King Fahd University of Petroleum & Minerals Computer Engineering Dept

Adaptive Power Allocation Algorithm To Support Absolute Proportional Rates Constraint For Scalable OFDM Systems

Ashraf S. Hasan Mahmoud, Ali Al-Rayyah
Email: {ashraf, g200604240}@kfupm.edu.sa

Presentation Plan

- Assumptions and Problem Definition
- Sub-carrier allocation by Rhee
- Optimal power allocation by Shen
- Proposed Methods:
  - Method 1: Optimal Power Allocation with strict QoS guarantee
  - Method 2: Slope method with subchannel auctioning
- Conclusions

Assumptions

- Downlink
- K users – total of N sub-carriers (called sub-channels)
- Total transmit power, Ptotal
- N0 is the power spectral density of AWGN
- B is total system bandwidth
- $p_{k,n}$ is power allocated for user k in sub-channel n
- $h_{k,n}$ is the channel gain for user k in sub-channel n
- $\rho_{k,n}$ is the sub-channel allocation indicator; equal to 1 if the nth sub-channel is allocated for kth user, 0 otherwise

NOTE:
$\rho_{k,n}$ is discrete – either 0 or 1 –
Sub-channels can not be shared!

Optimization Problem

Very hard problem to solve – involves both continuous variables $p_{k,n}$ and discrete one $\rho_{k,n}$

- The objective is to maximize equally weighted sum of capacities $\max_{p_{k,n},\rho_{k,n}} \sum_{k=1}^{K}\sum_{n=1}^{N} \frac{\rho_{k,n}}{N}\log_2\left(1+p_{k,n}H_{k,n}\right)$

where $H_{k,n} \triangleq h_{k,n}^2 / \left(N_0\left(B/N\right)\right)$ is the channel gain to noise power ratio for the nth subchannel as measured by kth user.

- Subject to the following constraints:

Power constraint $\Rightarrow$ $\begin{cases} \sum_{k=1}^{K}\sum_{n=1}^{N} p_{k,n} \le P_{\text{total}} \\ p_{k,n} \ge 0 \quad \text{for all } k,n \end{cases}$

Allocation constraint $\Rightarrow$ $\rho_{k,n} \in \{0,1\} \quad \text{for all } k,n \qquad \sum_{k=1}^{K}\rho_{k,n} = 1 \quad \text{for all } n$

Proportional Fairness $\Rightarrow$ $\frac{R_1}{\gamma_1} = \frac{R_2}{\gamma_2} = \cdots = \frac{R_K}{\gamma_K}$

Optimization Problem – cont'd

- The kth user total user rate, Rk, is given by

$$R_k = \sum_{n=1}^{N} \frac{\rho_{k,n}}{N} \log_2\left(1 + p_{k,n} H_{k,n}\right)$$

- The fairness index

$$F = \frac{\left(\sum_{k=1}^{K} R_k / \gamma_k\right)^2}{K \sum_{k=1}^{K} \left(R_k / \gamma_k\right)^2}$$

- The set $\{\gamma_k\}_{k=1}^{K}$ is given.

F = 1 ➔ Rates constraint is 100% satisfied
F = 0 ➔ Rates constraint is 100% NOT satisfied

Ideally, R_k/γ_k is equal for all k

- Note that if $\gamma_k = 1$ for all k, then the optimization problem reduces to the max-min problem (maximum fairness problem)

Comments on the Optimization Problem

- Contains both continuous and discrete (binary) variables
- The previous problem is known as "Mixed binary integer programming problem"
- The non-linear constraints increase the difficulty of finding the optimal solution
- For K users and N sub-channels, there are total of K^N possible subchannel allocations
  - The exhaustive search solution is nearly impossible for large K and N!!

