A Theoretical Framework for Capacity-Achieving Multi-User Waterfilling in OFDMA

Simon Görtzen, Anke Schmeink
Information Theory and Systematic Design of Communication Systems
UMIC Research Centre, RWTH Aachen University, 52056 Aachen, Germany
Email: {goertzen, schmeink}@umic.rwth-aachen.de

***Abstract*—This paper introduces a theoretical framework for subcarrier and power allocation algorithms in rate-adaptive OFDMA systems. The focal point is locating "capacity-achieving" waterlevels for a given allocation in order to minimize the distance to the boundary of the capacity region. We prove that it is possible to restrict the choice of waterlevels to an optimality polyhedron. This paper introduces weighted subcarrier allocations which have a natural correspondence to this polyhedron, and are therefore promising candidates for the above problem. Based on the introduced theory, a low-complexity algorithm is designed and shown to reliably locate capacity-achieving waterlevels.**

I. Introduction

In a single-user OFDM system, performing power allocation across subcarriers in the well-known waterfilling fashion provides the optimal solution [1]. In a multiple-access OFDMA environment, however, the situation is more complicated on different levels. First of all, there is the problem of assigning the available subcarriers to the different users. In addition, the available power has to be distributed in a way that ensures fairness and efficiency. Similar to the single-user case, power should be allocated across subcarriers in a waterfilling fashion, however, with different waterlevels for each user [2]. To be able to simultaneously solve these two resource allocation problems in a near-optimal way while maintaining low computational complexity is of great importance to any OFDMA system.

We focus on the class of users with non-real-time applications, like downloads, leading to the so-called *rate-adaptive* approach for a given power budget. Here an increase in data rate always has an immediate positive impact, for example by decreasing the duration of the download.

The structure of this paper is as follows: Section II covers the system model and problem formulation. In Sections III and IV, the theoretical framework regarding capacity-achieving waterlevels is developed and presented. Section V introduces weighted allocations, which are shown to be naturally linked to the theoretical results of Section IV. In Section VI all results are combined to design a low-complexity algorithm for OFDMA resource allocation with capacity-achieving waterlevels. Simulation results are presented in Section VII, and Section VIII provides concluding remarks.

This work was supported by the UMIC Research Centre at RWTH Aachen University in Germany.

II. System Model and Problem Formulation

We assume a basic OFDMA downlink model with K users, N subcarriers equally dividing the bandwidth, a total power constraint $p_{\max}$ and full channel state information at the transmitter. The channel is completely characterized by the channel gain to noise ratio (CNR) matrix $(c_{k,n}) \in \mathbb{R}_{>0}^{K\times N}$.

Let $p_{k,n}$ denote the power of user k on subcarrier n. Then, based on the Shannon capacity formula, the achievable rate for user k computes to

$$r_k = \sum_{n=1}^{N} \log(1 + p_{k,n} c_{k,n}). \tag{1}$$

It is required that each subcarrier n is allocated to at most one user k and the powers $p_{k,n}$ are assigned in a way that the total power constraint is met. Therefore,

$$\sum_{k=1}^{K}\sum_{n=1}^{N} p_{k,n} \le p_{\max} \tag{2}$$

$$p_{k,n}p_{l,n} = 0 \qquad \forall\, k \neq l,\, n = 1,\dots,N \tag{3}$$

$$p_{k,n} \ge 0 \qquad \forall\, k = 1,\dots,K. \tag{4}$$

In the rate-adaptive setting, one aims at maximizing the users' data rates for a given power budget. One approach is *sum rate maximization*, where the objective is the total data rate obtained. However, due to the nature of wireless communication, some users have a much higher channel quality than others and benefit greatly from sum rate maximization while users with poor channels might not get any data rate at all.

Therefore, we need a problem formulation which promotes fairness in multi-user rate-adaptive optimization. We introduce a weight vector $\mathbf{w} \in \mathbb{R}_{\ge 0}^{K}$ and obtain the weighted objective

$$\underset{\{p_{k,n}\}}{\text{maximize}} \quad \mathbf{w}^T\mathbf{r} = \sum_{k=1}^{K} w_k r_k, \tag{5}$$

which, subject to constraints (2)-(4), is the *weighted sum rate* optimization problem. Denote by $\mathsf{WSRmax}(\mathbf{w})$ the optimal value of $\mathbf{w}^T\mathbf{r}$ in (5). This value can be obtained by convex optimization techniques as shown in [3]. However, achieving the desired amount of fairness between users depends heavily on the choice of $\mathbf{w}$ and is influenced by other factors like users' channel qualities and available power. For a given subcarrier allocation, the optimal power distribution is multi-user waterfilling with individual waterlevels proportional to $\mathbf{w}$

(see Section III). Therefore, we will use the terms waterlevels and weight vectors synonymously. For simplicity, we assume all weight vectors in this paper to be normalized in the sense that $w_1 + \ldots + w_K = 1$.

