H_{LB}, of the last submerged roof stair plus the height of the darkened area. Here, the height of the darkened area is obtained by dividing the remaining water amount $(= P_R - F_L)$ with the

‡[A,B] represents the interval in between A and B, including A and B.

TABLE I
COMPUTATIONAL COMPLEXITIES (IN FLOPS) OF KNOWN SOLUTIONS FOR SOLVING CFP

Iterative Algorithms [18], [19]	**FWF [21]**	**GWFPP [22]**	**ACF**
iterations $\times(6M)$	iterations $\times(5M+6)$	$4M^2+7M$	*16M+9*

number of remaining resources ($= |I_P^c|$) since the width of all resources is 1. If we have $L = 0$, then the last level is N_K. Therefore the waterlevel for I_P^c resources is given by

$$\frac{1}{\lambda} = \begin{cases} H_{LB} + \dfrac{P_R - F_L}{|I_P^c|}, & L > 0; \\ N_K + \dfrac{P_R}{|I_P^c|}, & \text{otherwise.} \end{cases} \tag{20}$$

The powers are then allotted as follows:

$$P_i = \begin{cases} P_{it}, & i \in I_P; \\ \left(\dfrac{1}{\lambda} - N_i\right), & i \in I_P^c. \end{cases} \tag{21}$$

D. Computational Complexity of the CFP

Let us now calculate the computational complexity of both Algorithm 1 as well as of Algorithm 2 separately and then add the complexity of calculating the powers, as follows:

- Calculating H_i requires M adds.
- Observe that Algorithm 1 requires $K+1$ adds for calculating d_i's; K multiplies to find Q_i's; *maximum of K subtractions for calculating $Z_{m,i}$'s* and, in the worst case, $4K$ additions as well as K multiplications for calculating U_K: the proof*s* are given in Appendi*ces C and* D. So, algorithm 1 requires *6K + 1* additions and $2K$ multiplications for calculating K.
- Note that in Algorithm 2: 2 multiplies and $3 + |I_{R_{K-1}}|$ additions are needed for the calculation of P_R; 2 adds and 1 multiply for computing F_1, D_1; $4|I_B|$ adds and I_B multiples for evaluating the while loop. Since we have $|I_{R_{K-1}}|, |I_B| < K$, the worst case complexity of Algorithm 2 is given by $5K+5$ adds and $K+3$ multiplies.
- The computational complexity of calculating P_i using (3) is at-most K adds.
- The total computational complexity of solving our CFP of this paper, is *12K+6+M* adds and $3K+3$ multiplies. Since K is not known apriori, the worst case complexity is given by *13M+6* adds and $3M+3$ multiplies. Hence we have a complexity order of $O(M)$ floating point operations (flops).

Table I gives the number of flops required for iterative algorithm of [18] and [19], FWF of [21], GWFPP algorithm of [22] and of the proposed ACF algorithm. Observe the order of magnitude improvement for ACF.

Remark 3: Following the existing algorithms conceived for solving the CFP (like [2] and [22]), we do not consider the complexity of sorting N_i, as the channel gain sequences come from the eigenvalues of a matrix; and most of the algorithms compute the eigenvalues and eigenvectors in sorted order.

Remark 4: Observe that we have not included the complexity of sorting H_i at step 4 in Algorithm 2. This is because the sorting is implementation dependent. For fixed-point implementations, sorting can be performed with a worst case complexity of $O(M)$ comparisons using algorithms like Count Sort [28]. For floating point implementations, sorting can be performed with a worst case complexity of $O(M\log(M))$ comparisons [29]. Since, almost all implementations are of fixed-point representation: the overall complexity, including sorting of H_i would still be of $O(M)$.

III. WEIGHTED CFP

An interesting generalization for CFP is the scenario when the rates and the sum power are weighted, hence resulting in the Weighted CFP (WCFP), arising in the following context.

(a) In a CR network, a CR senses that some resources are available for it's use. Hence the CR allots powers to the available resources for a predefined amount of time while assuring that the peak power remains limited in order to keep the interference imposed on the PU remains within the limit. The weights w_i and x_i may be adjusted based on the resource's available time and on the sensing probabilities [30]–[32].

(b) In Sensor Network (SN) the resources have priorities according to their capability to transfer data. These priorities are reflected in the weights, w_i. The weights x_i's allow the sensor nodes to save energy, while avoiding interference with the other sensor nodes [33], [34].

The optimization problem constituted by weighted CFP is given by

$$\begin{aligned} \max_{\{P_i\}_{i=1}^M} \quad & C = \sum_{i=1}^{M} w_i \log_2\left(1 + \frac{P_i}{N_i}\right) \\ \text{subject to :} \quad & \sum_{i=1}^{M} x_i P_i \le P_t \\ & P_i \le P_{it}, \quad i \le M \\ & \text{and } P_i \ge 0, \quad i \le M, \end{aligned} \tag{22}$$

where again w_i and x_i are the weights of the i^{th} resource's capacity and allocated power, respectively. Similar to Theorem 1, we have

Theorem 4: The solution of the WCFP (22) *is of the 'form'*

$$\bar{P}_i = \begin{cases} \left(\dfrac{1}{\lambda} - \bar{N}_i\right), & 0 < \bar{P}_i < \bar{P}_{it}; \\ \bar{P}_{it}, & \dfrac{1}{\lambda} \ge \bar{H}_i \triangleq \left(\bar{P}_{it} + \bar{N}_i\right); \\ 0, & \dfrac{1}{\lambda} \le \bar{N}_i \end{cases} \tag{23}$$

where "$\frac{1}{\lambda}$ is the water level of the WCFP", $\bar{P}_i = \frac{P_i x_i}{w_i}$ is the weighted power, $\bar{P}_{it} = \frac{P_{it} x_i}{w_i}$ is weighted peak power, $\bar{N}_i = \frac{N_i x_i}{w_i}$ is the weighted interference plus noise level and $\bar{H}_i = \bar{N}_i + \bar{P}_{it}$ is the weighted height of i^{th} cave ceiling stair.

Proof: The proof is similar to Theorem 1 and has been omitted. □

The above solution *form* can be rewritten as

$$\bar{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+, \quad i = 1, \cdots, M; \quad and \tag{24}$$

$$\bar{P}_i \leq \bar{P}_{it}, \quad i = 1, \cdots, M \tag{25}$$

where we have $A^+ \triangleq \max(A, 0)$. The solution for (22) has a simple form for the case the 'implied' weighted power budget, $\bar{P}_{It}$ as defined as $\bar{P}_{It} = \sum_{i=1}^{M} w_i \bar{P}_{it}$ is less than or equal to P_t and is given in Proposition 3.

Proposition 3: If the 'implied' power budget is less than or equal to the power budget ($\sum_{i=1}^{M} w_i \bar{P}_{it} \leq P_t$), then peak power allocation to all the M resources gives optimal capacity.

Note that in this case the total power allocated is less than (or equal to) P_t. However, if $P_t < \sum_{i=1}^{M} w_i \bar{P}_{it}$, then all the M resources cannot be allocated peak powers since it violates the total sum power constraint in (22).

In what follows, we pursue the solution of (22) for the case

$$P_t < \sum_{i=1}^{M} w_i \bar{P}_{it}. \tag{26}$$

We have,

Proposition 4: The optimal powers and hence optimal capacities are achieved in (22) (under the constraint (26)) only if

$$\sum_{i=1}^{M} w_i \bar{P}_i = P_t. \tag{27}$$

It follows that the solution of (22) is given by

$$\bar{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+, \quad i = 1, \cdots, M; \tag{28}$$

$$\sum_{i=1}^{K} w_i \bar{P}_i = P_t; \tag{29}$$

$$\bar{P}_i \leq \bar{P}_{it}, \quad i = 1, \cdots, M. \tag{30}$$

Using the proposed area based approach, we can extend the ACF algorithm to the weighted case as shown in Fig. 3.

Observe that the width of the stairs is now given by w_i in contrast to CFP, and $Z_{i,k}$ is now scaled by a factor of $\frac{x_i}{w_i}$.

Also observe that the sorting order now depends on the $\bar{N}_i$ values, since sorting the $\bar{N}_i$ values in ascending order makes the first K of the $\bar{P}_i$ values positive, while the remaining $\bar{P}_i$ values are equal to zero as per (28).

In what follows, we assume that the parameters like $\bar{H}_i$, $\bar{P}_{it}$, w_i and $\bar{N}_i$ are sorted in the ascending order of $\bar{N}_i$ values in order to conserve the original relationship among parameters.

Comparing (28)-(30) to (3), (4) and (9); we can see that in addition to the scaling of the variables, (29) has a weighing factor of w_i. Most importantly, since the widths of the stairs is not unity, they affect the area under consideration. As a consequence, Algorithms 1 and 2 cannot be directly applied to this case. However, the interpretations are similar. Algorithm 3 details the ACF for WCFP while Algorithm 4, defines the corresponding 'step-based' waterfilling algorithm conceived for finding the optimal values of K and L, respectively.

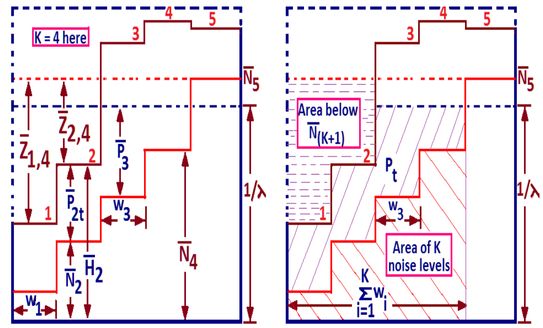


Fig. 3. Showing the effect of 'weights' in Weighted CFP.

Algorithm 3 ACF Algorithm for Obtaining K for WCFP

Require: Inputs required are M, P_t, $\bar{N}_i$, $\bar{H}_i$ & w_i (in ascending order of $\bar{N}_i$).
Ensure: Output is K, $\bar{I}_{R_{K-1}}$,$\bar{I}_{R_K}$, $\bar{d}_K$.
1: $i = 1$. Denote $\bar{d}_0 = P_t$, $W_0 = 0$, $\bar{U}_0 = 0$ and $\bar{I}_{R_0} = \emptyset$
2: Calculate $\bar{d}_i = \bar{d}_{i-1} + w_i \bar{N}_i$.
3: Calculate $W_i = W_{i-1} + w_i$
4: ▷ Calculate the area $\bar{U}_i = \sum_{m=1}^{i} w_m \bar{Z}_{m,i}^+$ as follows:
5: $\bar{I}_{R_i} = \{m;$ such that $\bar{N}_{i+1} > \bar{H}_m\}$, $W_{R_{i-1}} = \sum_{m \in \bar{I}_{R_{i-1}}} w_m$
$\bar{Z}_{m,i} = \bar{N}_{(i+1)} - \bar{H}_m, m \in (I_{R_i} - I_{R_{i-1}})$
6: $\bar{U}_i = \bar{U}_{i-1} + W_{R_{i-1}}(\bar{N}_{i+1} - \bar{N}_i) + \sum_{m \in (\bar{I}_{R_i} - \bar{I}_{R_{i-1}})} w_m \bar{Z}_{m,i}^+$
7: Calculate the area $\bar{Q}_i = W_i \bar{N}_{(i+1)}$
8: **if** $\bar{Q}_i \geq (\bar{d}_i + \bar{U}_i)$ **then**
9: $K \leftarrow i$. Exit the algorithm.
10: **else**
11: $i \leftarrow$ i+1, Go to 2
12: **end if**

Let us now formulate Theorem 5.

Theorem 5: The output of Algorithm 3 gives the optimal value K of the number of positive powers, as defined in Definition 1, for WCFP.

The proof is similar to that of the CFP case, with slight modifications concerning both the scaling and the width of the stairs w_i, hence it has been omitted.

Observe that the calculation of $\bar{P}_R$, $\bar{D}_m$ and $\bar{F}_m$ is affected by the weights w_i, since the areas depend on w_i.

Let us now state without proof that Algorithm 4 outputs the optimal value of L.

Theorem 6: Algorithm 4 delivers the optimal value L of the number of powers that are assigned peak powers, as defined in Definition 2, for WCFP.

Peak power allocated resources are $\bar{I}_P = \bar{I}_{R_{K-1}} \cup I_S(1 : L)$. Resources for which WFP allocates powers are $\bar{I}_P^c = [1, K] - \bar{I}_P$.

Algorithm 4 'Step-Based' Waterfilling Algorithm for Obtaining L for WCFP

Require: Inputs required are K, $\bar{d}_K$, $\bar{I}_{R_{K-1}}$, $\bar{I}_{R_K}$, W_K, $W_{R_{K-1}}$, $\bar{N}_i$, $\bar{H}_i$ & w_i (in ascending order of $\bar{N}_i$).
Ensure: Output is L, I_S.

Calculate $\bar{P}_R = \bar{d}_K - W_K\bar{N}_K + W_{R_{K-1}}\bar{N}_K - \sum_{m\in\bar{I}_{R_{K-1}}} w_m\bar{H}_m$
Calculate $\bar{I}_B = \bar{I}_{R_K} - \bar{I}_{R_{K-1}}$. $\bar{D}_1 = W_K - W_{R_{K-1}}$.
If $|\bar{I}_B| = 0$, set $L = 0$. Otherwise, if $|\bar{I}_B| > 0$, then only proceed with the following steps.
Sort $\{\bar{H}_m\}_{m\in\bar{I}_B}$ in ascending order and denote it as $\{\bar{H}_{mB}\}$ and the sorting index as I_S.
Initialize $m = 1$, $\bar{F}_m = (\bar{H}_{mB} - \bar{N}_K)\bar{D}_m$.
while $\bar{F}_m \le \bar{P}_R$ **do**
  $m = m + 1$. If $m > |\bar{I}_B|$, exit the while loop.
  $\bar{D}_m = \bar{D}_{m-1} - w_{I_S(m-1)}$
  $\bar{F}_m = \bar{F}_{m-1} + (\bar{H}_{mB} - \bar{H}_{(m-1)B})\bar{D}_m$
end while
$L = m - 1$.
calculate $\bar{D}_{L+1} = \bar{D}_L - w_{I_S(L)}$, only if $L = |\bar{I}_B|$.

The waterlevel for WCFP is given by

$$\frac{1}{\lambda} = \begin{cases} \bar{H}_{LB} + \dfrac{\bar{P}_R - \bar{F}_L}{\bar{D}_{L+1}}, & L > 0; \\ \bar{N}_K + \dfrac{\bar{P}_R}{\bar{D}_1}, & \text{L = 0.} \end{cases} \tag{31}$$

and the powers allocated are given by

$$P_i = \begin{cases} P_{it}, & i \in \bar{I}_P; \\ \dfrac{w_i}{x_i}\left(\dfrac{1}{\lambda} - \bar{N}_i\right), & i \in \bar{I}_P^c. \end{cases} \tag{32}$$

A. Computational Complexity of the WCFP

Let us now calculate the computational complexity of both Algorithm 3 and of Algorithm 4 and then add the complexity of calculating the powers, as follows:

- Calculating $\bar{N}_i$, $\bar{P}_{it}$ and $\bar{H}_i$ requires $3M$ multiplies and M adds.
- Observe that Algorithm 3 requires $(K + 1)$ adds and K multiplies for calculating $\bar{d}_i$, K multiplies to find $\bar{Q}_i$ and, in the worst case, $4K$ additions and $2K$ multiplications for calculating $\bar{Z}_{m,i}$'s & $\bar{U}_K$, the corresponding proof is given in Appendix VI-E; K additions for calculating W_K and at-most K additions for calculating $W_{R_{i-1}}$. Consequently Algorithm 3 requires $(7K + 1)$ additions and $4K$ multiplications for calculating K.
- Note that in Algorithm 4: 2 multiplies and $3 + |\bar{I}_{R_{K-1}}|$ additions are required for calculation of $\bar{P}_R$; at-most $(K + 1)$ adds and 1 multiply in computing $\bar{F}_1, \bar{D}_1$; $4|\bar{I}_B|$ adds and $\bar{I}_B$ multiples for evaluating the while loop. Since $|\bar{I}_{R_{K-1}}|, |\bar{I}_B| < K$, the worst case complexity of Algorithm 4 can be given as $(6K + 4)$ adds, $(K + 3)$ multiplies.
- The computational complexity of calculating P_i is at-most K adds and K multiplies.
- Consequently, the total computational complexity of solving the WCFP, considered is $(14K + 5 + M)$ adds and $(3M + 6K + 3)$ multiplies. Since K is not known apriori, the worst case complexity is given by $(15M + 5)$ adds and $(9M + 3)$ multiplies. i.e we have a complexity order of $O(M)$.

Explicitly, the proposed solution's computational complexity is of the order of M, whereas that of the GWFPP of [22] is of the order of M^2.

IV. WCFP Requiring Minimum Power

In this section we further extend the WCFP to the case where the resources/powers scenario of having both a Minimum and a Maximum Power (MMP) constraint. The resultant WCFP-MMP arises in the following context:

(a) In a CR network, CR senses that some resources are available for it's use and allocates powers to the available resources for a predefined amount of time while ensuring that the peak power constraint is satisfied, in order to keep the interference imposed on the PU with in the affordable limit. Again, the weights w_i and x_i represent the resource's available time and sensing probabilities. The minimum power has to be sufficient to support the required quality of service, such as the minimum transmission rate of each resource [30]–[32].

