Chapter 2
Geometric Water-Filling in RRM

In this chapter, we introduce water-filling algorithms to solve power allocation problems. Two water-filling approaches are presented. One is the conventional water-filling (CWF); and the other one is the proposed geometric water-filling (GWF). GWF is further extended to efficiently solve a class of power allocation problems with more complex structure which owns upper bounds of the power variables. Computational complexities are investigated.

2.1 Problem Statement and Water-Filling

The water-filling problem can be abstracted and generalized into the following problem: given $P > 0$, as the total power or volume of the water; the allocated power and the propagation path gain for the ith channel are given as s_i and a_i respectively, $i = 1, \ldots, K$; and K is the total number of channels. Letting $\{a_i\}_{i=1}^K$ be a sorted sequence, which is positive and monotonically decreasing, find that

$$\begin{aligned} &\max_{\{s_i\}_{i=1}^K} \ \textstyle\sum_{i=1}^K \log(1 + a_i s_i) \\ &\text{subject to} : 0 \le s_i, \ \forall i; \\ &\qquad\qquad\quad \textstyle\sum_{i=1}^K s_i = P. \end{aligned} \tag{2.1}$$

Since the constraints are that (i) the allocated power to be nonnegative; (ii) the sum of the power equals P, the problem (2.1) is called the water-filling (problem) with sum power constraint.

To find the solution to problem (2.1), we usually start from the Karush-Kuhn-Tucker (KKT) conditions of the problem, as a group of the optimality conditions, and derive the system (2.2) below from the KKT conditions,

P. He et al., *Radio Resource Management Using Geometric Water-Filling*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-04636-5_2, © The Author(s) 2014

$$\begin{cases} s_i = \left(\mu - \frac{1}{a_i}\right)^+, \text{ for } i = 1, \ldots, K, \\ \sum_{i=1}^{K} s_i = P, \\ \mu \geq 0, \end{cases} \tag{2.2}$$

where $(x)^+ = \max\{0, x\}$. μ is the water level chosen to satisfy the power sum constraints with equality ($\sum_{i=1}^{K} s_i = P$). The solution to (2.2) is referred as a solution of the CWF problem (2.1).

It can be seen that the implied system (2.2) has been used to find the optimal solution. The existence of its Lagrange multipliers and the implication mentioned above determine that enumeration can be utilized to find the water level μ. In [1], how to solve the problems has been discussed extensively. Complexity of the non-geometric approach to solve the problem (2.1) will be discussed in Sect. 2.5. In the sequel of the chapter, when water-filling problem is mentioned, the power sum constraint is always included.

2.2 Proposed Geometric Water-Filling Approach

In this chapter, we propose a novel approach to solve problem (2.1) based on geometric view. The proposed Geometric Water-Filling (GWF) approach eliminates the procedure to solve the non-linear system for the water level, and provides explicit solutions and helpful insights to the problem and the solution.

Figure 2.1a–c give an illustration of the proposed GWF algorithm. Suppose there are 4 steps/stairs ($K = 4$) with unit width inside a water tank. For the conventional approach, the dashed horizontal line, which is the water level μ, needs to be determined first and then the power allocated for each stair (water volume above the stair) is solved.

Let us use d_i to denote the "step depth" of the ith stair which is the height of the ith step to the bottom of the tank, and is given as

$$d_i = \frac{1}{a_i}, \text{ for } i = 1, 2, \ldots, K. \tag{2.3}$$

Since the sequence a_i is sorted as monotonically decreasing, the step depth of the stairs indexed as $\{1, \cdots, K\}$ is monotonically increasing. We further define $\delta_{i,j}$ as the "step depth difference" of the ith and the jth stairs, expressed as

$$\delta_{i,j} = d_i - d_j = \frac{1}{a_i} - \frac{1}{a_j}, \text{ as } i \geq j \text{ and } 1 \leq i, j \leq K. \tag{2.4}$$

Instead of trying to determine the water level μ, which is a real nonnegative number, we aim to determine water level step, which is an integer number from 1 to K, denoted by k^*, as the highest step under water. Based on the result of k^*, we can write out the solutions for power allocation instantly.

