King Fahd University of Petroleum & Minerals
Computer Engineering Dept

OFDMA Subchannel and Power Allocation Problem
COE 540 – Computer Networks
Term 112
Dr. Ashraf S. Hasan Mahmoud
Rm 22-420
Ext. 1724
Email: ashraf@kfupm.edu.sa

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 Hk,n is the channel gain to noise power ratio for the nth sub-channel as measured by kth user.

- Subject to the following constraints:

Power constraint ⇨

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

Allocation constraint ⇨

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

Proportional Rates Constraint (PRC) ⇨

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

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 (Jian’s) fairness index

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

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

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

- 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

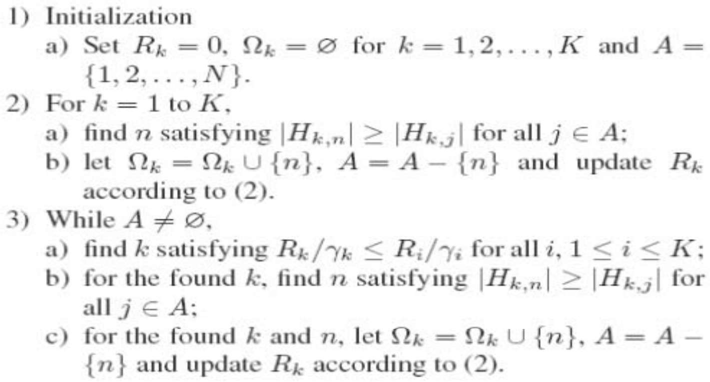
- For K users and N sub-channels, there are total of K^N possible subchannel allocations
    - The exhaustive search solution is nearly impossible for large K and N!!

Algorithm 1 – Rhee et al. [1]

- Simple low complexity steps
- Arrive at some suboptimal solution
- Splits the problem into two sub-problems:
  - Power allocation problem – Assume all sub-channels will receive EQUAL amount of power; i.e. $p_{k,n}$ = PTotal / N.
  - Sub-channel allocation problem – a simple low complexity procedure is used to compute the Ωk's

Algorithm 1 – Rhee et al. [1] – cont'd

- Procedure for computing the sub-channel allocations, Ωk's :



```
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 $R_k/\gamma_k \leq R_i/\gamma_i$ for all $i$, $1 \leq i \leq K$;
    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).
```

Algorithm 1 – Rhee et al. [1] – comments

- Step 2 – distributes the larges K sub-channel power gains to noise ratio to the K users; one sub-channel for each user
- Step 3 – consider the PRC and attempts to satisfy it while increasing the overall throughput at each step

Task 1 – Sat March 3^{rd}, 2012

- Implement Rhee's algorithm and evaluate its performance
- Produce curves for N = 64 sub-channels and a variable number of users K = 2 to 14 for both the average capacity figure and average fairness index.
  - Use the provided GetKNChannel.m routine to obtain realizations of the channel power gain to noise ratio matrix.
  - Average your results over enough number of realizations.

Algorithm 2 – Mohanram et al. [3]

- Yet another simple and relatively low complexity algorithm
- Arrive at some suboptimal solution
- Attempts to solve the power and sub-channel allocation problems **jointly**:
    - Power allocation problem
        - Initially assume a sub-channels is associated with an EQUAL amount of power; i.e. $p_{k,n}$ = PTotal / N.
        - Once assigned to the user, the user aggregates the total power in the assigned channels and performs **waterfilling** → to optimize capacity
    - Sub-channel allocation Ωk updated (similar to Rhee's algorithm) based on rates computed after **waterfilling** is applied

Algorithm 2 – Mohanram et al. [3] – cont'd

- What is waterfilling?
- What are the detailed steps of Algorithm 2?
    - Please note that waterfilling is only a step in Algorithm 2.

Waterfilling Algorithm

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

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

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

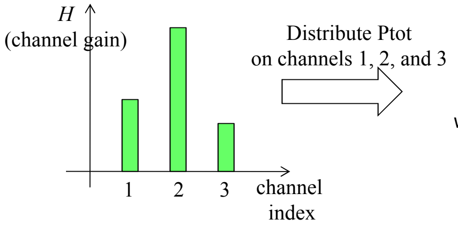
  - where p1 + p2 + … + pn = Ptot, and
  - C is the water level

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

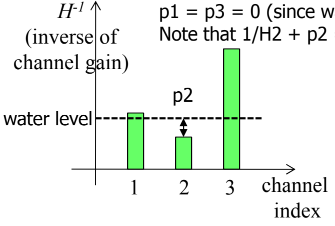
This is what Mohanram is using in step 4(e) of his algorithm!!

Waterfilling Algorithm (2)

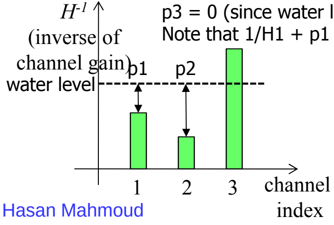
- Consider the following examples:



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

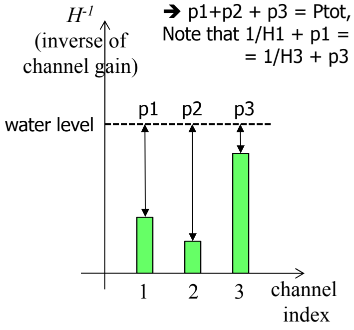


Case 2: Ptot medium
➔ p1+p2 = Ptot, and
p3 = 0 (since water level is not enough)
Note that 1/H1 + p1 = 1/H2 + p2 = wat



Waterfilling Algorithm (3)

- Observation on waterfilling
  - 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.
- 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 waterfilling algorithm.



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

Note: this code implements waterfilling – it does not implement the joint power and subchannel algorithm proposed by Mohanram and Bhashyam.

Waterfilling 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');
```

The idea of the code is as follows:

- Sort the channel gains, Hs
- Since $pi = (C – 1/Hi)^+$ for I =1, 2, …, n and Σpi = 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

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

Algorithm 2: Waterfilling 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 $x \geq 0$.

Ak is Ωk in Algorithm 1

Task 2 – Thursday March 22nd, 2012

- Modify/verify the operation of the waterfilling code provided. Test for several realization of the channels vectors.
- Implement Algorithm 2
- Repeat scenario used in Task 1 with two outputs now – one curve for Algorithm 1 and a a second for Algorithm 2.

References

1. W. Rhee and J. M. Cioffi, "Increase in capacity of multiuser OFDM system using dynamic subchannel allocation," in *Proc. IEEE Int. Vehicular Tech. Conf.*, Tokyo, Japan, May 2000, pp. 1085-1089.
2. Z. Shen*, et al.*, "Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints," *IEEE Transactions On Wireless Communications,* vol. 4, pp. 2726-2737, Nov. 2005.
3. C. Mohanram, S. Bhashyam, “A Sub-optimal Joint Subcarrier and Power Allocation Algorithm for Multiuser OFDM,” IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 8, AUGUST 2005, pp. 685-687.