Multi-Channel Power Allocation for Maximizing Energy Efficiency in Wireless Networks

Peter He, *Member, IEEE*, Shan Zhang, *Member, IEEE*, Lian Zhao, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—This paper aims at solving two classes of energy efficiency (EE) maximization problems in multiple channels wireless communication systems. Firstly, the EE maximization problem with sum power constraint is solved based on the geometric water-filling approach; and secondly, the approach is extended into the EE maximization problem with additional least throughput requirement constraint. Our proposed algorithms make use of the water-filling structure of the optimal solution and provide exact and computation efficient solution to the energy-efficient power allocation problems. The proposed algorithms also have excellent scalability, which is applicable for large scale wireless communication systems. Optimality of the proposed algorithms is strictly proved, and the proposed algorithms only require low degree polynomial computational complexity. Numerical results are presented to demonstrate the efficiency of the proposed algorithms. To the best of our knowledge, no prior algorithms in the existing literature could provide such solutions to the EE maximization problems under the merit of exactness and the efficiency.**

Keywords

Energy efficiency (EE), power allocation, water-filling, optimal solution, non-linear fractional optimization.

I. Introduction

As mobile networks continuously densify, the huge energy consumption has brought a heavy burden to operators, which may become the bottleneck of future network development [1], [2]. Therefore, green communications have drawn increasing research interests during recent years [3], [4]. Specifically, Energy Efficiency (EE), *i.e.*, the amount of transmitted data per unit energy consumption, has been considered as one of the key performance metrics in the upcoming 5G era and beyond [5], [6], [7].

A fundamental question for green communication is how to maximize the energy efficiency with sum power constraint [8], [9]. Conventional radio resource management has investigated how to maximize the system throughput with sum power constraint [10]–[13]. Some recent work has also explored how to minimize the transmit power consumption while satisfying throughput requirements [14], [15]. However, the EE maximization problem is of a non-linear fractional optimization, the objective functions of which own the numerators of a non-linear function form and the denominators of a sum power form [16]. Thus, it cannot be solved directly by the algorithms for the throughput or sum power optimizations. Also, a set of the Karush-Kuhn-Tucker (KKT) conditions [17] has not directly been used for solving these energy efficiency maximization problems, due to the fractional form of the EE maximization problem. As a result, existing algorithms seldom directly solve these EE maximization problems, including the widely-adopted Dinkelbach method [18], [19].

Some good efforts have been made for EE maximization from different perspectives [20]–[28] and the references cited in. A typical approach was to investigate convergence of the algorithm from the Dinkelbach method [20]–[23]. In [20], a combinatorial optimization problem was formulated to maximize the joint transmitter and receiver energy efficiency. Then a new divide-and-conquer approach was introduced to find a sub-optimal solution [20, p. 2728] to the energy efficiency maximization problem with the minimum throughput constraints. However, these state-of-the-art algorithms still show the limitations of high complexity and sub-optimality. In addition, there are other works such as [27] and [28], proposing different algorithms from Dinkelbach method with the ϵ-optimality.

In this paper, we investigate the geometry water-filling (GWF) approach to maximize EE in wireless systems with multiple channels, considering that the total power consumption of all the channels cannot exceed the budget. By embedding the geometric water-filling approach, a low-complexity power allocation algorithm, namely energy-efficient jumping water-filling (EE-JWF), is developed to obtain the exact solution. In addition, the algorithm is extended to solving the generalized case, where the minimal throughput is required as an additional constraint, considering the quality of service (QoS) requirements of mobile users. Although the water-filling approaches have been widely adopted in radio resource management to maximize the system throughput or minimize the total power consumption, this is the first exploration to maximize the energy efficiency. The rationale of the proposed EE-JWF algorithm is to use the water-filling-like architecture of the optimal solution, and it can locate the global optimal water level accurately with low complexity.

Peter He and Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1 (email: {z85he, sshen}@uwaterloo.ca).

Shan Zhang (corresponding author) is with the School of Computer Science and Engineering, Beihang University, Beijing, China, 100191. E-mail: zhangshan2007@gmail.com

Lian Zhao is with Dept. of Electrical and Computer Engineering, Ryerson University, Ontario, Canada, M5B 2K3. E-mail: l5zhao@ryerson.ca.

Specifically, the global optimal solution is obtained by first solving the local optimal ones, where the local optimal solutions form jumping water levels corresponds to each channel.

Compared with existing work on energy-efficient power allocation, the proposed algorithms own two distinct and important features, "exactness" and "low complexity". Exactness in this paper means that the error between our proposed solutions and the "theoretic" optimal solutions is the machine zero, no larger than 10^{-34} based on the standard of IEEE 754-2008 (for example, we treat $1.41421\cdots$ with 34 decimal digits as an exact representation of $\sqrt{2}$). Low complexity means that the proposed algorithms have low degree polynomial computational complexity with a concrete upper bound of the number of operations. In summary, the proposed algorithms can provide exact solution instead of sub-optimal ones, and with a low degree polynomial computational complexity. As a side note, although the proposed problems look simple, the existing algorithms do not have the mentioned two features to solve these problems, to the best of the authors' knowledge. Strict mathematical proofs and complexity analysis are provided to validate exactness and the low computational complexity of the proposed algorithms.

In the remaining of the paper, the statement of the proposed problem, and a review of our earlier proposed GWF are discussed in Section II. Section III is focused on solving the target EE maximizing problem with non-negative power and a sum power constraints, where the WW-JWF algorithm is proposed and discussed in details. Section IV extends the target problem introduced in Section II, with one more constraint of the throughput requirement, and then generalizes the proposed approach to compute the optimal power allocation solution to this extended problem. A review of the Dinkelbach method, as a comparison reference, and an analogous comparison, are provided. Section V presents numerical examples, performance and complexity analysis to illustrate the steps of the proposed algorithms and the advantages achieved of the proposed algorithms. Section VI concludes the paper.

II. Problem Statement

In this section, the target problem is introduced, followed by a brief review of our earlier proposed geometric water-filling approach [13] which is presented as a basis of the proposed method.

A simple form of the energy-efficiency maximization problems can be described by the following. Denote by P the total power budget (or upper power bound), $s_0 > 0$ the circuit power, s_i and a_i the allocated power and channel power gain of the ith channel [1], respectively, where $i = 1, \ldots, K$ and K is the total number of the channels. Letting $\{a_i\}_{i=1}^K$ be a sorted sequence with strictly monotonically decreasing (the indexes can be arbitrarily renumbered to satisfy this condition) without loss of generality, in which $a_i > 0, \forall i$, find a group of power $\{s_i\}$ to satisfy:

$$\begin{array}{ll} \max_{\{s_i\}_{i=1}^K} & \frac{\frac{1}{2}\sum_{i=1}^K \log(1+a_i s_i)}{s_0+\sum_{i=1}^K s_i} \\ \text{subject to:} & 0 \le s_i, \ \text{for } i = 1, \ldots, K; \\ & \sum_{i=1}^K s_i \le P, \end{array} \tag{1}$$

where the logarithm function $\log$ is assumed to be base 2 unless specified otherwise.

A. Concept of Water Tank and Geometric Relations of the Variables

A water tank is shown in Fig.1(a) with K steps/stairs, corresponding to the K channels. For the equally weighted case, each step/stair has a unit width. Let d_i denote the "step depth" of the ith stair, *i.e.*, the height of the ith step to the bottom of the tank, given as:

$$d_i = \frac{1}{a_i}, \quad i = 1, 2, \ldots, K. \tag{2}$$

Since the sequence $\{a_i\}$ is sorted in monotonically decreasing order, the step depth of the stairs indexed by $\{1, \cdots, K\}$ is monotonically increasing. When water (power) P is poured into the tank, a water level μ is obtained. The throughput-optimal power allocation to each channel corresponds to the area above the stair up to the water level[2].

GWF (Geometric Water Filling) algorithm was proposed in [13]. The main idea is summarized as follows. Let k^* denote the index of the highest (shallowest) step under water:

$$k^* = \max\left\{k \middle| P_u(k) > 0, \ 1 \le k \le K\right\}, \tag{3}$$

where $P_u(k)$ is a function in k, denoting the whole water volume above the kth step. From the geometric relationship, $P_u(k)$ can be obtained by

$$P_u(k) = \left[P - \sum_{i=1}^{k-1}\left(\frac{1}{a_k} - \frac{1}{a_i}\right)\right]^+, \quad \text{for } 1 \le k \le K. \tag{4}$$

Then the power allocated to the k^* step is

$$s_{k^*} = \frac{1}{k^*} P_2(k^*), \tag{5}$$

and the completed solution is given by

$$s_i = \begin{cases} s_{k^*} + \frac{1}{a_{k^*}} - \frac{1}{a_i}, & 1 \le i \le k^*; \\ 0, & k^* < i \le K. \end{cases} \tag{6}$$

The GWF algorithm is denoted by GWF($\{a_k\}_{k=1}^K, P$), *i.e.*, the mapping from $\{\{a_k\}_{k=1}^K, P\}$ to $\{k^*, P_u(k^*)\}$,

For a general weighted case, the objective function of problem (1) can be rewritten as

$$\max_{\{s_i\}_{i=1}^K} \frac{\frac{1}{2}\sum_{i=1}^K w_i \log(1+a_i s_i)}{s_0 + \sum_{i=1}^K s_i}, \tag{7}$$

[1] We assume perfect knowledge of channel power gain, which can be obtained by advanced channel estimation technologies.

[2] $S_3{}^*$ is denoted by the shadowed area. Since the width of the steps is one, the area is equivalent as the height as shown. This equivalence of the area with the corresponding height is used throughput the paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2803126, IEEE Transactions on Vehicular Technology

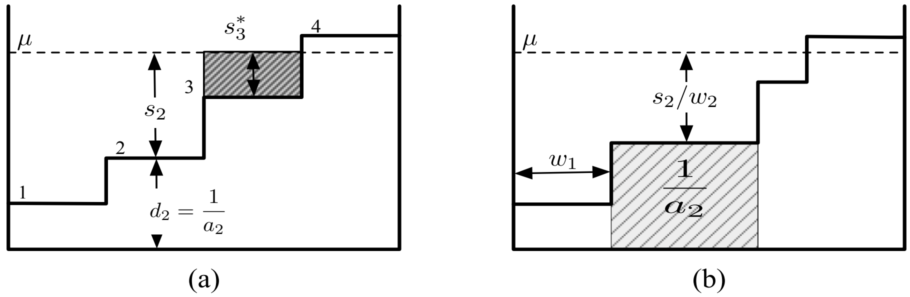


Fig. 1: Illustration for a water tank. (a) Water level step $k^* = 3$, allocated power for the third step s_3^*, and step/stair depth $d_i = \frac{1}{a_i}$. (b) The weighted case, the width of the ith step is denoted as w_i.

where the weight w_i reflects the importance of user/channel i. In this case, the width of the ith step depicts the weight w_i. The area right above this step to the water surface denotes the power allocated to the ith step, s_i. The height of the step to the water surface is then s_i/w_i. The area below the ith step to the bottom of the tank is $1/a_i$, as shown in Fig. 1(b). The depth of the ith step is then $1/(a_i w_i)$. Equipped with these geometric relations, the throughput-optimal power allocation can be obtained following the same idea of the equally-weighted case. Please refer to [13] for more details.

