Transmission Power Minimization for a Multi-Application User in OFDMA Systems

Koudjo Mawuefam Koumadi\textsuperscript{1}, Amevi Acakpovi\textsuperscript{2}, and Robert Adjetey Sowah\textsuperscript{1}

(1) University of Ghana, Computer Engineering Department
(2) Accra Polytechnic, Electrical and Electronics Department

\texttt{kmkoumadi@ug.edu.gh}

\textbf{Abstract}—When a user in Beyond third generation (B3G) mobile communication systems launches multiple applications simultaneously, the user’s radio resources (subcarriers, symbols, power, ...) need to be efficiently allocated to these applications while maintaining their quality of service (QoS) requirements and user utility. In this paper, a power minimization subcarrier, symbol, and power allocation problem is formulated, solved, and algorithms are derived. The problem formulation differs from those in existing works in that all the applications are from the same user. This relaxes some subcarrier allocation constraints that exist in multiuser single-application-per-user cases, leading to different solution techniques. Furthermore, the multiuser diversity which is the basis for many existing algorithms does not exist in the context of this paper.

\textbf{Index Terms}—power minimization, multi-application, single-user

\section{I. INTRODUCTION}

In B3G systems, how user resources (subcarrier, time slots, power, rate, ...) are shared among the various applications simultaneously enjoyed by the user in order to maximize system performance and/or user utility has become a subject of interest.

A great amount of publications were dedicated to allocation of power, rate and other radio resources allocation in order to meet service-specific QoS requirements. \cite{3} is an example of such work, where game-theory was used to allocate power to users with QoS requirements. \cite{1} was based on the packet-level QoS model proposed in \cite{2} to highlight the need of a user-centric utility function.

Subcarrier and power allocation has been another well-investigated topic in wireless communications. Some works focus on system throughput maximization (e.g. \cite{4}), some on utility maximization (e.g. \cite{7}), and some others on transmit power minimization (e.g. \cite{6}, \cite{10}). \cite{6} minimized total system transmit power by iteratively assigning a set of subcarriers to each user and by determining the number of bits and the amount of power for each subcarrier. Arbitrary water-levels are first assumed and updated after bit loading is performed. The process is repeated until the target rate is reached. Many other power minimization algorithms were proposed (e.g. \cite{8}) and most of them suggested a bit-by-bit bit loading using the algorithm proposed in \cite{9}. Though useful and powerful techniques and algorithms were developed, most of these works addressed multiuser resource allocation. Even \cite{1} (which considered single-user with multi-application) just allocated total user-rate to the application without indicating how subcarriers, time slots and/or user-power will be allocated in order to attain the derived optimal rates with the required application-specific bit-error-rates (BER) to the applications. Although many of these existing algorithms can be modified and adapted to single-user multi-application case, some aspects of the problem formulation and the solution are different. Note for instance that the principle of multiuser diversity, on which lays the majority of the multiuser subcarrier allocation algorithms, collapses as far as single-user multi-application is concerned. This is because from the user’s stand point, the channel gain on each subcarrier is identical for all the applications. This is contrary to multiuser environments where different users have different channel gains over the same subcarrier. Another interesting point is that a subcarrier cannot be shared by multiple users in most of the works on multiuser subchannel allocation. This constraint will not be applicable to a single-user’s applications where symbols from different applications can be transmitted over the same subcarrier at different times. Because of the above reasons, we find appropriate to formulate and solve a resource allocation problem with power minimization objective, for a single-user with multiple simultaneous QoS-constrained applications.

Our proposed algorithm may be very useful in solving the multiuser-multiplication cases.

This article mathematically derives an algorithm to jointly allocate subcarriers, symbols, and power to the applications, with transmission power minimization as objective. The performance of the proposed algorithm is compared with existing schemes through simulation. Moreover, instead of iteratively getting the water-level of each application (as traditionally done), we propose a way to get the number of \textit{utilizable} RUs, given a number of allocated RUs. [Note that a resource unit (RU) is assumed to be a symbol duration over a subcarrier]. Once this number is known, the water-level can be calculated using a closed form formula. Thus, it is clear that our algorithm also shares the well-known \textit{water-filling} principle, while short-cutting the exhaustive search of the \textit{water-levels}.