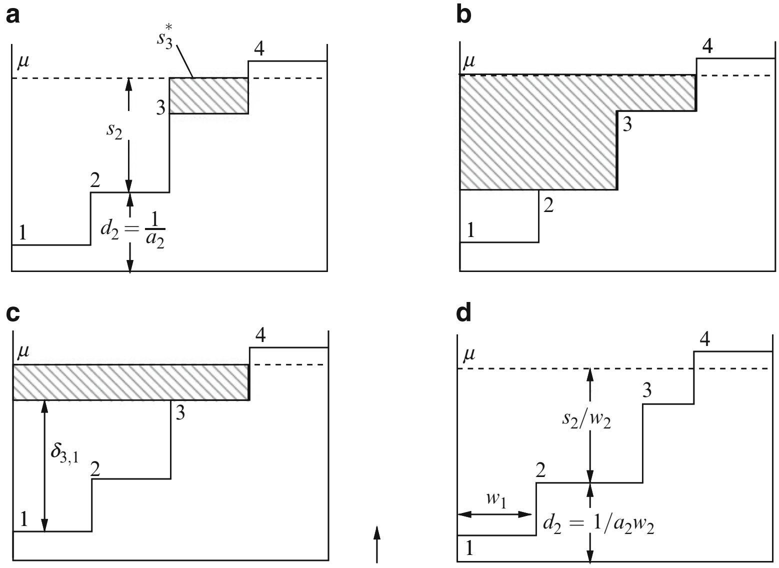

Fig. 2.1 Illustration for the proposed geometric water-filling (GWF) algorithm. (**a**) Illustration of water level step $k^* = 3$, allocated power for the third step s_3^*, and step/stair depth $d_i = 1/a_i$. (**b**) Illustration of $P_2(k)$ (*shadowed area*, representing the total water/power above step k) when $k = 2$. (**c**) Illustration of $P_2(k)$ when $k = 3$. (**d**) Illustration of the weighted case

Figure 2.1a illustrates the concept of k^*. Since the third level is the highest level under water, we have $k^* = 3$. The shaded area denotes the allocated power for the third step by s_3^*.

In the following, we explain how to find the water level step k^* without the knowledge of the water level μ. Let $P_2(k)$ denote the water volume above step k or zero, whichever is greater. The value of $P_2(k)$ can be solved by subtracting the volume of the water under step k from the total power P, as

$$P_2(k) = \left\{P - \left[\sum_{i=1}^{k-1}\left(\frac{1}{a_k} - \frac{1}{a_i}\right)\right]\right\}^+ = \left\{P - \left[\sum_{i=1}^{k-1}\delta_{k,i}\right]\right\}^+, \text{ for } k = 1, \ldots, K. \tag{2.5}$$

Due to the definition of $P_2(k)$ being the power (water volume) above step k, it cannot be a negative number. Therefore we use $\{\cdot\}^+$ in (2.5) to assign 0 to $P_2(k)$ if the result inside the bracket is negative. The corresponding geometric meaning is that the kth level is above water. Note a reminder of the definition of a special case for the summation is:

$$\sum_{i=m}^{n} b_i = 0, \text{ as } m > n. \tag{2.6}$$

Figure 2.1b, c illustrate the concept of $P_2(k)$ for $k = 2$ and $k = 3$ respectively by the shadowed area. As an example of Fig. 2.1c, the water volume under step 3 can be expressed as the sum of the two terms: (i) the step depth difference between the 3rd and the 1st step, $\delta_{3,1}$, and (ii) the step depth difference between the 3rd and the 2nd step, $\delta_{3,2}$. Thus, $P_2(k = 3)$ can be written as

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

and the above result is the shadowed area in Fig. 2.1c, which is also an expansion of the composite form of (2.5). Then, we are ready to have the following proposition:

Proposition 2.1. *The explicit solution to (2.1) is:*

$$s_i = \begin{cases} s_{k^*} + (d_{k^*} - d_i) & 1 \le i \le k^* \\ 0, & k^* < i \le K, \end{cases} \tag{2.7}$$

where the water level step k^* is given as

$$k^* = \max \left\{ k \middle| P_2(k) > 0, \;\; 1 \le k \le K \right\} \tag{2.8}$$

and the power level for this step is

$$s_{k^*} = \frac{1}{k^*} P_2(k^*). \tag{2.9}$$

It is easy to interpret Proposition 2.1 from Fig. 2.1. The first step of the proposed approach is to find the water level step k^*. From Fig. 2.1, we can find that $k = 3$ is the maximal index that makes $P_2(k)$ greater than zero. Therefore, based on (2.8), $k^* = 3$ can be determined. Then the power at this step s_{k^*} can be determined based on (2.9). For those steps with index higher than k^*, no power is assigned. For those steps with index lower than k^*, their power levels are obtained by adding s_{k^*} with the corresponding level depth difference with the k^*th step as shown in (2.7).

Proposition 2.1 provides an explicit constructed solution rather than the implicit solution. The procedure eliminates solving the nonlinear equation as shown in (2.2) and the real number water level μ. The proof of the optimality of the solution will be left to the next subsection when we discuss the weighted case.

2.3 Generalize to Weighted Case

For the weighted case, the generalized problem can be stated as: given $P > 0$, as the total power or volume of the water; the allocated power and the propagation path gain for the ith antenna are given as s_i and a_i respectively, $i = 1, \ldots, K$; and K is the total number of the transmit antennas. Furthermore, the weighted coefficients $w_i > 0, i \in \{1,\ldots,K\}$, and $\{a_i w_i\}_{i=1}^K$ being monotonically decreasing, find that

$$\begin{aligned} &\max_{\{s_i\}_{i=1}^K} \quad \textstyle\sum_{i=1}^K w_i \log(1+a_i s_i) \\ &\text{subject to}: 0 \le s_i,\ \forall i; \\ &\qquad\qquad\quad \textstyle\sum_{i=1}^K s_i = P. \end{aligned} \tag{2.10}$$

Using the proposed geometric approach, we can extend the geometric relation for the weighted case as shown in Fig. 2.1d to obtain the corresponding solution to (2.10).

In Fig. 2.1d, the width of the ith stair/step is denoted as w_i. The value of $1/a_i$ denotes the volume under the ith step to the bottom of the tank. Hence, the step depth of the ith step is given as

$$d_i = \frac{1}{a_i w_i}, \quad i = 1,\cdots,K. \tag{2.11}$$

Then, $P_2(k)$, the water volume above step k, can be obtained using the similar approach as in the previous subsection considering the step depth difference and the width of the stairs as,

$$P_2(k) = \left[P - \textstyle\sum_{i=1}^{k-1} (d_k - d_i)\, w_i\right]^+, \text{ for } k = 1,\ldots,K. \tag{2.12}$$

As an example in Fig. 2.1d, the water volume above step 1 and below step 3 with the width w_1 can be found as: the step depth difference, $(d_3 - d_1)$ multiplying the width of the step, w_1. Therefore, the corresponding $P_2(k=3)$ can be expressed as,

$$P_2(k=3) = [P - (d_3 - d_1)w_1 - (d_3 - d_2)w_2]^+,$$

which is an expansion of (2.12). Then we have the following proposition.

Proposition 2.2. *The explicit solution to (2.10) is:*

$$\begin{cases} s_i = [\frac{s_{k^*}}{w_{k^*}} + (d_{k^*} - d_i)]w_i, \text{ as } 1 \le i \le k^*; \\ s_i = 0, \text{ as } k^* < i \le K, \end{cases} \tag{2.13}$$

where

$$k^* = \max\left\{k \middle| P_2(k) > 0,\ 1 \le k \le K\right\} \tag{2.14}$$

and the power level for this step is

$$s_{k^*} = \frac{w_{k^*}}{\sum_{i=1}^{k^*} w_i} P_2(k^*). \tag{2.15}$$

Proof of Proposition 2.2. System (2.13) implies that

$$\frac{w_{k^*}}{\frac{1}{a_{k^*}}+s_{k^*}} = \frac{w_i}{\frac{1}{a_i}+s_i}, \text{ as } 1 \le i \le k^*. \tag{2.16}$$