III. Solving EE Maximization Problem

In this section, we propose algorithms based on geometry water filling to solve the energy efficiency maximization problem. The intuition is that the EE-optimal power allocation satisfies the water-filling-like structure with the water tank model, whereas the key is to determine the water level. In our proposed algorithm, we start from solving local EE-optimal power allocation when the water level is constrained between two neighboring step depths. The global optimal solution, at which the objective function achieves the global maximum value, can be obtained by selecting the one from the local optimal solutions. As a side note, here the mentioned local optimal solution is different from the regular one discussed in any textbook. It means the global optimal solution(s) over a compact subset of the feasible set.

A. Local EE Optimal Solution

We still use a water tank with unit width and monotonically increasing steps to illustrate the geometric relationship of the variables as in Fig. 2. Let K denote the total number of the steps in the tank. Assume a certain amount of water is poured into the tank, making water level μ between the Nth and the $(N+1)$th step.

For the ith step ($i \leq N$), the power allocated is $s_i = \mu - d_i$. An auxiliary variable $\triangle s$, shown as the shadowed area in Fig. 2, denotes the entire volume of the water (total power) above the Nth step. Since each step is assumed to have a unit width, the allocated power for the Nth step, is

$$s_N = \frac{\triangle s}{N}. \tag{8}$$

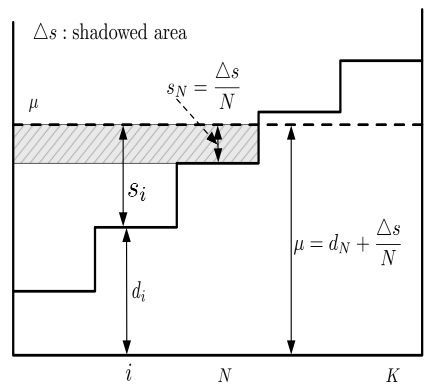


Fig. 2: Illustration of geometric relation and auxiliary variable $\triangle s$.

From the geometric relationship, $\triangle s$ has a domain, denoted by $[\triangle s_{\min}, \triangle s_{\max}]$, where

$$\triangle s_{\min} = 0, \quad \triangle s_{\max} = N\left(d_{N+1} - d_N\right). \tag{9}$$

$\triangle s_{\min}$ occurs when the water-level μ is at the Nth step, and $\triangle s_{\max}$ happens when the water-level μ reaches the $(N+1)$th step. For a given N, we introduce a function $g(\triangle s)$, where $\triangle s$ is the variable with the geometric meaning shown as Fig. 2,

$$\begin{aligned} g(\triangle s) =& (\triangle s + d_N \cdot N) \log\left(d_N + \frac{\triangle s}{N}\right) \\ &- \left(d_N + \frac{\triangle s}{N}\right) \sum_{k=1}^{N} \log(d_k) - S_T, \end{aligned} \tag{10}$$

where s_i is the allocated power for the ith channel,

$$s_i = (d_N - d_i) + \frac{\triangle s}{N}, \quad \forall\, 1 \leq i \leq N, \tag{11}$$

and S_T is the total power allocated plus circuit power consumption, as

$$S_T = s_0 + \sum_{i=1}^{N} s_i. \tag{12}$$

In Section IV-E, the insights of the function $g(\triangle s)$ will be further discussed. In Appendix A, it is shown that $g(\triangle s)$ exhibits a desired monotonically increasing property in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2803126, IEEE Transactions on Vehicular Technology

range of $(\triangle s_{\min}, \triangle s_{\max})$. Following Lemma gives the relation of the local optimal power allocation with the defined $g(\triangle s)$ function.

Lemma 1: With the domain of $\triangle s$, the EE-optimal power allocation is equivalent to

$$\begin{aligned} &\min |g(\triangle s)| \\ \text{s.t.} \quad &0 \leq \triangle s \leq N\left(d_{N+1} - d_N\right). \end{aligned} \tag{13}$$

Proof. See Appendix A, the final three cases of which just correspond to the relationship of (13). □

(13) is applied as a necessary and sufficient condition on the local optimal solutions to EE-maximization problem, based on which we can directly calculate the exact solution with high efficiency of the N loop operations. The function $g(\triangle s)$ is monotonically increasing, and the optimal solution to (13) is classified into three cases. In addition, Proposition 1 gives how to calculate $\triangle s$ when the optimal solution to (13) is not at the boundaries. *As a reminder, the concept of the mentioned local optimal solution has been defined in the beginning paragraph of this section.* For example, when minimizing $f(x) = x$ over [0, 2] with the two compact subsets [0, 1] and [1,2], $x = 1$ is the local optimal solution corresponding to the subset [1,2], $x = 0$ is the local optimal solution over [0,1], and $x = 0$ is the global optimal solution.

Fig. 3 depicts the monotonic trend of $g(\triangle s)$ and illustrates the possible three situations when solving (13). It is noted that we only need to evaluate the signs of $g(\triangle s)$ at the minimal and the maximal values of $\triangle s$ to branch into one of the three cases below:

(1) If $g(0) > 0$, as shown in Fig. 3 (a), the solution to (13) (making $g(\triangle s)$ closest to zero) is $\triangle s = 0$, and the complexity to evaluate $g(0)$ is $O(N)$ according to Eq. (10);
(2) If $g(\triangle s_{\max}) < 0$, as shown in Fig. 3 (b), the solution to (13) is $\triangle s = \triangle s_{\max}$.
(3) If $g(0) < 0$ and $g(\triangle s_{\max}) > 0$, as shown in Fig. 3 (c). The solution is given in Proposition 1 below.

Proposition 1: The solution to (13) when $g(0) < 0$ and $g(\triangle s_{\max}) > 0$ is computed through the following iteration,

$$\triangle s_{n+1} = \triangle s_n - \frac{g(\triangle s_n)}{g'(\triangle s_n)}, \quad \forall n \in \mathbb{Z}^+, \tag{14}$$

where $\mathbb{Z}^+$ is the set of non-negative integers. The subscript of $\triangle s_n$ is the iteration index. Details of the iteration steps see Appendix B.

Proof. See Appendix B. □

Remark 1. Proposition 1 calculates the local EE-optimal power level/allocation when the corresponding solution does not appear at the boundaries at the stairs, with the given parameters, such as $\{\{d_n\}, P\}$, *etc.*. Here, Proposition 1 applies (14) with N_t (derived in Appendix B) loop operations, to compute the exact solution to (13). The formula of $N_t(= N_1 + N_2)$ implies the low computational complexity of $5K[\max_{1\leq N\leq M}\{\frac{d_{N+1}}{d_N}\} + 35]$ basic (arithmetic, logical and basic function evaluation) operations, where $5K$ means that each of the N_t loops has $5K$ basic operations at most, without any exponential level in each of the system parameters. Therefore, Proposition 1 computing the local EE-optimal power level/allocation has the computational complexity of $\mathrm{O}(K)$.

B. Global EE-Optimal Power Allocation

By applying Lemma 1, N local EE-optimal power allocations can be obtained, forming jumping power levels corresponding to each channel. Then, the global optimal solution to the target problem (1) can be find within these N local optimal solutions. The Algorithm EE-JWF is described as follows.

Algorithm EE-JWF:

0) Pre-processing:

$$\begin{aligned} &\{k^*, P_u(k^*)\} = \text{GWF}(\{a_k\}_{k=1}^K, P), \\ &k^* \to M \text{ and } \tfrac{P_u(M)}{M} + \tfrac{1}{a_M} \to \tfrac{1}{a_{M+1}}, \end{aligned} \tag{15}$$

where the symbol "→" denotes the assignment operation.

1) Input:

$s_0, \{d_k = 1/a_k\}_{k=1}^M, n = 1$ and assigning a $K \times K$ matrix $\mathbf{S}$ a zero matrix.

2) Loop for n from 1 to M: for each loop, one of the following three branches is executed to update the nth row of the matrix $\mathbf{S}$:

2.1) If $g(\triangle s_{\min}) = g(0) > 0$,

$$S_{n,j} = \frac{1}{a_n} - \frac{1}{a_j}, \quad j = 1, \cdots, n, \tag{16}$$

and then go to Step 3); else

2.2) if $g(\triangle s_{\max}) = g\left(n\left(\frac{1}{a_{n+1}} - \frac{1}{a_n}\right)\right) < 0$,

$$S_{n,j} = \frac{1}{a_{n+1}} - \frac{1}{a_j}, \quad j = 1, \cdots, n, \tag{17}$$

and then go to item 3); else

2.3) if

$$g(0) \cdot g\left(n\left(\frac{1}{a_{n+1}} - \frac{1}{a_n}\right)\right) < 0, \tag{18}$$

then solve $\triangle s$ by applying Proposition 1 and

$$S_{n,j} = \frac{\triangle s}{n} + \left(\frac{1}{a_n} - \frac{1}{a_j}\right), \quad j = 1, \cdots, n. \tag{19}$$

3) If $n = M$, compute

$$n^* = \arg \max_{\{n|1\leq n\leq M\}} \left\{ \frac{\frac{1}{2}\sum_{i=1}^K \log_2(1 + a_i S_{n,i})}{s_0 + \sum_{i=1}^K S_{n,i}} \right\}, \tag{20}$$

and then output $S_{n^*,k}, \forall k$ as solution; else let $n+1 \to n$ and go to Step 2).

Proposition 2. EE-JWF outputs the exact optimal solution to (1) with a finite amount of computation.