Based on the above model, define the *capacity region* $\mathcal{C}$ as the set of achievable rate vectors:

$$\mathcal{C} = \{\mathbf{r} = (r_1, \ldots, r_K) \,|\, \text{s.t. (2), (3), (4)}\} \subseteq \mathbb{R}^K_{\geq 0}.$$

From a geometric point of view, a rate vector $\mathbf{r}_{\text{opt}}$ maximizing (5) implies that there is no point in $\mathcal{C}$ lying above the hyperplane with normal $\mathbf{w}$ that contains $\mathbf{r}_{\text{opt}}$. In particular, for every weight vector $\mathbf{w}$, the optimal solution is located on the boundary of $\mathcal{C}$, while suboptimal solutions $\mathbf{r}_{\text{sub}}$, by definition, satisfy $\mathbf{w}^T\mathbf{r}_{\text{sub}} < \mathbf{w}^T\mathbf{r}_{\text{opt}}$.

Based on the above, we introduce an intuitive performance measure for suboptimal solutions to the weighted sum rate problem. The ratio

$$q(\mathbf{w}, \mathbf{r}) = \frac{\mathbf{w}^T\mathbf{r}}{\mathbf{w}^T\mathbf{r}_{\text{opt}}} = \frac{\mathbf{w}^T\mathbf{r}}{\mathsf{WSRmax}(\mathbf{w})} \tag{6}$$

measures the relative distance between the parallel hyperplanes that run through $\mathbf{r}$ and $\mathbf{r}_{\text{opt}}$, respectively. Independent of the problem, the closer $q(\mathbf{w}, \mathbf{r})$ is to 1, the better the solution. We call a suboptimal solution $\mathbf{r}$ with $q(\mathbf{w}, \mathbf{r}) \approx 1$ *capacity-achieving* to stress its quasi-optimality.

Clearly, $q(\mathbf{w}, \mathbf{r})$ is an appropriate quality measure for solutions that aim at maximizing $\mathbf{w}^T\mathbf{r}$. For practical purposes, however, low-complexity algorithms have to be designed to work without a predetermined weight vector $\mathbf{w}$, instead relying on good subcarrier allocations to provide fair and power-efficient results. The most common power distribution scheme for suboptimal algorithms is single-level waterfilling, which maximizes the total sum rate and therefore corresponds to a weight vector $\mathbf{m} = (\frac{1}{K}, \ldots, \frac{1}{K})$. Recall that this is a weight vector for which (5) greatly favors the users with the best channel quality.

This has multiple consequences, which are the main motivation of this paper. The subcarrier allocation, which is performed in the first step, is solely responsible for the fairness of the solution. Conducting next a waterfilling which naturally favors strong users is questionable if not contradictory. Finally, for lack of a better measure, suboptimal algorithms are often evaluated and compared based on the total sum rate achieved, which further supports the idea that high sum rates imply an efficient utilization of available power.

This paper analyzes the relationship between subcarrier allocations and waterfilling. We prove that any subcarrier allocation has a so-called *optimality polyhedron*, a natural set of waterlevels and thus, weight vectors, which includes the weights $\mathbf{w}$ for which the obtained rate vectors are closest to the boundary of the capacity region (as indicated by $q(\mathbf{w}, \mathbf{r})$). Based on a special class of allocations we show that it is possible to obtain capacity-achieving waterlevels for an allocation without reverting to convex optimization. In fact, the algorithms of Section VI not only offer satisfying results, but at the same time a low complexity even when compared to algorithms utilizing much simpler schemes.

III. A Necessary Condition for Optimality

In the following, problem (5) is analyzed for a fixed weight vector $\mathbf{w}$. As two users i and j with $w_i = w_j$ can be regarded as a single user for the purpose of multi-user resource allocation, we assume $w_i \neq w_j$ for all $i \neq j$ without loss of generality.

Definition 1. *A function $a\colon \{1, \ldots, N\} \to \{1, \ldots, K\}$, together with the condition*

$$p_{k,n} = 0 \quad \forall\, k \neq a(n),\, n = 1, \ldots, N$$

is called an allocation.