Solution Proposed by Rhee [1]

- Assume that ALL subchannels carry same power share equal to Ptotal/N
- We need to distribute the N subchannels to K users.
- Compute Ω_k (set of allocated subchannels for kth user) for k = 1, 2, …, K as follows:

1) Initialization
   a) Set $R_k = 0$, $\Omega_k = \varnothing$ for $k = 1, 2, \ldots, K$ and $A = \{1, 2, \ldots, N\}$.
2) For $k = 1$ to K,
   a) find n satisfying $|H_{k,n}| \geq |H_{k,j}|$ for all $j \in A$;
   b) let $\Omega_k = \Omega_k \cup \{n\}$, $A = A - \{n\}$ and update R_k according to (2).
3) While $A \neq \varnothing$,
   a) find k satisfying $R_k/\gamma_k < R_i/\gamma_i$ for all i, $1 < i < K$;
   b) for the found k, find n satisfying $|H_{k,n}| \geq |H_{k,j}|$ for all $j \in A$;
   c) for the found k and n, let $\Omega_k = \Omega_k \cup \{n\}$, $A = A - \{n\}$ and update R_k according to (2).

Optimal Power Distribution for FIXED Sub-channel Allocation (Shen's Method [2][3])

- Rhee's algorithm is used to obtain Ωk (set of allocated subchannels for kth user) for k = 1, 2, …, K
- Now we need to compute pk,n that fulfills the following optimization problem is:

$$\max_{p_{k,n}} \sum_{k=1}^{K} \sum_{n \in \Omega_k} \frac{1}{N} \log_2\left(1 + p_{k,n} H_{k,n}\right)$$

- Subject to $\sum_{k=1}^{K} \sum_{n \in \Omega_k} p_{k,n} \leq P_{\text{total}}$ and $p_{k,n} \geq 0 \quad$ for all k, n

$$\Omega_k \quad \text{are disjoint for all } k$$

$$\Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_K \subseteq \{1, 2, \cdots N\}$$

$$\frac{R_1}{\gamma_1} = \frac{R_2}{\gamma_2} = \cdots = \frac{R_K}{\gamma_K}$$

Optimal Power Distribution - Solution [2][3]

- The paper focuses on how to solve this optimization problem using (semi)analytic method
- Using *conventional* Lagrange Multipliers method, the corresponding cost function is given by

$$L=\sum_{k=1}^{K}\sum_{n\in\Omega_k}\frac{1}{N}\log_2\left(1+p_{k,n}H_{k,n}\right)+\lambda_1\left(\sum_{k=1}^{K}\sum_{n\in\Omega_k}p_{k,n}-P_{\text{total}}\right)$$

$$+\sum_{k=2}^{K}\lambda_k\left(\sum_{n\in\Omega_1}\frac{1}{N}\log_2\left(1+p_{1,n}H_{1,n}\right)-\frac{\gamma_1}{\gamma_k}\sum_{n\in\Omega_k}\frac{1}{N}\log_2\left(1+p_{k,n}H_{k,n}\right)\right)$$

The total power constraint

The K-1 Rate constraints

The set of $\{\lambda_k\}$ are the set of Lagrange Multipliers
The unknowns are power allocations $p_{k,n}$'s

Optimal Power Distribution - Solution [2][3] – cont'd(1)

- Differentiate L with respect to $p_{k,n}$ and equate each derivative to zero, we get

$$\frac{H_{k,m}}{1+H_{k,m}p_{k,m}}=\frac{H_{k,n}}{1+H_{k,n}p_{k,n}}\qquad\forall m,n\in\Omega_k,k=1,2,\cdots,K$$

- If we order the normalized channel power gains in the set Ω_k (of size N_k) such that $H_{k,1} \leq H_{k,2} \leq \ldots \leq H_{k,Nk}$, then the previous equation can be rewritten as:

Power distribution for single user

$$p_{k,n}=p_{k,1}+\frac{H_{k,n}-H_{k,1}}{H_{k,n}H_{k,1}}\qquad n=1,2,\ldots,N_k \text{ and } k=1,2,\cdots,K$$

Eq (10)

Optimal Power Distribution - Solution [2][3] – cont'd(2)

- The power distribution for single user – shows that more power will be assigned to sub-channels with higher $H_{k,n}$ ➔ WATER-FILLING Algorithm
- The total power assigned to a user k is then given by

$$P_{k,\text{tot}} = \sum_{n=1}^{N_k} p_{k,n} = N_k p_{k,1} + \sum_{n=1}^{N_k} \frac{H_{k,n} - H_{k,1}}{H_{k,n} H_{k,1}} \qquad k = 1, 2, \cdots, K$$

- We can not use this relation ALONE to find $P_{k,tot}$ – $p_{k,1}$ is STILL UNKNOWN