Proof. EE-JWF enumerates the K intervals of $\{[d_n, d_{n+1}]\}_{n=1}^K$ to compute the local optimums. According

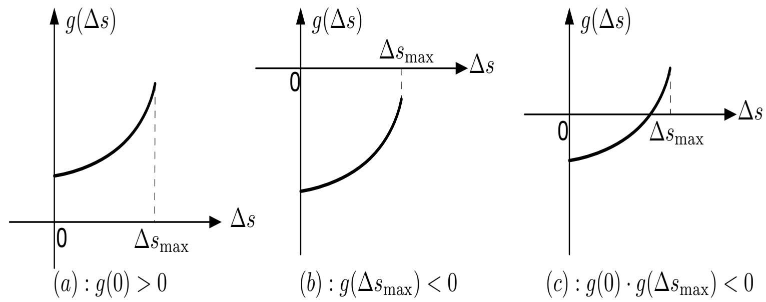


Fig. 3: Illustration for solving conditions (13).

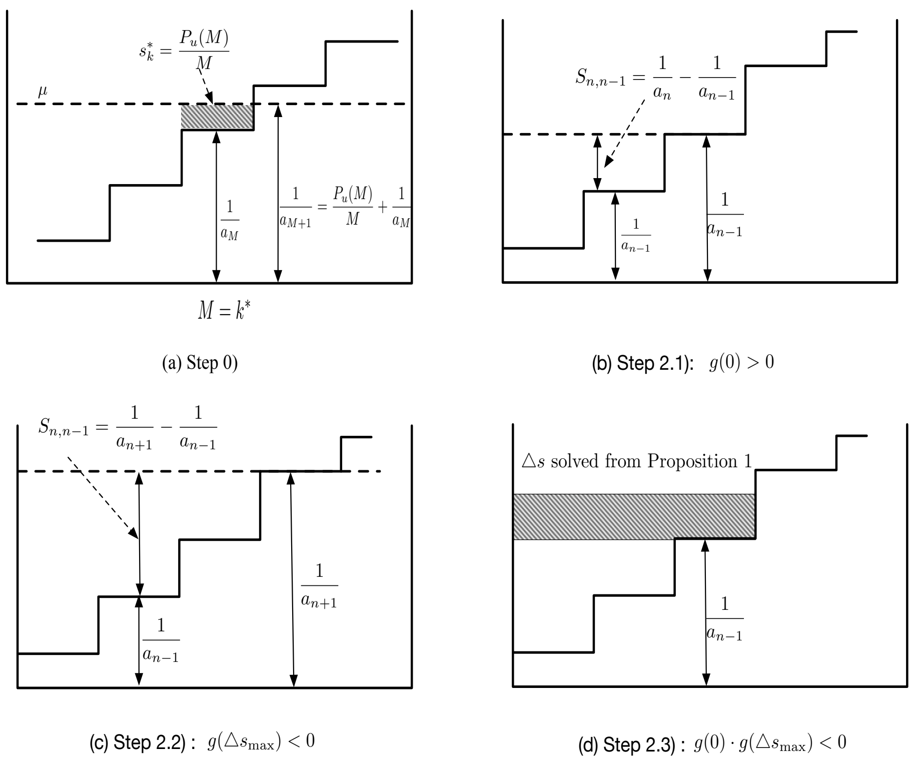


Fig. 4: Illustration for Algorithm EE-JWF.

to Lemma 1, without case 3 appearing, it is easy to see that $K+1$ evaluations, at most, of $g(x)$ in μ are needed. If case 3 appears, Proposition 1 implies that the solution of $g(x) = 0$ with the error of machine zero is obtained by a finite amount of computation. Therefore, the conclusion of Proposition 2 is true, noting that the solution to (1) exists and it must satisfy one of the three cases. □

In the Pre-processing Step, GWF algorithm solves water level step, k^*, and assigns it to M. The water level index, k^*, denotes the maximum number of the channels, we would consider for power allocation. For those channels with higher index ($i > k^*$), the power is allocated with zeros, as illustrated in Fig. 4 (a). The physical meaning of Pre-process Step is that only the channels with high channel gains are considered for power allocation, since allocating power to channels under bad condition is not energy-efficient. Note that Pre-processing in EE-JWF assumed all available power being allocated, which may not be EE-optimal. As a side note, $0 \le a_n \le 1, \forall n$, means $d_n \ge 1$. Thus, $0 \le d_M - d_1 = (d_M - d_{M-1}) + \cdots + (d_2 - d_1) \le P$. Then $d_M \le d_1 + P$ implies $N_t \le \frac{d_1+P}{d_1} + 35 \le 36 + \lceil P \rceil$, *i.e.*, $N_t = \mathrm{O}(1)$, indeed. Here $\lceil x \rceil$ means the ceiling function, and the power budget P is given for the proposed problems in this paper.

The second assignment operation (*i.e.*, Step 2)) further considers the cases when the available power is not completely allocated, based on system (13). For any loop index n in Step 2), it denotes that we only allocate the first n channels with positive power and assign zeros for the remaining channels, as illustrated in Fig. 4. Correspondingly, $\triangle s$ in the function $g(\triangle s)$ ranges as

$$\triangle s_{\min} = 0, \quad \triangle s_{\max} = n\left(\frac{1}{a_{n+1}} - \frac{1}{a_n}\right).$$

In Step 2, one of the branches is executed depending on the values of the function $g(\triangle s)$. In Branches 2.1) and 2.2), the values of $g(\triangle s_{\min}) = g(0) > 0$ or $g(\triangle s_{\max}) < 0$ respectively. In these two cases, there is no solution to $g(\triangle s) = 0$ subject to $\triangle s \in [\triangle s_{\min}, \triangle s_{\max}]$. The corresponding $\triangle s$ selection making $g(\triangle s)$ closest to 0 is $s_{\min} = 0$ and $s_{\max}$, respectively. The power allocation is shown as Fig. 4 (b) and (c) respectively. In 2.1), the water-level μ is at the nth step, and the corresponding non-zero power from step 1 to step $(n-1)$ is readily obtained from (16). In 2.2), the water-level μ is at the $(n+1)$th step, and then the non-zero power for the first n steps is obtained.

In Branch 2.3), the condition satisfies the existence of the feasible solution for $g(\triangle s) = 0$. Based on Proposition 1, the solution $\triangle s$ is solved as shown in Fig. 4 (d). From the geometric relationship, the power allocated to the first n channels is obtained, as shown in Fig. 4 (d).

In Step 3), when the loop index reaches M, the algorithm determines the n^*th row from the first M rows of the matrix $\mathbf{S}$ which leads to the maximum objective function value as the output of the algorithm. This step compares the M power allocation schemes and selects the one which can achieve the maximal EE. Based on the proof of Lemma 1, we can come to the conclusion that the optimal solution to the EE optimization problem presents water-filling-like power allocation architecture.

Remark 2. The proposed EE-JWF algorithm makes use of geometric relationships of the variables. This feature makes the proposed algorithm easily to be extended to the weighted EE maximization problem. When the width of the steps is considered, the water level μ is updated as

$$\mu = \frac{\triangle s}{\sum_{i=1}^{N} w_i} + d_N. \tag{21}$$

The corresponding $g(\triangle s)$ is written as

$$g(\triangle s) = \mu \sum_{i=1}^{N} \log\left(\frac{\mu}{d_i}\right)^{w_i} - S_T. \tag{22}$$

Remaining steps follow a similar approach to the unweighted case discussed above. This also reflects one of the advantages of using a geometric-based approach method.

Proposition 3: If there exists $g(\triangle s) = 0$ between steps n and $n+1$, Proposition 1 is applied to compute the solution $\triangle s$. The corresponding power level $\mu = (d_n + \triangle s/n)$ determines the global optimal solution to (1): $s_j = \mu - d_j$, for $j = 1, \ldots, n$; and $s_j = 0$, for $j = n+1, \ldots, K$. Furthermore, there is at most one interval which needs to apply Proposition 1 to compute $\triangle s$.

Proof. Denote by $G_n^{\min} = g(0)$ and $G_n^{\max} = g(\triangle s_{\max}) = g(n(d_{n+1} - d_n))$ for step n, where $n = 1, 2, \cdots, K$. Specifically, $G_n^{\min}$ corresponds to the value of $g(\triangle s)$ at the lowest water level ($\mu = d_n$ and $\triangle s = 0$), while $G_n^{\max}$ corresponds to the value of $g(\triangle s)$ at the highest water level ($\mu = d_{n+1}$ and $\triangle s = n(d_{n+1} - d_n)$). Notice that

$$\begin{aligned} G_n^{\min} &= d_n \sum_{i=1}^{n} \log\left(\frac{d_n}{d_i}\right) - \left[s_0 + \sum_{i=1}^{n}(d_n - d_i)\right] \\ &= d_n \sum_{i=1}^{n-1} \log\left(\frac{d_n}{d_i}\right) - \left[s_0 + \sum_{i=1}^{n-1}(d_n - d_i)\right] \\ &= G_{n-1}^{\max}. \end{aligned} \tag{23}$$

Furthermore, $g(\triangle s)$ is a monotone increasing function in terms of $\triangle s$ within range $[0, n(d_{n+1} - d_n)]$ at step n, for $n = 1, 2, \cdots, K$. Therefore, Eq. (23) indicates that the minimal value of $g(\triangle s)$ at step n is equal to the maximal value of $g(\triangle s)$ at step $n-1$. Thus, the value of $g(\triangle s)$ increases with the step index n. In the step 2 of EE-JWF algorithm, if case 3 holds for step n^* (i.e., $g(0) \cdot g\left(n^*\left(\frac{1}{a_{n^*+1}} - \frac{1}{a_{n^*}}\right)\right) < 0$), we have $G_{n^*}^{\min} < 0$ and $G_{n^*}^{\max} > 0$. Accordingly, $G_n^{\max} < 0$ for $n = 1, 2, \cdots, n^*-1$, and $G_n^{\min} > 0$ for $n = n^*+1, n^*+2, \cdots, K$. As a result, step 2.3 appears at most once in the EE-JWF algorithm. □

Accordingly, the proposed algorithm obtains the exact solution with great efficiency. In fact, the rationale of EE-JWF algorithm is two-fold, (1) the optimal solution presents water-filling-like power allocation architecture, and (2) the continuous water level is reduced to N jumping local optimal water levels, by applying the transformed problem (13) and the monotone property.

IV. Extended EE Maximization Problem

In this section, the EE maximization problem (1) is extended, adding the throughput (least) requirement, B, as one more constraint. Firstly, the extended problem statement is introduced; secondly, our previously proposed geometric water-filling for the sum power minimization (P-GWF) [15] is briefly reviewed; then, the algorithm that combines Algorithm EE-JWF with P-GWF, is proposed, to compute the solution to the extended problem exactly and efficiently; and finally, the Dinkelbach approach is reviewed, followed by an analogous comparison of our proposed algorithm to clearly illustrate the advantages of the proposed approach.