Given an allocation a, the optimal solution is obtained by *multi-user waterfilling*:

$$p_{a(n),n} = \left(w_{a(n)}\nu - \frac{1}{c_{a(n),n}}\right)^+ \tag{7}$$

$$\text{with} \quad \nu \quad \text{s.t.} \quad \sum_{n=1}^{N} p_{a(n),n} = p_{\max},$$

where $x^+ = \max(x, 0)$. The value $w_k\nu$ is the individual waterlevel of user k and the waterlevels are proportional to $\mathbf{w}$. Given power distribution (7), rate vector $\mathbf{r}$ is computed from (1). Denote the obtained weighted sum rate $\mathbf{w}^T\mathbf{r}$ by $\mathsf{WF}(\mathbf{w}, a)$. This allows us to reformulate (6) for suboptimal solutions obtained by waterfilling over allocation a:

$$q(\mathbf{w}, a) = \frac{\mathsf{WF}(\mathbf{w}, a)}{\mathsf{WSRmax}(\mathbf{w})}. \tag{8}$$

Finding the maximum value of $\mathsf{WF}(\mathbf{w}, a)$ over all allocations $a \in K^N$ yields the optimal solution to the weighted sum rate maximization problem through an exhaustive search, which is computationally prohibitive. In Theorem 1 we take a first step to reduce the problem size.

Theorem 1 (Optimality Condition)**.** *Let weight vector $\mathbf{w}$, allocation a, subcarrier n with $p_{a(n),n} > 0$ and two users i and j with*

$$w_i > w_j \quad \textit{and} \quad w_i c_{i,n} \geq w_j c_{j,n} \tag{9}$$

be given. Then, $a(n) \neq j$ is a necessary condition for optimality.

Proof: Assume $a(n) = j$ with $p_{j,n} > 0$. We show that the allocation

$$a^*(x) = \begin{cases} a(x), & x \neq n, \\ i, & x = n \end{cases}$$

satisfies

$$\mathsf{WF}(\mathbf{w}, a) < \mathsf{WF}(\mathbf{w}, a^*). \tag{10}$$

It holds that

$$\mathsf{WF}(\mathbf{w}, a) = \sum_{n=1}^{N} w_{a(n)} \log(1 + p_{a(n),n} c_{a(n),n})$$

but any power allocation provides a lower bound for $\mathsf{WF}(\mathbf{w}, a^*)$ *as well:*

$$\mathsf{WF}(\mathbf{w}, a^*) \geq \sum_{n=1}^{N} w_{a^*(n)} \log(1 + p_{a(n),n} c_{a^*(n),n}).$$

Therefore,

$$\begin{aligned} &\mathsf{WF}(\mathbf{w}, a^*) - \mathsf{WF}(\mathbf{w}, a) \\ &\geq \sum_{n=1}^{N} w_{a^*(n)} \log(1 + p_{a(n),n} c_{a(n),n}) \\ &\quad - \sum_{n=1}^{N} w_{a(n)} \log(1 + p_{a(n),n} c_{a^*(n),n}) \\ &= w_i \log(1 + p_{j,n} c_{i,n}) - w_j \log(1 + p_{j,n} c_{j,n}). \end{aligned} \tag{11}$$

Evidently, (11) is zero for $p_{j,n} = 0$*. We omit index* n *and show that (11) is strictly increasing in* p_j*, which proves (10).*

$$\begin{aligned} &\frac{\partial}{\partial p_j} (w_i \log(1 + p_j c_i) - w_j \log(1 + p_j c_j)) \\ &= \frac{w_i c_i}{1 + p_j c_i} - \frac{w_j c_j}{1 + p_j c_j} \\ &= \frac{w_i c_i (1 + p_j c_j) - w_j c_j (1 + p_j c_i)}{(1 + p_j c_i)(1 + p_j c_j)} \\ &= \frac{\overbrace{w_i c_i - w_j c_j}^{\geq 0} + p_j c_i c_j (\overbrace{w_i - w_j}^{>0})}{(1 + p_j c_i)(1 + p_j c_j)} > 0. \end{aligned}$$

■

Lemma 1. *Theorem 1 implies that only the Pareto optimal elements of the set* $\{(w_i, w_i c_{i,n}) \mid i = 1, \ldots, K\}$ *are potentially optimal. Accordingly, we call a user-subcarrier-pair* (j, n) *Pareto optimal when there is no user* i *such that (9) holds.*

Lemma 2. *Theorem 1 reduces the size of the solution space of the exhaustive search from* K^N *to*

$$\prod_{n=1}^{N} \left| \{ j \mid (j, n) \text{ is Pareto optimal} \} \right|.$$

IV. The Location of Capacity-Achieving Waterlevels

Most suboptimal algorithms are focused on finding good subcarrier allocations. Once one of these is found, the problem of power distribution remains. It is well-known that multi-user waterfilling provides the optimum solution. However, it is evident from the waterfilling formula (7) that the weight vector $\mathbf{w}$ has to be known to obtain the optimal individual waterlevels $w_k \nu$. Therefore, as long as one is not solving an explicit weighted sum rate problem, allocating subcarriers is only the first step.