We show that solving WCFP-MMP can be reduced to solving WCFP with the aid of an appropriate transformation. Hence, Section III can be used for this case. Mathematically, the problem can be formulated as

$$\max_{\{P_i\}_{i=1}^M} \; C = \sum_{i=1}^{M} w_i \log_2\left(1 + \frac{P_i}{N_i}\right)$$
$$\text{subject to}: \; \sum_{i=1}^{M} x_i P_i \le P_t \tag{33}$$
$$P_{ib} \le P_i \le P_{it}, \quad i \le M$$
$$\text{and } P_i \ge 0, \quad i \le M,$$

where $P_{ib} \le P_{it}$ and P_{ib} is the lower bound while P_{it} is the upper bound of the i^{th} power. w_i and x_i are weights of the i^{th} resource's capacity and i^{th} resource's allotted power, respectively. Using the KKT, the solution of this case can be written as

$$\bar{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+, \quad i = 1, \cdots, M; \tag{34}$$

$$\sum_{i=1}^{K} w_i\bar{P}_i = P_t; \tag{35}$$

$$\bar{P}_{ib} \le \bar{P}_i \le \bar{P}_{it}, \quad i = 1, \cdots, M, \tag{36}$$

where $\bar{P}_i = \frac{P_i x_i}{w_i}$ is the weighted power, $\bar{P}_{it} = \frac{P_{it} x_i}{w_i}$ is weighted peak power, $\bar{P}_{ib} = \frac{P_{ib} x_i}{w_i}$ is the weighted minimum power and $\bar{N}_i = \frac{N_i x_i}{w_i}$ is the weighted noise.

Let us now formulate Theorem 7.

Theorem 7: For every WCFP-MMP given by (33)*, there exists a WCFP, whose solution will result in a solution to the WCFP-MMP.*

Proof: Consider the solution to WCFP-MMP given by (34)-(36). Defining $\hat{P}_i = \bar{P}_i - \bar{P}_{ib}$ and substituting it into (34)-(36), we arrive at:

$$\hat{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+ - \bar{P}_{ib}, \quad i = 1, \cdots, M; \tag{37}$$

$$\sum_{i=1}^{K} w_i(\hat{P}_i + \bar{P}_{ib}) = P_t; \tag{38}$$

$$0 \le \hat{P}_i \le \left(\bar{P}_{it} - \bar{P}_{ib}\right), \quad i = 1, \cdots, M. \tag{39}$$

Using (37) and the definition of $()^+$, we can rewrite (37)–(39) as

$$\hat{P}_i = \left(\frac{1}{\lambda} - \underbrace{\{\bar{N}_i + \bar{P}_{ib}\}}_{\hat{N}_i}\right)^+, \quad i = 1, \cdots, M; \tag{40}$$

$$\sum_{i=1}^{K} w_i \hat{P}_i = \underbrace{\left(P_t - \sum_{i=1}^{K} w_i \bar{P}_{ib}\right)}_{\hat{P}_t}; \tag{41}$$

$$0 \le \hat{P}_i \le \underbrace{\left(\bar{P}_{it} - \bar{P}_{ib}\right)}_{\hat{P}_{it}}, \quad i = 1, \cdots, M. \tag{42}$$

Comparing (40)-(42) to (28)-(30), we can observe that this is a solution for a WCFP with variables $\hat{P}_i$, $\hat{N}_i$, $\hat{P}_{it}$ and $\hat{P}_t$. It follows then that we can solve the WCFP-MMP by solving the WCFP, whose solution is given by (40)-(42). □

Note that the effect of the lower bound is that of increasing the height of the floor stairs for the corresponding WCFP at a concomitant reduction of the total power constraint.

A. Computaional Complexity of the WCFP-MMP

Solving WCFP-MMP requires $4M$ additional adds, to compute $\hat{P}_i$, $\hat{N}_i$, $\hat{P}_{it}$ as well as $\hat{P}_t$, and K adds to recover P_i from $\hat{P}_i$; as compared to WCFP. Hence the the worst case complexity of solving the WCFP-MMP is given by $(19M+6)$ adds and $(8M+3)$ multiplies. i.e we have a complexity of $O(M)$.

V. Simulation Results

Our simulations have been carried out in MATLAB R2010b software. To demonstrate the operation of the proposed algorithm, some numerical examples are provided in this section.

Example 1: Illustration of the CFP is provided by the following simple example:

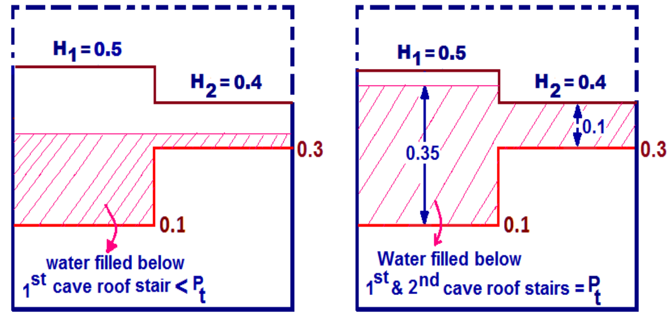
$$\max_{\{P_i\}_{i=1}^2} \; C = \sum_{i=1}^{2} \log_2\left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints}: \sum_{i=1}^{2} P_i \le 0.45;$$

$$P_i \le 0.7 - 0.3i, \quad i \le 2$$

$$\text{and } P_i \ge 0, \quad i \le 2. \tag{43}$$

Assuming $N_i = \{0.1, 0.3\}$, we have $H_i = \{0.5, 0.4\}$. For the example of (43), water is filled above the first floor stair, as shown in Fig. 4a. This quantity of water is less than P_t. Hence, we fill the water above the second floor stair until the



(a) (b)

Fig. 4. Illustration for Example 1: (a) Water filled above floor stairs 1 and 2, without peak constraint. (b) Water filled above floor stairs 2 only.

water level reaches 0.45. At this point the peak constraint for the second resource comes into force and the water can only be filled above second floor stair, as shown in Fig. 4b. Now, this amount of water becomes equal to P_t giving $K = 2$. We can observe that the first resource has a power determined by the 'waterlevel', while the second resource is assigned the peak power.

In Algorithm 1, we have $U_1 = 0$ as $Z_{1,1}^+ = 0$ and $I_{R_1} = 0$. $d_1 = P_t + N_1 = 0.55$, while $Q_1 = 1 \times N_2 = 0.3$. We can check that $Q_1 \not\geq (d_1 + U_1)$ which indicates that $K > 1$. Hence, we get $K = 2$.

Let us now use Algorithm 2 to find the specific resources that are to be allocated the peak powers. We have $I_{R_{K-1}} = 0$ as $N_K < H_1$. The remaining power P_R in Algorithm 2 is 0.25. The resource indices to check for the peak power allocation are $I_B = \{1, 2\}$. From $H_m|_{m \in I_B}$, we get $[H_{1B}, H_{2B}] = \{0.4, 0.5\}$ and $I_S = \{2, 1\}$. We can check that $F_1 = 0.2 < P_R$ and $F_2 = 0.3 > P_R$. This gives $L = 1$. Hence we allocate the peak power to the $I_S(L)$ or second resource, i.e. we have $P_2 = P_{2t} = 0.1$. The first resource can be assigned the remaining power of $P_1 = P_t - P_{2t} = 0.35$.

Example 2: A slightly more involved example of the CFP, with more resources is illustrated here:

$$\max_{\{P_i\}_{i=1}^8} \; C = \sum_{i=1}^{8} \log_2\left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints}: \sum_{i=1}^{8} P_i \le 6;$$

$$P_i \le P_{it}, \quad i \le 8$$

$$\text{and } P_i \ge 0, \quad i \le 8. \tag{44}$$

In (44); we have $N_i = 2i - 1, \forall i$ and $P_{it} = \{8, 1, 3, 3, 6, 3, 4, 1\}$. The heights of the cave roof stairs are $H_i = \{9, 4, 8, 10, 15, 14, 17, 16\}$.

In Fig. 5, when the water is filled below the third cave roof stair, the amount of water is $P_t = 6$, which fills above the three cave floor stairs, hence giving $K = 3$. The same can be obtained from Algorithm 1. Using Algorithm 1, the $(d_i + U_i)$ and the Q_i values are obtained which are shown in Table II. Since the $(d_i + U_i)$ values are $\{7, 11, 18\}$, while the Q_i are $\{3, 10, 21\}$, we have $Q_3 > (d_3 + U_3)$ and $Q_i < (d_i + U_i)$, $i = 1, 2$. This gives $K = 3$.

As we have $N_K = 5 > H_2 = 4$, $I_{R_{K-1}} = 2$; the second resource is to be assigned the peak power.

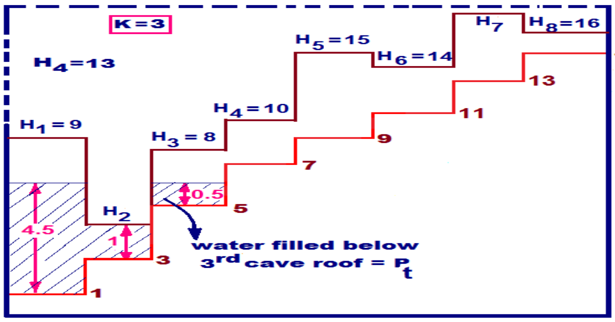


Fig. 5. Illustration of Example 2: Water filled below the roof stair 3 gives $K = 3$.

TABLE II

RESULTS FOR EXAMPLE 2:

Parameter	**Values of the parameters for** (44)
$(d_i + U_i), i \leq K$	7, 11, 18
$Q_i, \quad i \leq K$	3, 10, 21
Peak power based resources	2
Water filling based resources	1, 3
Powers of the resources $P_i, i \in [1, K]$	4.5, 1, 0.5
Capacities of the resources $i \in [1, K]$	2.4594,2.8745,3.0120

Similarly, as $N_{K+1} (= 7) > H_i, i \in [1, K]$ is satisfied for $i = 2$ resource, we have $I_{R_K} = 2$. Since $I_B = I_{R_K} - I_{R_{K-1}} = \emptyset$, there are no resources that have $H_i, i \in [1, K]$ values in between N_K and N_{K+1}. Thus, there is no need to invoke the 'step-based water filling' of Algorithm 2, which gives $L = 0$.

Now, peak power based resources are $I_P = I_{R_{K-1}} = \{2\}$. The water filling algorithm allocates powers for the $I_P^c = [1, K] - I_P = \{1, 3\}$ resources.

The peak power based resources and water filling based resources are shown in Table II. For the remaining power, $P_R = 1$, the water level obtained for the I_P^c resources (with $L = 0$) is 5.5. The powers allocated to the resources $\{1, 3\}$ using this water level are $\{4.5, 0.5\}$. The powers and corresponding throughputs are shown in Table II.

Example 3: The weighted CFP is illustrated by the following simple example:

$$\max_{\{P_i\}_{i=1}^{5}} \quad C = \sum_{i=1}^{5} w_i \log_2 \left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints :} \sum_{i=1}^{5} x_i P_i \leq 5;$$

$$P_i \leq 2, \quad i \leq 5$$

$$\text{and } P_i \geq 0, \quad i \leq 5. \qquad (45)$$

In (45); lets us consider $N_i = [0.2, 0.1, 0.4, 0.3, 0.5]$, $w_i = 6 - i$ and $x_i = i$, $\forall i$. The $\bar{N}_i$ values are [0.04, 0.05, 0.4, 0.6, 2.5], while the $\bar{H}_i$ values are [0.44, 1.05, 2.40, 4.60, 12.5]. Applying the ACF algorithm, we arrive at $K = 4$.

We have $\bar{H}_i < \bar{N}_K, i \in [1, K]$ for the 1^{st} resource. The 'step-based' waterfilling algorithm confirms that 1^{st} resource is indeed the resource having the peak power. The remaining $2^{nd}, 3^{rd}$ and 4^{th} resources are allocated their powers using the water filling algorithm. For the water level of 0.62222, powers allotted for {2,3,4} resources are [1.1444, 0.22222, 0.011111].

Example 4: Another example for the weighted CFP associated with random weights:

$$\max_{\{P_i\}_{i=1}^{64}} \quad C = \sum_{i=1}^{64} w_i \log_2 \left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints :} \sum_{i=1}^{64} x_i P_i \leq 1;$$

$$P_i \leq P_{it}, \quad i \leq 64$$

$$\text{and } P_i \geq 0, \quad i \leq 64. \qquad (46)$$

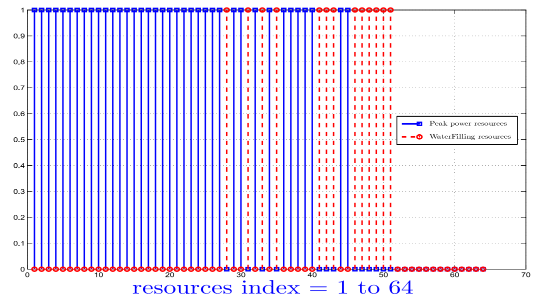


Fig. 6. Index of the peak power based resources (continuous lines) and waterfilling allotted resources (dashed lines) for Example 4.

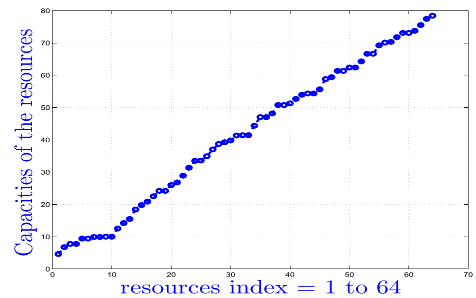


Fig. 7. Throughputs of the resources for Example 4.

In this example, we assume $N_i = \frac{\sigma^2}{h_i}$ while h_i, w_i and x_i are exponentially distributed with a mean of 1. Furthermore $\sigma^2 = 10^{-2}$ and P_{it}, $\forall i$ are random values in the range $[10^{-3}, 5 \times 10^{-2}]$.

Now applying the ACF algorithm, we get $K = 51$ for a particular realization of h_i, w_i and x_i. For this realization, from the $[1, K]$ resources, 38 resources are to be allocated with the peak powers and 13 resources get powers from the waterfilling algorithm. These resources are shown in Fig. 6. The achieved throughput of the resources is given in Fig. 7 for the proposed algorithm. The results match with the values obtained for known algorithms.

Table III gives the actual number of flops required by the proposed solution and the other existing algorithms for

TABLE III

COMPUTATIONAL COMPLEXITIES OF EXISTING ALGORITHMS AND THE PROPOSED SOLUTION FOR $\omega_i = x_i = 1, \forall i$

M → K	**Number of flops in algorithms of [18], [19]**[§]	**Number of flops in FWF of [21]**[¶]	**Number of flops in GWFPP of [22]**	**Number of flops in in proposed solution** [‖]
64 → 46	14985216 (39024)	7824 (24)	16832	541 (24,6)
128 → 87	70563072 (91879)	33592 (52)	66432	956 (31,1)
256 → 135	291746304 (189939)	96450 (75)	263936	1513 (13,4)
512 → 210	$1.5115 \times 10^{+09}$ ($4.9203 \times 10^{+05}$)	156526 (61)	1052160	2432 (21,0)
1024 → 334	$1.6165 \times 10^{+10}$ ($2.6311 \times 10^{+06}$)	271678 (53)	4201472	4059 (15,1)

Example 4 with different M values. Since some of the existing algorithms do not support $\omega_i \neq 1$ and $x_i \neq 1, \forall i$; we assume $\omega_i = x_i = 1, \forall i$ for Table III.

It can be observed from Table III that the number of flops imposed by the sub-gradient algorithm of [18] and [19] is more than 10^4 times that of the proposed solution. The number of flops required for the FWF of [21] and for the GWFPP of [22] are more than 10^2 times that of the proposed solution. This is because the proposed solution's computational complexity is $O(M)$, whereas the best known existing algorithms have an $O(M^2)$ order of computational complexity; as listed in Table I.

It has also been observed from the above examples that $|I_B| = |I_{R_K} - I_{R_{K-1}}|$ values are very small as compared to M. As such L has been obtained from Algorithm 2 within two iterations of the while loop.

VI. CONCLUSIONS

In this paper, we have proposed algorithms for solving the CFP at a complexity order of $O(M)$. The approach was then generalized to the WCFP and to the WCFP-MMP. Since the best known solutions solve these three problems at a complexity order of $O(M^2)$, the proposed solution results in a significant reduction of the complexity imposed. The complexity reduction attained is also verified by simulations.

APPENDIX

A. Proof of Theorem 1

Proof: Lagrange's equation for (1) is

$$L(P_i, \lambda, \omega_i, \gamma_i) = \sum_{i=1}^{M} \log_2 \left(1 + \frac{P_i}{N_i}\right) - \lambda \left(\sum_{i=1}^{M} P_i - P_t\right) - \sum_{i=1}^{M} \omega_i (P_i - P_{it}) - \sum_{i=1}^{M} \gamma_i (0 - P_i) \quad (47)$$

[§] λ is initialized to 5×10^{-1}.

[§,¶] Number of iterations is given in brackets.

[‖] $|I_{R_{K-1}}|$ and $|I_B|$ are given in brackets. Actual number of flops is $M + 9K + 5|I_B| + |I_{R_{K-1}}| + 9$.

Karush-Kuhn-Tucker (KKT) conditions for (47) are [3], [35]

$$\frac{\partial L}{\partial P_i} = 0 \Rightarrow \frac{1}{N_i + P_i} - \lambda - \omega_i + \gamma_i = 0, \quad (48)$$

$$\lambda \left(P_t - \sum_{i=1}^{M} P_i\right) = 0, \quad (49)$$

$$\omega_i (P_{it} - P_i) = 0, \quad \forall i \quad (50)$$

$$\gamma_i P_i = 0, \quad \forall i \quad (51)$$

$$\lambda, \omega_i \ \& \ \gamma_i \geq 0, \quad \forall i \quad (52)$$

$$P_i \leq P_{it}, \quad \forall i, \quad (53)$$

$$\sum_{i=1}^{M} P_i \leq P_t. \quad (54)$$

In what follows we show that the KKT conditions result in a simplified *'form'* for the solution of CFP which is similar to the conventional WFP. *The proof is divided into three parts corresponding to the three possibilities for P_i: that is 1) Equivalent constraint for $P_i < 0$ in terms of the 'water level' $\frac{1}{\lambda}$ and the corresponding solution form, 2) Equivalent constraint for $P_i \leq P_{it}$ in terms of the 'water level' and and the corresponding solution form, and 3) Equivalent form for $P_i < P_i < P_{it}$ in terms of the 'water level' and the corresponding solution form.*

1) Simplification for $P_i \geq 0$: Multiplying (48) with P_i and substituting (51) in it, we obtain

$$P_i \left(\frac{1}{N_i + P_i} - \lambda - \omega_i\right) = 0 \quad (55)$$

In order to satisfy (55), either P_i or $(\frac{1}{N_i+P_i} - \lambda - \omega_i)$ should be zero. Having $P_i = 0, \forall i$ does not solve the optimization problem. Hence, we obtain

$$\left(\frac{1}{N_i + P_i} - \lambda - \omega_i\right) = 0, \quad when\ P_i > 0. \quad (56)$$

Since $\omega_i \geq 0$, (56) can be re-written as $(\frac{1}{N_i+P_i} - \lambda) \geq 0$. Furthermore, taking $P_i > 0$ in this, we attain

$$\frac{1}{\lambda} > N_i, \quad when\ P_i > 0. \quad (57)$$

The opposite of this is

$$\frac{1}{\lambda} \leq N_i, \quad when\ P_i \leq 0. \tag{58}$$

We can observe that (57) and (58) are equations related to the conventional WFP.