A. Statement of Extended Problem

Letting all the parameters be assumed the same as those in the target EE maximization problem (1), find a group of power $\{s_i\}$ to satisfy:

$$\begin{aligned} &\max_{\{s_i\}_{i=1}^K} && \frac{\frac{1}{2}\sum_{i=1}^{K}\log(1+a_i s_i)}{s_0 + \sum_{i=1}^{K} s_i} \\ &\text{subject to:} && 0 \le s_i, \text{ for } i = 1, \ldots, K; \\ &&& \textstyle\sum_{i=1}^{K} s_i \le P; \\ &&& \tfrac{1}{2}\textstyle\sum_{i=1}^{K}\log(1+a_i s_i) \ge B, \end{aligned} \tag{24}$$

where the non-negative number, B, denotes the minimal throughput requirement.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2803126, IEEE Transactions on Vehicular Technology

The EE maximization problem (24) is an independent problem on the target problem of (1). At the same time, if the minimal throughput B is set as zero, this EE maximization problem is regressed into the target problem (1). Therefore, (24) is a more general form of EE maximization.

B. A Concise Review of P-GWF

The P-GWF algorithm has been proposed to compute the sum power minimization problem with the throughput requirement constraint [15]. The problem is stated as follows:

$$\begin{array}{ll} \min_{\{s_i\}_{i=1}^K} & \sum_{i=1}^K s_i \\ \text{subject to:} & 0 \le s_i, \quad \text{for } i = 1, \ldots, K; \\ & \frac{1}{2}\sum_{i=1}^K w_i \log(1 + a_i s_i) \ge B. \end{array} \tag{25}$$

Since (25) may be regarded as a duality of the throughput maximization problem [13, (1)], there are concepts, like the duality of those which appear in (2)-(6). Their concrete expressions may refer to [15, (5), (27)-(30)]. In [15], Algorithm P-GWF has been proven to provide the optimal solution to problem (25). It needs $8K$ operations, which consist of K basic (elementary) function evaluation operations (BEs), $5K$ arithmetic operations (AOs), and $2K$ logical operations (LOs), at most. Similarly, Algorithm P-GWF provides the mapping of $(\{a_k\}_{k=1}^K, B)$ to the exact solution $\{s_i\}$ and k^*.

C. EEE-JWF, Algorithm of Extended Problem

Based on both P-GWF and EE-WF, an algorithm is proposed to solve the EE maximization problem (24). This algorithm is denoted by EEE-JWF.

The interesting treatment is reflected on the fact that P-GWF sets up an updated "stair heights". Fig. 5(a) shows the application of P-GWF algorithm, we can have a water level μ_1 and corresponding power allocation to meet the minimum throughput requirement B. A new index $\underline{k}$ is used to divide all the channels into two categories. All the channels with index smaller than $\underline{k}$ are treated as one category, whose stair heights are updated as μ_1. Then μ_1 is used as the lowest stair height as shown in Fig. 5(b), where the dashed stairs denote the original channels; the bold solid stairs denote the updated stairs after applying P-GWF. In this way, we can guarantee the constraint in (24) can be satisfied.

Then under these updated stairs, the extended problem could be solved using algorithm EE-JWF. This method is indeed a novel approach. Algorithm details is listed below.

Algorithm EEE-JWF:

0) Pre-processing:

$$\begin{array}{l} \{k^*, P_u(k^*)\} = \text{GWF}(\{a_k\}_{k=1}^K, P), \\ k^* \to M \text{ and } \frac{P_u(M)}{M} + \frac{1}{a_M} \to \frac{1}{a_{M+1}}. \\ \{k^*, \{s_k\}_{k=1}^K\} = \text{P-GWF}(\{a_k\}_{k=1}^K, B), \\ \min\{k | s_{k^*} + \frac{1}{a_{k^*}} < \frac{1}{a_k}, 1 \le k \le K\} - 1 \to \underline{k}. \\ s_{k^*} + \frac{1}{a_{k^*}} \to d_k, \text{ for } 1 \le k \le \underline{k}; \\ \text{while } \frac{1}{a_k} \to d_k, \text{ for } \underline{k} < k \le K. \end{array} \tag{26}$$

1) Input:

Let $n = \underline{k}$ and assign a $(K + 1 - \underline{k}) \times K$ matrix $\mathbf{S}$ a zero matrix.

2) Loop for n from $\underline{k}$ to M: for each loop, one of the following three branches being executed to update the $n - \underline{k} + 1$th row of the matrix $\mathbf{S}$:

2.1) If $g(\triangle s_{\min}) = g(0) > 0$,

$$S_{n,j} = d_n - d_j, \quad j = 1, \cdots, n, \tag{27}$$

and then go to Step 3); else

2.2) if $g(\triangle s_{\max}) = g\left(n\left(d_{n+1} - d_n\right)\right) < 0$,

$$S_{n,j} = d_{n+1} - d_j, \quad j = 1, \cdots, n, \tag{28}$$

and then go to item 3); else

2.3) if

$$g(0) \cdot g\left(n\left(d_{n+1} - d_n\right)\right) < 0, \tag{29}$$

then solve $\triangle s$ by applying Proposition 1 and

$$S_{n,j} = \frac{\triangle s}{n} + \left(d_n - d_j\right), \quad j = 1, \cdots, n. \tag{30}$$

3) If $n = M$, compute

$$n^* = \arg \max_{\{n | \underline{k} \le n \le M\}} \left\{ \frac{\frac{1}{2}\sum_{i=1}^K \log(1 + a_i S_{n,i})}{s_0 + \sum_{i=1}^K S_{n,i}} \right\}, \tag{31}$$

then output $S_{n^*,k} + d_{k^*} - \frac{1}{a_k}$, for $1 \le k \le \underline{k}$, as the preceding $\underline{k}$ entries of the solution, and $S_{n^*,k}$, for $\underline{k} < k \le K$, as the following $K - \underline{k}$ entries of the solution; else let $n + 1 \to n$ and go to Step 2).

Proposition 4. EEE-JWF outputs the solution to (24) with a finite amount of computation.

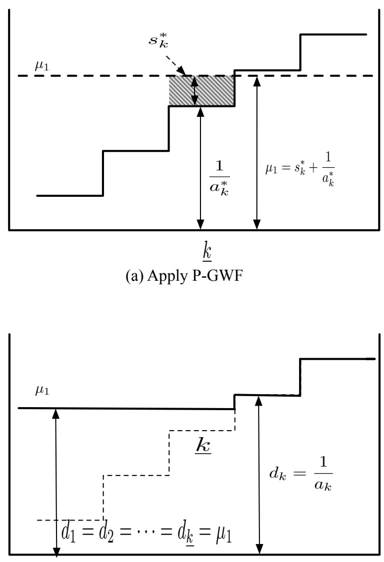
Proof. Referring to the poof of Proposition 2, and noting that P-GWF provides the initial interval $[d_{\underline{k}}, d_{\underline{k}+1}]$, it is then trivially seen that Proposition 4 holds. ☐

D. Dinkelbach Approach and Its Iterations

The Dinkelbach approach has been well applied to solve the energy efficiency maximization problems, *e.g.*, the target problem (24), by making use of the following equations and iterations.

Let $\mathcal{F}$ be the feasible set of problem (24). This equation, including a convex optimization operation, in q^* is:

$$\begin{array}{l} \max_{\{s_i\}_{i=1}^K \in \mathcal{F}} \\ \{\sum_{i=1}^K \frac{1}{2}\log(1 + a_i s_i) - q^*(s_0 + \sum_{i=1}^K s_i)\} = 0. \end{array} \tag{32}$$



(b) Updated stair heights

Fig. 5: Illustration of Algorithm EEE-JWF.

If the q^* is regarded as a parameter, the left hand side (LHS) of equation (32) is a convex optimization with respect to variables $\{s_i\}$. Denote by $J(q^*)$ the corresponding maximum value, *i.e.*,

$$J(q^*) \triangleq \max_{\{s_i\}_{i=1}^K \in \mathcal{F}} \{\sum_{i=1}^K \tfrac{1}{2}\log(1+a_i s_i) - q^*(s_0 + \sum_{i=1}^K s_i)\}, \tag{33}$$

and then equation (32) can be simplified as $J(q^*) = 0$. The extended problem (24) can be solved based on equation (32) and the KKT conditions of (33). Combining equation (32) with the KKT conditions of (33), we have

$$\begin{cases} \sum_{i=1}^K \frac{1}{2}\log[1+a_i(\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})^+] - \\ \quad q^*[s_0 + \sum_{i=1}^K(\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})^+] = 0, \\ \lambda\cdot[P - \sum_{i=1}^K(\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})^+] = 0, \\ \sum_{i=1}^K(\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})^+ \le P, \\ \nu\cdot\left[\sum_{i=1}^K \frac{1}{2}\log\left(1+a_i(\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})\right) - B\right] = 0, \\ \sum_{i=1}^K \frac{1}{2}\log(1+a_i(\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})) \ge B, \\ \lambda \ge 0, \quad \nu \ge 0, \quad q^* > 0, \end{cases} \tag{34}$$

where $s_i = (\frac{\nu+1}{\lambda+q^*}-\frac{1}{a_i})^+$, for $i = 1,\dots,K$; λ is the optimal dual variable that corresponds to the sum power constraint of $\mathcal{F}$; and ν is the optimal dual variable that corresponds to the throughput requirement constraint of $\mathcal{F}$ [3]. The system (34) consists of three non-linear and non-smooth equalities as well as five inequalities in λ, ν and q^*. There seems no prior method in the open literature to compute the solution to (34) or (24), under the merits of exactness and polynomial computational complexity.

The Dinkelbach method has been proposed to solve the non-linear problem (34) based on iteration. The outline is: (a) Initialize $q^{(0)}$; (b) Assume $q^{(n)}$ to be given, compute $J(q^{(n)})$ and denote the corresponding optimal solution by $\{s_i^{(n)}\}_{i=1}^K$; and (c) Update $q^{(n)}$ into $q^{(n+1)}$, where

$$q^{(n+1)} = \frac{\sum_{i=1}^K \frac{1}{2}\log_2(1+a_i s_i^{(n)})}{s_0 + \sum_{i=1}^K s_i^{(n)}}. \tag{35}$$

Different from the Dinkelbach method, we proposed algorithms to directly compute the solution to the target problems by applying the geometry-based machinery, with exactness and low-degree polynomial complexity. Following subsection gives an analogous comparison of these two methods to demonstrate the advantages of the proposed method.