Let

$$\lambda = \frac{w_{k^*}}{\frac{1}{a_{k^*}}+s_{k^*}}. \tag{2.17}$$

From a geometric view, λ is the reciprocal of water level μ. According to the definitions of k^* and s_{k^*}, for $k^* < i \le K$, $\frac{w_{k^*}}{\frac{1}{a_{k^*}}+s_{k^*}} > \frac{w_i}{\frac{1}{a_i}+s_i}$ and $s_i = 0$.

Let

$$\sigma_i = \frac{w_{k^*}}{\frac{1}{a_{k^*}}+s_{k^*}} - \frac{w_i}{\frac{1}{a_i}+s_i}. \tag{2.18}$$

Then

$$\begin{cases} \sigma_i > 0, \text{ as } k^* < i \le K \\ \sigma_i = 0, \text{ as } 1 \le i \le k^*. \end{cases} \tag{2.19}$$

Therefore, the following system holds:

$$\begin{cases} \frac{w_i}{\frac{1}{a_i}+s_i} - \lambda + \sigma_i = 0, \text{ as } 1 \le i \le K & \\ s_i \ge 0, & \forall i \\ \sigma_i s_i = 0, & \forall i \\ \sigma_i \ge 0, & \forall i \\ \sum_{i=1}^{K} s_i = P, & \lambda \in \mathbb{R}. \end{cases} \tag{2.20}$$

By observation, the equation and inequality set above is just a set of the KKT conditions of the problem in Proposition 2.2 and the water level μ is equal to the reciprocal of the Lagrange multiplier λ mentioned above. Note that the Lagrange function of the problem in Proposition 2.2 is

$$L(\{s_i\},\lambda,\{\sigma_i\}) = \sum_{i=1}^{K} w_i \log(1+a_i s_i) - \lambda\left(\sum_{i=1}^{K} s_i - P\right) + \sum_{i=1}^{K} \sigma_i s_i. \tag{2.21}$$

Since it is a differentiable convex optimization problem with linear constraints, not only are the KKT conditions mentioned above sufficient, but they are also necessary for optimality. Note that the constraint qualification of the problem (2.10) holds. Proposition 2.2 hence is proved.

Similar to the unweighted case, the first step is to calculate $P_2(k)$, then find the water level step, k^*, from (2.14), which is the maximal index making $P_2(k)$ nonnegative. The corresponding power level for this step, s_{k^*}, can be obtained by applying (2.15). Then for those steps with index higher than k^*, the power level is assigned with zero. For those steps below k^*, the power level is assigned as in (2.13). The first term (s_{k^*}/w_{k^*}) inside the square bracket denotes the depth of the k^*th step to the surface of the water. The second term inside the square bracket denotes the step depth difference of the k^*th step and the ith step. Therefore, the sum inside the square bracket means the depth of the ith step to the surface of the water. When this quantity is multiplied with the width of this step, the volume of the water above this step (allocated power) can be then readily obtained.

With the proposed GWF approach, the weighted problem could be solved straightforwardly, avoiding complicated derivation and calculation. When the weighting factors are set to ones, the corresponding unweighted case is obtained. In the following description of algorithm implementation and proof, we only provide weighted case.

From Proposition 2.2, when k^* is obtained, $P_2(k^*)$ is given. Then it is memorized and only multiplied by a constant to compute s_{k^*}. Thus, how to search k^* is a key point for the proposed GWF and the procedure is stated as follows:

1. Initialize $W_s = 0; P_M = P^* = P; i = 1$.
2. Compute $W_s <= W_s + w_i; P^* <= P^* - (d_{i+1} - d_i)W_s$. Then $i <= i+1$, where the symbol "$<=$" represents the assignment operation.
3. If $P^* > 0$ and $i \leq K$, $P_M = P^*$, and repeat the step 2; else, output $k^* = i-1, W_s = W_s - w_i$ and $s_{k^*} = \frac{w_{k^*}}{W_s} P_M$.