The model under consideration and the problem formulation follow in the next section. The solution and proposed algorithms are presented in section III while simulation and results follow in section IV. Concluding remarks in section V close this article.
II. SYSTEM MODEL AND PROBLEM FORMULATION

The model used for a multi-application user is shown in Figure 1, with the example of a user with three applications. The user-channel is composed of multiple time-varying subcarriers with different gains. It is assumed that the BS already allocated subcarriers to the user. We also assume that the condition of each subcarrier is known over all the duration under consideration, and that the state of each subcarrier remains constant over at least one symbol duration. The user is assumed to have a set $\Omega$ of $N$ RUs to be shared among her/his $I$ applications. The set of these applications is denoted by $\mathcal{I}$. Let the minimum and maximum rates of application $i$ ($a_i$) be denoted by $\alpha_i$ and $\beta_i$, respectively. The subset of RUs allocated to $a_i$ is $\Omega_i \subseteq \Omega$ and $b$ denotes the bandwidth of each subcarrier. Note from Figure 1 that some RUs may be unallocated because they experience deep fading and do not suite for data transmission.

\[
\text{Min } P = \sum_{a_i \in \mathcal{I}} \sum_{u_z \in \Omega} p_{iz} \tag{1}
\]

Subject to:

\[
p_{iz} \geq 0, \forall a_i \in \mathcal{I}, u_z \in \Omega \tag{2}
\]

\[
\alpha_i \leq r_i = \sum_{u_z \in \Omega_i} b \log_2 \left(1 + \frac{p_{iz}g_z}{\Gamma_i}\right) \leq \beta_i, \forall a_i \in \mathcal{I} \tag{3}
\]

\[
\bigcup_{a_i \in \mathcal{I}} \Omega_i \subseteq \Omega \tag{4}
\]

\[
\bigcap_{a_i \in \mathcal{I}} \Omega_i = \emptyset \tag{5}
\]

where $\Gamma_i = -\ln(5BER_i)/1.5$ and $BER_i$ is the maximum tolerable bit-error-rate for $a_i$, ([11]). $p_{iz}$ and $g_z$ are the power over RU $z$ ($u_z$) and the channel gain on $u_z$, respectively. Continuous achievable rate is assumed. Note that Eq. (3), which is from [1], not only accounts for the BER requirement, but also considers minimum and maximum rates of each application. Eq. (5) prevents two applications from sharing the same RU while Eq. (4) states that the union of all allocated RUs must be a subset of $\Omega$. This means that only the RUs allocated to the user can be allocated to his/her applications.

III. PROBLEM SOLUTION

A. Mathematical Analysis

We start the resolution of problem (1)-(5) by noting that in order to minimize power, each application should be transmitted at its minimal possible rate. Therefore, constraints (3)-(5) can be reformulated as

\[
- \sum_{u_z \in \Omega_i} w_{iz} \log_2 \left(1 + \frac{p_{iz}g_z}{\Gamma_i}\right) = -\alpha_i, \forall i \in \mathcal{I} \tag{6}
\]

\[
\sum_{a_i \in \mathcal{I}} w_{iz} \leq 1, \forall u_z \in \Omega \tag{7}
\]

\[
0 \leq w_{iz} \leq 1, \forall a_i \in \mathcal{I}, u_z \in \Omega \tag{8}
\]

where the real number $w_{iz}$ is the allocation variable. $w_{iz} = 1$ if $u_z$ is allocated to $a_i$ or $w_{iz} = 0$ if $u_z$ is not allocated to $a_i$. Note that $b$ is set to unity, for simplicity. One can argue that $0 < w_{iz} < 1$ is not feasible since two applications cannot share the same RU. Indeed, $w_{iz}$ is allowed to take non-integer values for the sake of convexity. This relaxation will be considered in the solution to the problem. It is not hard, in fact, to see that Problem $\{\text{(1)}, \text{(2)}, \text{(6)}, \text{(7)}, \text{(8)}\}$ is convex, in both $p_{iz}$ and $w_{iz}$. Let $\Omega_i^+(\Omega_i^+ \subseteq \Omega_i)$ be the subset of RUs allocated to $a_i$, and which can actually be used for transmission, and let $N_i$ be their number. The following theorem gives the optimality conditions