2) Simplification for $P_i \leq P_{it}$: Multiplying (48) with $P_{it} - P_i$ and substituting (50) in it, we attain

$$(P_{it} - P_i)\left(\frac{1}{N_i + P_i} - \lambda + \gamma_i\right) = 0 \tag{59}$$

In (59), two cases arise:

(a) If $P_{it} > P_i$, then $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) = 0$ becomes true.
Since $\gamma_i \geq 0$, $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) = 0$ is taken as $(\frac{1}{N_i+P_i} - \lambda) < 0$. Further Simplifying this and substituting $P_i < P_{it}$, we get

$$\frac{1}{\lambda} < H_i \triangleq (P_{it} + N_i), \quad if\ P_i < P_{it}. \tag{60}$$

(b) If $P_{it} = P_i$, then $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) \geq 0$ becomes true in (59).
As $\gamma_i \geq 0$, $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) \geq 0$ is re-written as $(\frac{1}{N_i+P_i} - \lambda) \geq 0$. Substituting $P_{it} = P_i$ and simplifying this further, we obtain

$$\frac{1}{\lambda} \geq H_i \triangleq (P_{it} + N_i), \quad if\ P_i = P_{it}. \tag{61}$$

3) Simplification for $0 < P_i < P_{it}$:

(a) In (51); if γ_i is equal to zero, then $P_i > 0$. Combining this relation with (57), we can conclude that

$$\frac{1}{\lambda} > N_i, \quad if\ \gamma_i = 0. \tag{62}$$

(b) Similarly, in (50), if $\omega_i = 0$, then $P_{it} > P_i$ follows. Using this relation in (60), we acquire

$$\frac{1}{\lambda} < H_i, \quad if\ \omega_i = 0. \tag{63}$$

(c) Combining (62) and (63), we have

$$N_i < \frac{1}{\lambda} < H_i, \quad if\ \omega_i = \gamma_i = 0. \tag{64}$$

Using (64) in (48) and then re-arranging it gives

$$P_i = \frac{1}{\lambda} - N_i, \quad if\ N_i < \frac{1}{\lambda} < H_i. \tag{65}$$

Combining (57), (58), (60), (61) and (65), powers are obtained as

$$P_i = \begin{cases} \left(\frac{1}{\lambda} - N_i\right), & N_i < \frac{1}{\lambda} < H_i \text{ or } \\ & 0 < P_i < P_{it}; \\ P_{it}, & \frac{1}{\lambda} \geq H_i; \\ 0, & \frac{1}{\lambda} \leq N_i. \end{cases} \tag{66}$$

□

B. Proof of Proposition 2

Proof: The proof is by contradiction. Assume that $P_i^\star$, $i \leq M$ is the optimal solution for (1) such that $\sum_{i=1}^{M} P_i^\star < P_t$. We now prove that as $P_i^\star$ powers fulfil $\sum_{i=1}^{M} P_i^\star < P_t$, there exists $P_i^\diamond$ that has greater capacity. Define

$$P_i^\diamond = P_i^\star + \triangle P_i^\star, \quad \forall i \tag{67}$$

such that

$$\sum_{i=1}^{M} P_i^\diamond = P_t \quad \text{and } P_i^\diamond \leq P_{it}, \quad \forall i \tag{68}$$

where $\triangle P_i^\star \geq 0, \forall i$. From (7) there exists atleast one i such that $P_i^\star < P_{it}$. It follows that $\triangle P_i^\star > 0$ for atleast one i. The capacity of M resources for $P_i^\diamond$ allotted powers is

$$C\left(P_i^\diamond\right) = \sum_{i=1}^{M} \log_2\left(1 + \frac{P_i^\diamond}{N_i}\right) \tag{69}$$

Substituting (67) in (69), we get

$$C\left(P_i^\diamond\right) = \sum_{i=1}^{M} \log_2\left(1 + \frac{P_i^\star}{N_i} + \frac{\triangle P_i^\star}{N_i}\right) \tag{70}$$

Re-writing the above, we obtain

$$C\left(P_i^\diamond\right) = \sum_{i=1}^{M} \log_2\left[\left(1 + \frac{P_i^\star}{N_i}\right)\left(1 + \frac{\frac{\triangle P_i^\star}{N_i}}{1 + \frac{P_i^\star}{N_i}}\right)\right] \tag{71}$$

Following 'log(ab) = log(a) + log(b)' in the above, we acquire

$$C\left(P_i^\diamond\right) = \sum_{i=1}^{M} \log_2\left(1 + \frac{P_i^\star}{N_i}\right) + \sum_{i=1}^{M} \log_2\left(1 + \frac{\frac{\triangle P_i^\star}{N_i}}{1 + \frac{P_i^\star}{N_i}}\right) \tag{72}$$

As $\triangle P_i^\star > 0$ for atleast one i, the second term on the R.H.S. of (72) is always positive. We have

$$C\left(P_i^\diamond\right) > C\left(P_i^\star\right) \tag{73}$$

In other words, $\sum_{i=1}^{M} P_i^\diamond = P_t$ produces optimal capacity; completing the proof. □

C. The Computational Complexity of Calculating $Z_{m,i}$ for CFP

Below, it is shown that the worst case computational complexity of calculating $Z_{m,i}$; $m \leq i$ and $i \leq K$ for CFP is K subtractions.

- *In Algorithm 1, we first check if $N_{i+1} > H_m$. I_{R_i} is taken as 'm' values for which $N_{i+1} > H_m$. Note also that $I_{R_{i-1}} \subset I_{R_i}$. This is because if $Z_{m,i} = N_{i+1} - H_m > 0$, then $Z_{m,j}$; $j = i+1, \cdots, K$ is always positive since $N_j > N_i$, $j > i$. Hence, in the worst case, $K\log(K)$ comparisons are required. The cost of a comparison, is typically lower than that of an addition [36]. Hence it has not been included in the flop count.*
- *As per Algorithm 1, we calculate $Z_{m,i}$'s only for $m \in (I_{R_i} - I_{R_{i-1}})$. Furthermore, if we have $Z_{m,i} = N_{i+1} - H_m > 0$, then $Z_{m,j}$; $j = i+1, \cdots, K$ is always positive*

since $N_j > N_i$, $j > i$. In other words, if $I_{R_{i-1}}$ gets some 'x' values, then the same 'x' values will also be there in I_{R_i} and the contribution of this part to the overall area, U_i is $|I_{R_{i-1}}|(N(i+1) - N_i)$; which is calculated in Step 5. This implies that if $Z_{m,i}$ is calculated for $m \in I_{R_i}$, then there is no need to calculate $Z_{m,i}$ for $m \in I_{R_{i+1}}, I_{R_{i+2}}, \ldots I_{R_K}$. Hence, for a given m, $Z_{m,i}$ is calculated, in the worst case, once; for one 'i' only. As such, the worst case complexity of calculating $Z_{m,i}$ is as low as that of K subtractions.

D. The Computational Complexity of Calculating U_K for CFP

Here we show that the worst case computational complexity of calculating U_K for CFP is $4K$ adds and K multiplies. Note that in each iteration of Algorithm 1 the following is calculated:

$$U_i = U_{i-1} + |I_{R_{i-1}}|\,(N_{i+1} - N_i) + \sum_{m \in (I_{R_i} - I_{R_{i-1}})}^{i} Z^{+}_{m,i}. \quad (74)$$

There are three terms in (74) and we calculate the complexity of each term separately, as follows:

- The first term of (74), U_{i-1}, is already computed in the $(i-1)$-th iteration, hence involves no computation during the i-th iteration.
- The second term, $|I_{R_{i-1}}|(N_{i+1} - N_i)$, is taking care of the increase in reference height from N_i to N_{i+1} for those roof stairs, which are already below the reference level N_i. The computation of this term requires only a single multiplication and addition.
- The third term gives the areas of the roof stairs which are below N_{i+1} but not N_i. The number of additions in this is $A_i = |I_{R_i} - I_{R_{i-1}}| - 1$.
- Taking into account the two adds per iteration required for adding all the three terms, the total computational complexity of calculating U_i, given U_{i-1} is 1 multiply and $3 + A_i$ adds.

Since KU_i's are calculated; the total computational complexity of calculating all U_i's will be $\sum_{i=1}^{K} 3 + A_i = 3K + |I_{R_K}| \leq 4K$ adds and K multiplies.

E. The Computational Complexity of Calculating $\bar{U}_K$ for WCFP

Here we show that the worst case computational complexity of calculating $\bar{U}_K$ for WCFP is $4K$ adds $2K$ multiplies. Note that in each iteration of Algorithm 3 the following is calculated:

$$\bar{U}_i = \bar{U}_{i-1} + W_{R_{i-1}}\left(\bar{N}_{i+1} - \bar{N}_i\right) + \sum_{m \in (\bar{I}_{R_i} - I_{R_{i-1}})}^{i} w_m \bar{Z}^{+}_{m,i}. \quad (75)$$

There are three terms in (75) and we calculate the complexity of each term separately, as follows:

- The first term of (75), $\bar{U}_{i-1}$, is already computed in $i-1$-th iteration, hence involves no computation during the i-th iteration.
- The computation of second term, $W_{R_{i-1}}(\bar{N}_{i+1} - \bar{N}_i)$, requires only a single multiplication and addition.
- The third term gives the areas of the roof stairs which are below $\bar{N}_{i+1}$ but not $\bar{N}_i$. The number of additions in this is $A_i = |\bar{I}_{R_i}| - |\bar{I}_{R_{i-1}}|$. The corresponding number of multiplications is one.
- Taking into account the two adds per iteration required for adding all the three terms, the total computational complexity of calculating U_i, given U_{i-1} is 2 multiply and $3 + A_i$ adds.

Since KU_i's are calculated; the total computational complexity of calculating all U_i's will be $\sum_{i=1}^{K} 3 + A_i = 3K + |I_{R_K}| \leq 4K$ adds and $2K$ multiplies.

REFERENCES

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, May 2005.

[2] D. P. Palomar and J. R. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 686–695, Feb. 2005.

[3] F. Gao, T. Cui, and A. Nallanathan, "Optimal training design for channel estimation in decode-and-forward relay networks with individual and total power constraints," *IEEE Trans. Signal Process.*, vol. 56, no. 12, pp. 5937–5949, Dec. 2008.

[4] A. A. D'Amico, L. Sanguinetti, and D. P. Palomar, "Convex separable problems with linear constraints in signal processing and communications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 6045–6058, Nov. 2014.

[5] E. Altman, K. Avrachenkov, and A. Garnaev, "Closed form solutions for water-filling problems in optimization and game frameworks," *Telecommun. Syst.*, vol. 47, nos. 1–2, pp. 153–164, 2011.

[6] R. Zhang, "On peak versus average interference power constraints for protecting primary users in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 2112–2120, Apr. 2009.

[7] X. Kang, R. Zhang, Y.-C. Liang, and H. K. Garg, "Optimal power allocation strategies for fading cognitive radio channels with primary user outage constraint," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 374–383, Feb. 2011.

[8] G. Bansal, M. J. Hossain, and V. K. Bhargava, "Optimal and suboptimal power allocation schemes for OFDM-based cognitive radio systems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4710–4718, Nov. 2008.

[9] N. Kalpana, M. Z. A. Khan, and U. B. Desai, "Optimal power allocation for secondary users in CR networks," in *Proc. IEEE Adv. Netw. Telecommun. Syst. Conf. (ANTS)*, Bengaluru, India, Dec. 2011, pp. 1–6.

[10] H. Zhang and D. L. Goeckel, "Peak power reduction in closed-loop MIMO-OFDM systems via mode reservation," *IEEE Commun. Lett.*, vol. 11, no. 7, pp. 583–585, Jul. 2007.

[11] C. Studer and E. G. Larsson, "PAR-aware large-scale multi-user MIMO-OFDM downlink," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 303–313, Feb. 2013.

[12] N. Andgart, B. S. Krongold, P. Ödling, A. Johansson, and P. O. Börjesson, "PSD-constrained PAR reduction for DMT/OFDM," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 10, pp. 1498–1507, 2004. AQ:4

[13] A. Amirkhany, A. Abbasfar, V. Stojanović, and M. A. Horowitz, "Practical limits of multi-tone signaling over high-speed backplane electrical links," in *Proc. ICC*, Jun. 2007, pp. 2693–2698.

[14] V. M. K. Chan and W. Yu, "Multiuser spectrum optimization for discrete multitone systems with asynchronous crosstalk," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5425–5435, Nov. 2007.

[15] L. Fang and R. J. P. de Figueiredo, "Energy-efficient scheduling optimization in wireless sensor networks with delay constraints," in *Proc. ICC*, Jun. 2007, pp. 3734–3739.

[16] A. Roumy and D. Gesbert, "Optimal matching in wireless sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 725–735, Dec. 2007.

[17] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher, "RID: Radio interference detection in wireless sensor networks," in *Proc. IEEE Adv. Netw. Telecommun. Syst. Conf. (ANTS)*, Bangalore, India, Dec. 2011. AQ:5

[18] M. Arulraj and T. S. Jeyaraman, "MIMO radar waveform design with peak and sum power constraints," *EURASIP J. Adv. Signal Process.*, vol. 2013, no. 1, p. 127, 2013.

[19] L. Zhang, Y. Xin, Y.-C. Liang, and H. V. Poor, "Cognitive multiple access channels: Optimal power allocation for weighted sum rate maximization," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2754–2762, Sep. 2009.

[20] E. Yaacoub and Z. Dawy, *Resource Allocation in Uplink OFDMA Wireless Systems: Optimal Solutions and Practical Implementations*. New York, NY, USA: Wiley, 2012.

[21] X. Ling, B. Wu, P.-H. Ho, F. Luo, and L. Pan, "Fast water-filling for agile power allocation in multi-channel wireless communications," *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1212–1215, Aug. 2012.

[22] P. He, L. Zhao, S. Zhou, and Z. Niu, "Water-filling: A geometric approach and its application to solve generalized radio resource allocation problems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3637–3647, Jul. 2013.

[23] R.-R. Chen and Y. Lin, "Optimal power control for multiple access channel with peak and average power constraints," in *Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput.*, vol. 2. Jun. 2005, pp. 1407–1411.

[24] N. Papandreou and T. Antonakopoulos, "Bit and power allocation in constrained multicarrier systems: The single-user case," *EURASIP J. Adv. Signal Process.*, vol. 2008, Jan. 2008, Art no. 11.

[25] X. Zhou, R. Zhang, and C. K. Ho, "Wireless information and power transfer in multiuser OFDM systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 4092–4097.

[26] N. Kalpana and M. Z. A. Khan, "Fast Computation of Generalized Waterfilling Problems," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1884–1887, Nov. 2015.

[27] N. Kalpana and M. Z. A. Khan, "Weighted water-filling algorithm with reduced computational complexity," in *Proc. ICCIT Conf.*, May 2015. AQ:6

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.

[29] D. E. Knuth, *The Art of Computer Programming: Sorting Searching*, vol. 3, 2nd ed. Boston, MA, USA: Addison-Wesley, 1998.

[30] L. Zhang, Y.-C. Liang, and Y. Xin, "Joint beamforming and power allocation for multiple access channels in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 38–51, Jan. 2008.

[31] S. Stotas and A. Nallanathan, "Optimal sensing time and power allocation in multiband cognitive radio networks," *IEEE Trans. Commun.*, vol. 59, no. 1, pp. 226–235, Jan. 2011.

[32] Z. Tang, G. Wei, and Y. Zhu, "Weighted sum rate maximization for OFDM-based cognitive radio systems," *Telecommun. Syst.*, vol. 42, nos. 1–2, pp. 77–84, Oct. 2009.

[33] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[34] R. Rajesh, V. Sharma, and P. Viswanath. (2012). "Information capacity of energy harvesting sensor nodes." [Online]. Available: http://arxiv.org/abs/1009.5158

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] A. Bellaouar and M. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. New York, NY, USA: Springer, 1995.

AQ:7

Kalpana Naidu received the Ph.D. degree from IIT Hyderabad, in 2016. Since 2016, she has been an Associate Professor with the VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad. The focus of her research is on resource allocation in wireless communication, HetNets, cognitive radio networking, and signal processing applied to wireless networks.

Mohammed Zafar Ali Khan received the B.E. degree in electronics and communications from Osmania University, Hyderabad, India, in 1996, the M.Tech. degree in electrical engineering from IIT Delhi, Delhi, India, in 1998, and the Ph.D. degree in electrical and communication engineering from the Indian Institute of Science, Bangalore, India, in 2003. In 1999, he was a Design Engineer with Sasken Communication Technologies, Ltd., Bangalore. From 2003 to 2005, he was a Senior Design Engineer with Silica Labs Semiconductors India Pvt. Ltd., Bangalore. In 2005, he was a Senior Member of the Technical Staff with Hellosoft, India. From 2006 to 2009, he was an Assistant Professor with IIIT Hyderabad. Since 2009, he has been with the Department of Electrical Engineering, IIT Hyderabad, where he is currently a Professor. He has more than ten years of experience in teaching and research and the space-time block codes that he designed have been adopted by the WiMAX Standard. He has been a Chief Investigator for a number of sponsored and consultancy projects. He has authored the book entitled *Single and Double Symbol Decodable Space-Time Block Codes* (Germany: Lambert Academic). His research interests include coded modulation, space-time coding, and signal processing for wireless communications. He serves as a Reviewer for many international and national journals and conferences. He received the INAE Young Engineer Award in 2006.