To solve this dilemma, there are two approaches. The first approach simply assigns power in a single-user-waterfilling-fashion with a uniform waterlevel. Recall from Section II that this power assignment is the optimizer of the sum rate maximization problem, which heavily favors users with good channel quality.

The second approach is to compute the individual waterlevels based on the problem formulation or the choice of allocation. It is much more complicated and very problem-specific. See [4] for an example of waterlevel computation to achieve proportional fairness between users' data rates. However, to the best of our knowledge, there is no general theory about how to find capacity-achieving waterlevels and how to evaluate the results.

For notational simplicity only, we assume the weight vector $\mathbf{w}$ to be ordered, that is, $w_1 > w_2 > \ldots > w_K$. Additionally, we make the reasonable assumption that $\mathbf{w}$ has to be chosen anti-proportional to the average channel gains of the users to promote fairness. Hence, we assume user 1 to have the worst average channel-to-noise-ratio, while user K has the best.

Recall that Theorem 1 reduces the number of potentially optimal allocations for a given weight vector. In practice, however, the task is to find a good weight vector for a given allocation. This is made possible by Theorem 2.

Theorem 2 (Optimality Polyhedron)**.** *Given an allocation* a*, define the polyhedron* $P(a)$ *by linear inequalities*

$$w_j - w_{j-1} \leq 0 \tag{12}$$

$$w_i - \phi_{j,i} w_j \leq 0 \tag{13}$$

for $i, j \in \{1, \ldots, K\}$*,* $i < j$*, where*

$$\phi_{j,i} = \min_{\substack{n \\ a(n)=j}} \frac{c_{j,n}}{c_{i,n}}. \tag{14}$$

Then, for any weight vector $\mathbf{w}$ *not in the interior of* $P(a)$*, there exists an allocation* a^* *with*

$$\mathsf{WF}(\mathbf{w}, a) < \mathsf{WF}(\mathbf{w}, a^*). \tag{15}$$

Proof: Assume $\mathbf{w}$ *is not in the interior of* $P(a)$*. Inequalities (12) are fulfilled by the ordering assumption, therefore at least one of the inequalities (13) is violated. Pick subcarrier* n *with* $a(n) = j$*,* $\phi_{j,i} = \frac{c_{j,n}}{c_{i,n}}$ *and*

$$w_i - \frac{c_{j,n}}{c_{i,n}} w_j \geq 0.$$

Clearly, this is equivalent to

$$w_i c_{i,n} \geq w_j c_{j,n}$$

and therefore (9) is satisfied. Apply Theorem 1 to conclude that

$$a^*(x) = \begin{cases} a(x), & x \neq n, \\ i, & x = n. \end{cases}$$

satisfies (15).

■

Lemma 3. *We make two observations:*

i) *An allocation* a *with* $P(a) = \emptyset$ *is always suboptimal.*
ii) *If* $P(a) \neq \emptyset$*, then* $\mathbf{m} = (\frac{1}{K}, \ldots, \frac{1}{K})$ *is a corner point of* $P(a)$*.*

Refer to Figure 1 for a typical example of an optimality polyhedron. The relevance of weight vector $\mathbf{v}$ is explained in Section V.

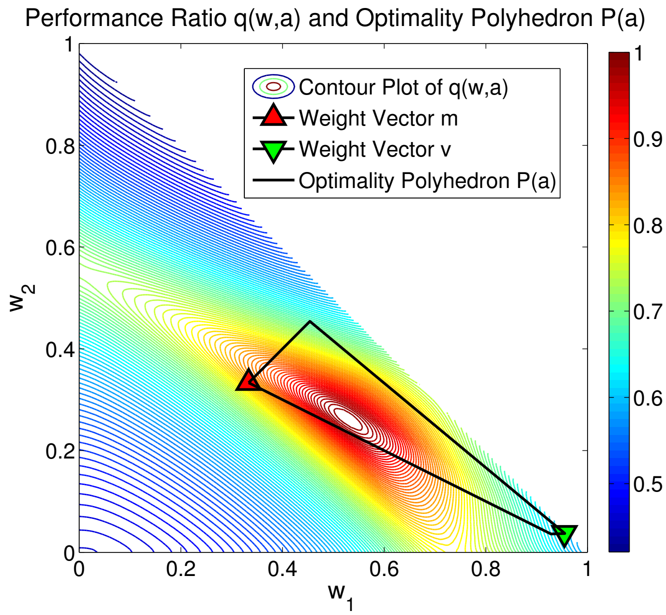

Fig. 1. Optimality Polyhedron for $K = 3$ with $w_3 = 1 - w_1 - w_2$.

V. Weighted Allocations

In this section we introduce a class of subcarrier allocations a for which $P(a) \neq \emptyset$ always holds. These allocations have a geometric property which proves useful in Section VI, where an algorithm for computing capacity-achieving waterlevels is designed. In addition, the computation of these allocations neither requires sorting nor an iterated assignment of subcarriers, leading to very low computational complexity.