This relation computes the total power assigned to EVERY user k

Optimal Power Distribution - Solution [2][3] – cont'd(3) {KEY RESULT IN PAPER}

- If we have $\{P_{k,tot}\}$ for all k, then we can use the previous relation to determine pk,1 and then all pk,n
- How to find $\{P_{k,tot}\}$?
- Solution: use the previous formula and substitute in the capacity (proportions) equation

$$\frac{1}{\gamma_1}\frac{N_1}{N}\left[\log_2\left(1 + H_{1,1}\frac{P_{1,\text{tot}} - V_1}{N_1}\right) + \log_2 W_1\right]$$

Eq (12)

$$= \frac{1}{\gamma_k}\frac{N_k}{N}\left[\log_2\left(1 + H_{k,1}\frac{P_{k,\text{tot}} - V_k}{N_k}\right) + \log_2 W_k\right]$$

THE KEY RESULT IN PAPER

For k=2,3,…, K; where Vk and Wk are given by

$$V_k = \sum_{n=2}^{N_k} \frac{H_{k,n} - H_{k,1}}{H_{k,n} H_{k,1}}$$ **Eq (13)**

$$W_k = \left(\prod_{n=2}^{N_k} \frac{H_{k,n}}{H_{k,1}}\right)^{\frac{1}{N_k}}$$ **Eq (14)**

For a given allocation of sub-channels Vk and Wk are constants!

Optimal Power Distribution - Solution [2][3] – cont'd(4)

- The "Eq (12)" describes a set of K-1 non-linear equations
- The power constraint represents the K^{th} needed equation is

$$\sum_{k=1}^{K} P_{k,\mathrm{tot}} = P_{\mathrm{total}} \qquad \textbf{Eq (15)}$$

- The above system of K non-linear equations need to be solved for $\{P_{k,tot}\}$
  - Use Iterative methods such as Newton-Raphson
  - Refer to Appendix II of paper for Shen's implementation of Newton-Raphson method
  - Matlab can also be used

Optimal Power Distribution - Solution [2][3] – cont'd(5)

- Shen in his paper DID NOT solve this system of non-linear equations ***for an arbitrary case***

- Shen in his paper DID simplify this system for two cases:
  - Linear case – refer to backup slides
  - High Hk,n case – refer to backup slides

Observation on Optimal Power Distribution Solution

- Note: In general a solution for the original K nonlinear equation (or even for the two special cases) MAY NOT exist for the specified Ωk's
  - In Shen's Matlab code: when a solution does not exist, then the problem is solved using Rhee's algorithm.
  - Therefore, it is an open issue of **how to** update Ωk
    - Intuitively, this is done by dropping weak channels first, recalculate Nk, Vk, and Wk and attempt to solve
      - For what k? (i.e. user)
  - Repeat until a solution is found
- When evaluating the final solution {pk,n} with regard to the Rate Proportionality constraint (i.e. computing fairness index F on slide 5), we find that the constraint is not always satisfied
  - Due to the update mechanism for the Ωk's

Proposed Method for Strict QoS Guarantee

- Starting with the system of non-linear equations:

$$\frac{1}{\gamma_1}\frac{N_1}{N}\left[\log_2\left(1+H_{1,1}\frac{P_{1,\text{tot}}-V_1}{N_1}\right)+\log_2 W_1\right]=\frac{1}{\gamma_k}\frac{N_k}{N}\left[\log_2\left(1+H_{k,1}\frac{P_{k,\text{tot}}-V_k}{N_k}\right)+\log_2 W_k\right] \quad (2)$$

for k=2, 3, …, K; and

$$\sum_{k=1}^{K} P_{k,\text{tot}} = P_{\text{total}}$$

- Let us define the quantity Xk given by

$$X_k = 1 + H_{k,1}\frac{P_{k,\text{tot}}-V_k}{N_k}$$

Proposed Method for Strict QoS Guarantee – cont'd (2)

- Then $P_{k,tot}$ can be computed in terms of Xk using

$$P_{k,\text{tot}} = \frac{N_k}{H_{k,1}}\left(X_k - 1\right) + V_k \quad (3)$$

- Substituting (3) into (1) we find

$$X_k = \left[\left(X_i W_i\right)^{(\gamma_k N_i)/(\gamma_i N_k)}\right] \Big/ W_k \quad (4)$$