E. Analogous Comparison with Dinkelbach Method

For convenience of analogous comparison, assume that $B = 0$ in (24) without loss of generality. From analysis of the Dinkelbach method, solving KKT conditions of (33) leads to the power allocation solution following a water-filling like structure. Applying the geometric relation, considering the fact that only the first N channels out of K channels are allocated with non-zero power with the water level μ, we can update the summation range from $[1,K]$ to $[1,N]$. Then the Dinkelbach method (33) can be written as

$$\begin{aligned} J(q^*) &\triangleq \max_{\{s_i\}_{i=1}^N \in \mathcal{F}} \left\{\sum_{i=1}^N \frac{1}{2}\log\left(1+\frac{s_i}{d_i}\right) - q^* S_T\right\} \quad (36) \\ &= \max_{\{s_i\}_{i=1}^N \in \mathcal{F}} \left\{\sum_{i=1}^N \frac{1}{2}\log\left(1+\frac{\mu-d_i}{d_i}\right) - q^* S_T\right\} \\ &= \max_{\{s_i\}_{i=1}^N \in \mathcal{F}} \left\{\sum_{i=1}^N \frac{1}{2}\log\left(\frac{\mu}{d_i}\right) - q^* S_T\right\}, \quad (37) \end{aligned}$$

where we applied $s_i = (\mu - d_i)$. For easy presentation, we define Dinkelbach operator, $\mathcal{D}$, as

$$\mathcal{D} = \sum_{i=1}^N \frac{1}{2}\log\left(\frac{\mu}{d_i}\right) - q^* S_T. \tag{38}$$

Now reviewing $g(\triangle s)$ defined in (10), and using the concept of the water-level μ, which is equal to $(d_N + \frac{\triangle s}{N})$ as illustrated in Fig. 2, $g(\triangle s)$ can be expressed below with a better geometric vision as,

$$\begin{aligned} g(\triangle s) &= \mu N\log(\mu) - \mu\sum_{k=1}^N \log(d_k) - S_T \quad (39) \\ &= \mu\left[\log(\mu)^N - \log(d_1 d_2\cdots d_N)\right] - S_T \\ &= \mu\left[\log\frac{\mu^N}{d_1 d_2\cdots d_N}\right] - S_T \\ &= \mu\sum_{i=1}^N \log\left(\frac{\mu}{d_i}\right) - S_T. \quad (40) \end{aligned}$$

Comparing (40) with (38), the relationship between Dinkelbach operator and $g(\triangle s)$ is given by

$$g(\triangle s) = 2\mu\cdot\mathcal{D}|_{q^*=\frac{1}{2\mu}}. \tag{41}$$

[3]Here the function of $(x)^+$ is defined by $(x)^+ = x$ for $x \ge 0$; and $(x)^+ = 0$ for $x < 0$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2803126, IEEE Transactions on Vehicular Technology

Thus, the main advantages of our approach over Dinkelbach method are as follows:

(a) Search range: the Dinkelbach method is to search q^* through (35). In this search, q^* cannot be obtained from a finite discrete point set. On the other hand, in our proposed approach, we determine the water level step index, k^*, from GWF. The searching space is narrowed down to (k^*) intervals, specified by $[d_1, d_2], [d_2, d_3], \cdots, [d_{k^*-1}, d_{d^*}]$ and $[d_{k^*}, d_{k^*+1}]$. This significantly reduces the searching effort.

(b) For a given q, the Dinkelbach method needs to compute the exact solution to every convex optimization problem with a "$\max$" operator in (33). However, it is difficult to obtain an exact solution, especially remarkable for more complicated constraints being met in problem (33). This non-exact solution can impair convergence of the Dinkelbach algorithm (refer to [18, (B) on p. 495]). On the other hand, in our proposed approach, for each searching interval, we solve $\min |g(\triangle s)|$. For all (k^*) intervals, there is at most one interval which utilizes Proposition 1 to solve $\triangle s$. For all other $(k^* - 1)$ intervals, our algorithm only needs to execute Step 2.1) or Step 2.2) as shown in Figs. 3 (a) and (b), or Figs. 4 (b) and (c) respectively. The computation effort for these $(k^* - 1)$ intervals is almost negligible.

V. Numerical Results and Complexity Analysis

A few numerical examples are presented in this Section to illustrate the steps of the proposed algorithms. As a positive constant factor does not affect the optimal allocation, the objective functions of the following examples use the natural logarithm for convenience.

Example 1. Instantiate an EEE-JWF problem:

$$\begin{array}{ll} \max_{\{s_i\}_{i=1}^2} & (\eta =) \frac{\log(1+s_1)+\log(1+0.5s_2)}{1+s_1+s_2} \\ \text{subject to:} & s_i \geq 0, \forall i; \\ & s_1 + s_2 \leq 2. \end{array} \tag{42}$$

The reciprocals $\{d_k = \frac{1}{a_k}\}$ of initial channel power gains are shown in Fig. 6 (1.a). The procedures to solve the problem are illustrated in Table 1, where the first two rows represent the results for Step 2) in Algorithm EE-JWF. The last row lists the output by Step 3) of EE-JWF. The column "Branch in Step 2)" lists the corresponding "If" condition being met in Step 2) , and the corresponding subfigures in Fig. 6.

In the last row, since $\eta(n = 2) > \eta(n = 1)$, the output is $n^* = 2$, and the second row of $\mathbf{S}$ as the solution of power allocation for the two channels.

Note that this example has a unique optimal solution as the global solution, without any other local or global optimal solutions.

Example 2. Instantiate another EE-JWF problem:

$$\begin{array}{ll} \max_{\{s_i\}_{i=1}^2} & (\eta =) \frac{\log(1+s_1)+\log(1+0.5s_2)}{1+s_1+s_2} \\ \text{subject to:} & s_i \geq 0, \forall i; \\ & s_1 + s_2 \leq 3. \end{array} \tag{43}$$

TABLE I: Iteration results for Example 1

Iteration	Branch in Step 2)	$\mathbf{S}$	η
$n = 1$	2.2, Fig. 6 (1.b)	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{\log(2)}{2} \\ 0 \end{bmatrix}$
$n = 2$	2.2, Fig. 6 (1.c)	$\begin{bmatrix} 1 & 0 \\ 1.5 & 0.5 \end{bmatrix}$	$\begin{bmatrix} \frac{\log(2)}{2} \\ \frac{\log(3.125)}{3} \end{bmatrix}$
output	Fig. 6 (1.c)	$[1.5 \quad 0.5]$	$\frac{\log(3.125)}{3}$

This example is similar to Example 1, except that the upper bound of the sum power is 3.

The reciprocals $\{d_k = \frac{1}{a_k}\}$ of the channel power gains are the same as those of Example 1 in Fig. 6 (1.a). The solving procedures are listed in Table 2.

TABLE II: Iteration results for Example 2

n	Step 2) Branch	$\mathbf{S}$	η
1	2.2 Fig. 6 (1.b)	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{\log(2)}{2} \\ 0 \end{bmatrix}$
2	2.3, Fig. 6 (2.a)	$\begin{bmatrix} 1 & 0 \\ 1.62729 & 0.62729 \end{bmatrix}$	$\begin{bmatrix} \frac{\log(2)}{2} \\ \frac{\log(3.45133)}{3.25458} \end{bmatrix}$
	Fig. 6 (2.a)	$[1.62729 \quad 0.62729]$	0.38062

With the machine computing, the results are represented by five decimal digits. Example 2 has a unique global optimal solution as shown in the third row of Table 2, corresponding to the maximal value of the objective function. It does not have any other local or global optimal solutions either. Example 2 indicates that the optimal solution does not always use out the available total power for allocation. In this example, the total power used for allocation is 2.25458 with an objective function value as 0.38062. As a comparison, if we use up all the available power $P = 3$, the allocation is illustrated in Fig. 6 (2.c) as $[s_1 = 2, s_2 = 1]$, leading to the corresponding objective function value as 0.3760.

Example 3. Instantiate a weighted case of EE-JWF:

$$\begin{array}{ll} \max_{\{s_i\}_{i=1}^2} & (\eta =) \frac{\frac{2}{3}\log(1+s_1)+\log(1+0.5s_2)}{1+s_1+s_2} \\ \text{subject to:} & s_i \geq 0, \forall i; \\ & s_1 + s_2 \leq 2. \end{array} \tag{44}$$

The reciprocals $\{d_k = \frac{1}{a_k}\}$ are illustrated in Fig. 6 (3.a). The solving procedure is listed in Table 3. This example also has the unique global optimal solution.

TABLE III: Iteration results for Example 3

n	Step 2) Branch	$\mathbf{S}$	η
1	2.2 Fig. 6 (3.b)	$\begin{bmatrix} 1/3 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{\log(4/3)}{2} = 0.14 \\ 0 \end{bmatrix}$
2	2.2, Fig. 6 (3.c)	$\begin{bmatrix} 1/3 & 0 \\ 1.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} \frac{\log(4/3)}{2} = 0.14 \\ \frac{3\log 3 - \log 2}{9} = 0.29 \end{bmatrix}$
	Fig. 6 (3.c)	$[1.0 \quad 1.0]$	0.29

Following, we present computational complexity analysis of the algorithms. Fig. 7 assumes that the number of the channels, K, changes from 100 to 200. The parameters $\{a_k\}$

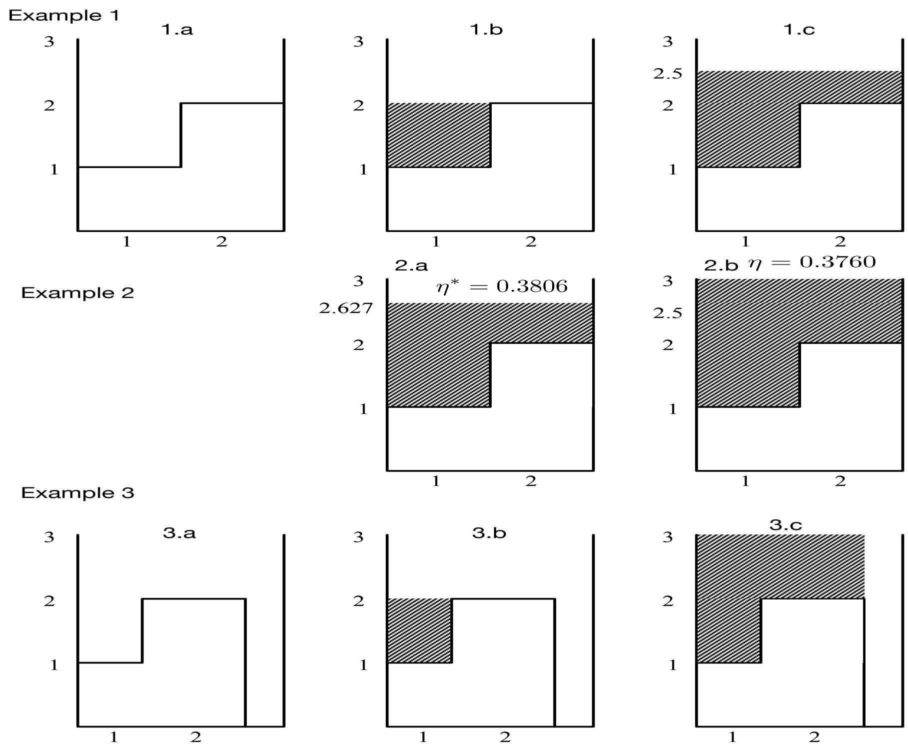


Fig. 6: Illustration of optimal power allocation for examples 1-3.