Lajos Hanzo (F'–) received the degree in electronics in 1976, the Ph.D. degree in 1983, and the Honorary Doctorate degree from the Technical University of Budapest, in 2009, while by the University of Edinburgh, in 2015. During his 38-year career in telecommunications, he has held various research and academic positions in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the Chair in Telecommunications. He has successfully supervised about 100 Ph.D. students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10000 pages, published over 1500 research entries at the IEEE Xplore, acted both as a TPC and General Chair of the IEEE conferences, presented keynote lectures, and has received a number of distinctions. He directs a 60-strong academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by the industry, the Engineering and Physical Sciences Research Council, U.K., the European Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. He is an Enthusiastic Supporter of industrial and academic liaison and he offers a range of industrial courses. He is a fellow of REng, IET, and EURASIP. He is also a Governor of the IEEE VTS. From 2008 to 2012, he was the Editor-in-Chief of the IEEE PRESS and a Chaired Professor with Tsinghua University, Beijing. His research is funded by the European Research Council's Senior Research Fellow Grant. He has 24000 citations. AQ:8 AQ:9

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

AQ:1 = Please be advised that per instructions from the Communications Society this proof was formatted in Times Roman font and therefore some of the fonts will appear different from the fonts in your originally submitted manuscript. For instance, the math calligraphy font may appear different due to usage of the usepackage[mathcal]euscript. We are no longer permitted to use Computer Modern fonts.

AQ:2 = Please confirm whether the financial section retained as in the metadata is OK.

AQ:3 = Note that if you require corrections/changes to tables or figures, you must supply the revised files, as these items are not edited for you.

AQ:4 = Please confirm the volume no. for refs. [12], [18], and [24].

AQ:5 = Please confirm the conference title, month, and year for ref. [17]. Also provide the page range.

AQ:6 = Please confirm the author names, article title, conference title, month, and year for ref. [27]. Also provide the page range.

AQ:7 = Current affiliation in biography of Kalpana Naidu does not match First Footnote. Please check.

AQ:8 = Please confirm whether the edits made in the sentence “Lajos Hanzo received . . . Edinburgh in 2015” are OK.

AQ:9 = Please provide the membership year for the author “Lajos Hanzo.”

An Efficient Direct Solution of Cave-Filling Problems

Kalpana Naidu, *Student Member, IEEE*, Mohammed Zafar Ali Khan, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

AQ:1

***Abstract*— Waterfilling problems subjected to peak power constraints are solved, which are known as cave-filling problems (CFP). The proposed algorithm finds both the optimum number of positive powers and the number of resources that are assigned the peak power before finding the specific powers to be assigned. The proposed solution is non-iterative and results in a computational complexity, which is of the order of M, $O(M)$, where M is the total number of resources, which is significantly lower than that of the existing algorithms given by an order of M^2, $O(M^2)$, under the same memory requirement and sorted parameters. The algorithm is then generalized both to weighted CFP (WCFP) and WCFP requiring the minimum power. These extensions also result in a computational complexity of the order of M, $O(M)$. Finally, simulation results corroborating the analysis are presented.**

***Index Terms*— Weighted waterfilling problem, Peak power constraint, less number of flops, sum-power constraint, cave waterfilling.**

I. INTRODUCTION

WATERFILLING Problems (WFP) are encountered in numerous communication systems, wherein specifically selected powers are allotted to the resources of the transmitter by maximizing the throughput under a total sum power constraint. Explicitly, the classic WFP can be visualized as filling a water tank with water, where the bottom of the tank has stairs whose levels are proportional to the channel quality, as exemplified by the Signal-to-Interference Ratio (SIR) of the Orthogonal Frequency Division Multiplexing (OFDM) sub-carriers [1], [2].

This paper deals with the WaterFilling Problem under Peak Power Constraints (WFPPPC) for the individual resources. In contrast to the classic WFP where the 'tank' has a 'flat lid', in WFPPPC the 'tank' has a 'staircase shaped lid', where the steps are proportional to the individual peak power constraint. This scenario is also metaphorically associated with a 'cave' where the stair-case shaped ceiling represents the peak power that can be assigned, thus fulfilling all the requirements of WFPPPC. Thus WFPPPC is often referred to as a 'Cave-Filling Problem' (CFP) [3], [4].

In what follows, we will use the 'cave-filling' metaphor to develop insights for solving the WFPPPC. Again, the user's resources can be the sub-carriers in OFDM or the tones in a Digital Subscriber Loop (DSL) system, or alternatively the same sub-carriers of distinct time slots [5].

More broadly, the CFP occurs in various disciplines of communication theory. A few instances of these are:

a) protecting the primary user (PU) in Cognitive Radio (CR) networks [6]–[9];
b) when reducing the Peak-to-Average-Power Ratio (PAPR) in Multi-Input-Multi-Output (MIMO)-OFDM systems [10], [11];
c) when limiting the crosstalk in Discrete Multi-Tone (DMT) based DSL systems [12]–[14];
d) in energy harvesting aided sensors; and
e) when reducing the interference imposed on nearby sensor nodes [15]–[17].

Hence the efficient solution of CFP has received some attention in the literature, which can be classified into iterative and exact direct computation based algorithms.

Iterative algorithms conceived for CFP have been considered in [18]–[20], which may exhibit poor accuracy, unless the initial values are carefully selected. Furthermore, they may require an extremely high number of iterations for their accurate convergence.

Exact direct computation based algorithms like the Fast WaterFilling (FWF) algorithm of [21], the Geometric WaterFilling with Peak Power (GWFPP) constraint based algorithm of [22] and the Cave-Filling Algorithm (CFA) obtained by minimizing Minimum Mean-Square Error (MMSE) of channel estimation in [3] solve CFPs within limited number of steps, but impose a complexity on the order of $O(M^2)$.

All the existing algorithms solve the CFPs by evaluating the required powers multiple times, whereas the proposed algorithm directly finds the required powers in a single step. Explicitly, the proposed algorithm reduces the number of Floating point operations (flops) by first finding the number of positive powers to be assigned, namely K, and the number of powers set to the maximum possible value, which is denoted by L. This is achieved in two (waterfilling) steps. First we use 'coarse' waterfilling to find the number of positive powers to

AQ:2

Manuscript received November 30, 2015; revised March 14, 2016; accepted April 24, 2016. This work was supported in part by the Engineering and Physical Sciences Research Council EP/Noo4558/1 and EP/L018659/1, in part by the European Research Council advanced fellow grant under Beam-me-up, and in part by the Royal Society under Wolfson research merit award. The associate editor coordinating the review of this paper and approving it for publication was M. Tao.

K. Naidu and M. Z. Ali Khan are with the Department of Electrical Engineering, IIT Hyderabad, Hyderabad 502205, India (e-mail: ee10p002@iith.ac.in; zafar@iith.ac.in).

L. Hanzo is with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: lh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2016.2560813

be assigned and then we embark on step-by-step waterfilling to find the number of positive powers that have to be set to the affordable peak powers.

In this paper we present an algorithm designed for the efficient solution of CFPs. The proposed solution is then generalized for **conceiving** both a Weighted CFP (WCFP) and a WCFP having both a Minimum and a Maximum Power (WCFP-MMP) constraint. It is demonstrated that the maximum throughput is achieved at a complexity order of $O(M)$ by all the three algorithms proposed.

The outline of the paper is as follows. Section II outlines our system model and develops the algorithms for solving the CFP. In Section III we conceive the WCFP, while Section IV presents our WCFP-MMP. Our simulation results are provided in Section V, while Section VI concludes the paper.

II. The Cave-Filling Problem

In Subsection II-A, we introduce the CFP. The computation of the number of positive powers is presented in Subsection II-B, while that of the number of powers set to the maximum is presented in Subsection II-C. Finally, the computational complexity is evaluated in Subsection II-D.

A. The CFP

The CFP maximizes the attainable throughput, C, while satisfying the sum power constraint; Hence, the sum of powers allocated is within the prescribed power budget, P_t, while the power, $P_i, \forall i$ assigned for the i^{th} resource is less than the peak power, P_{it}, $\forall i$. Our optimization problem is then formulated as:

$$\begin{aligned} \max_{\{P_i\}_{i=1}^{M}} \; & C = \sum_{i=1}^{M} \log_2\left(1 + \frac{P_i}{N_i}\right) \\ \text{subject to :} \; & \sum_{i=1}^{M} P_i \leq P_t; \\ & P_i \leq P_{it}, \quad i \leq M, \\ & \text{and } P_i \geq 0, \quad i \leq M, \end{aligned} \tag{1}$$

where M is the total number of resources (such as OFDM sub-carriers) and $\{N_i\}_{i=1}^{M}$ is the sequence of interference plus noise samples. The above optimization problem occurs in the following scenarios:

(a) In the downlink of a wireless communication system, where the base station (BS) assigns a resource (e.g. frequency band) to a user and allocates a certain power, P_i, to the i^{th} resource while obeying the total power budget (P_t). The BS ensures that $P_i \leq P_{it}$ for avoiding the near-far problem [23].

(b) In an OFDM system, a transmitter assigns specific powers to the resources (e.g. sub-carriers) for satisfying the total power budget, P_t. Furthermore, to reduce the PAPR problem, the maximum powers assigned are limited to be within the peak powers [24], [25].

Theorem 1: The solution of the CFP (1) *is of the 'form'*

$$P_i = \begin{cases} \left(\frac{1}{\lambda} - N_i\right), & 0 < P_i < P_{it}; \\ P_{it}, & \frac{1}{\lambda} \geq H_i \triangleq (P_{it} + N_i); \\ 0, & \frac{1}{\lambda} \leq N_i \end{cases} \tag{2}$$

where "$\frac{1}{\lambda}$ is the water level of the CFP".

Proof: The proof is in Appendix VI-A. □

Remark 1: Note that as in the case of conventional water-filling, the solution of CFP is of the form (2)*. The actual solution is obtained by solving the solution form along with the primal feasibility constraints. Furthermore, for the set of primal feasibility constraints of our CFP, the Peak Power Constraint of $P_i \leq P_{it}, \forall i$ is incorporated in the solution form. By contrast, the sum power constraint is considered along with* (2) *to obtain the solution in Propositions 1 and 2.*

Remark 2: Observe from (2) *that for $0 < P_i < P_{it}$, $P_i = (\frac{1}{\lambda} - N_i)$ which allows $\frac{1}{\lambda}$ to be interpreted as the 'water level'. However, in contrast to conventional water-filling, the 'water level' is upper bounded by* $\max_i P_{it}$*. Beyond this value, no 'extra' power can be allocated and the 'water level' cannot increase. The solution of this case is considered in Proposition 1.*

It follows that (2) *has a nice physical interpretation, namely that if the 'water level' is below the noise level N_i, no power is allocated. When the 'water level' is between N_i and P_{it}, the difference of the 'water level' and the noise level is allocated. Finally, when the 'water level' is higher than the 'peak level', H_i; the peak power P_{it} is allocated.*

The above solution 'form' can be rewritten as

$$P_i = \left(\frac{1}{\lambda} - N_i\right)^{+}, \quad i = 1, \cdots, M; \quad \textit{and} \tag{3}$$

$$P_i \leq P_{it}, \quad i = 1, \cdots, M \tag{4}$$

where we have $A^{+} \triangleq \max(A, 0)$. The solution for (1) has a simple form for the case the 'implied' power budget, P_{It} as defined as $P_{It} = \sum_{i=1}^{M} P_{it}$ is less than or equal to P_t and is given in Proposition 1.

Proposition 1: If the 'implied' power budget is less than or equal to the power budget ($\sum_{i=1}^{M} P_{it} \leq P_t$), then peak power allocation to all the M resources gives optimal capacity.

Proof: Taking summation on both sides of $P_i \leq P_{it}, \forall i$, we obtain the 'implied' power constraint

$$\sum_{i=1}^{M} P_i \leq \underbrace{\sum_{i=1}^{M} P_{it}}_{P_{IT}} . \tag{5}$$

However from (1) we have

$$\sum_{i=1}^{M} P_i \leq P_t. \tag{6}$$

Consequently, if $P_{It} \leq P_t$, then peak power allocation to all the M resources (i.e. $P_i = P_{it}$, $\forall i$) fulfils all the constraints of (1). Consequently, the total power allocated to M resources $\sum_{i=1}^{M} P_{it}$. Since the maximum power that can be allocated to any resource is it's peak power, peak power allocation to all the M resources produces optimal capacity. □

Note that in this case the total power allocated is less than (or equal to) P_t. However, if $P_t < \sum_{i=1}^{M} P_{it}$, then all the M resources cannot be allocated peak powers since it violates the total sum power constraint in (1).

In what follows, we pursue the solution of (1) for the case

$$P_t < \sum_{i=1}^{M} P_{it}. \tag{7}$$

We have,

Proposition 2: The optimal powers and hence optimal capacities are achieved in (1) *(under the assumption* (7)*) only if*

$$\sum_{i=1}^{M} P_i = P_t. \tag{8}$$

Proof: The proof is in Appendix VI-B. □

Since finding both the number of positive powers and the number of powers that are set to the maximum is crucial for solving the CFP, we formally introduce the following definitions.

Definition 1 (The Number of Positive Powers, K): Let $\mathcal{I} = \{i;$ such that $P_i > 0\}$ be the set of resource indices, where P_i is positive. Then the number of positive powers, $K = |\mathcal{I}|$, is given by the cardinality, $|\mathcal{I}|$, of the set.

Definition 2 (The Number of Powers Set to the Peak Power, L): Let $\mathcal{I_P} = \{i;$ such that $P_i = P_{it}\}$ be the set of resource indices, where P_i has the maximum affordable value of P_{it}. Then the number of powers set to the peak power, $L = |\mathcal{I_P}|$, is the cardinality, $|\mathcal{I_P}|$ of the set.

Without loss of generality, we assume that the interference plus noise samples N_i are sorted in ascending order, so that the first K powers are positive, while the remaining ones are set to zero. Then, (8) becomes

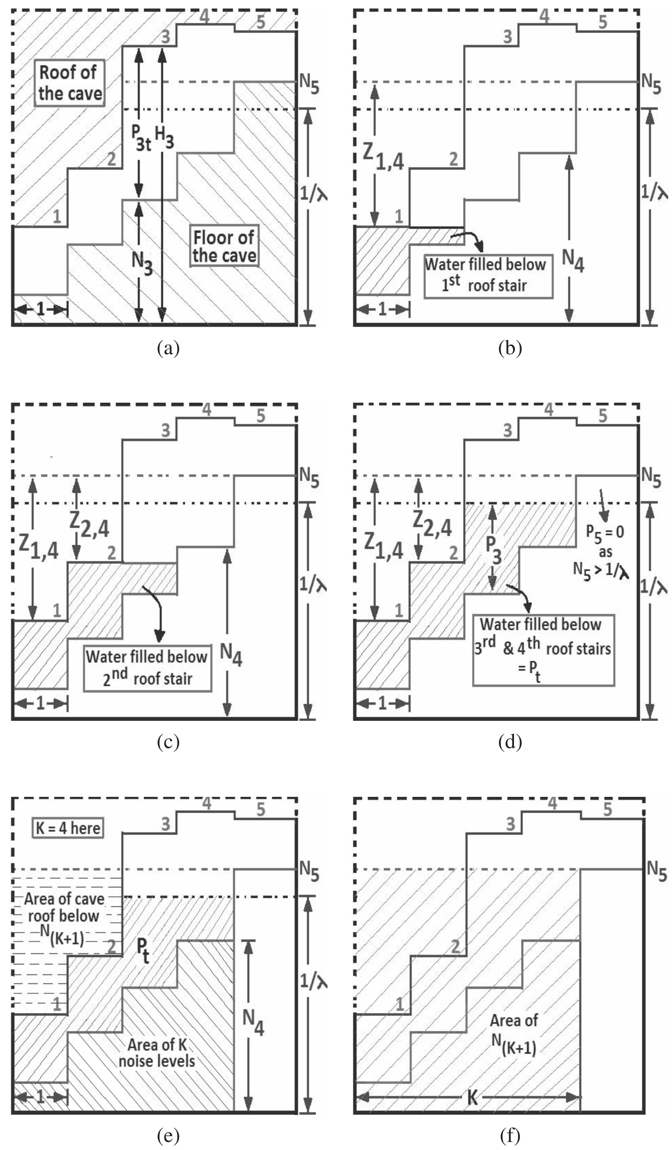
$$\sum_{i=1}^{K} P_i = P_t. \tag{9}$$

Note that H_i and P_{it} are also arranged in the ascending order of N_i, in order to preserve the original relationship between H_i and N_i.

B. Computation of the Number of Positive Powers

The CFP can be visualized as shown in Fig. 1a. In a cave, the water is filled i.e. the power is apportioned between the floor of the cave and the ceiling of the cave. The levels of the i^{th} 'stair' of the floor staircase and of the ceiling staircase are N_i and $H_i \triangleq (P_{it} + N_i)$, respectively. The widths of all stairs are assumed to be 1. Since the power gap between the floor stair and the ceiling stair is P_{it}, the allocated power has to satisfy $P_i \leq P_{it}$.

As the water is poured into the cave, observe from Fig. 1b that it obeys the classic waterfilling upto the point where the 'waterlevel' $(\frac{1}{\lambda})$ reaches the ceiling stair of the 1^{st} resource. From this point onwards, water can only be stored above the second stair, as depicted in Fig. 1c upto a point where the water has filled the gap between the floor stair and the ceiling stair of both the first and the second stairs. In terms of power, we have $P_i = P_{it}$ for the resources $i = 1$ and 2. Mathematically, we have $P_i = P_{it}$ if $H_i \leq \frac{1}{\lambda}$.