- Now, using the total power constraint, we write

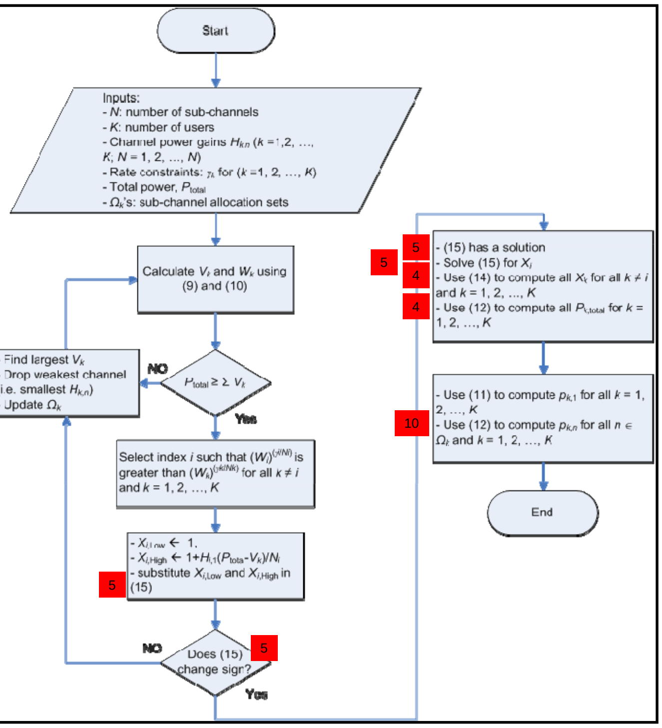
$$\sum_{k=1}^{K}\left[N_k\left(\left(\left[\left(X_i W_i\right)^{(\gamma_k N_i)/(\gamma_i N_k)}\right] \Big/ W_k - 1\right) + V_k\right) \Big/ H_{k,1}\right] - P_{\text{total}} = 0 \quad (5)$$

Proposed Method for Strict QoS Guarantee – cont'd (3)

- Equation (5) is has one unknown (Xi) – can be solved numerically using Matlab's "fzero".
- However, since (5) is derived from the original problem formulation given by (2) → a solution is not guaranteed for the given Ωk's
- We devise an algorithm for updating Ωk's and solving (5) such that the rate proportionality constraint is satisfied 100%

Proposed Algorithm

- The equation references in the flowchart are different than the numbers in this presentation – The correct number is written on the side
- The updating of Ωk's



Sample Results

- P_{total} = 1 Watts; N_0 =-65 dBW per Hz. N = 64 sub-channels in a 1 MHz bandwidth configuration.

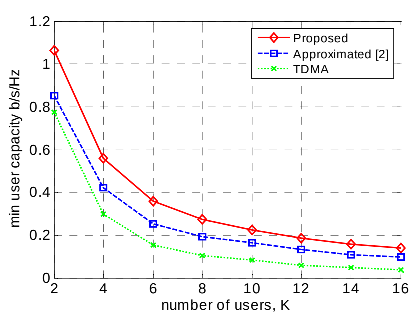


Fig.1: minimum user capacity for multiuser OFDM versus number of users for proposed algorithm and other methods.

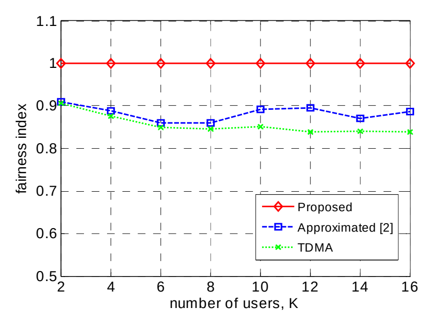


Fig.2: Fairness index and adherence to the proportional rates constraint for proposed algorithm and original algorithm.

Further Work

- The slope method
  - Combines subchannel and power allocation
  - Iterative method.
- Allocation for Heterogeneous (Real-time and Non real-time users)
- Extension to MIMO systems

References

1. Wonjong Rhee and John M. Cioffi, "Increase in Capacity of Multiuser OFDM System Using Dynamic Subchannel Allocation," VTC 2000, Tokyo, Japan. {main reference on sub-carrier allocation}
2. Zukang Shen, Jeffrey G. Andrews, and Brian L. Evans , "Optimal Power Allocation in Multiuser OFDM Systems," Globecom '03,
3. Zukang Shen, Jeffrey G. Andrews, and Brian L. Evans, "Adaptive Resource Allocation in Multiuser OFDM Systems With Proportional Rate Constraints," IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 4, NO. 6, NOVEMBER 2005. {same as the Globecom paper but expanded}
   1. http://users.ece.utexas.edu/~bevans/projects/ofdm/software/ - {matlab code for above two papers}!!