are assigned at random, where the square root of each entry of $\{a_k\}$ is drawn independently from the standard Gaussian distribution and then squared (due to a_k being a channel power-gain, $\forall k$). We use $n1$ ($O(K^2)$, details of which are provided in the complexity analysis of next section) to denote the number of basic operations needed by the proposed algorithm to obtain the global optimal solution to (1). Fig. 7 compares the achieved EE of the proposed algorithm (circle marked curve) with those of Dinkelbach method. The lower two curves are the corresponding EE values achieved by using Dinkelbach algorithm under the number of the basic operations $(2 \times n1)$ and $n1$ respectively. Fig. 7 shows that the gain in EE is significant over Dinkelbach method under the same number of the basic operations, or doubled the number of the basic operations.

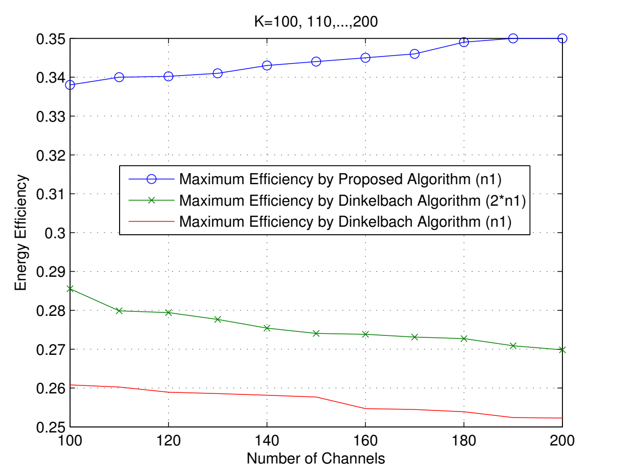


Fig. 7: Maximum energy efficiency by EE-JWF (proposed) vs. Dinkelbach algorithm.

For the EE-JWF Algorithm, we have the following computational complexity analysis. Steps 2.1) and 2.2) of EE-JWF has experienced $K+1$ evaluations, which need a total of basic arithmetic, logical and basic function evaluation operations (BAO, BLO, and BFVO) of $4K^2+4K$ times, at most. Step 2.3), only used once at most, needs $N = N_1 + N_2$ (defined in Appendix B) loops to output the solution to the system of $g(\triangle s) = 0$ and $\triangle s_{\min} \leq \triangle s \leq \triangle s_{\max}$. As the EE-JWF algorithm has K loops at most, the computational complexity is $5K\left[(\max_{\{1\leq N\leq M\}}\{\frac{d_{N+1}}{d_N}\}+35)\right] + [4K^2 + 4K] = \mathrm{O}(K) + \mathrm{O}(K^2) = \mathrm{O}(K^2)$. Sorting $\{d_n\}$ needs the complexity of $\mathrm{O}(K\log(K))$. It implies that EE-JWF has the computational complexity of $\mathrm{O}(K^2) + \mathrm{O}(K\log(K)) = \mathrm{O}(K^2)$. It is a loose result but clearly expresses the result of the computational complexity: $\mathrm{O}(K^2)$. In addition, the computational complexities of EE-JWF and EEE-JWF are the same, since only $\mathrm{O}(K)$ computational complexity is added. For the Dinkelbach method, PD-IPM algorithm is utilized. The computational complexity of running one PD-IPM algorithm is $\mathrm{O}(K^{3.5})\log(1/\epsilon)$ [17], [29]. The proposed approach is to compute the solution with the error of machine zero, whereas PD-IPM is to compute an ϵ-error solution. Therefore, the proposed approach is a significant stepforward

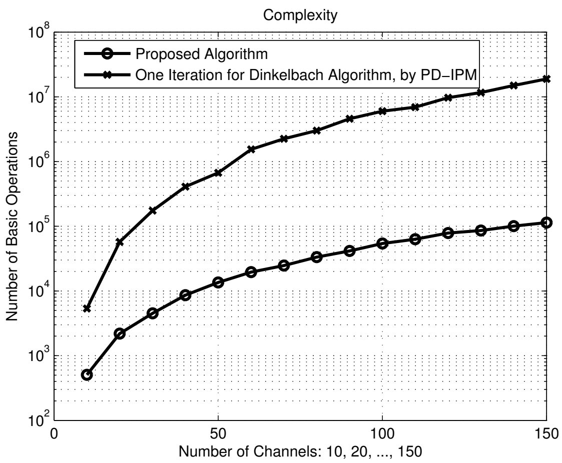


Fig. 8: Comparing computational complexity of EE-JWF (proposed) with that of one iteration of Dinkelbach algorithm.

over the Dinkelbach method, to solve EE maximization problems. Specifically, we have following proposition.

Proposition 5. For the proposed problems, EE-JWF and EEE-JWF both output the solutions with errors of machine zero and the computational complexities of $\mathrm{O}(K^2)$; while PD-IPM outputs an ϵ solution, with the computational complexity of $\mathrm{O}(K^{3.5})\log(1/\epsilon)$.

As discussed before, the weighted cases can be straightforwardly acquired from the equal weighted cases. The weighted case is considered in the following figure. The parameters $\{a_k, w_k\}$ are assigned at random, where the square root of each entry of $\{a_k\}$ is drawn independently from the standard Gaussian distribution and then squared (due to a_k being a channel power-gain, $\forall k$), and each entry of $\{w_k\}$ independently from a uniform distribution of $U[0, 1]$. The experiment is conducted based on 100 samples, with the sum power budget that is set up from 10 dBm to 150 dBm (*i.e.*, 1dBm×the number of the channels). In Fig. 8, the cross marked curve illustrates the corresponding computational complexity of Dinkelbach algorithm, to compute an ϵ solution with $\epsilon = 0.01$ to one of the mentioned family of the optimization operation using. The results of Fig. 8 demonstrate that the computational complexity of one Dinkelbach iteration is much higher than that of the proposed EE-JWF algorithm, especially when the number of channels is large. If accuracy requirement is more strict, the computational complexity of the Dinkelbach method can be even higher. Thus, the proposed EE-JWF algorithm can significantly reduce the complexity to compute the energy-efficiency optimal power allocation, which is especially favourable in large scale systems.

The proposed algorithms can compute the exact solutions with excellent scalability, considering their low computational complexity. Further, the proposed algorithms can be computed in a parallel manner. Therefore, EE-JWF can be applied to solve super-large-scale problems with high efficiency and exactness, such as the massive Multiple Input Multiple Output (MIMO) systems.

VI. Conclusion

In this paper, we have proposed the algorithms, EE-JWF and EEE-JWF, to efficiently compute the optimal solutions to the energy efficiency maximization problems with the sum power upper bound constraints and the added throughput requirement constraint, respectively. Compared with existing energy-efficient power allocation algorithms, the proposed EE-JWF and EEE-JWF algorithms have common two-fold benefits of exactness and low complexity. Optimality of the proposed algorithms has been proved strictly, and the low computational complexity has been analyzed, proven and validated through numerical results. As the proposed algorithms can provide exact solution through low degree polynomial-complexity parallel computations, they can be applied to EE-maximal power allocation in large-scale wireless systems to realize effective green communication.

Appendix A
Proof of Lemma 1

It is seen that since the objective function in (1) is continuous over the feasible set that is compact, there exists an optimal solution to (1). Let $\{s_k^*\}_{k=1}^K$ denote an optimal solution to (1) under the meaning of global optimality. Thus, there exists $n \in \mathbb{Z}^+$ with $1 \le n \le K$ such that $\sum_{k=1}^n(\frac{1}{a_n} - \frac{1}{a_k}) \le \sum_{k=1}^K s_k^* \le \sum_{k=1}^{n+1}(\frac{1}{a_{n+1}} - \frac{1}{a_k})$. The optimal solution of $\{s_k^*\}_{k=1}^K$ implies that it is also the solution to the following problem:

$$\begin{aligned} \max_{\{s_i\}_{i=1}^K} \quad & \tfrac{1}{2}\textstyle\sum_{i=1}^K \log_2(1 + a_i s_i) \\ \text{subject to:} \quad & 0 \le s_i, \ \forall i; \\ & \textstyle\sum_{i=1}^K s_i = \sum_{k=1}^K s_k^*. \end{aligned} \tag{45}$$

Successively, (45) and GWF (refer to (3)-(6) or the details in [13]) result in the facts that both there exists a $\triangle s^*$ with $0 \le \triangle s^* \le \triangle s_{\max}$ such that $s_k^* = (\frac{1}{a_n} - \frac{1}{a_k}) + \frac{\triangle s^*}{n}$, for $k = 1, \ldots, n$, and $s_k^* = 0$, for $n < k \le K$ if available.