AQ:3

Fig. 1. Geometric Interpretation of CFP for $K = 4$. (a) Heights of i^{th} stair in cave floor staircase and cave roof staircase are N_i and $H_i (= P_{it} + N_i)$. (b) Water is filled (Power is allotted) in between the cave roof stair and cave floor stair, at this waterlevel the peak power constraint for P_1 constraints further allocation to P_1. (c) A similar issue occurs to P_2 also.Observe that the variable $Z_{m,4}$ represents the height of m^{th} cave roof stair below the $(4+1)^{th}$ cave floor stair. (d) Power allotted for i^{th} resource is $P_i = min\{\frac{1}{\lambda}, H_i\} - N_i$. Observe the waterlevel between 4^{th} and 5^{th} resource. (e) The area $\frac{1}{\lambda}K$, shown in this figure, is smaller than the area $N_{K+1}K$ shown in (f).

As more water is poured, observe from Fig. 1d that for the third and the fourth stairs, we have $H_i > \frac{1}{\lambda}$. It is clear from the above observations (also from (2)) that the power assigned to the i^{th} resource becomes:

$$P_i = min\left\{\frac{1}{\lambda}, H_i\right\} - N_i, \quad i \leq K. \tag{10}$$

In Fig. 1d, the height of the fifth floor stair exceeds $\frac{1}{\lambda}$. As water can only be filled below the level $\frac{1}{\lambda}$, no water is

Algorithm 1 ACF Algorithm for Obtaining K

Require: Inputs required are M, P_t, N_i & H_i (in ascending order of N_i).

Ensure: Output is K, $I_{R_{K-1}}$,I_{R_K}, d_K.

1: $i = 1$. Denote $d_0 = P_t$, $U_0 = 0$ and $I_{R_0} = \emptyset$
2: Calculate $d_i = d_{i-1} + N_i$.
3: $\triangleright$ Calculate the area $U_i = \sum_{m=1}^{i} Z_{m,i}^{+}$ as follows:
4: $I_{R_i} = I_{R_{i-1}} \cup \{m;$ *such that* $N_{i+1} > H_m$ & $m \notin I_{R_{i-1}}\}$;
$Z_{m,i} = N_{(i+1)} - H_m, m \in (I_{R_i} - I_{R_{i-1}})$
5: $U_i = U_{i-1} + |I_{R_{i-1}}|(N_{i+1} - N_i) + \sum_{m\in(I_{R_i} - I_{R_{i-1}})} Z_{m,i}^{+}$
6: Calculate the area $Q_i = i\,N_{(i+1)}$
7: **if** $Q_i \geq (d_i + U_i)$ **then**
8: $K \leftarrow i$. Exit the algorithm.
9: **else**
10: $i \leftarrow$ i+1, Go to 2
11: **end if**

Algorithm 2 'Step-Based' Waterfilling Algorithm for Obtaining L

Require: Inputs required are K, d_K, $I_{R_{K-1}}$, I_{R_K}, N_i & H_i (in ascending order of N_i)

Ensure: Output is L, I_S.

1: Calculate $P_R = d_K - KN_K + |I_{R_{K-1}}|N_K - \sum_{m\in I_{R_{K-1}}} H_m$
2: Calculate $I_B = I_{R_K} - I_{R_{K-1}}$ & $D_1 = K - |I_{R_{K-1}}|$.
3: If $|I_B| = 0$, set $L = 0$, $I_S = \emptyset$. Exit the algorithm.
4: Sort $\{H_m\}_{m\in I_B}$ in ascending order and denote it as $\{H_{mB}\}$ and the sorting index as I_S.
5: Initialize $m = 1$, $F_m = (H_{mB} - N_K)D_m$.
6: **while** $F_m < P_R$ **do**
7: $m = m + 1$.
8: $D_m = D_{m-1} - 1$
9: $F_m = F_{m-1} + (H_{mB} - H_{(m-1)B})D_m$
10: **end while**
11: $L = m - 1$.

filled above the fifth bottom stair. This results in $K = 4$, as shown in Fig. 1d. The area of the water-filled cave cross-section becomes equal to P_t.

Fig. 1c also introduces the variable $Z_{i,k}$ as the depth of the i^{th} ceiling stair below the $(k+1)^{st}$ bottom stair; that is, we have:

$$Z_{i,k} = N_{(k+1)} - H_i, \quad i \leq k. \tag{11}$$

The variable $Z_{i,k}$ allows us to have a reference, namely a constant roof ceiling of N_{i+1}, while verifying whether $K = i$. Figure 1c depicts this dynamic for $i = 4$. The constant roof reference is given at N_{i+1}. Observe that we have $Z_{i,k}^{+} > 0$ for $i = 1, 2$ and $Z_{i,k}^{+} = 0$ for $i = 3, 4$ with $k = 4$. This allows us to quantify the total cave cross-section area in Fig 1e, upto the i^{th} step in three parts:

- the area occupied by roof stairs below the constant roof reference, given by $\sum_{k=1}^{i} Z_{k,i}^{+}$;
- the area occupied by the 'water', given by P_t;
- the area occupied by the floor stairs, $\sum_{k=1}^{i} N_k$.

This is depicted in Fig. 1e. Observe from Fig. 1e that if the waterlevel of $\frac{1}{\lambda}$ is less than the $(i+1)^{st}$ water level ($i + 1 = 5$ in this case), then the cave cross-section area given by $\sum_{k=1}^{i} Z_{k,i}^{+} + P_t + \sum_{k=1}^{i} N_k$ (shown in Fig. 1e) would be less than the total area of $i\,N_{i+1}$, as shown in Fig. 1f. Furthermore, if the waterlevel $\frac{1}{\lambda}$ is higher than the $(i+1)^{st}$ water level ($i + 1 = 2, 3, 4$ in this case), then the area given by $\sum_{k=1}^{i} Z_{k,i}^{+} + P_t + \sum_{k=1}^{i} N_k$ would be higher than the total area of $i\,N_{i+1}$, as shown in Fig. 1f.

Based on the insight gained from the above geometric interpretation of the CFP, we develop an algorithm for finding K for any arbitrary CFP, which we refer to as the **Area based Cave-Filling (ACF)** of Algorithm 1.

Note that d_0 in Algorithm 1 represents an initialization step that eliminates the need for the addition of P_t at every resource-index i and the set I_{R_i} contains the indices of the ceiling steps, whose 'height' is below N_{i+1}. Furthermore, the additional outputs of Algorithm 1 are required for finding the number of roof stairs that are below the waterlevel in Algorithm 2. We now prove that Algorithm 1 indeed finds the optimal value of K.

Theorem 2: The Algorithm 1 delivers the optimal value of the number of positive powers, K, as defined in Definition 1.

Proof: We prove Theorem 2 by first proving that $\phi(i) = d_i + U_i$, is a monotonically increasing function of the resource-index i. It then follows that $Q_i \geq (d_i + U_i)$ gives the first i, for which the waterlevel is below the next step. Consider

$$\begin{aligned}
&\phi(i) - \phi(i-1) \\
&\quad = d_i - d_{i-1} + U_i - U_{i-1} && (12)\\
&\quad = N_i + |I_{R_{i-1}}|\,(N_{i+1} - N_i) + \sum_{m\in(I_{R_i} - I_{R_{i-1}})}^{i} Z_{m,i}^{+} && (13)\\
&\quad > 0, && (14)
\end{aligned}$$

where (13) follows from (12) by using the definitions of d_i and U_i in Algorithm 1. Since the interference plus noise levels N_i are positive, we have $(N_{i+1} - N_i) \geq 0$, and since the N_i's are in ascending order, (14) follows from (13).

Let us now consider the reference area, $Q_i = i\,N_{i+1}$. Within this reference area; certain parts are occupied by the floor stairs, others by the projections of the ceiling stairs and finally by the space in between the floor and the ceiling; filled by 'water'. This is given by $W_i = Q_i - \sum_{m}^{i} N_m - U_i$. Recall that the total amount of water that can be stored is P_t. If we have $P_t > W_i$, then there is more water than the space available, hence the water will overflow to the next stair(s). Otherwise, if we have $P_t \leq W_i$, all the water can be contained within the space above this stair and the lower stairs. Substituting the value of W_i in this inequality, we have

$$P_t \leq Q_i - \sum_{m}^{i} N_m - U_i \tag{15}$$

$$\begin{aligned}
\Rightarrow P_t + \sum_{m}^{i} N_m + U_i &\leq Q_i && (16)\\
d_i + U_i &\leq Q_i && (17)
\end{aligned}$$

where (16) is obtained from (15) by rearranging. Then using the definition of d_i in Algorithm 1, we arrive at (17).

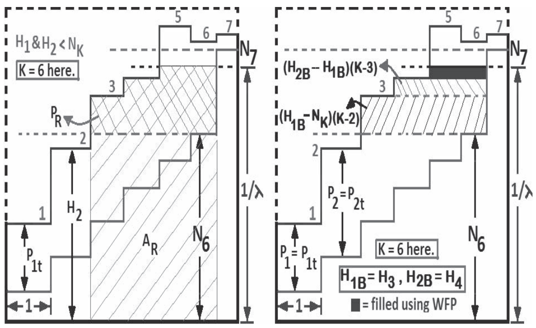


Fig. 2. Peak power allocation for resources having their H_i's in between N_K and $N_{(K+1)}$.

Since Algorithm 1 outputs the (first) smallest value of the resource-index i for which (17) is satisfied, it represents the optimal value of K.

This completes the proof. □

Once K is obtained, it might appear straightforward to obtain the values of P_i, $i \in [1, K]$[‡] as in [26] and [27]; but in reality it is not. This is because of the need to find the specific part of the cave roof, which is below the 'current' waterlevel.

Note that $I_{R_{K-1}} \subset I_P \subset I_{R_K}$ where I_P is the set of roof stairs below the current waterlevel and I_{R_K} is the set of roof stairs below N_{K+1}. This is because the waterlevel of $\frac{1}{\lambda}$ is between N_K and N_{K+1}.

C. Waterfilling for Finding the Number of Powers Having the Peak Allocation

In order to develop an algorithm for finding L, we first consider the geometric interpretation of an example shown in Fig. 2. Note that the H_m's below N_K, $(N_K - H_m) > 0$, belong to $I_{R_{K-1}}$ and the H_m values above N_{K+1} belong to I_{U_K}. This is clearly depicted in Fig. 2 for $K = 6$, where $I_{R_{K-1}} = \{1, 2\}$ and $I_{U_K} = \{5, 6\}$.

The contentious H_m's are those whose heights lie between N_K and N_{K+1}. The indices of these H_m's are denoted by I_B (in Fig. 2, $I_B = \{3, 4\}$). Without loss of generality, we assume that B roof stairs, H_m's, lie between N_K and N_{K+1}. We now have to find among these B stairs, those particular ones whose heights lie below the water level, $\frac{1}{\lambda}$ (for which peak powers are allotted). Note that $B = |I_{R_K}| - |I_{R_{K-1}}|$ and $I_B = [1, K] - I_{R_{K-1}} - I_{U_K} = I_{R_K} - I_{R_{K-1}}$.

This is achieved by a 'second' waterfilling style technique as detailed below.

Clearly, the resources that belong to the set $I_{R_{K-1}}$ are allotted with peak powers as $(H_m - \frac{1}{\lambda}) < 0, m \in I_{R_{K-1}}$. The remaining ceiling stairs in I_B will submerge one by one as the waterlevel increases from N_K. For this reason; the heights $\{H_m\}_{m \in I_B}$ are sorted in ascending order to obtain H_{mB} and I_S is the sort index for H_{mB}.

After allotting $I_{R_{K-1}}$ resources with peak powers, whose sum is equal to $\sum_{m \in I_{R_{K-1}}} P_{mt}$, we can allocate $(N_K - N_m)^+, m \in I^c_{R_{K-1}}$ power to the remaining resources indexed by $I^c_{R_{K-1}}$, where for a set A, $A^c = [1, K] - A$ represents its complement. That is we allot power to remaining resources with the 'present' waterlevel being N_K. The power that remains to be allocated for $I^c_{R_{K-1}}$ resources is given by

$$P_R = P_t - \sum_{m \in I_{R_{K-1}}} P_{mt} - \sum_{m \in I^c_{R_{K-1}}} (N_K - N_m)^+ \quad (18)$$

$$= P_t + \sum_{m=1}^{K} N_m - K N_K + |I_{R_{K-1}}| N_K - \sum_{m \in I_{R_{K-1}}} H_m. \quad (19)$$

Equation (19) is obtained using a geometric interpretation as follows; the term $d_K = P_t + \sum_{m=1}^{K} N_m$ is the sum of total water and K floor stairs. Subtracting from it the reference area of $K N_K$ gives the excess water that is in excess amount; without considering the ceiling stairs. Further subtracting the specific part of the ceiling stairs that are below N_K namely $\sum_{m \in I_{R_{K-1}}} H_m - |I_{R_{K-1}}| N_K$ gives the residual 'water' amount, P_R.

Note from Fig. 2 that once P_R amount of 'water' has been poured, and provided that $P_R < (K - |I_{R_{K-1}}|)(H_{1B} - N_K)$ is satisfied, then we have $L = |I_{R_{K-1}}|$ and hence no more 'water' is left to be poured. Otherwise, $F_1 = (K - |I_{R_{K-1}}|)(H_{1B} - N_K)$ amount of 'water' is used for completely submerging the 1^{st} ceiling stair (H_{1B}) and the 'present' waterlevel increases to H_{1B}. Similarly, $F_2 = (K - |I_{R_{K-1}}| - 1)(H_{2B} - H_{1B})$ amount of water is used for submerging the second ceiling stair and hence the waterlevel increases to H_{2B}. This process continues until all the 'water' has been poured. We refer to this process as 'step-based' waterfilling since the waterlevel is changed in steps given by the size of the roof stairs.

The formal algorithm, which follows the above geometric interpretation but it aims for a low complexity, is given in Algorithm 2. Let us now prove that Algorithm 2 delivers the optimal value of L.

Theorem 3: Algorithm 2 finds the optimal value L of the number of powers that are assigned peak powers, where L is defined in Definition 2 .

Proof: First observe that the F_m values are monotonically increasing functions of the index m. Since the H_{mB} values are sorted in ascending order, the water filling commences from $m = 1$. The condition $F_m < P_R$ is true, as long as the total amount of water required to submerge the m^{th} roof stair, F_m, is less than the available water. It follows then that the algorithm outputs the largest m, for which the inequality is satisfied which hence represents the optimal value of L. □

The resources for which peak powers are allotted are indexed by $I_P = I_{R_{K-1}} \cup I_S(1 : L)$, where $I_S(1 : L)$ stands for the first 'L' resources of I_S. The remaining resources, indexed by $I^c_P = [1, K] - I_P$, are allotted specific powers using waterfilling.

In Fig. 2, the I^c_P resources are 5 and 6 with associated 'L' = 2 while $P_R - F_L$ represents the darkened area in Fig. 2. The waterlevel for I^c_P resources is equal to the height, H_{LB}, of the last submerged roof stair plus the height of the darkened area. Here, the height of the darkened area is obtained by dividing the remaining water amount ($= P_R - F_L$) with the

‡[A,B] represents the interval in between A and B, including A and B.

TABLE I
COMPUTATIONAL COMPLEXITIES (IN FLOPS) OF KNOWN SOLUTIONS FOR SOLVING CFP

Iterative Algorithms [18], [19]	**FWF [21]**	**GWFPP [22]**	**ACF**
iterations $\times(6M)$	iterations $\times(5M+6)$	$4M^2+7M$	*16M+9*

number of remaining resources ($= |I_P^c|$) since the width of all resources is 1. If we have $L = 0$, then the last level is N_K. Therefore the waterlevel for I_P^c resources is given by

$$\frac{1}{\lambda} = \begin{cases} H_{LB} + \dfrac{P_R - F_L}{|I_P^c|}, & L > 0; \\ N_K + \dfrac{P_R}{|I_P^c|}, & \text{otherwise.} \end{cases} \tag{20}$$

The powers are then allotted as follows:

$$P_i = \begin{cases} P_{it}, & i \in I_P; \\ \left(\dfrac{1}{\lambda} - N_i\right), & i \in I_P^c. \end{cases} \tag{21}$$

D. Computational Complexity of the CFP

Let us now calculate the computational complexity of both Algorithm 1 as well as of Algorithm 2 separately and then add the complexity of calculating the powers, as follows:

- Calculating H_i requires M adds.
- Observe that Algorithm 1 requires $K + 1$ adds for calculating d_i's; K multiplies to find Q_i's; *maximum of K subtractions for calculating $Z_{m,i}$'s* and, in the worst case, $4K$ additions as well as K multiplications for calculating U_K: the proof*s* are given in Appendi*ces C and* D. So, algorithm 1 requires *6K* + 1 additions and $2K$ multiplications for calculating K.
- Note that in Algorithm 2: 2 multiplies and $3 + |I_{R_{K-1}}|$ additions are needed for the calculation of P_R; 2 adds and 1 multiply for computing F_1, D_1; $4|I_B|$ adds and I_B multiples for evaluating the while loop. Since we have $|I_{R_{K-1}}|, |I_B| < K$, the worst case complexity of Algorithm 2 is given by $5K + 5$ adds and $K + 3$ multiplies.
- The computational complexity of calculating P_i using (3) is at-most K adds.
- The total computational complexity of solving our CFP of this paper, is *12K+6+M* adds and $3K+3$ multiplies. Since K is not known apriori, the worst case complexity is given by *13M+6* adds and $3M+3$ multiplies. Hence we have a complexity order of $O(M)$ floating point operations (flops).

Table I gives the number of flops required for iterative algorithm of [18] and [19], FWF of [21], GWFPP algorithm of [22] and of the proposed ACF algorithm. Observe the order of magnitude improvement for ACF.