BACKUP SLIDES

Solution Steps Proposed by Shen [2][3]

Two Special Cases of The Key Result – Linear Case

- Linear Case: if N1:N2:…:NK=γ1:γ2:…:γK, then the system of Eq (12) can be written as

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & a_{K,K} \end{bmatrix} \begin{bmatrix} P_{1,\text{tot}} \\ P_{2,\text{tot}} \\ \vdots \\ P_{K,\text{tot}} \end{bmatrix} = \begin{bmatrix} P_{\text{total}} \\ b_2 \\ \vdots \\ b_K \end{bmatrix} \quad \textbf{Eq (16)}$$

where $a_{k,k}$ and b_k (for k=2, 3, …, K) are given by

$$a_{k,k} = -\frac{N_1}{N_k}\frac{H_{k,1}W_k}{H_{1,1}W_1} \qquad b_k = \frac{N_1}{H_{1,1}W_1}\left(W_k - W_1 + \frac{H_{1,1}V_1W_1}{N_1} - \frac{H_{k,1}V_kW_k}{N_k}\right)$$

- Solving the above system yields the required $P_{k,tot}$ variables – and then all $p_{k,n}$ can be found as before.

Two Special Cases of The Key Result – High Hk,n Case

- High Hk,n Case:
  - If $H_{k,n} \sim H_{k,1}$ ➔ $V_k \sim 0$,
  - If $H_{k,1} P_{k,tot}/N_k >> 1$
- Then the key equation Eq (*) can be rearranged and simplified to be

$$\left(\frac{H_{1,1}W_1}{N_1}\right)^{\frac{N_1}{\gamma_1}} \left(P_{1,\text{tot}}\right)^{\frac{N_1}{\gamma_1}} = \left(\frac{H_{k,1}W_k}{N_k}\right)^{\frac{N_k}{\gamma_k}} \left(P_{k,\text{tot}}\right)^{\frac{N_k}{\gamma_k}}$$

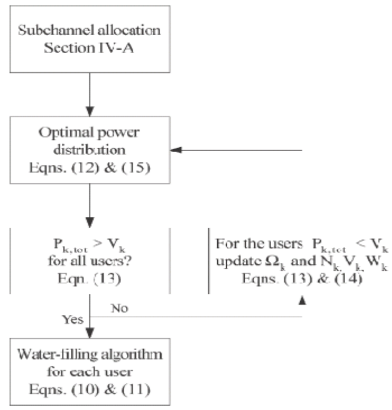
for k=2,3, …, K. Since sum of all $P_{k,tot}$ must be P_{total} then one can write

$$P_{1,\text{tot}} + \sum_{k=2}^{K} \frac{\left(H_{1,1}W_1/N_1\right)^{\frac{N_1\gamma_k}{\gamma_1 N_k}}}{H_{k,1}W_k/N_k} \left(P_{1,\text{tot}}\right)^{\frac{N_1\gamma_k}{\gamma_1 N_k}} - P_{\text{total}} = 0 \quad \textbf{Eq (20)}$$

- The zero P1,tot of above equation can be found using numerical iterative techniques (check Matlab)
- After P1,tot is found, then all Pk,tot can be found from the first equation.

Existence of Solution for Single User – For Optimal Power Allocation Method by Shen [2][3]

- A power allocation for user k, Pk,tot exists only if Vk is LESS THAN Pk,tot – Refer to the KEY RESULT SLIDE
- If this is not the case, the sub-channel algorithm (Step 1) need to be rerun and the allocated sub-channels (corresponding Ωk) need to be changed – this means update Ωk, then update Nk, Vk, and Wk accordingly
  - In Shen's Matlab code – Ωk was not updated – but power allocations and rates were computed using Rhee's algorithm!!
- Therefore, it remains an open issue of how to update Ωk's.
  - Logically you want to drop the weakest subchannels first.