Definition 2. *Given a weight vector* $\mathbf{v} \in \mathbb{R}^K_{\geq 0}$, *define the* weighted allocation $a_{\mathbf{v}}$ *by*

$$a_{\mathbf{v}}(n) = \arg\max_k v_k c_{k,n}, \quad n = 1, \dots, N.$$

Theorem 3 (Optimality of weighted allocations)**.** *For any* $\mathbf{v}$ *with* $v_1 \geq \ldots \geq v_K$, *the subcarrier allocation* $a_{\mathbf{v}}$ *satisfies* $\mathbf{v} \in P(a_{\mathbf{v}})$. *In particular,* $P(a_{\mathbf{v}}) \neq \emptyset$.

Proof: We show $\mathbf{v} \in P(a_{\mathbf{v}})$. *By assumption, linear inequality (12) holds for* $\mathbf{v}$. *Inequality (13) is equivalent to*

$$\begin{aligned} & \frac{v_i}{v_j} \leq \min_{\substack{n \\ a_{\mathbf{v}}(n)=j}} \frac{c_{j,n}}{c_{i,n}} \\ \Leftrightarrow \quad & \frac{v_i}{v_{a_{\mathbf{v}}(n)}} \leq \frac{c_{a_{\mathbf{v}}(n),n}}{c_{i,n}} \qquad \forall n \\ \Leftrightarrow \quad & v_i c_{i,n} \leq v_{a_{\mathbf{v}}(n)} c_{a_{\mathbf{v}}(n),n} \quad \forall n \end{aligned}$$

which holds because

$$v_{a_{\mathbf{v}}(n)} c_{a_{\mathbf{v}}(n),n} = \max_k v_k c_{k,n} \quad \forall n$$

by definition of $a_{\mathbf{v}}$. ■

Next, we analyze the relevance of the distinct polyhedron points $\mathbf{m}$ and $\mathbf{v}$ provided by Theorem 3. The point $\mathbf{m}$ is clearly a corner point of $P(a_{\mathbf{v}})$, as it fulfills all inequalities (12) with equality. However, $\mathbf{v}$ has a similar property with respect to inequalities (13). To see this, for $i < j$, define

$$\gamma = \min_{\substack{n \\ a_{\mathbf{v}}(n)=j}} (v_j c_{j,n} - v_i c_{i,n}).$$

Clearly, the constant γ depends on the number of subcarriers involved. Assuming a random CNR matrix, the more subcarriers are allocated to user j, the closer γ is to zero. In fact, applying (14) yields

$$\lim_{\gamma \to 0} \phi_{j,i} = \frac{v_i}{v_j},$$

which shows that $\mathbf{v}$ fulfills inequalities (13) approximately with equality. Therefore, $\mathbf{v}$ is a good approximation of another corner point of $P(a_{\mathbf{v}})$. An example of the distinct location of $\mathbf{v}$ in $P(a_{\mathbf{v}})$ can be seen in Figure 1.

Summarizing, this means that we have not only located two points $\mathbf{m}$ and $\mathbf{v}$ in $P(a_{\mathbf{v}})$, but these two points can, in a way, be considered "opposing" corner points. The relationship between $\mathbf{m}$ and $\mathbf{v}$ supports the approach to search for an optimal solution only on the line connecting these points.

Lemma 4. *For any weighted subcarrier allocation* $a_{\mathbf{v}}$, *the points* $\mathbf{m}$ *and* $\mathbf{v}$ *lie in* $P(a_{\mathbf{v}})$, *which is a polyhedron and therefore convex. It follows that the line* L *connecting* $\mathbf{m}$ *and* $\mathbf{v}$ *also runs through* $P(a_{\mathbf{v}})$:

$$L(\lambda) = (1 - \lambda)\mathbf{m} + \lambda \mathbf{v} \in P(a_{\mathbf{v}}) \quad \forall \lambda \in [0, 1].$$

VI. Algorithm Design

The algorithm we present in this section is a two-step algorithm that first assigns the subcarriers to the users and deals with the optimal waterlevels afterwards. However, Theorem 2 and Theorem 3 imply a natural correspondence between the choice of allocation $a_{\mathbf{v}}$ and the choice of waterlevels $\mathbf{w}$. First, we deal with the problem of locating a capacity-achieving waterlevel for a given weighted allocation $a_{\mathbf{v}}$ on the line of waterlevels between $\mathbf{m}$ and $\mathbf{v}$. In the spirit of (8), we formulate the objective as

$$\underset{\lambda \in [0,1],\, \mathbf{w} = L(\lambda)}{\text{maximize}} \quad q(\mathbf{w}, a_{\mathbf{v}}) = \frac{\mathsf{WF}(\mathbf{w}, a_{\mathbf{v}})}{\mathsf{WSRmax}(\mathbf{w})} \tag{16}$$

Of course, no resource allocation algorithm can compute the denominator $\mathsf{WSRmax}(\mathbf{w})$ without resorting to high-complexity convex optimization techniques. Therefore, we have arrived at a particularly challenging hurdle in optimization: The problem of optimizing over an *unknown* objective function.