**Theorem 1. [RU and Optimal Power Allocation]**

- $w_{iz}^* = 1 [(i^*, z^*) = (i, z)]$ if

$$
(i^*, z^*) = \arg \max_{(i,z)} \Theta_{iz} \text{ with } u_z \in \Omega_i^+ \tag{9}
$$

where $\Omega_i^+$ is defined as in Theorem 2.

$$
\Theta_{iz} = \frac{\lambda_i}{\ln 2} \ln \left(\frac{\lambda_i}{\ln 2} \frac{g_z}{\Gamma_i}\right) \tag{10}
$$

and

$$
\lambda_i = \ln \left[\prod_{u_z \in \Omega_i^+(g_z)} \frac{2^\alpha_i}{\lambda_i}\right] \tag{11}
$$

where $N_i$ is found as in Theorem 2.

- $\forall a_i$, $\forall u_z, u_z' \in \Omega_i^+$, $g_z > g_z' \Rightarrow \Theta_{iz} > \Theta_{iz'}$

- The optimum power allocation to RU $z \in \Omega_i$ is $p_{iz}^*$

\[
p_{iz}^* = \left\{ \begin{array}{ll}
\frac{\lambda_i}{\ln 2} - \frac{\Gamma_i}{g_z} & i \in u_z \in \Omega_i^+ \\
0 & i \not\in u_z \notin \Omega_i^+
\end{array} \right. \tag{12}
\]

**Proof:** The proof to this theorem is done by using convex optimization techniques, starting with the Karush-Kuhn-Tucker (KKT) conditions ([5]).

It can be seen that the above theorem has some unknown parameter($\Omega_i^+$ and $N_i$). The following theorem helps find them. Let $Z_i$ be the number of RUs in $\Omega_i$.

**Theorem 2. [Defining $\Omega_i^+$ and Finding $N_i$]**

By assuming that $g_z$'s ($u_z \in \Omega_i$) are ordered in a decreasing order as:

$$
g_1 \geq g_2 \geq \cdots \geq g_{N_i} \geq g_{(N_i+1)} \geq \cdots \geq g_{Z_i} \tag{13}
$$

$\Omega_i^+$ is defined as

$$
\Omega_i^+ = \{ u_z \in \Omega_i / z \leq N_i \} \tag{14}
$$

where $N_i$ is defined by the following two conditions:

$$
\prod_{z=1}^{N_i} (g_z) < (g_{N_i})^{N_i} \tag{15}
$$

$$
\prod_{z=1}^{N_i+1} (g_z) \geq (g_{(N_i+1)})^{(N_i+1)} \tag{16}
$$

2012 IEEE 4th International Conference on Adaptive Science & Technology (ICAST)
4. The single-user RU allocation (SRUA) algorithm

Proof: The proof to this proposition can be found in Appendix A.

The main point of Theorem 2 is the following. (14) must be satisfied for an RU \( u_z \) to be suitable for data transmission. If RUs \( (u_z \not\in \Omega^+_i) \) are ordered as in Eq. (12), one of the ways to find \( N_i \) is to test \( u_z \) against condition (14), starting from \( g_1 \), until one \((g_{N_i+1})\) fails. The proof in Appendix A will show that if \( g_{N_i+1} \) fails \( (u_{N_i+1} \not\in \Omega^+_i) \), then \( g_{N_i+2} \) will also fail. Another way to find \( N_i \) is to start the test against condition (14) from \( g_2 \). RUs will fail the test \((u_z \not\in \Omega^+_i) \) until one \((u_{N_i}) \) passes it. In both cases, \( \Omega^+_i = \{u_z \in \Omega_i | z \leq N_i\} \).

B. Algorithm Design

We first present the routine which generates \( \Omega^+_i \) and \( N_i \), given \( \Omega \) and the target rate \( \alpha_i \). Two ways were suggested in the section above. We estimate that a majority of RUs will be used \((N_i > Z_i/2)\). Thus, we choose the second approach, meaning that we start the test from \( z = Z_i \). This way, the number of RUs to be tested will be reduced. This will fasten the execution of the routine, which we call basic routine.