Remark 3: Following the existing algorithms conceived for solving the CFP (like [2] and [22]), we do not consider the complexity of sorting N_i, as the channel gain sequences come from the eigenvalues of a matrix; and most of the algorithms compute the eigenvalues and eigenvectors in sorted order.

Remark 4: Observe that we have not included the complexity of sorting H_i at step 4 in Algorithm 2. This is because the sorting is implementation dependent. For fixed-point implementations, sorting can be performed with a worst case complexity of $O(M)$ comparisons using algorithms like Count Sort [28]. For floating point implementations, sorting can be performed with a worst case complexity of $O(M\log(M))$ comparisons [29]. Since, almost all implementations are of fixed-point representation: the overall complexity, including sorting of H_i would still be of $O(M)$.

III. WEIGHTED CFP

An interesting generalization for CFP is the scenario when the rates and the sum power are weighted, hence resulting in the Weighted CFP (WCFP), arising in the following context.

(a) In a CR network, a CR senses that some resources are available for it's use. Hence the CR allots powers to the available resources for a predefined amount of time while assuring that the peak power remains limited in order to keep the interference imposed on the PU remains within the limit. The weights w_i and x_i may be adjusted based on the resource's available time and on the sensing probabilities [30]–[32].

(b) In Sensor Network (SN) the resources have priorities according to their capability to transfer data. These priorities are reflected in the weights, w_i. The weights x_i's allow the sensor nodes to save energy, while avoiding interference with the other sensor nodes [33], [34].

The optimization problem constituted by weighted CFP is given by

$$\begin{aligned} \max_{\{P_i\}_{i=1}^M} \; & C = \sum_{i=1}^{M} w_i \log_2\left(1 + \frac{P_i}{N_i}\right) \\ \text{subject to :} \; & \sum_{i=1}^{M} x_i P_i \le P_t \\ & P_i \le P_{it}, \quad i \le M \\ & \text{and } P_i \ge 0, \quad i \le M, \end{aligned} \tag{22}$$

where again w_i and x_i are the weights of the i^{th} resource's capacity and allocated power, respectively. Similar to Theorem 1, we have

Theorem 4: The solution of the WCFP (22) *is of the 'form'*

$$\bar{P}_i = \begin{cases} \left(\dfrac{1}{\lambda} - \bar{N}_i\right), & 0 < \bar{P}_i < \bar{P}_{it}; \\ \bar{P}_{it}, & \dfrac{1}{\lambda} \ge \bar{H}_i \triangleq \left(\bar{P}_{it} + \bar{N}_i\right); \\ 0, & \dfrac{1}{\lambda} \le \bar{N}_i \end{cases} \tag{23}$$

where "$\frac{1}{\lambda}$ is the water level of the WCFP", $\bar{P}_i = \frac{P_i x_i}{w_i}$ is the weighted power, $\bar{P}_{it} = \frac{P_{it} x_i}{w_i}$ is weighted peak power, $\bar{N}_i = \frac{N_i x_i}{w_i}$ is the weighted interference plus noise level and $\bar{H}_i = \bar{N}_i + \bar{P}_{it}$ is the weighted height of i^{th} cave ceiling stair.

Proof: The proof is similar to Theorem 1 and has been omitted. □

The above solution *form* can be rewritten as

$$\bar{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+, \quad i = 1, \cdots, M; \quad and \tag{24}$$

$$\bar{P}_i \le \bar{P}_{it}, \quad i = 1, \cdots, M \tag{25}$$

where we have $A^+ \triangleq \max(A, 0)$. The solution for (22) has a simple form for the case the 'implied' weighted power budget, $\bar{P}_{It}$ as defined as $\bar{P}_{It} = \sum_{i=1}^{M} w_i \bar{P}_{it}$ is less than or equal to P_t and is given in Proposition 3.

Proposition 3: If the 'implied' power budget is less than or equal to the power budget ($\sum_{i=1}^{M} w_i \bar{P}_{it} \le P_t$), then peak power allocation to all the M resources gives optimal capacity.

Note that in this case the total power allocated is less than (or equal to) P_t. However, if $P_t < \sum_{i=1}^{M} w_i \bar{P}_{it}$, then all the M resources cannot be allocated peak powers since it violates the total sum power constraint in (22).

In what follows, we pursue the solution of (22) for the case

$$P_t < \sum_{i=1}^{M} w_i \bar{P}_{it}. \tag{26}$$

We have,

Proposition 4: The optimal powers and hence optimal capacities are achieved in (22) *(under the constraint* (26)*) only if*

$$\sum_{i=1}^{M} w_i \bar{P}_i = P_t. \tag{27}$$

It follows that the solution of (22) is given by

$$\bar{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+, \quad i = 1, \cdots, M; \tag{28}$$

$$\sum_{i=1}^{K} w_i \bar{P}_i = P_t; \tag{29}$$

$$\bar{P}_i \le \bar{P}_{it}, \quad i = 1, \cdots, M. \tag{30}$$

Using the proposed area based approach, we can extend the ACF algorithm to the weighted case as shown in Fig. 3.

Observe that the width of the stairs is now given by w_i in contrast to CFP, and $Z_{i,k}$ is now scaled by a factor of $\frac{x_i}{w_i}$.

Also observe that the sorting order now depends on the $\bar{N}_i$ values, since sorting the $\bar{N}_i$ values in ascending order makes the first K of the $\bar{P}_i$ values positive, while the remaining $\bar{P}_i$ values are equal to zero as per (28).

In what follows, we assume that the parameters like $\bar{H}_i$, $\bar{P}_{it}$, w_i and $\bar{N}_i$ are sorted in the ascending order of $\bar{N}_i$ values in order to conserve the original relationship among parameters.

Comparing (28)-(30) to (3), (4) and (9); we can see that in addition to the scaling of the variables, (29) has a weighing factor of w_i. Most importantly, since the widths of the stairs is not unity, they affect the area under consideration. As a consequence, Algorithms 1 and 2 cannot be directly applied to this case. However, the interpretations are similar. Algorithm 3 details the ACF for WCFP while Algorithm 4, defines the corresponding 'step-based' waterfilling algorithm conceived for finding the optimal values of K and L, respectively.

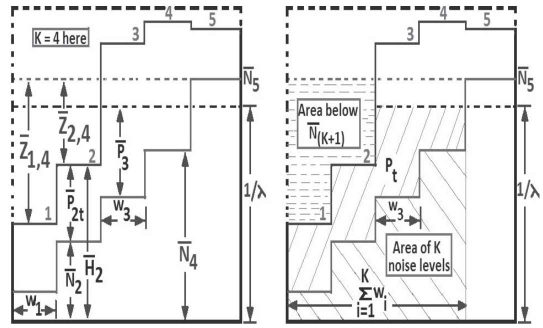


Fig. 3. Showing the effect of 'weights' in Weighted CFP.

Algorithm 3 ACF Algorithm for Obtaining K for WCFP

Require: Inputs required are M, P_t, $\bar{N}_i$, $\bar{H}_i$ & w_i (in ascending order of $\bar{N}_i$).

Ensure: Output is K, $\bar{I}_{R_{K-1}}$, $\bar{I}_{R_K}$, $\bar{d}_K$.

1: $i = 1$. Denote $\bar{d}_0 = P_t$, $W_0 = 0$, $\bar{U}_0 = 0$ and $\bar{I}_{R_0} = \emptyset$
2: Calculate $\bar{d}_i = \bar{d}_{i-1} + w_i \bar{N}_i$.
3: Calculate $W_i = W_{i-1} + w_i$
4: ▷ Calculate the area $\bar{U}_i = \sum_{m=1}^{i} w_m \bar{Z}_{m,i}^+$ as follows:
5: $\bar{I}_{R_i} = \{m;$ such that $\bar{N}_{i+1} > \bar{H}_m\}$, $W_{R_{i-1}} = \sum_{m \in \bar{I}_{R_{i-1}}} w_m$ $\bar{Z}_{m,i} = \bar{N}_{(i+1)} - \bar{H}_m, m \in (I_{R_i} - I_{R_{i-1}})$
6: $\bar{U}_i = \bar{U}_{i-1} + W_{R_{i-1}}(\bar{N}_{i+1} - \bar{N}_i) + \sum_{m \in (\bar{I}_{R_i} - \bar{I}_{R_{i-1}})} w_m \bar{Z}_{m,i}^+$
7: Calculate the area $\bar{Q}_i = W_i \bar{N}_{(i+1)}$
8: **if** $\bar{Q}_i \ge (\bar{d}_i + \bar{U}_i)$ **then**
9: $K \leftarrow i$. Exit the algorithm.
10: **else**
11: $i \leftarrow$ i+1, Go to 2
12: **end if**

Let us now formulate Theorem 5.

Theorem 5: The output of Algorithm 3 gives the optimal value K of the number of positive powers, as defined in Definition 1, for WCFP.

The proof is similar to that of the CFP case, with slight modifications concerning both the scaling and the width of the stairs w_i, hence it has been omitted.

Observe that the calculation of $\bar{P}_R$, $\bar{D}_m$ and $\bar{F}_m$ is affected by the weights w_i, since the areas depend on w_i.

Let us now state without proof that Algorithm 4 outputs the optimal value of L.

Theorem 6: Algorithm 4 delivers the optimal value L of the number of powers that are assigned peak powers, as defined in Definition 2, for WCFP.

Peak power allocated resources are $\bar{I}_P = \bar{I}_{R_{K-1}} \cup I_S(1 : L)$. Resources for which WFP allocates powers are $\bar{I}_P^c = [1, K] - \bar{I}_P$.

Algorithm 4 'Step-Based' Waterfilling Algorithm for Obtaining L for WCFP

Require: Inputs required are K, $\bar{d}_K$, $\bar{I}_{R_{K-1}}$, $\bar{I}_{R_K}$, W_K, $W_{R_{K-1}}$, $\bar{N}_i$, $\bar{H}_i$ & w_i (in ascending order of $\bar{N}_i$).
Ensure: Output is L, I_S.
Calculate $\bar{P}_R = \bar{d}_K - W_K\bar{N}_K + W_{R_{K-1}}\bar{N}_K - \sum_{m\in \bar{I}_{R_{K-1}}} w_m \bar{H}_m$
Calculate $\bar{I}_B = \bar{I}_{R_K} - \bar{I}_{R_{K-1}}$. $\bar{D}_1 = W_K - W_{R_{K-1}}$.
If $|\bar{I}_B| = 0$, set $L = 0$. Otherwise, if $|\bar{I}_B| > 0$, then only proceed with the following steps.
Sort $\{\bar{H}_m\}_{m\in \bar{I}_B}$ in ascending order and denote it as $\{\bar{H}_{mB}\}$ and the sorting index as I_S.
Initialize $m = 1$, $\bar{F}_m = (\bar{H}_{mB} - \bar{N}_K)\bar{D}_m$.
while $\bar{F}_m \le \bar{P}_R$ **do**
  $m = m + 1$. If $m > |\bar{I}_B|$, exit the while loop.
  $\bar{D}_m = \bar{D}_{m-1} - w_{I_S(m-1)}$
  $\bar{F}_m = \bar{F}_{m-1} + (\bar{H}_{mB} - \bar{H}_{(m-1)B})\bar{D}_m$
end while
$L = m - 1$.
calculate $\bar{D}_{L+1} = \bar{D}_L - w_{I_S(L)}$, only if $L = |\bar{I}_B|$.

The waterlevel for WCFP is given by

$$\frac{1}{\lambda} = \begin{cases} \bar{H}_{LB} + \dfrac{\bar{P}_R - \bar{F}_L}{\bar{D}_{L+1}}, & L > 0; \\ \bar{N}_K + \dfrac{\bar{P}_R}{\bar{D}_1}, & \text{L = 0.} \end{cases} \tag{31}$$

and the powers allocated are given by

$$P_i = \begin{cases} P_{it}, & i \in \bar{I}_P; \\ \dfrac{w_i}{x_i}\left(\dfrac{1}{\lambda} - \bar{N}_i\right), & i \in \bar{I}_P^c. \end{cases} \tag{32}$$

A. Computational Complexity of the WCFP

Let us now calculate the computational complexity of both Algorithm 3 and of Algorithm 4 and then add the complexity of calculating the powers, as follows:

- Calculating $\bar{N}_i$, $\bar{P}_{it}$ and $\bar{H}_i$ requires $3M$ multiplies and M adds.
- Observe that Algorithm 3 requires $(K+1)$ adds and K multiplies for calculating $\bar{d}_i$, K multiplies to find $\bar{Q}_i$ and, in the worst case, $4K$ additions and $2K$ multiplications for calculating $\bar{Z}_{m,i}$'s & $\bar{U}_K$, the corresponding proof is given in Appendix VI-E; K additions for calculating W_K and at-most K additions for calculating $W_{R_{i-1}}$. Consequently Algorithm 3 requires $(7K+1)$ additions and $4K$ multiplications for calculating K.
- Note that in Algorithm 4: 2 multiplies and $3 + |\bar{I}_{R_{K-1}}|$ additions are required for calculation of $\bar{P}_R$; at-most $(K+1)$ adds and 1 multiply in computing $\bar{F}_1$, $\bar{D}_1$; $4|\bar{I}_B|$ adds and $\bar{I}_B$ multiples for evaluating the while loop. Since $|\bar{I}_{R_{K-1}}|, |\bar{I}_B| < K$, the worst case complexity of Algorithm 4 can be given as $(6K+4)$ adds, $(K+3)$ multiplies.
- The computational complexity of calculating P_i is at-most K adds and K multiplies.
- Consequently, the total computational complexity of solving the WCFP, considered is $(14K+5+M)$ adds and $(3M+6K+3)$ multiplies. Since K is not known apriori, the worst case complexity is given by $(15M+5)$ adds and $(9M+3)$ multiplies. i.e we have a complexity order of $O(M)$.

Explicitly, the proposed solution's computational complexity is of the order of M, whereas that of the GWFPP of [22] is of the order of M^2.

IV. WCFP Requiring Minimum Power

In this section we further extend the WCFP to the case where the resources/powers scenario of having both a Minimum and a Maximum Power (MMP) constraint. The resultant WCFP-MMP arises in the following context:

(a) In a CR network, CR senses that some resources are available for it's use and allocates powers to the available resources for a predefined amount of time while ensuring that the peak power constraint is satisfied, in order to keep the interference imposed on the PU with in the affordable limit. Again, the weights w_i and x_i represent the resource's available time and sensing probabilities. The minimum power has to be sufficient to support the required quality of service, such as the minimum transmission rate of each resource [30]–[32].

We show that solving WCFP-MMP can be reduced to solving WCFP with the aid of an appropriate transformation. Hence, Section III can be used for this case. Mathematically, the problem can be formulated as

$$\begin{aligned} \max_{\{P_i\}_{i=1}^M} \; & C = \sum_{i=1}^{M} w_i \log_2\left(1 + \frac{P_i}{N_i}\right) \\ \text{subject to :} \; & \sum_{i=1}^{M} x_i P_i \le P_t \qquad (33) \\ & P_{ib} \le P_i \le P_{it}, \quad i \le M \\ & \text{and } P_i \ge 0, \quad i \le M, \end{aligned}$$

where $P_{ib} \le P_{it}$ and P_{ib} is the lower bound while P_{it} is the upper bound of the i^{th} power. w_i and x_i are weights of the i^{th} resource's capacity and i^{th} resource's allotted power, respectively. Using the KKT, the solution of this case can be written as

$$\bar{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+, \quad i = 1, \cdots, M; \tag{34}$$

$$\sum_{i=1}^{K} w_i \bar{P}_i = P_t; \tag{35}$$

$$\bar{P}_{ib} \le \bar{P}_i \le \bar{P}_{it}, \quad i = 1, \cdots, M, \tag{36}$$

where $\bar{P}_i = \frac{P_i x_i}{w_i}$ is the weighted power, $\bar{P}_{it} = \frac{P_{it} x_i}{w_i}$ is weighted peak power, $\bar{P}_{ib} = \frac{P_{ib} x_i}{w_i}$ is the weighted minimum power and $\bar{N}_i = \frac{N_i x_i}{w_i}$ is the weighted noise.

Let us now formulate Theorem 7.

Theorem 7: For every WCFP-MMP given by (33)*, there exists a WCFP, whose solution will result in a solution to the WCFP-MMP.*

Proof: Consider the solution to WCFP-MMP given by (34)-(36). Defining $\hat{P}_i = \bar{P}_i - \bar{P}_{ib}$ and substituting it into (34)-(36), we arrive at:

$$\hat{P}_i = \left(\frac{1}{\lambda} - \bar{N}_i\right)^+ - \bar{P}_{ib}, \quad i = 1, \cdots, M; \tag{37}$$

$$\sum_{i=1}^{K} w_i(\hat{P}_i + \bar{P}_{ib}) = P_t; \tag{38}$$

$$0 \leq \hat{P}_i \leq \left(\bar{P}_{it} - \bar{P}_{ib}\right), \quad i = 1, \cdots, M. \tag{39}$$

Using (37) and the definition of $()^+$, we can rewrite (37)–(39) as

$$\hat{P}_i = \left(\frac{1}{\lambda} - \underbrace{\{\bar{N}_i + \bar{P}_{ib}\}}_{\hat{N}_i}\right)^+, \quad i = 1, \cdots, M; \tag{40}$$

$$\sum_{i=1}^{K} w_i \hat{P}_i = \underbrace{\left(P_t - \sum_{i=1}^{K} w_i \bar{P}_{ib}\right)}_{\hat{P}_t}; \tag{41}$$

$$0 \leq \hat{P}_i \leq \underbrace{\left(\bar{P}_{it} - \bar{P}_{ib}\right)}_{\hat{P}_{it}}, \quad i = 1, \cdots, M. \tag{42}$$

Comparing (40)-(42) to (28)-(30), we can observe that this is a solution for a WCFP with variables $\hat{P}_i$, $\hat{N}_i$, $\hat{P}_{it}$ and $\hat{P}_t$. It follows then that we can solve the WCFP-MMP by solving the WCFP, whose solution is given by (40)-(42). □

Note that the effect of the lower bound is that of increasing the height of the floor stairs for the corresponding WCFP at a concomitant reduction of the total power constraint.