Resource Allocation Algorithm

Algorithm 1

Optimal Power Allocation – Water Filling

- Water filling procedure provides the optimal power allocation for a given subchannel assignment.
- The cost function specified on slide 9 along with the constraints is given by

$$L = \sum_{k=1}^{K}\sum_{n\in\Omega_k}\frac{1}{N}\log_2\left(1+p_{k,n}H_{k,n}\right)+\lambda_1\left(\sum_{k=1}^{K}\sum_{n\in\Omega_k}p_{k,n}-P_{\text{total}}\right)$$

$$+\sum_{k=2}^{K}\lambda_k\left(\sum_{n\in\Omega_1}\frac{1}{N}\log_2\left(1+p_{1,n}H_{1,n}\right)-\frac{\gamma_1}{\gamma_k}\sum_{n\in\Omega_k}\frac{1}{N}\log_2\left(1+p_{k,n}H_{k,n}\right)\right)$$

- Differentiating L with respect to the unknown $p_{k,n}$ results in the following equality

$$\frac{H_{k,m}}{1+H_{k,m}p_{k,m}}=\frac{H_{k,n}}{1+H_{k,n}p_{k,n}} \qquad \forall m,n\in\Omega_k, k=1,2,\cdots,K$$

or

$$\frac{1}{H_{k,m}}+p_{k,m}=\frac{1}{H_{k,n}}+p_{k,n}=\text{constant} \qquad \forall m,n\in\Omega_k, k=1,2,\cdots,K$$

- This constant is referred to as the "water level"

Water Filling Algorithm

- Given
  - A set of channel gains: H1, H2, …, Hn
  - A total power Ptot
- Required – user water filling to find optimal allocations p1, p2, …, pn.

- Solution:
- we must find p1, p2, …, pn such that

$$1/H1 + p1 = 1/H2 + p2 = \ldots = 1/Hn + pn = C$$

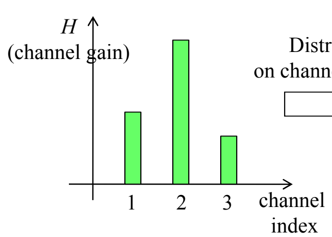
  - where p1 + p2 + … + pn = Ptot, and
  - C is the water level

- Note the pi = max(C – 1/Hi, 0) = $(C\text{-}1/Hi)^{+}$ for i = 1, 2, …, n.

This is what Mohanram is using in step 4(e) of his algorithm!!

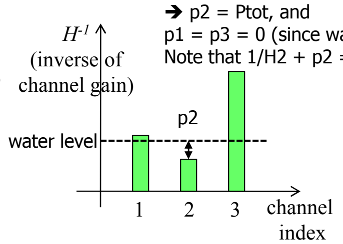
Water Filling Algorithm (2)

- Consider the following examples:

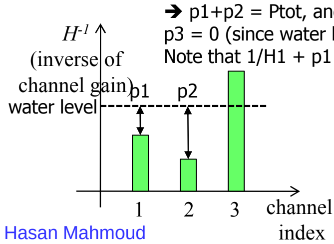


Distribute Ptot on channels 1, 2, and 3

Case 1: Ptot small
→ p2 = Ptot, and
p1 = p3 = 0 (since water level is low)!!
Note that 1/H2 + p2 = water level

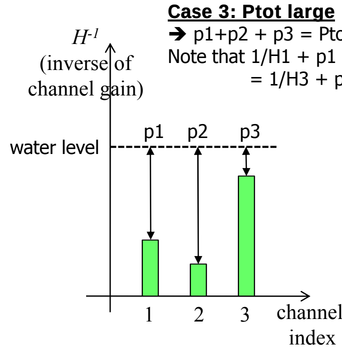


Case 2: Ptot medium
→ p1+p2 = Ptot, and
p3 = 0 (since water level is not enough)
Note that 1/H1 + p1 = 1/H2 + p2 = wat



Water Filling Algorithm (3)

- Observation on water filling
  - The algorithm starts by allocating power to the strongest channels
  - The stronger channel is always allocated power greater than that allocated for relatively weaker channel
  - If Ptot is not sufficient (i.e. water level is not high), for weak channels $(C - 1/Hi)^+$ will be equal to zero – i.e. the power allocation will be zero.