We can observe that $\frac{s_{k^*}}{w_{k^*}} + d_{k^*}$ is the water level due to $\frac{s_{k^*}}{w_{k^*}} + d_{k^*} = \frac{s_i}{w_i} + d_i$, for $1 \leq i \leq k^*$.

As an alternative to the enumeration search in the Algorithm GWF, a Fibonacci-like search is possibly used to speed up finding k^* due to (non-increasing) monotonicity of the sequence $\{P_2(k)\}$. Without loss of generality, let Fibonacci approximation ratios be $\frac{1}{3}$ and $\frac{2}{3}$ for searching k^*. The method can be described as:

1st Step. Assume that $a = 1$ and $b = K$.
2nd Step. If $a = b$, then $k^* = a$ and go to Step 3 of GWF.
Else, $a_1 = \lfloor a + \frac{1}{3}(b-a) \rfloor$, $b_1 = \lceil a + \frac{2}{3}(b-a) \rceil$.
3rd Step. If $P_2(a_1) \leq 0$, then $b = a_1 - 1$ and go to the 2nd Step;
If $P_2(b_1) > 0$, then $a = b_1$ and go to the 2nd Step;
If $P_2(a_1) > 0$ and $P_2(b_1) \leq 0$, then $a = a_1$, $b = b_1 - 1$ and go to the 2nd Step.

The number of loops to search k^* is reduced into a complexity level of $\log_3(K)$.

2.4 Weighted Water-Filling with Individual Peak Power Constraints

In this section, we extend the CWF problem to include individual peak power constraints (WFPP).

The weighted WFPP problem is stated as follows. Given $P > 0$, as the total power or volume of the water; the allocated power and the propagation path gain for the ith antenna are given as s_i and a_i respectively, $i = 1, \dots, K$; and K is the total number of the transmit antennas. Also, the weights $w_i > 0, \forall i$, and without loss of generality, $\{a_i \cdot w_i\}_{i=1}^{K}$ being positive and monotonically decreasing, find that

$$\begin{aligned} &\max_{\{s_i\}_{i=1}^{K}} && \textstyle\sum_{i=1}^{K} w_i \log(1 + a_i s_i) \\ &\text{subject to}: && 0 \le s_i \le P_i,\ \forall i; \\ & && \textstyle\sum_{i=1}^{K} s_i \le P. \end{aligned} \tag{2.22}$$

Comparing the problem (2.22) with (2.10), the constraint of $0 \le s_i$ is extended to $0 \le s_i \le P_i$, i.e., additional individual peak power constraints, and $\sum_{i=1}^{K} s_i = P$ to $\sum_{i=1}^{K} s_i \le P$. The problem (2.22) is thus referred to as (weighted) water-filling with sum and individual peak power constraints (WFPP). In this section, we discuss the solution to the WFPP problem.

Proposition 2.2 in Sect. 2.3 provides an explicit solution using geometric view approach. Interestingly, the proposed GWF can be applied to the WFPP problem with some modifications. The following presents an algorithm which is a modification of the above discussed GWF and it is termed as the GWFPP.

For convenience, the expression (2.12) can further be extended into the expression:

$$P_2(i_k) = \left[P - \textstyle\sum_{t=1}^{|E|-1} \left(d_{i_k} - d_{i_t}\right) w_{i_t}\right]^{+}, \text{for } k = 1, \dots, |E|, \tag{2.23}$$

where E is a subsequence of the sequence $\{1, 2, \dots, K\}$, $|E|$ is the cardinality of the set E, so E can be expressed as $\{i_1, i_2, \dots, i_{|E|}\}$. Especially, if E is taken as the sequence $\{1, 2, \dots, K\}$, then the extended expression is regressed into the original expression (2.12). Similarly, some corresponding changes in (2.13)–(2.15) are also done (i.e., the subscripts of sequence are replaced with those of the subsequence). For avoiding tediousness, these extended expressions are still labelled as (2.13)–(2.15) in the following statement of Algorithm GWFPP.