We introduce a way out of this dilemma with Algorithm 1, the core algorithm for capacity-achieving waterlevels (CAWL). Given $a_{\mathbf{v}}$, the goal is to find a weight vector $\mathbf{w} = L(\lambda)$ such that $q(\mathbf{w}, a_{\mathbf{v}})$ in (16) is maximized. For notational simplicity, we define

$$q(\lambda, \psi) = q(L(\lambda), a_{L(\psi)}),$$

such that the weights $\mathbf{w} = L(\lambda)$ and weighted allocations $a_{L(\psi)}$ are identified with their parameters λ and ψ, respectively.

Algorithm 1 CAWL

Require: $\mathbf{v} \geq 0,\ v_1 + \ldots + v_K = 1$.
  Compute[1] $\varepsilon_1,\ \varepsilon_2 > 0$ based on $(c_{k,n})$ and $\mathbf{v}$.
  for $i = 1, 2$ **do**
    $\psi_i \leftarrow 1 + (-1)^i \varepsilon_i$
    $L(\psi_i) \leftarrow (1 - \psi_i)\mathbf{m} + \psi_i \mathbf{v}$
    $a_{L(\psi_i)}(n) \leftarrow \arg\max_k L(\psi_i)_k c_{k,n}, \quad n = 1, \ldots, N$
  end for
  $\delta_0 \leftarrow \mathsf{WF}(\mathbf{m}, a_{L(\psi_1)}) - \mathsf{WF}(\mathbf{m}, a_{L(\psi_2)})$
  $\delta_1 \leftarrow \mathsf{WF}(\mathbf{v}, a_{L(\psi_1)}) - \mathsf{WF}(\mathbf{v}, a_{L(\psi_2)})$
  $\lambda^* \leftarrow \delta_0 / (\delta_0 - \delta_1)$
  $\mathbf{w} \leftarrow L(\lambda^*)$
  return $\mathbf{w}$

In the following, the workings of Algorithm 1 are explained in detail. To simplify notation, define the function

$$f(\psi) = \arg\max_{\lambda \in [0,1]} q(\lambda, \psi), \quad \psi \geq 0. \tag{17}$$

Based on the convexity of $\mathcal{C}$ and the interpretation of $q(\lambda, \psi)$ as a measure of distance, we assume $q(\lambda, \psi)$ to be unimodal (quasiconcave) for fixed ψ. Furthermore, we assume $f(\psi)$ to be monotonically increasing. Figure 2 shows how $f(\psi)$ maximizes each $q(\lambda, \psi)$, resulting in a monotonic function. Clearly, $f(1)$ is the optimal solution to our optimization problem as $a_{\mathbf{v}} = a_{L(1)}$.

We pick two allocations $a_{L(\psi_1)}$ and $a_{L(\psi_2)}$ surrounding $a_{\mathbf{v}}$. Given $\varepsilon_1, \varepsilon_2 > 0$, define $\psi_1 = 1 - \varepsilon_1$ and $\psi_2 = 1 + \varepsilon_2$. In the following, we show that it is possible to obtain a very good approximation of the optimal solution with the help of these neighboring allocations. Starting with the monotonicity of $f(\psi)$,

$$f(\psi_1) \leq f(1) \leq f(\psi_2)$$

with

$$q(f(\psi_1), \psi_1) \approx q(f(1), 1) \approx q(f(\psi_2), \psi_2)$$

for small values of ε_1 and ε_2. Thus,

$$\begin{aligned} &q(f(\psi_1), \psi_1) - q(f(\psi_1), \psi_2) \\ &\approx q(f(\psi_2), \psi_2) - q(f(\psi_1), \psi_2) \geq 0 \end{aligned}$$

and

$$\begin{aligned} &q(f(\psi_2), \psi_1) - q(f(\psi_2), \psi_2) \\ &\approx q(f(\psi_2), \psi_1) - q(f(\psi_1), \psi_1) \leq 0 \end{aligned}$$

by (17). As $q(\lambda, \psi)$ is continuous for every ψ, the difference $q(\lambda, \psi_1) - q(\lambda, \psi_2)$ is also continuous. By the intermediate value theorem there exists $\lambda^* \in [f(\psi_1), f(\psi_2)]$ with

$$q(\lambda^*, \psi_1) - q(\lambda^*, \psi_2) = 0.$$

[1]The computation of ε_1 and ε_2 is not complex, however, $a_{L(\psi_1)} \neq a_{\mathbf{v}} \neq a_{L(\psi_2)}$ has to be ensured while maintaining small values for ε_1 and ε_2. We omit the details due to lack of space.