A. Computaional Complexity of the WCFP-MMP

Solving WCFP-MMP requires $4M$ additional adds, to compute $\hat{P}_i$, $\hat{N}_i$, $\hat{P}_{it}$ as well as $\hat{P}_t$, and K adds to recover P_i from $\hat{P}_i$; as compared to WCFP. Hence the the worst case complexity of solving the WCFP-MMP is given by $(19M+6)$ adds and $(8M + 3)$ multiplies. i.e we have a complexity of $O(M)$.

V. Simulation Results

Our simulations have been carried out in MATLAB R2010b software. To demonstrate the operation of the proposed algorithm, some numerical examples are provided in this section.

Example 1: Illustration of the CFP is provided by the following simple example:

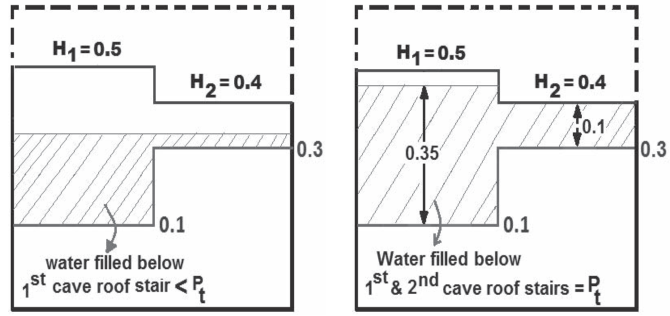
$$\max_{\{P_i\}_{i=1}^2} \quad C = \sum_{i=1}^{2} \log_2\left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints :} \sum_{i=1}^{2} P_i \leq 0.45;$$

$$P_i \leq 0.7 - 0.3i, \quad i \leq 2$$

$$\text{and } P_i \geq 0, \quad i \leq 2. \tag{43}$$

Assuming $N_i = \{0.1, 0.3\}$, we have $H_i = \{0.5, 0.4\}$. For the example of (43), water is filled above the first floor stair, as shown in Fig. 4a. This quantity of water is less than P_t. Hence, we fill the water above the second floor stair until the water level reaches 0.45. At this point the peak constraint for the second resource comes into force and the water can only be filled above second floor stair, as shown in Fig. 4b. Now, this amount of water becomes equal to P_t giving $K = 2$. We can observe that the first resource has a power determined by the 'waterlevel', while the second resource is assigned the peak power.



(a) (b)

Fig. 4. Illustration for Example 1: (a) Water filled above floor stairs 1 and 2, without peak constraint. (b) Water filled above floor stairs 2 only.

In Algorithm 1, we have $U_1 = 0$ as $Z_{1,1}^+ = 0$ and $I_{R_1} = 0$. $d_1 = P_t + N_1 = 0.55$, while $Q_1 = 1 \times N_2 = 0.3$. We can check that $Q_1 \ngtr (d_1+U_1)$ which indicates that $K > 1$. Hence, we get $K = 2$.

Let us now use Algorithm 2 to find the specific resources that are to be allocated the peak powers. We have $I_{R_{K-1}} = 0$ as $N_K < H_1$. The remaining power P_R in Algorithm 2 is 0.25. The resource indices to check for the peak power allocation are $I_B = \{1, 2\}$. From $H_m|_{m\in I_B}$, we get $[H_{1B}, H_{2B}] = \{0.4, 0.5\}$ and $I_S = \{2, 1\}$. We can check that $F_1 = 0.2 < P_R$ and $F_2 = 0.3 > P_R$. This gives $L = 1$. Hence we allocate the peak power to the $I_S(L)$ or second resource, i.e. we have $P_2 = P_{2t} = 0.1$. The first resource can be assigned the remaining power of $P_1 = P_t - P_{2t} = 0.35$.

Example 2: A slightly more involved example of the CFP, with more resources is illustrated here:

$$\max_{\{P_i\}_{i=1}^8} \quad C = \sum_{i=1}^{8} \log_2\left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints :} \sum_{i=1}^{8} P_i \leq 6;$$

$$P_i \leq P_{it}, \quad i \leq 8$$

$$\text{and } P_i \geq 0, \quad i \leq 8. \tag{44}$$

In (44); we have $N_i = 2i - 1, \forall i$ and $P_{it} = \{8, 1, 3, 3, 6, 3, 4, 1\}$. The heights of the cave roof stairs are $H_i = \{9, 4, 8, 10, 15, 14, 17, 16\}$.

In Fig. 5, when the water is filled below the third cave roof stair, the amount of water is $P_t = 6$, which fills above the three cave floor stairs, hence giving $K = 3$. The same can be obtained from Algorithm 1. Using Algorithm 1, the $(d_i + U_i)$ and the Q_i values are obtained which are shown in Table II. Since the $(d_i + U_i)$ values are {7, 11, 18}, while the Q_i are {3, 10, 21}, we have $Q_3 > (d_3 + U_3)$ and $Q_i < (d_i + U_i)$, $i = 1, 2$. This gives $K = 3$.

As we have $N_K = 5 > H_2 = 4$, $I_{R_{K-1}} = 2$; the second resource is to be assigned the peak power.

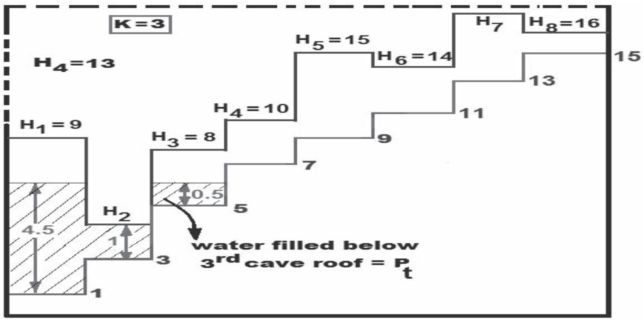


Fig. 5. Illustration of Example 2: Water filled below the roof stair 3 gives $K = 3$.

TABLE II
RESULTS FOR EXAMPLE 2:

Parameter	**Values of the parameters for** (44)
$(d_i + U_i), i \leq K$	7, 11, 18
$Q_i, \;\; i \leq K$	3, 10, 21
Peak power based resources	2
Water filling based resources	1, 3
Powers of the resources $P_i, i \in [1, K]$	4.5, 1, 0.5
Capacities of the resources $i \in [1, K]$	2.4594,2.8745,3.0120

Similarly, as $N_{K+1}(= 7) > H_i, i \in [1, K]$ is satisfied for $i = 2$ resource, we have $I_{R_K} = 2$. Since $I_B = I_{R_K} - I_{R_{K-1}} = \emptyset$, there are no resources that have $H_i, i \in [1, K]$ values in between N_K and N_{K+1}. Thus, there is no need to invoke the 'step-based water filling' of Algorithm 2, which gives $L = 0$.

Now, peak power based resources are $I_P = I_{R_{K-1}} = \{2\}$. The water filling algorithm allocates powers for the $I_P^c = [1, K] - I_P = \{1, 3\}$ resources.

The peak power based resources and water filling based resources are shown in Table II. For the remaining power, $P_R = 1$, the water level obtained for the I_P^c resources (with $L = 0$) is 5.5. The powers allocated to the resources $\{1, 3\}$ using this water level are $\{4.5, 0.5\}$. The powers and corresponding throughputs are shown in Table II.

Example 3: The weighted CFP is illustrated by the following simple example:

$$\max_{\{P_i\}_{i=1}^{5}} \; C = \sum_{i=1}^{5} w_i \log_2 \left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints :} \; \sum_{i=1}^{5} x_i P_i \leq 5;$$

$$P_i \leq 2, \quad i \leq 5$$

$$\text{and } P_i \geq 0, \quad i \leq 5. \qquad (45)$$

In (45); lets us consider $N_i = [0.2, 0.1, 0.4, 0.3, 0.5]$, $w_i = 6 - i$ and $x_i = i$, $\forall i$. The $\bar{N}_i$ values are [0.04, 0.05, 0.4, 0.6, 2.5], while the $\bar{H}_i$ values are [0.44, 1.05, 2.40, 4.60, 12.5]. Applying the ACF algorithm, we arrive at $K = 4$.

We have $\bar{H}_i < \bar{N}_K, i \in [1, K]$ for the 1^{st} resource. The 'step-based' waterfilling algorithm confirms that 1^{st} resource is indeed the resource having the peak power. The remaining 2^{nd}, 3^{rd} and 4^{th} resources are allocated their powers using the water filling algorithm. For the water level of 0.62222, powers allotted for {2,3,4} resources are [1.1444, 0.22222, 0.011111].

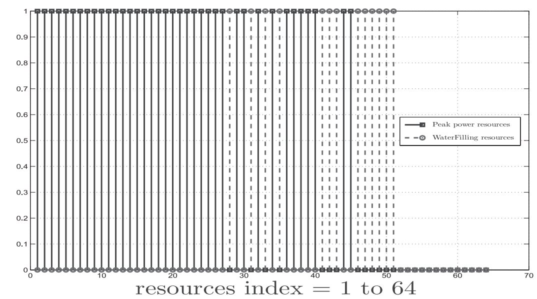


Fig. 6. Index of the peak power based resources (continuous lines) and waterfilling allotted resources (dashed lines) for Example 4.

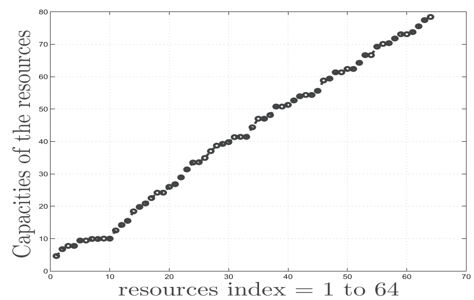


Fig. 7. Throughputs of the resources for Example 4.

Example 4: Another example for the weighted CFP associated with random weights:

$$\max_{\{P_i\}_{i=1}^{64}} \; C = \sum_{i=1}^{64} w_i \log_2 \left(1 + \frac{P_i}{N_i}\right)$$

$$\text{with constraints :} \; \sum_{i=1}^{64} x_i P_i \leq 1;$$

$$P_i \leq P_{it}, \quad i \leq 64$$

$$\text{and } P_i \geq 0, \quad i \leq 64. \qquad (46)$$

In this example, we assume $N_i = \frac{\sigma^2}{h_i}$ while h_i, w_i and x_i are exponentially distributed with a mean of 1. Furthermore $\sigma^2 = 10^{-2}$ and P_{it}, $\forall i$ are random values in the range $[10^{-3}, 5 \times 10^{-2}]$.

Now applying the ACF algorithm, we get $K = 51$ for a particular realization of h_i, w_i and x_i. For this realization, from the $[1, K]$ resources, 38 resources are to be allocated with the peak powers and 13 resources get powers from the waterfilling algorithm. These resources are shown in Fig. 6. The achieved throughput of the resources is given in Fig. 7 for the proposed algorithm. The results match with the values obtained for known algorithms.

Table III gives the actual number of flops required by the proposed solution and the other existing algorithms for

TABLE III

COMPUTATIONAL COMPLEXITIES OF EXISTING ALGORITHMS AND THE PROPOSED SOLUTION FOR $\omega_i = x_i = 1, \forall i$

M → K	**Number of flops in algorithms of [18], [19]**§	**Number of flops in FWF of [21]**¶	**Number of flops in GWFPP of [22]**	**Number of flops in in proposed solution** ‖
$64 \rightarrow 46$	14985216 (39024)	7824 (24)	16832	541 (24,6)
$128 \rightarrow 87$	70563072 (91879)	33592 (52)	66432	956 (31,1)
$256 \rightarrow 135$	291746304 (189939)	96450 (75)	263936	1513 (13,4)
$512 \rightarrow 210$	$1.5115 \times 10^{+09}$ ($4.9203 \times 10^{+05}$)	156526 (61)	1052160	2432 (21,0)
$1024 \rightarrow 334$	$1.6165 \times 10^{+10}$ ($2.6311 \times 10^{+06}$)	271678 (53)	4201472	4059 (15,1)

Example 4 with different M values. Since some of the existing algorithms do not support $\omega_i \neq 1$ and $x_i \neq 1, \forall i$; we assume $\omega_i = x_i = 1, \forall i$ for Table III.

It can be observed from Table III that the number of flops imposed by the sub-gradient algorithm of [18] and [19] is more than 10^4 times that of the proposed solution. The number of flops required for the FWF of [21] and for the GWFPP of [22] are more than 10^2 times that of the proposed solution. This is because the proposed solution's computational complexity is $O(M)$, whereas the best known existing algorithms have an $O(M^2)$ order of computational complexity; as listed in Table I.

It has also been observed from the above examples that $|I_B| = |I_{R_K} - I_{R_{K-1}}|$ values are very small as compared to M. As such L has been obtained from Algorithm 2 within two iterations of the while loop.

VI. CONCLUSIONS

In this paper, we have proposed algorithms for solving the CFP at a complexity order of $O(M)$. The approach was then generalized to the WCFP and to the WCFP-MMP. Since the best known solutions solve these three problems at a complexity order of $O(M^2)$, the proposed solution results in a significant reduction of the complexity imposed. The complexity reduction attained is also verified by simulations.

APPENDIX

A. Proof of Theorem 1

Proof: Lagrange's equation for (1) is

$$L(P_i, \lambda, \omega_i, \gamma_i) = \sum_{i=1}^{M} \log_2\left(1 + \frac{P_i}{N_i}\right) - \lambda\left(\sum_{i=1}^{M} P_i - P_t\right) - \sum_{i=1}^{M} \omega_i\,(P_i - P_{it}) - \sum_{i=1}^{M} \gamma_i\,(0 - P_i) \tag{47}$$

§ λ is initialized to 5×10^{-1}.

§,¶ Number of iterations is given in brackets.

‖ $|I_{R_{K-1}}|$ and $|I_B|$ are given in brackets. Actual number of flops is $M + 9K + 5|I_B| + |I_{R_{K-1}}| + 9$.

Karush-Kuhn-Tucker (KKT) conditions for (47) are [3], [35]

$$\frac{\partial L}{\partial P_i} = 0 \Rightarrow \frac{1}{N_i + P_i} - \lambda - \omega_i + \gamma_i = 0, \tag{48}$$

$$\lambda\left(P_t - \sum_{i=1}^{M} P_i\right) = 0, \tag{49}$$

$$\omega_i\,(P_{it} - P_i) = 0, \quad \forall i \tag{50}$$

$$\gamma_i P_i = 0, \quad \forall i \tag{51}$$

$$\lambda, \omega_i \;\&\; \gamma_i \geq 0, \quad \forall i \tag{52}$$

$$P_i \leq P_{it}, \quad \forall i, \tag{53}$$

$$\sum_{i=1}^{M} P_i \leq P_t. \tag{54}$$

In what follows we show that the KKT conditions result in a simplified *'form'* for the solution of CFP which is similar to the conventional WFP. *The proof is divided into three parts corresponding to the three possibilities for P_i: that is 1) Equivalent constraint for $P_i < 0$ in terms of the 'water level' $\frac{1}{\lambda}$ and the corresponding solution form, 2) Equivalent constraint for $P_i \leq P_{it}$ in terms of the 'water level' and and the corresponding solution form, and 3) Equivalent form for $P_i < P_i < P_{it}$ in terms of the 'water level' and the corresponding solution form.*

1) Simplification for $P_i \geq 0$: Multiplying (48) with P_i and substituting (51) in it, we obtain

$$P_i\left(\frac{1}{N_i + P_i} - \lambda - \omega_i\right) = 0 \tag{55}$$

In order to satisfy (55), either P_i or $(\frac{1}{N_i+P_i} - \lambda - \omega_i)$ should be zero. Having $P_i = 0, \forall i$ does not solve the optimization problem. Hence, we obtain

$$\left(\frac{1}{N_i + P_i} - \lambda - \omega_i\right) = 0, \quad when\ P_i > 0. \tag{56}$$

Since $\omega_i \geq 0$, (56) can be re-written as $(\frac{1}{N_i+P_i} - \lambda) \geq 0$. Furthermore, taking $P_i > 0$ in this, we attain

$$\frac{1}{\lambda} > N_i, \quad when\ P_i > 0. \tag{57}$$

The opposite of this is

$$\frac{1}{\lambda} \leq N_i, \quad when \ P_i \leq 0. \tag{58}$$

We can observe that (57) and (58) are equations related to the conventional WFP.

2) Simplification for $P_i \leq P_{it}$: Multiplying (48) with $P_{it} - P_i$ and substituting (50) in it, we attain

$$(P_{it} - P_i)\left(\frac{1}{N_i + P_i} - \lambda + \gamma_i\right) = 0 \tag{59}$$

In (59), two cases arise:

(a) If $P_{it} > P_i$, then $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) = 0$ becomes true.

Since $\gamma_i \geq 0$, $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) = 0$ is taken as $(\frac{1}{N_i+P_i} - \lambda) < 0$. Further Simplifying this and substituting $P_i < P_{it}$, we get

$$\frac{1}{\lambda} < H_i \triangleq (P_{it} + N_i), \quad if \ P_i < P_{it}. \tag{60}$$

(b) If $P_{it} = P_i$, then $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) \geq 0$ becomes true in (59).