Algorithm GWFPP:

Input: vector $\{d_i\}, \{w_i\}, \{P_i\}$ for $i = 1, 2, \dots, K$, the set $E = \{1, 2, \dots, K\}$, and P.

1. Utilize (2.13)–(2.15) to compute $\{s_i\}$.
2. The set Λ is defined by the set $\{i | s_i > P_i, i \in E\}$. If Λ is the empty set, output $\{s_i\}_{i=1}^{K}$; else, $s_i = P_i$, as $i \in \Lambda$.
3. Update E with $E \setminus \Lambda$ and P with $P - \sum_{t \in \Lambda} P_t$. Then return to (1) of the GWFPP.

Remark 2.1. Algorithm GWFPP is a dynamic power distribution process. The state of this process is the difference between the individual peak power sequence and the current power distribution sequence obtained by the Algorithm GWF. The control of this process is to use (2.13)–(2.15) of the Algorithm GWF based on the state mentioned above. Thus, a new state for next time stage appears. Therefore, an optimal dynamic power distribution process, the GWFPP, with the state feedback is formed. Since the finite set E is getting smaller and smaller until the set Λ is empty, Algorithm GWFPP carries out K loops to compute the optimal solution, at most.

Similar to the proof of the Proposition 2.2, we can obtain the following conclusion:

Proposition 2.3. *Algorithm GWFPP can provide the optimal solution to the problem (2.22).*

Proof of Proposition 2.3. If the final set E in Algorithm GWFPP is empty, it implies that $\sum_{i=1}^{K} P_i \leq P$. Then it is easy to see the optimal solution $s_i = P_i$, for any i.

If it is non-empty, observing the stricture of (2.22), Proposition 2.3 is easily proved, similarly to the previous one.

2.5 Complexity Analysis

As stated in [1] (Sect. 2.3), the conventional WF algorithm had an exponential worst-case complexity of 2^K, where K is the number of the channels, even though the channel gains had been sorted in decreasing order. Pointing to this case, [1] proposed an improved algorithm with worst-case complexity of K iterations. Since each iteration consists of multiple arithmetic and logical operations, here we use total number of operations as a measure of the complexity level (See [2], Chap. 8).

The CWF approach has a worst-case complexity of K iterations, i.e., total $O(K^2)$ fundamental arithmetic and logical operations under the $2(K+1)$ memory requirement and the sorted parameters $\{w_k a_k\}_{k=1}^{K}$ (e.g. see [3], page 137, for more details).

The proposed GWF algorithm occupies less computational resource. It is seen that it needs K loops at most to search k^* and it needs four arithmetic operations and two logical operations to complete each loop. Thus, the worst-case computational complexity of the proposed solution is $8K+3$ (from the operations of $6K+3+2K$) fundamental arithmetical and logical operations under the $2(K+1)$ memory units to store $\{d_i\}$, $\{w_i\}$, W_s, and P_M.

For the GWFPP, it needs K loops to compute the optimal solution, at most. The required number of operations is, at worst, $\sum_{i=1}^{K}(8i+3) = 4K^2+7K$ fundamental arithmetical and logical operations.

Note that the content of this chapter comes partially from [4] and references therein.

References

1. D. Palomar, “Practical algorithms for a family of waterfilling solutions,” IEEE Transactions on Signal Processing, vol. 53, pp. 686–695, 2005.
2. C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Unabridged edition, Dover Publications, Mineola, 1998.
3. D. Palomar, PhD Thesis: A Unified Framework for Communications through MIMO Channels, Universitat Politecnica De Catalunya, Spain, 2003.
4. 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 Transactions on Wireless Communications, vol. 12, pp. 3637–3647, 2013.

Springer

http://www.springer.com/978-3-319-04635-8

Radio Resource Management Using Geometric
Water-Filling
He, P.; Zhao, L.; Zhou, S.; Niu, Z.
2014, XI, 87 p. 18 illus., 10 illus. in color., Softcover
ISBN: 978-3-319-04635-8