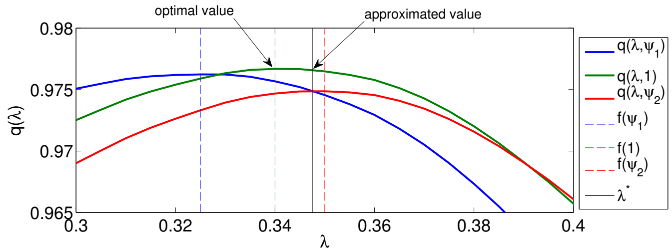

Fig. 2. Approximating $f(1)$ with the point of intersection λ^*.

Finally,

$$\begin{aligned} &q(\lambda^*, \psi_1) = q(\lambda^*, \psi_2) \\ \overset{(8)}{\Leftrightarrow}\ &\mathsf{WF}(L(\lambda^*), a_{L(\psi_1)}) = \mathsf{WF}(L(\lambda^*), a_{L(\psi_2)}), \end{aligned}$$

which means that the root of the unknown function $q(\lambda, \psi_1) - q(\lambda, \psi_2)$ coincides with the root of the readily computed function

$$\delta(\lambda) = \mathsf{WF}(L(\lambda), a_{L(\psi_1)}) - \mathsf{WF}(L(\lambda), a_{L(\psi_2)}),$$

which denotes the difference in weighted sum rate between allocations $a_{L(\psi_1)}$ and $a_{L(\psi_2)}$ for waterlevel $L(\lambda)$. By the optimality condition from Theorem 1 it holds that

$$\delta(0) > 0 \quad \text{and} \quad \delta(\lambda) < 0 \quad \forall\, \lambda \geq \psi_2. \tag{18}$$

Algorithm 1 is based on the additional assumption that δ is monotonically decreasing. The monotonicity of δ is not only supported by (18), but also from a geometrical point of view: The allocation $a_{L(\psi_1)}$ is closer to $\mathbf{m}$ than $a_{L(\psi_2)}$ and therefore provides better results for waterlevel $\mathbf{m} = L(0)$. Once we increase λ, the advantage of allocation $a_{L(\psi_1)}$ decreases and at some (unknown) point λ^* both allocations achieve the same weighted sum rate, i.e., $\delta(\lambda^*) = 0$. From this point on, allocation $a_{L(\psi_2)}$ provides better weighted sum rates.

Lemma 5. *For the monotonically decreasing function δ defined above and λ^* with $\delta(\lambda^*) = 0$, it follows that*

$$\lambda^* \in [f(\psi_1), f(\psi_2)],$$

which means that up to a small approximation error of

$$|\lambda^* - f(1)| \leq f(\psi_2) - f(\psi_1),$$

λ^* is equal to $f(1)$, the optimizer of (16).

Refer to Figure 2 for an example of the approximation of the optimizer $f(1)$ by the point of intersection λ^*. Keep in mind that not the plotted functions themselves, but only their points of intersection are computable by the algorithm.

Instead of exactly computing the root of δ, a linear approximation technique with values $\lambda = 0$ and $\lambda = 1$ is applied. With the help of four waterfillings, $\delta_0 = \delta(0)$ and $\delta_1 = \delta(1)$ are computed. We approximate the root of δ by

$$\lambda^* = \delta_0 / (\delta_0 - \delta_1).$$

It is possible to iterate this root-finding-process at the cost of computing two waterfilling solutions per iteration. However, this only marginally improves simulation results which we take as evidence for the approximate linearity of δ.

The computed waterlevel $\mathbf{w} = L(\lambda^*)$ is an accurate low-complexity approximation of the waterlevel that maximizes (16), which is the waterlevel that intrinsically and optimally corresponds to allocation $a_{\mathbf{v}}$ based on the ideas of Section II.

We conclude this section with Algorithm 2, the most natural way to apply the CAWL-algorithm. The weight vector $\mathbf{v}$ is chosen anti-proportional to the average CNRs. This quasi-normalization averages the amount of subcarriers per user in the case of identically distributed normalized channel gains.