Algorithm 1 Basic Routine

1. \( \Omega^+_i = \Omega_i \), \( N = Z_i \)
2. \( z = \min \arg \frac{1}{2} \cdot \frac{N_i}{\prod_{u_z \in \Omega^+_i} g_z} \), \( (u_z \not\in \Omega^+_i) \)
3. IF \( \prod_{u_z \in \Omega^+_i} g_z < [g_z]^N \) \( N_i \) \( N \), \( \Omega^+_i \) is the current one, RETURN!
4. ELSE
5. \( \Omega^+_i = \Omega^+_i \setminus \{u_z\} \), \( N = N - 1 \), Go back to step 2
6. ENDIF

With \( \Omega^+_i \) and \( N_i \) got from the basic routine, the water-level is calculated using Eq. (11). Intuitively, this routine makes the calculation of the water-level much simpler, as compared to the traditional iterative searching. A small number (largely inferior to \( Z_i \)) of tests is needed and the test condition is simple. Furthermore, the traditional iterative searching would require a step where an exhaustive bit-by-bit loading (until \( \Omega_i \) is reached) has to be performed in order to determine the number of bits to be loaded on each RU. Our proposed basic routine also short-cuts this step. Additionally, if the calculation of the total amount of power needed by the application is required (for further use), using the traditional methods would require the calculation of the power needed to transmit the loaded number of bits on each RU, before adding them up. With our proposed method, this total power can be directly calculated and can be proved to be

\[
p_i = N_i \frac{\lambda_i}{\ln 2} - \Gamma_i \sum_{u_z \in \Omega^+_i} \frac{1}{g_z}
\]

(16)

Now, let us analyse the RU and power allocation in Algorithm 2. The single-user RU allocation (SRUA) algorithm is run when only RU allocation is needed. This may occur when the algorithm is a sub-routine of a larger algorithm. The single-user resource allocation (SRA) is run when both RU and power allocations are needed.

Algorithm 2 Single-user RU Allocation (SRUA) [or Single-user Resource Allocation (SRA)]

1. DO \( \Omega^- = \Omega, \forall a_i \in I, \Omega^+_i = \emptyset \)
2. WHILE \( \{\Omega^- \neq \emptyset\} \)
3. \( \pi = \max \arg \frac{1}{2} \cdot \frac{u_z}{\prod_{u_z \in \Omega^+_i} g_z}, (u_z \not\in \Omega^-) \)
4. FOR\( \{i = 1 \to I\} \)
5. \( \Omega_i = \Omega^+_i \cup \Omega^- \)
6. Use \( \Omega_i \) to get \( \Omega^+_i \) and \( N_i \) from “textit{basic routine}”
7. \( \Theta_\pi = \left\{ \begin{array}{ll}
\frac{\lambda_i}{\ln 2} \cdot \ln \left( \frac{\lambda_i}{\ln 2} \cdot \frac{\pi}{g_z} \right) & \text{if } \pi \in \Omega^+_i \\
-\infty & \text{if } \pi \not\in \Omega^+_i
\end{array} \right. \)
8. RU allocation \( a_i \) is completed (\( \Omega^+_i \) is the current one) if \( u_{\pi} \not\in \Omega^+_i \)
9. ENDFOR
10. IF \( \{\max_i \Theta_\pi = -\infty\} \)
11. \( \bar{\pi} = \arg \max_i \Theta_\pi \left\{ a_i, u_z \right\} \in (I, \Omega^-) \)
12. \( \Omega^- \leftarrow \Omega^+_i \cup \{u_z\}, \Omega^+_i \leftarrow \Omega^- \setminus \{u_z\} \)
13. ELSE
14. Go to step 17
15. ENDIF
16. ENDWHILE
17. \( \forall a_i \in I, \Omega^+_i \) is the current one, RETURN! [END OF SRUA]
18. \( p_{i_z} = \frac{\lambda_i}{\ln 2} - \frac{\Gamma_i}{g_z}, \forall a_i \in I \) and \( \forall u_z \in \Omega^+_i \)
19. \( p_{i_z} = 0, \forall u_z \in \Omega^-, \forall a_i, \) RETURN! [END OF SRA]

It can be seen that the proposed algorithm takes a greedy allocation approach and greedy algorithms are not always optimal, we shall briefly show that Algorithm 2 will yield at least a near-optimal solution. The followings should be noted:

1) At each iteration (say \( t \)), a greedy choice (say \( [a_i^*(t), u_z^*(t)] \)), which satisfies the optimal condition in Eqs. (9)-(11), is made on the set (say \( \Omega^-(t) \)) of non-allocated RUs. Also, all the applications are considered at each iteration and after each iteration, an optimal solution to the remaining sub-problem can be obtained. Note also that at the initial stage, \( \Omega^-(t = 1) = \Omega \).