Case 3: Ptot large
➔ p1+p2 + p3 = Ptot, and
Note that 1/H1 + p1 = 1/H2 + p2
= 1/H3 + p3 = water level

- We need a Matlab code that takes a vector of Hs and Ptot as input and returns the corresponding vector of power allocations, Ps computed as per the water filling algorithm.

Water Filling Algorithm (4) – Matlab Code for Calculation of Pi

```
function [Ps, C] = MyWaterFilling(Hs, Ptot);
% return Ps - the power allocations corresponding to the Hs
%         C  - the water level
n  = length(Hs);
Ps = zeros(size(Hs));

if (Ptot == 0)
    return; % will be zero for all channels'
end
% Ptot is NOT equal to zero –
%the at least the strongest channel will have power!!
[SortedHs, Indices]  = sort(Hs); % store the indices
C = (Ptot + sum(1./SortedHs))/n;
P = C - 1./SortedHs; % temporary power calculation
Sign = (P > 0); k = 0; % test for elimination of weak channels
while (sum(Sign) ~= (n-k)) && (k <= n)
    % eliminate the weakest channel
    k = k + 1; SHT = SortedHs(k+1:n);
    C = (Ptot + sum(1./SHT))/(n-k);
    P = C - 1./SHT;
    Sign = (P > 0);
end
Ps(Indices(k+1:n)) = P;
%fprintf('strongest k           = %3d users were allocated
     power\n', n-k);
%fprintf('Water level           = %7.3f\n', C);
%fprintf('Total allocated power = %7.3f\n', sum(P));
%fprintf('Allocations: ');
%for i=k+1:n, fprintf('P[%2d] = %7.3f, ', i-k, P(i-k)); end
%fprintf('\n');
```

<u>The idea of the code is as follows:</u>
- Sort the channel gains, Hs
- Since $pi = (C - 1/Hi)^+$ for I =1, 2, …, n and $\Sigma pi = Ptot$, then

$$\Sigma\, pi = n\,C - \Sigma\,(1/Hi) = Ptot,$$

therefore,

$$C = (Ptot + \Sigma\,(1/Hi))/n$$

This is provided that C is the true water level and all pi's are positive.

Therefore, we iteratively compute C and pi's until all pi's are positive. For every failed, iteration we eliminate the weakest channel out of the remaining channels

<u>Note</u> that if Ptot is NOT zero, then at least we can allocate the entire Ptot to the strongest channel!!

Water Filling Algorithm in Mohanram's Paper

- The channel gains are denoted by γk,n
- Every allocated channel comes with Ptotal/N share – these shares are accumated in Pk, which is the total power for the kth user.
- Water filling is used to distribute this Pk over the channels owned by the kth user (step 4(e).
  - γ is the water level (previously called constant C)

The joint subcarrier and power allocation strategy is as follows.

1. Initialize $A = \{1, 2, 3, ...N\}$
2. $\forall\, k = 1$ to K, $A_k = \phi, P_k = 0$
3. $\forall\, k = 1$ to K,
   (a) $\gamma_k = \max_n \gamma_{k,n}$ for $n \in A$
   (b) $A_k = A_k \cup \{n\}$, $P_k = P_k + \frac{P_{total}}{N}$
   (c) $R_k = \log_2(1 + P_k\gamma_k)$
   (d) $A = A - \{n\}$
4. While $A \neq \phi$,
   (a) find i such that $\frac{R_i}{\alpha_i} \leq \frac{R_k}{\alpha_k}\ \forall\, k, i = 1$ to K
   (b) for the above i, find n such that
   $$\gamma_{i,n} \geq \gamma_{i,m}\ \forall\, n, m \in A$$
   (c) $A_i = A_i \cup \{n\}$, $P_i = P_i + \frac{P_{total}}{N}$
   (d) $A = A - \{n\}$
   (e) $R_i = \sum_{n \in A_i} log_2(1 + P_{i,n}\gamma_{i,n})$ where $P_{i,n} = \left(\gamma - \frac{1}{\gamma_{i,n}}\right)^+$ and $\sum_{n \in A_i} P_{i,n} = P_i$

The $f(x) = (x)^+$ operator indicates that $f(x) = 0$ when $x < 0$ and $f(x) = x$ when $x \geq 0$.