Algorithm 2 NORM-CAWL

Require: $(c_{k,n}) \in \mathbb{R}^{K\times N}_{>0}$
$v_k \leftarrow (\sum_{n=1}^{N} c_{k,n})^{-1}, \quad k = 1, \ldots, K$
$\mathbf{v} \leftarrow (v_1, \ldots, v_K)/(\sum_{k=1}^{K} v_k)$
$a_{\mathbf{v}}(n) \leftarrow \arg\max_k v_k c_{k,n}, \quad n = 1, \ldots, N$
$\mathbf{w} \leftarrow \mathsf{CAWL}(\mathbf{v})$
return $\mathsf{WF}(\mathbf{w}, a_{\mathbf{v}})$

Every call to $\mathsf{WF}(\mathbf{w}, a)$ in Algorithm 1 can be solved by elementary arithmetics with a worst-case complexity of $\mathcal{O}(N)$ regardless of K as shown in [5] for single-level single-constraint waterfilling problems. This reduces the impact of K to the computation of weighted allocations, which is a series of low-level comparisons of constant complexity $\mathcal{O}(KN)$. Algorithm 2, calling $\mathsf{WF}(\mathbf{w}, a)$ five times and computing a total of three weighted allocations, therefore has a worst-case complexity of $\mathcal{O}(KN)$, with the (relatively small) value of K having a much lower impact on the average runtime, which is exceptionally low because no sorting operations are being invoked.

VII. Simulation Results

In this section we present simulation results for Algorithm 2. We assume K users to be uniformly positioned in a circular cell. The minimal distance to the base station is $d = 50\,\text{m}$, while the maximum distance is $d = 1000\,\text{m}$. With a path loss exponent of $\alpha = 2$, this leads to a maximum difference of 26 dB between the channel gains of best and worst users. For simplicity, we set the average channel gain on the boundary of the cell to 0 dB and the total power is set to $p_{\max} = 10N$, such that an average signal-to-noise-ratio of 10 dB per subcarrier is guaranteed even on the boundary of the cell.

The channel gains (in decimal notation) of each user are assumed to be $\text{Exp}(\frac{1}{\mu})$-distributed, where $\mu = 10^6 d^{-2}$ denotes the average CNR based on the path loss model. Note that this model does not take correlation between subcarriers into account.

Figure 3 shows the simulation results for Algorithm 2. To compute the denominator in (8) we employ the WSRmax-algorithm of [3]. The fact that the performance slowly degrades with the number of users is quite natural: The more users are positioned in the cell, the larger the solution space becomes and large channel gain differences between users become more likely and further complicate the subcarrier allocation. Nonetheless, note that the worst average performance is still above 97.5% of the optimum value.

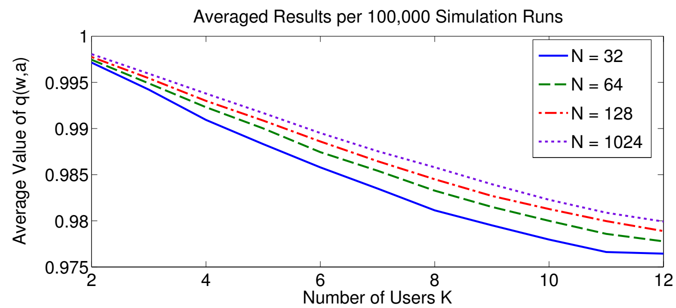

Fig. 3. Averaged Results of NORM-CAWL.

VIII. Conclusion

Evaluating resource allocation algorithms based on the achieved sum rate is not always adequate, as the only solutions where spent power is fully utilized are the ones that are capacity-achieving in the sense that they approach the boundary of the capacity region. However, no suboptimal algorithm can verify if this is the case. We prove that weighted allocations have a natural geometric correspondence to their optimality polyhedron. Thus, they are the perfect candidates to search for a capacity-achieving waterlevel on a straight line in $(K-1)$-dimensional space. This leads to a one-dimensional optimization problem with an unknown objective function. Despite this obstacle, we show that it is possible to achieve results that are close to optimal and therefore capacity-achieving with the design of a low-complexity algorithm. This paper focuses on the theoretical foundation of capacity-achieving waterlevels. The most practical results are the incorporation of weighted allocations and their algorithmic uses, and it will be interesting to see how well the concepts of this paper carry over to a more realistic system, where fairness, power efficiency and algorithm speed are just some of many crucial parameters. As a next step, we plan a direct comparison with other suboptimal resource allocation algorithms.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley, 1991.

[2] R. Cheng and S. Verdu, "Gaussian multiaccess channels with ISI: capacity region and multiuser water-filling," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 773–785, May 1993.

[3] K. Seong, M. Mohseni, and J. Cioffi, "Optimal resource allocation for OFDMA downlink systems," in *IEEE International Symposium on Information Theory*, July 2006, pp. 1394–1398.

[4] I. Wong, Z. Shen, B. Evans, and J. Andrews, "A low complexity algorithm for proportional resource allocation in OFDMA systems," in *IEEE Workshop on Signal Processing Systems*, Oct. 2004.

[5] D. Palomar and J. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 686–695, Feb. 2005.