2) The union of the optimal \( u_z^*(t) \) and the remaining subset of RUs (say \( \Omega^-(t + 1) \)) form the original set \( \Omega \). Additionally, \( [a_i^*(t), u_z^*(t)] \) combines with (and enters the calculation of) the solution of the remaining sub-problem to form the final solution.

Therefore, Algorithm 2 leads to at least a near-optimal solution.

The following other interesting features of the proposed algorithm can be observed:

- The RU with the greatest channel gain is always allocated first, as a direct consequence of the second point of Theorem 1. This is a great difference with existing multiuser algorithms, where the RU with the greatest gain is different from a user to another. Knowing which RU is under consideration is a big deal since this allow us to calculate \( \Theta_\pi \) only \( I \) times, instead of \( I \cdot Z \) times if that RU were not known. This is even more important
because a user normally would launch around $I = 3$ or $I = 4$ applications while there might be thousands of RUs.

- Another advantage of knowing that the $u_z$ with the greatest $g_z$ is always the only one under consideration is that if an RU $u_z$ to be allocated is not suitable for use by an application, $(u_z \notin \Omega^+ \forall a_i)$, the implication is that RU allocation to that particular application has to be halted because all the remaining RUs will not be suitable for data transmission either. This also will shorten the algorithm’s running time.

- If an RU is allocated to an application $a_i$, this additional RU results in an increase in $N_i$, as well as an increase in $\prod_{u_z \in \Omega^+}(g_z)$. According to Eq. (11), these two increases yield a decrease in the water-level $\frac{\lambda_i}{\ln 2}$ of this application. As a result, the chances of this application to get the next RU are reduced, giving the opportunity to other applications to get the necessary RUs to meet their respective QoSs. This phenomenon will increase fairness among applications.

- Applications with high minimum rates and those with very strict BER requirements will have more chances to get high gain RUs because they will have high water-levels (Eq. 11), yielding high $\Theta_{iz}$ for them.

IV. SIMULATION AND RESULTS

The optimality of the proposed algorithm was demonstrated. Additionally, we provide numerical results to corroborate the mathematical analysis. The following four algorithm are under investigation: 1) an adaptation of Lee’s multiuser suboptimal subcarrier allocation algorithm ([8]), 2) a random RU allocation algorithm, 3) an adaptation of Wong’s optimal multiuser subcarrier allocation algorithm ([6]), and 4) our proposed RU allocation algorithm. The total user transmission power and the average RU utilization ratio (number of RUs used for transmission by an application over the number of RUs allocated to this application).

A. Simulation Environment

The simulation environment consists of a single cell of radius 1km, where a single user is randomly generated. The total bandwidth of the system is 5MHz, divided into 16 subcarriers. 16 symbols are simulated. The channel is frequency selective and the channel gain of each subcarrier varies across the 16 symbols. The duration of simulated time frame is $T = 1s$. The duration of an RU is $\tau$. The user launches three applications following application-related constraints ([10]): $a_1[BER_1 = 10^{-6}, \alpha_1 = 200kbps]$, $a_1[BER_1 = 10^{-3}, \alpha_1 = 64kbps]$ and $a_1[BER_1 = 10^{-3}, \alpha_1 = 10kbps]$. As $\tau/T$ varies, we collect the total power consumption and the RU utilization ratio for each of the four algorithms under consideration.

B. Results and Discussions

Figure 2 and Figure 3 show a single user’s power consumption and RU utilization, respectively, as function of the $\tau/T$ ratio. Based on the results in these graphs, the following observations can be made:

- For all the algorithms, the power consumption increases as $\tau/T$ increase. This is because the resource granularity increases with $\tau/T$. The channel gains become averages of more and more symbols and RUs become bigger blocks. Therefore the RU allocation becomes less efficient. For instance, two consecutive symbols will be allocated together as an RU to a single application. If the RU were a single symbol, the two RUs might be allocated to different applications with an improved efficiency.