On the other hand, it is seen that the optimal value $\frac{\frac{1}{2}\sum_{i=1}^K \log_2(1+a_i s_i^*)}{s_0 + \sum_{i=1}^K s_i^*}$ of (1) is not less than the optimal value of the problem:

$$\begin{aligned} \max_{\{s_i\}_{i=1}^K} \quad & \frac{\frac{1}{2}\sum_{i=1}^K \log_2(1+a_i s_i)}{s_0 + \sum_{i=1}^K s_i} \\ \text{subject to:} \quad & 0 \le s_i, \ \forall i; \\ & \textstyle\sum_{i=1}^K s_i = V_n, \end{aligned} \tag{46}$$

for any V_n where $V_n \in [\sum_{k=1}^n(\frac{1}{a_n} - \frac{1}{a_k}), \sum_{k=1}^{n+1}(\frac{1}{a_{n+1}} - \frac{1}{a_k})]$. An optimal solution to (46) is denoted by $\{\underline{s}_k\}$. Thus, similarly, there exists $\underline{s}_k$ with $0 \le \triangle s \le \triangle s_{\max}$ such that $\underline{s}_k = (\frac{1}{a_n} - \frac{1}{a_k}) + \frac{\triangle s}{n}$, for $k = 1, \ldots, n$, and $\underline{s}_k = 0$, for $n < k \le K$ if available, from GWF. These are due to $\{\underline{s}_k\}$ also being the solution to

$$\begin{aligned} \max_{\{s_i\}_{i=1}^K} \quad & \tfrac{1}{2}\textstyle\sum_{i=1}^K \log_2(1 + a_i s_i) \\ \text{subject to:} \quad & 0 \le s_i, \ \forall i; \\ & \textstyle\sum_{i=1}^K s_i = V_n. \end{aligned} \tag{47}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2803126, IEEE Transactions on Vehicular Technology

Due to the mentioned relationship between the two optimal values of (1) and (46), we have: $\frac{\sum_{i=1}^{K}\log(1+a_i s_i^*)}{s_0+\sum_{i=1}^{K} s_i^*} \geq \frac{\sum_{i=1}^{K}\log(1+a_i \underline{s}_i)}{s_0+\sum_{i=1}^{K} \underline{s}_i}$. That is to say, $\triangle s^*$ is the maximum point to the problem:

$$\begin{array}{ll} \max_{\triangle s} & \frac{\sum_{i=1}^{n}\log[1+a_i((\frac{1}{a_n}-\frac{1}{a_i})+\frac{\triangle s}{n})]}{s_0+\sum_{i=1}^{n}[(\frac{1}{a_n}-\frac{1}{a_i})+\frac{\triangle s}{n}]} \\ \text{subject to:} & 0 \leq \triangle s \leq \triangle s_{\max}, \end{array} \tag{48}$$

interestingly being changed into this optimization problem in only a single optimization variable. Let us denote the objective function of (48) by $f(\triangle s)$. The derivative of $f(\triangle s)$, $f'(\triangle s)$, is expressed into:

$$f'(\triangle s) = \frac{[s_0+\sum_{i=1}^{n}(\frac{1}{a_n}-\frac{1}{a_i})+\triangle s]-(\frac{1}{a_n}+\frac{\triangle s}{n})[\sum_{i=1}^{n}\log(a_i)+n\log(\frac{1}{a_n}+\frac{\triangle s}{n})]}{(\frac{1}{a_n}+\frac{\triangle s}{n})[s_0+\sum_{i=1}^{n}(\frac{1}{a_n}-\frac{1}{a_i})+\triangle s]^2}, \tag{49}$$

where $0 \leq \triangle s \leq \triangle s_{\max}$. Since the denominator part keeps the positive sign, the zero and the positive or negative values of $f'(\triangle s)$ only depend on the numerator part. Thus, for convenience and simplification, the numerator part times "-1" is denoted by $g(\triangle s)$ which is the same as that defined in Proposition 1, based on the reference being selected by minimizing $-f(\triangle s)$. Of course, somebody may also choose the symmetric maximization of $f(\triangle s)$. As mentioned before, $g'(0) \geq 0, g'(\triangle s) \uparrow$, for $\triangle s \in [0, \triangle s_{\max}]$, and then $g''(\triangle s) > 0$, within valid range of $\triangle s$. Therefore,

- if $g(0) > 0$, $\triangle s = 0$ and then $s_k^* = \frac{1}{a_n} - \frac{1}{a_k} = \underline{s}_k$, for $k = 1, \ldots, n$; while $s_k^* = 0 = \underline{s}_k$, for $n < k \leq K$ if available;
- if $g(\triangle s_{\max}) < 0$, $\triangle s = \triangle s_{\max}$ and then $s_k^* = \frac{1}{a_n} - \frac{1}{a_k} + \frac{\triangle s}{n} = \frac{1}{a_{n+1}} - \frac{1}{a_k} = \underline{s}_k$, for $k = 1, \ldots, n$; while $s_k^* = 0 = \underline{s}_k$, for $n < k \leq K$.
- If $g(0) \cdot g(\triangle s_{\max}) < 0$, apply Proposition 1 to calculate $\triangle s$, and $s_k^* = \frac{\triangle s}{n} + (\frac{1}{a_n} - \frac{1}{a_k}) = \underline{s}_k$, for $k = 1, \ldots, n$; while $s_k^* = 0 = \underline{s}_k$, for $n < k \leq K$.

Due to the monotonicity of $g(\triangle s)$, the point which satisfies each of the three cases just mentioned above, must be the local optimal solution.

Lemma 1 is hence proved.

Appendix B
Proof of Proposition 1

$g(\triangle s)$ is given in (10), where the other parameters, $\{d_k\}$, of $g(\triangle s)$ depend on their subscripts k up to N. Its derivative $g'(\triangle s)$ at $\triangle s = 0$ is

$$g'(0) = \frac{\sum_{k=1}^{N}\log(\frac{d_N}{d_k})}{N} \tag{50}$$

and then $g'(0) \geq 0$. At the same time,

$$g''(\triangle s) = \frac{\frac{1}{N}}{d_N + \frac{\triangle s}{N}} = \frac{1}{\mu N} > 0 \text{ for } \triangle s \in (0, \triangle s_{\max}). \tag{51}$$

Thus, $g'(\triangle s) > 0$ for $\triangle s \in (0, \triangle s_{\max})$, and it is strictly monotonically increasing in this $\triangle s$ range. Therefore, if

$$g(0) \cdot g(\triangle s_{\max}) = 0$$

as a trivial case, the solution can take 0 or $\triangle s_{\max}$; and if

$$g(0) \cdot g(\triangle s_{\max}) > 0$$

then (10) does not have any solution. On the other hand, we only require to consider the case of $g(0) < 0$ and $g(\triangle s_{\max}) > 0$. The case of $g(0) > 0$ and $g(\triangle s_{\max}) < 0$ does not exists, due to $g'(\triangle s) > 0$. Therefore, it is seen that the existence of the solution is guaranteed if $g(0) < 0$ and $g(\triangle s_{\max}) > 0$. Also uniqueness of the solution is guaranteed if the condition above holds.

If $g(0) < 0$ and $g(\triangle s_{\max}) > 0$, an algorithm is introduced as follows for the solution.

$$\triangle s_{n+1} = \triangle s_n - \frac{g(\triangle s_n)}{g'(\triangle s_n)}, \forall n \in \mathbb{Z}^+, \tag{52}$$

where $\mathbb{Z}^+$ is the set of non-negative integers and let us take any $\triangle s_0$ from the interval of $(\triangle s^*, \triangle s_{\max}]$, *i.e.*, $g(\triangle s_0) > 0$ and $\triangle s^* < \triangle s_0 \leq \triangle s_{\max}$. Here $\triangle s^*$ is the solution to the system (10) and greater than zero. Thus, $0 < \triangle s^* \leq \triangle s_{n+1} < \triangle s_n < \triangle s_{\max}, \forall n$. This point can be proven through mathematical induction as follows. According to the definition of $\triangle s_0$, $g(\triangle s_0) > 0$, and the properties of both $g'(\triangle s)$ and $g''(\triangle s)$ mentioned above, it is seen that $0 < \triangle s^* < \triangle s_1 = \triangle s_0 - \frac{g(\triangle s_0)}{g'(\triangle s_0)} < \triangle s_0$. As a side note, the second and the third inequalities in the family, just mentioned above, of inequalities result from a fact. This fact is that there exists $\eta_0 \in (\triangle s^*, \triangle s_0)$, with $\triangle s^*$ being greater than zero, such that $\triangle s_0 - \triangle s^* > \triangle s_1 - \triangle s^* = (\triangle s_0 - \triangle s^*)[1 - \frac{g'(\eta_0)}{g'(\triangle s_0)}] > 0$. Assume that $0 < \triangle s^* < \triangle s_{n+1} < \triangle s_n < \triangle s_{\max}$ with $\triangle s_{n+1} - \triangle s^* = (\triangle s_n - \triangle s^*)[1 - \frac{g'(\eta_n)}{g'(\triangle s_n)}]$, where $\eta_n \in (\triangle s^*, \triangle s_n)$. The following is to prove that $0 < \triangle s^* < \triangle s_{n+2} < \triangle s_{n+1} < \triangle s_{\max}$. This system of inequalities comes from a similar fact to the one mentioned above: $\triangle s_{n+1} - \triangle s^* > \triangle s_{n+2} - \triangle s^* = (\triangle s_{n+1} - \triangle s^*)[1 - \frac{g'(\eta_{n+1})}{g'(\triangle s_{n+1})}] > 0$, where $\eta_{n+1} \in (\triangle s^*, \triangle s_{n+1})$. Therefore, the algorithm of (52) with the corresponding aforementioned assumptions owns the property of $0 < \triangle s^* < \triangle s_{n+1} < \triangle s_n < \triangle s_{\max}, \forall n$. Also as a by-product, the equations

$$\triangle s_{n+1} - \triangle s^* = (\triangle s_n - \triangle s^*)\left[1 - \frac{g'(\eta_n)}{g'(\triangle s_n)}\right], \forall n, \tag{53}$$

have been proven. In (53), since

$$1 - \frac{g'(\eta_0)}{g'(\triangle s_0)} = \frac{g'(\triangle s_0) - g'(\eta_0)}{g'(\triangle s_0)}, \tag{54}$$

it is seen that

$$0 < \frac{g'(\triangle s_0) - g'(\eta_0)}{g'(\triangle s_0)} < \frac{g'(\triangle s_0) - g'(\eta_0)}{g'(\triangle \underline{s})}, \tag{55}$$

where

$$\triangle \underline{s} = \frac{g(0)N(d_{N+1} - d_N)}{g(0) - g(N(d_{N+1} - d_N))} \tag{56}$$

and $0 < \triangle \underline{s} < \triangle s^*$. In addition, for the equation $g(\triangle s) = 0$, first, we may use the bisection method [30] over the initial interval of $[g'(\triangle \underline{s}), g'(\triangle s_{\max})]$ repeatedly, and an interval denoted by $[g'(a), g'(b)]$ is obtained such that this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2018.2803126, IEEE Transactions on Vehicular Technology

interval contains $g'(\triangle s^*)$ with $0 < g'(b) - g'(a) < \frac{1}{10} g'(\triangle \underline{s})$ (where $g'(\triangle \underline{s})$ is given). This obtaining needs