As $\gamma_i \geq 0$, $(\frac{1}{N_i+P_i} - \lambda + \gamma_i) \geq 0$ is re-written as $(\frac{1}{N_i+P_i} - \lambda) \geq 0$. Substituting $P_{it} = P_i$ and simplifying this further, we obtain

$$\frac{1}{\lambda} \geq H_i \triangleq (P_{it} + N_i), \quad if \ P_i = P_{it}. \tag{61}$$

3) Simplification for $0 < P_i < P_{it}$:

(a) In (51); if γ_i is equal to zero, then $P_i > 0$. Combining this relation with (57), we can conclude that

$$\frac{1}{\lambda} > N_i, \quad if \ \gamma_i = 0. \tag{62}$$

(b) Similarly, in (50), if $\omega_i = 0$, then $P_{it} > P_i$ follows. Using this relation in (60), we acquire

$$\frac{1}{\lambda} < H_i, \quad if \ \omega_i = 0. \tag{63}$$

(c) Combining (62) and (63), we have

$$N_i < \frac{1}{\lambda} < H_i, \quad if \ \omega_i = \gamma_i = 0. \tag{64}$$

Using (64) in (48) and then re-arranging it gives

$$P_i = \frac{1}{\lambda} - N_i, \quad if \ N_i < \frac{1}{\lambda} < H_i. \tag{65}$$

Combining (57), (58), (60), (61) and (65), powers are obtained as

$$P_i = \begin{cases} \left(\frac{1}{\lambda} - N_i\right), & N_i < \frac{1}{\lambda} < H_i \text{ or } \\ & 0 < P_i < P_{it}; \\ P_{it}, & \frac{1}{\lambda} \geq H_i; \\ 0, & \frac{1}{\lambda} \leq N_i. \end{cases} \tag{66}$$

□

B. Proof of Proposition 2

Proof: The proof is by contradiction. Assume that $P_i^\star$, $i \leq M$ is the optimal solution for (1) such that $\sum_{i=1}^M P_i^\star < P_t$. We now prove that as $P_i^\star$ powers fulfil $\sum_{i=1}^M P_i^\star < P_t$, there exists $P_i^\diamond$ that has greater capacity. Define

$$P_i^\diamond = P_i^\star + \triangle P_i^\star, \quad \forall i \tag{67}$$

such that

$$\sum_{i=1}^M P_i^\diamond = P_t \ \text{ and } P_i^\diamond \leq P_{it}, \quad \forall i \tag{68}$$

where $\triangle P_i^\star \geq 0, \forall i$. From (7) there exists atleast one i such that $P_i^\star < P_{it}$. It follows that $\triangle P_i^\star > 0$ for atleast one i. The capacity of M resources for $P_i^\diamond$ allotted powers is

$$C\left(P_i^\diamond\right) = \sum_{i=1}^M \log_2\left(1 + \frac{P_i^\diamond}{N_i}\right) \tag{69}$$

Substituting (67) in (69), we get

$$C\left(P_i^\diamond\right) = \sum_{i=1}^M \log_2\left(1 + \frac{P_i^\star}{N_i} + \frac{\triangle P_i^\star}{N_i}\right) \tag{70}$$

Re-writing the above, we obtain

$$C\left(P_i^\diamond\right) = \sum_{i=1}^M \log_2\left[\left(1 + \frac{P_i^\star}{N_i}\right)\left(1 + \frac{\frac{\triangle P_i^\star}{N_i}}{1 + \frac{P_i^\star}{N_i}}\right)\right] \tag{71}$$

Following 'log(ab) = log(a) + log(b)' in the above, we acquire

$$C\left(P_i^\diamond\right) = \sum_{i=1}^M \log_2\left(1 + \frac{P_i^\star}{N_i}\right) + \sum_{i=1}^M \log_2\left(1 + \frac{\frac{\triangle P_i^\star}{N_i}}{1 + \frac{P_i^\star}{N_i}}\right) \tag{72}$$

As $\triangle P_i^\star > 0$ for atleast one i, the second term on the R.H.S. of (72) is always positive. We have

$$C\left(P_i^\diamond\right) > C\left(P_i^\star\right) \tag{73}$$

In other words, $\sum_{i=1}^M P_i^\diamond = P_t$ produces optimal capacity; completing the proof. □

C. The Computational Complexity of Calculating $Z_{m,i}$ for CFP

Below, it is shown that the worst case computational complexity of calculating $Z_{m,i}$; $m \leq i$ and $i \leq K$ for CFP is K subtractions.

- *In Algorithm 1, we first check if $N_{i+1} > H_m$. I_{R_i} is taken as 'm' values for which $N_{i+1} > H_m$. Note also that $I_{R_{i-1}} \subset I_{R_i}$. This is because if $Z_{m,i} = N_{i+1} - H_m > 0$, then $Z_{m,j}$; $j = i+1, \cdots, K$ is always positive since $N_j > N_i$, $j > i$. Hence, in the worst case, $K\log(K)$ comparisons are required. The cost of a comparison, is typically lower than that of an addition [36]. Hence it has not been included in the flop count.*
- *As per Algorithm 1, we calculate $Z_{m,i}$'s only for $m \in (I_{R_i} - I_{R_{i-1}})$. Furthermore, if we have $Z_{m,i} = N_{i+1} - H_m > 0$, then $Z_{m,j}$; $j = i+1, \cdots, K$ is always positive*

since $N_j > N_i$, $j > i$. In other words, if $I_{R_{i-1}}$ gets some 'x' values, then the same 'x' values will also be there in I_{R_i} and the contribution of this part to the overall area, U_i is $|I_{R_{i-1}}|(N(i+1) - N_i)$; which is calculated in Step 5. This implies that if $Z_{m,i}$ is calculated for $m \in I_{R_i}$, then there is no need to calculate $Z_{m,i}$ for $m \in I_{R_{i+1}}, I_{R_{i+2}}, \ldots I_{R_K}$. Hence, for a given m, $Z_{m,i}$ is calculated, in the worst case, once; for one 'i' only. As such, the worst case complexity of calculating $Z_{m,i}$ is as low as that of K subtractions.

D. The Computational Complexity of Calculating U_K for CFP

Here we show that the worst case computational complexity of calculating U_K for CFP is $4K$ adds and K multiplies. Note that in each iteration of Algorithm 1 the following is calculated:

$$U_i = U_{i-1} + |I_{R_{i-1}}|\,(N_{i+1} - N_i) + \sum_{m\in(I_{R_i} - I_{R_{i-1}})}^{i} Z^+_{m,i}. \quad (74)$$

There are three terms in (74) and we calculate the complexity of each term separately, as follows:

- The first term of (74), U_{i-1}, is already computed in the $(i-1)$-th iteration, hence involves no computation during the i-th iteration.
- The second term, $|I_{R_{i-1}}|(N_{i+1} - N_i)$, is taking care of the increase in reference height from N_i to N_{i+1} for those roof stairs, which are already below the reference level N_i. The computation of this term requires only a single multiplication and addition.
- The third term gives the areas of the roof stairs which are below N_{i+1} but not N_i. The number of additions in this is $A_i = |I_{R_i} - I_{R_{i-1}}| - 1$.
- Taking into account the two adds per iteration required for adding all the three terms, the total computational complexity of calculating U_i, given U_{i-1} is 1 multiply and $3 + A_i$ adds.

Since KU_i's are calculated; the total computational complexity of calculating all U_i's will be $\sum_{i=1}^{K} 3 + A_i = 3K + |I_{R_K}| \leq 4K$ adds and K multiplies.

E. The Computational Complexity of Calculating $\bar{U}_K$ for WCFP

Here we show that the worst case computational complexity of calculating $\bar{U}_K$ for WCFP is $4K$ adds $2K$ multiplies. Note that in each iteration of Algorithm 3 the following is calculated:

$$\bar{U}_i = \bar{U}_{i-1} + W_{R_{i-1}}\left(\bar{N}_{i+1} - \bar{N}_i\right) + \sum_{m\in(\bar{I}_{R_i} - I_{R_{i-1}})}^{i} w_m \bar{Z}^+_{m,i}. \quad (75)$$

There are three terms in (75) and we calculate the complexity of each term separately, as follows:

- The first term of (75), $\bar{U}_{i-1}$, is already computed in $i-1$-th iteration, hence involves no computation during the i-th iteration.
- The computation of second term, $W_{R_{i-1}}(\bar{N}_{i+1} - \bar{N}_i)$, requires only a single multiplication and addition.
- The third term gives the areas of the roof stairs which are below $\bar{N}_{i+1}$ but not $\bar{N}_i$. The number of additions in this is $A_i = |\bar{I}_{R_i}| - |\bar{I}_{R_{i-1}}|$. The corresponding number of multiplications is one.
- Taking into account the two adds per iteration required for adding all the three terms, the total computational complexity of calculating U_i, given U_{i-1} is 2 multiply and $3 + A_i$ adds.

Since KU_i's are calculated; the total computational complexity of calculating all U_i's will be $\sum_{i=1}^{K} 3 + A_i = 3K + |I_{R_K}| \leq 4K$ adds and $2K$ multiplies.

References

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, May 2005.

[2] D. P. Palomar and J. R. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 686–695, Feb. 2005.

[3] F. Gao, T. Cui, and A. Nallanathan, "Optimal training design for channel estimation in decode-and-forward relay networks with individual and total power constraints," *IEEE Trans. Signal Process.*, vol. 56, no. 12, pp. 5937–5949, Dec. 2008.

[4] A. A. D'Amico, L. Sanguinetti, and D. P. Palomar, "Convex separable problems with linear constraints in signal processing and communications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 6045–6058, Nov. 2014.

[5] E. Altman, K. Avrachenkov, and A. Garnaev, "Closed form solutions for water-filling problems in optimization and game frameworks," *Telecommun. Syst.*, vol. 47, nos. 1–2, pp. 153–164, 2011.

[6] R. Zhang, "On peak versus average interference power constraints for protecting primary users in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 2112–2120, Apr. 2009.

[7] X. Kang, R. Zhang, Y.-C. Liang, and H. K. Garg, "Optimal power allocation strategies for fading cognitive radio channels with primary user outage constraint," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 374–383, Feb. 2011.

[8] G. Bansal, M. J. Hossain, and V. K. Bhargava, "Optimal and suboptimal power allocation schemes for OFDM-based cognitive radio systems," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4710–4718, Nov. 2008.

[9] N. Kalpana, M. Z. A. Khan, and U. B. Desai, "Optimal power allocation for secondary users in CR networks," in *Proc. IEEE Adv. Netw. Telecommun. Syst. Conf. (ANTS)*, Bengaluru, India, Dec. 2011, pp. 1–6.

[10] H. Zhang and D. L. Goeckel, "Peak power reduction in closed-loop MIMO-OFDM systems via mode reservation," *IEEE Commun. Lett.*, vol. 11, no. 7, pp. 583–585, Jul. 2007.

[11] C. Studer and E. G. Larsson, "PAR-aware large-scale multi-user MIMO-OFDM downlink," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 303–313, Feb. 2013.

[12] N. Andgart, B. S. Krongold, P. Ödling, A. Johansson, and P. O. Börjesson, "PSD-constrained PAR reduction for DMT/OFDM," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 10, pp. 1498–1507, 2004. AQ:4

[13] A. Amirkhany, A. Abbasfar, V. Stojanović, and M. A. Horowitz, "Practical limits of multi-tone signaling over high-speed backplane electrical links," in *Proc. ICC*, Jun. 2007, pp. 2693–2698.

[14] V. M. K. Chan and W. Yu, "Multiuser spectrum optimization for discrete multitone systems with asynchronous crosstalk," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5425–5435, Nov. 2007.

[15] L. Fang and R. J. P. de Figueiredo, "Energy-efficient scheduling optimization in wireless sensor networks with delay constraints," in *Proc. ICC*, Jun. 2007, pp. 3734–3739.

[16] A. Roumy and D. Gesbert, "Optimal matching in wireless sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 725–735, Dec. 2007.

[17] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher, "RID: Radio interference detection in wireless sensor networks," in *Proc. IEEE Adv. Netw. Telecommun. Syst. Conf. (ANTS)*, Bangalore, India, Dec. 2011. AQ:5

[18] M. Arulraj and T. S. Jeyaraman, "MIMO radar waveform design with peak and sum power constraints," *EURASIP J. Adv. Signal Process.*, vol. 2013, no. 1, p. 127, 2013.

[19] L. Zhang, Y. Xin, Y.-C. Liang, and H. V. Poor, "Cognitive multiple access channels: Optimal power allocation for weighted sum rate maximization," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2754–2762, Sep. 2009.

[20] E. Yaacoub and Z. Dawy, *Resource Allocation in Uplink OFDMA Wireless Systems: Optimal Solutions and Practical Implementations*. New York, NY, USA: Wiley, 2012.

[21] X. Ling, B. Wu, P.-H. Ho, F. Luo, and L. Pan, "Fast water-filling for agile power allocation in multi-channel wireless communications," *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1212–1215, Aug. 2012.

[22] P. He, L. Zhao, S. Zhou, and Z. Niu, "Water-filling: A geometric approach and its application to solve generalized radio resource allocation problems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3637–3647, Jul. 2013.

[23] R.-R. Chen and Y. Lin, "Optimal power control for multiple access channel with peak and average power constraints," in *Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput.*, vol. 2. Jun. 2005, pp. 1407–1411.

[24] N. Papandreou and T. Antonakopoulos, "Bit and power allocation in constrained multicarrier systems: The single-user case," *EURASIP J. Adv. Signal Process.*, vol. 2008, Jan. 2008, Art no. 11.

[25] X. Zhou, R. Zhang, and C. K. Ho, "Wireless information and power transfer in multiuser OFDM systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 4092–4097.

[26] N. Kalpana and M. Z. A. Khan, "Fast Computation of Generalized Waterfilling Problems," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1884–1887, Nov. 2015.

[27] N. Kalpana and M. Z. A. Khan, "Weighted water-filling algorithm with reduced computational complexity," in *Proc. ICCIT Conf.*, May 2015. AQ:6

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.

[29] D. E. Knuth, *The Art of Computer Programming: Sorting Searching*, vol. 3, 2nd ed. Boston, MA, USA: Addison-Wesley, 1998.

[30] L. Zhang, Y.-C. Liang, and Y. Xin, "Joint beamforming and power allocation for multiple access channels in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 38–51, Jan. 2008.

[31] S. Stotas and A. Nallanathan, "Optimal sensing time and power allocation in multiband cognitive radio networks," *IEEE Trans. Commun.*, vol. 59, no. 1, pp. 226–235, Jan. 2011.

[32] Z. Tang, G. Wei, and Y. Zhu, "Weighted sum rate maximization for OFDM-based cognitive radio systems," *Telecommun. Syst.*, vol. 42, nos. 1–2, pp. 77–84, Oct. 2009.

[33] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[34] R. Rajesh, V. Sharma, and P. Viswanath. (2012). "Information capacity of energy harvesting sensor nodes." [Online]. Available: http://arxiv.org/abs/1009.5158

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] A. Bellaouar and M. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*. New York, NY, USA: Springer, 1995.

AQ:7

Kalpana Naidu received the Ph.D. degree from IIT Hyderabad, in 2016. Since 2016, she has been an Associate Professor with the VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad. The focus of her research is on resource allocation in wireless communication, HetNets, cognitive radio networking, and signal processing applied to wireless networks.

Mohammed Zafar Ali Khan received the B.E. degree in electronics and communications from Osmania University, Hyderabad, India, in 1996, the M.Tech. degree in electrical engineering from IIT Delhi, Delhi, India, in 1998, and the Ph.D. degree in electrical and communication engineering from the Indian Institute of Science, Bangalore, India, in 2003. In 1999, he was a Design Engineer with Sasken Communication Technologies, Ltd., Bangalore. From 2003 to 2005, he was a Senior Design Engineer with Silica Labs Semiconductors India Pvt. Ltd., Bangalore. In 2005, he was a Senior Member of the Technical Staff with Hellosoft, India. From 2006 to 2009, he was an Assistant Professor with IIIT Hyderabad. Since 2009, he has been with the Department of Electrical Engineering, IIT Hyderabad, where he is currently a Professor. He has more than ten years of experience in teaching and research and the space-time block codes that he designed have been adopted by the WiMAX Standard. He has been a Chief Investigator for a number of sponsored and consultancy projects. He has authored the book entitled *Single and Double Symbol Decodable Space-Time Block Codes* (Germany: Lambert Academic). His research interests include coded modulation, space-time coding, and signal processing for wireless communications. He serves as a Reviewer for many international and national journals and conferences. He received the INAE Young Engineer Award in 2006.

Lajos Hanzo (F'–) received the degree in electronics in 1976, the Ph.D. degree in 1983, and the Honorary Doctorate degree from the Technical University of Budapest, in 2009, while by the University of Edinburgh, in 2015. During his 38-year career in telecommunications, he has held various research and academic positions in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the Chair in Telecommunications. He has successfully supervised about 100 Ph.D. students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totaling in excess of 10 000 pages, published over 1500 research entries at the IEEE Xplore, acted both as a TPC and General Chair of the IEEE conferences, presented keynote lectures, and has received a number of distinctions. He directs a 60-strong academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by the industry, the Engineering and Physical Sciences Research Council, U.K., the European Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. He is an Enthusiastic Supporter of industrial and academic liaison and he offers a range of industrial courses. He is a fellow of REng, IET, and EURASIP. He is also a Governor of the IEEE VTS. From 2008 to 2012, he was the Editor-in-Chief of the IEEE PRESS and a Chaired Professor with Tsinghua University, Beijing. His research is funded by the European Research Council's Senior Research Fellow Grant. He has 24 000 citations.

AQ:8 AQ:9

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

AQ:1 = Please be advised that per instructions from the Communications Society this proof was formatted in Times Roman font and therefore some of the fonts will appear different from the fonts in your originally submitted manuscript. For instance, the math calligraphy font may appear different due to usage of the usepackage[mathcal]euscript. We are no longer permitted to use Computer Modern fonts.

AQ:2 = Please confirm whether the financial section retained as in the metadata is OK.

AQ:3 = Note that if you require corrections/changes to tables or figures, you must supply the revised files, as these items are not edited for you.

AQ:4 = Please confirm the volume no. for refs. [12], [18], and [24].

AQ:5 = Please confirm the conference title, month, and year for ref. [17]. Also provide the page range.

AQ:6 = Please confirm the author names, article title, conference title, month, and year for ref. [27]. Also provide the page range.

AQ:7 = Current affiliation in biography of Kalpana Naidu does not match First Footnote. Please check.

AQ:8 = Please confirm whether the edits made in the sentence “Lajos Hanzo received . . . Edinburgh in 2015” are OK.

AQ:9 = Please provide the membership year for the author “Lajos Hanzo.”