- The proposed algorithm yields the same power consumption and the same RU utilization as Wong’s optimal algorithm. This is a confirmation of at worst the near-optimality of the proposed scheme.

- Lee’s sub-optimal algorithm naturally performs worse than the optimal and near-optimal schemes in terms of power minimization. However, it has a better RU utilization ratio than the optimal schemes do.

- Obviously, the random RU allocation performs the worst in terms of power minimization. It also has the lowest RU
utilization ratio. This is because most of the RUs received by an application don’t suit for transmission. These RUs will not receive any power because their channel gain is too low.

Although many existing algorithms are adaptable to solve this problem, we solve it using a different approach, which we consider to be new. While 1) iterative search for a water-level and 2) bit-by-bit bit loading are the main approaches used in existing algorithms, we propose a way to get the number of RUs which can be used (given a set of user-RUs). Once this number is known, the water-level can be calculated using a closed form formula. This should short-cut the exhaustive search for optimal water-levels.

V. CONCLUSIONS

This article discusses resource units allocation to a single multi-application user for transmission power minimization. Although some existing multiuser algorithms can be adapted for single user RU allocation, a more specific algorithm is proposed here and shown to be at least near-optimal. It is shown that the fact that all user-applications face the same user-channel can be exploited to design a faster RU allocation algorithm. This single user algorithm can be used in designing resource allocation algorithms for multiuser multi-application environments, which is the topic of our current research. Moreover, a strict mathematical proof of the optimality (or not) of the algorithm is being worked on.

APPENDIX A

Proof: (Proof of Theorem 2) It can be derived from the proof of Theorem 1 that the set \( \Omega_i^+ \) of RUs which are used for data transmission, is composed of RUs verifying the condition

\[
g_z > \Gamma_i \cdot \ln \frac{2}{\lambda_i}
\]

By substituting \( \Gamma_i \cdot \ln \frac{2}{\lambda_i} \) by its value (deduced from Eq. (11)), we get

\[
g_z > \left[ \prod_{u_i \in \Omega_i^+} (g_z) \right]^{-1} \cdot \frac{2}{\alpha_i}.
\]

Taking power \( N_i \) of both sides yields

\[
(g_z)^{N_i} > \left[ \prod_{u_i \in \Omega_i^+} (g_z) \right]^{-1} \cdot \frac{1}{2^{\alpha_i}}.
\]

The idea of Theorem 2 is that \( g_z \)'s are first put in a decreasing order. Starting from the biggest \( g_z \)'s, RUs are tested against condition (19) until one (say \( g_K \)) fails the test. Therefore, for the ordering \( g_1 \geq ... \geq g_K \geq g_{K+1} \), we just have to prove that if \( u_K \) fails the test, then \( u_{K+1} \) will also fail it. In this particular case, \( N_i = K - 1 \). Mathematically, the proof comes down to showing that

\[
[g_K]^K \leq \frac{\prod_{i=1}^{K} (g_z)}{2^{\alpha_i}} \Rightarrow [g_{K+1}]^{(K+1)} \leq \frac{\prod_{i=1}^{K+1} (g_z)}{2^{\alpha_i}}.
\]

By multiplying both sides of the first inequality in (20) by \([g_{K+1}]^{(K+1)}\), we have

\[
[g_{K+1}]^{(K+1)} \cdot [g_K]^K \leq \prod_{i=1}^{K} \frac{(g_z)}{2^{\alpha_i}} \cdot [g_{K+1}]^{(K+1)}
\]

By dividing both sides by \([g_K]^K\), we have

\[
[g_{K+1}]^{(K+1)} \leq \prod_{i=1}^{K+1} \frac{(g_z)}{2^{\alpha_i}}
\]

because \( g_{K+1} < g_K \) and thus \([g_{K+1}]^{K} < 1\),

\[
\prod_{i=1}^{K+1} \frac{(g_z)}{2^{\alpha_i}} \leq \prod_{i=1}^{K+1} \frac{(g_z)}{g_K}.
\]

This finally leads to

\[
[g_{K+1}]^{(K+1)} \leq \prod_{i=1}^{K+1} \frac{(g_z)}{2^{\alpha_i}}.
\]

\[\blacksquare\]

REFERENCES