$$N_1 = \left\lceil \frac{\log\left[10\left(\frac{g'(\triangle s_{\max}) - g'(0)}{g'(\triangle \underline{s})}\right)\right]}{\log 2} \right\rceil + 1 \quad (57)$$

loop operations, at most, *i.e.*, a finite amount of operations. Here, the notation $\lceil \cdot \rceil$ denotes the ceil function. Then, let $\triangle s_0 = b$. Thus, stemming from (53), we have:

$$\begin{aligned} 0 < & \quad \triangle s_n - \triangle s^* \\ = & \quad (\triangle s_0 - \triangle s^*) \Pi_{k=1}^{n} \left[\frac{g'(\triangle s_{k-1}) - g'(\eta_{k-1})}{g'(\triangle s_{k-1})}\right] \\ < & \quad (\triangle s_0 - \triangle \underline{s}) \times \left(\frac{1}{10}\right)^n, \forall n, \end{aligned} \quad (58)$$

with the proven property: $0 < \triangle s^* < \triangle s_{n+1} < \triangle s_n < \triangle s_{\max}, \forall n$. Thus, it is seen that, for any $\epsilon > 0$, there exists

$$N_2(\epsilon) = \left\lceil \frac{\log\left(\frac{\triangle s_0 - \triangle \underline{s}}{\epsilon}\right)}{\log(10)} \right\rceil + 1 = \left\lceil \lg\left(\frac{\triangle s_0 - \triangle \underline{s}}{\epsilon}\right) \right\rceil + 1 \quad (59)$$

such that as $n \geq N_2, 0 < \triangle s_n - \triangle s^* < \epsilon$, where $\lg(x)$ is the logarithm in x with the base of 10. Thus, through the mentioned $N_t (= N_1 + N_2)$ loop operations above, at most, with letting $\epsilon = 10^{-34}$, (52) with setting $\triangle s_0$ computes the practical exact solution with error of machine zero by a finite amount of operations. In addition, the practical exact solution only uses 34 more loops than the case of $\epsilon = 1$, at most.

Proposition 1 is hence proved.

References

[1] S. Lambert, W. Van Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, "Worldwide electricity consumption of communication networks," *Optics express*, vol. 20, no. 26, pp. B513–B524, 2012.

[2] "C-RAN: the road towards green RAN," Tech. Rep., China Mobile, 2011.

[3] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Commun. Surveys Tut.*, vol. 13, no. 4, pp. 524–540, Nov. 2011.

[4] M. Ismail, W. Zhuang, E. Serpedin, and K. Qaraqe, "A survey on green mobile networking: From the perspectives of network operators and mobile users," *IEEE Commun. Surveys Tut.*, vol. 17, no. 3, pp. 1535–1556, 2015.

[5] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. CK. Soong, and J. Z. Zhang, "What will 5G be?," *IEEE J. Select. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, 2014.

[6] D. Tsilimantos, J. Gorce, K. Jaffrès-Runser, and H. V. Poor, "Spectral and energy efficiency trade-offs in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 54–66, 2016.

[7] Q. Wu, G. Y. Li, W. Chen, D. W. K. Ng, and R. Schober, "An overview of sustainable green 5G networks," *IEEE Wireless Commun.*, vol. 24, pp. 72–80, 2017.

[8] Y. Soh, T. Quek, M. Kountouris, and H. Shin, "Energy efficient heterogeneous cellular networks," *IEEE J. Select. Areas Commun.*, vol. 31, no. 5, pp. 840–850, 2013.

[9] D. Feng, C. Jiang, G. Lim, L. Cimini, G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Commun. Surveys Tut.*, vol. 15, no. 1, pp. 167–178, 2013.

[10] P. He and L. Zhao, "Improved sum power iterative water-filling with rapid convergence and robustness for multi-antenna Gaussian broadcast channels," in *IEEE VTC'10 Spring*, Taipei,Taiwan, May 2010, IEEE, pp. 1–5.

[11] V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel state information for throughput maximization," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2566–2582, 2004.

[12] S. Ekbatani, F. Etemadi, and H. Jafarkhani, "Throughput maximization over slowly fading channels using quantized and erroneous feedback," *IEEE Trans. Commun.*, vol. 57, no. 9, 2009.

[13] 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, 2013.

[14] M. Ismail, A. T. Gamage, W. Zhuang, X. Shen, E. Serpedin, and K. Qaraqe, "Uplink decentralized joint bandwidth and power allocation for energy-efficient operation in a heterogeneous wireless medium," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1483–1495, 2015.

[15] P. He and L. Zhao, "Solving a class of sum power minimization problems using generalized water-filling," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6792–6804, 2015.

[16] K. Cheung, S. Yang, and L. Hanzo, "Achieving maximum energy-efficiency in multi-relay ofdma cellular networks: A fractional programming approach," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2746–2757, 2013.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.

[18] W. Dinkelbach, "On nonlinear fractional programming," *Management Science*, vol. 13, no. 7, pp. 492–498, 1967.

[19] J. P. Crouzeix and J. A. Ferland, "Algorithms for generalized fractional programming," *Mathematical Programming*, vol. 52, pp. 191–207, 1991.

[20] Q. Wu, M. Tao, and W. Chen, "Joint Tx/Rx energy-efficient scheduling in multi-radio wireless networks: A divide-and-conquer approach," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2727–2740, 2016.

[21] Q. Wu, M. Tao, D. Ng, W. Chen, and R. Schober, "Energy-efficient resource allocation for wireless powered communication networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2312–2327, 2016.

[22] Q. Wu, W. Chen, M. Tao, J. Li, H. Tang, and J. Wu, "Resource allocation for joint transmitter and receiver energy efficiency maximization in downlink OFDMA systems," *IEEE Trans. Commun.*, vol. 63, no. 2, pp. 416–430, 2015.

[23] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in OFDMA systems with large numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3292–3304, 2012.

[24] D. Zhang, Z. Chen, J. Ren, N. Zhang, M. Awad, H. Zhou, and X. Shen, "Energy harvesting-aided spectrum sensing and data transmission in heterogeneous cognitive radio sensor network," *IEEE Trans. Vehicular Tech.*, vol. PP, no. 99, pp. 1–1, 2016.

[25] J. Ren, Y. Zhang, N. Zhang, D. Zhang, and X. Shen, "Dynamic channel access to improve energy efficiency in cognitive radio sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3143–3156, 2016.

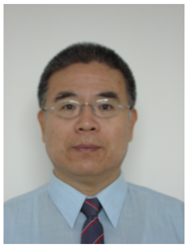
[26] A. Zappone, B. Matthiesen, and E. A. Jorswieck, "Energy efficiency in MIMO underlay and overlay device-to-device communications and cognitive radio systems," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 1026–1041, 2017.

[27] J. Mao, G. Xie, J. Gao, and Y. Liu, "Energy efficiency optimization for ofdm-based cognitive radio systems: A water-filling factor aided search method," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2366–2375, 2013.

[28] Y. Wang, W. Xu, K. Yang, and J. Lin, "Optimal energy-efficient power allocation for ofdm-based cognitive radio networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1420–1423, 2012.

[29] 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, pp. 2754–2762, 2009.

[30] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics, 2nd Edition*, Springer Berlin Heidelberg, 2010.



Peter He (S'15-M'15) received the M.A.Sc. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2009. He received the Ph.D. degree in electrical engineering from Ryerson University, Toronto, Canada, in 2015. His interests are the union of large-scale optimization, information theory, communications, and smart power grid. During his Ph.D. studies, he has been continuously awarded Graduate Research Excellence Award in Department of Electrical and Computer Engineering at Ryerson University and OGS (Ontario Graduate Scholarship) awards. He was awarded NSERC (Natural Sciences and Engineering Research Council) Postdoctoral Fellowship in 2016. He is a recipient of the Best Paper Award of the 2013 International Conference on Wireless Communications and Signal Processing (WCSP).

Shan Zhang (S'13-M'16) received her Ph.D. degree in Department of Electronic Engineering from Tsinghua University and B.S. degree in Department of Information from Beijing Institute Technology, Beijing, China, in 2016 and 2011, respectively. She is currently with the School of Computer Science and Engineering, Beihang University, Beijing, China. From August 2016 to December 2017, she was a post doctoral fellow in Department of Electronical and Computer Engineering, University of Waterloo, Ontario, Canada. Her research interests include network resource and traffic management, network virtualization and integration. Dr. Zhang received the Best Paper Award at the Asia-Pacific Conference on Communication in 2013.

Lian Zhao (S'99-M'03-SM'06) received the Ph.D. degree from the Department of Electrical and Computer Engineering (ELCE), University of Waterloo, Canada, in 2002. She joined the Department of Electrical and Computer Engineering at Ryerson University, Toronto, Canada, in 2003 and a Professor in 2014. Her research interests are in the areas of wireless communications, radio resource management, power control, cognitive radio and cooperative communications, optimization for complicated systems.

She received the Best Land Transportation Paper Award from IEEE Vehicular Technology Society in 2016; Top 15 Editor in 2015 for IEEE Transaction on Vehicular Technology; Best Paper Award from the 2013 International Conference on Wireless Communications and Signal Processing (WCSP) and Best Student Paper Award (with her student) from Chinacom in 2011; the Ryerson Faculty Merit Award in 2005 and 2007; the Canada Foundation for Innovation (CFI) New Opportunity Research Award in 2005, and Early Tenure and promotion to Associate Professor in 2006. She has been an Editor for IEEE TRANSACTION ON VEHICULAR TECHNOLOGY since 2013; co-chair for IEEE ICC 2018 Wireless Communication Symposium; workshop co-chair for IEEE/CIC ICCC 2015; local arrangement co-chair for IEEE VTC Fall 2017 and IEEE Infocom 2014; co-chair for IEEE Global Communications Conference (GLOBECOM) 2013 Communication Theory Symposium. She served as a committee member for NSERC (Natural Science and Engineering Research Council of Canada) Discovery Grants Evaluation Group for Electrical and Computer Engineering since 2015. She is a licensed Professional Engineer in the Province of Ontario, a senior member of the IEEE Communication and Vehicular Society.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. Dr. Shens research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom16, Infocom14, IEEE VTC10 Fall, and Globecom07, the Symposia Chair for IEEE ICC10, the Tutorial Chair for IEEE VTC11 Spring and IEEE ICC08, the General CoChair for ACM Mobihoc15, Chinacom07 and QShine06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; an Associate Editor-in-Chief for IEEE Internet of Things Journal, a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.