# Scheduling Algorithms for Point-to-Multipoint Operation in IEEE 802.16 Networks

R. Iyengar, K. Kar, B. Sikdar
Rensselaer Polytechnic Institute, Troy, NY 12180
Email: iyengr@rpi.edu

***Abstract*— We study the resource allocation problem in OFDMA based 802.16 broadband wireless access systems. Frequency and time resources must be allocated by a central controller (Base Station) to a number of users. We consider variations of a resource allocation problem, some of which are difficult to solve. Situations in which only the objective of the Base Station need to be maximized are easily dealt with as are cases where all the users perceive the same channel conditions. Scenarios where both the objectives of the BS as well as those of the end users must be met simultaneously require more complicated solutions since individual users experience different channel conditions. We present linear programming relaxations for the resource allocation problem. While solving the LP using standard techniques like ellipsoidal algorithm can provide optimal allocations for all users, it can be expensive in terms of computing overhead as the number of users in the system increase. Therefore we present an efficient algorithm which performs well even as the number of clients $n$ in the system increases. We also present a heuristic based on the interpretation of the linear programming relaxation as a concurrent flow problem. We note that in numerical experiments, the performance of the heuristic closely matches the optimal solution to the linear programming relaxation.**

## I. Introduction

The IEEE 802.16 is an emerging suite of standards for point to multipoint (PMP) broadband wireless access (BWA). The 802.16e amendment to the 802.16-2004 specification enables support for combined fixed and mobile operation for licensed and license-exempt frequencies below 11 GHz. IEEE 802.16 is likely to emerge as a preeminent technology for cost-competitive ubiquitous broadband wireless access supporting fixed, nomadic, portable and fully mobile operations offering integrated voice, video and data services. The technology is likely to be considered in a variety of deployment scenarios, such as standalone IP core based networks and as a data overlay over existing broadband and cellular networks. Initial deployments are likely to be based on fixed/nomadic operation with fully mobile usage to follow. Of the three different PHYs specified in the standard, OFDM multi-access (OFDMA) is likely to emerge as the most preferred PHY supporting all usage models.

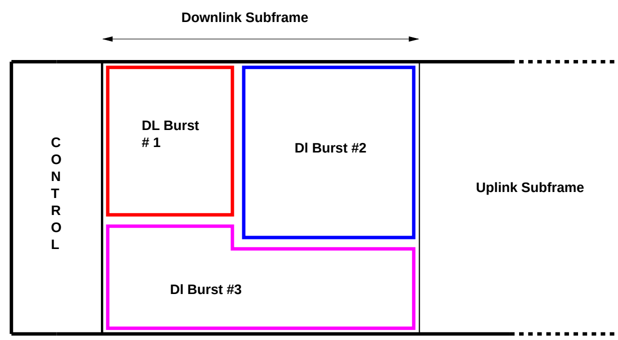


Fig. 1. Typical Frame Structure in Wireless Scenarios

The frame structure typically used in IEEE 802.16 based wireless systems is shown in Figure 1. The initial portion of the frame (control) consists of the Downlink Map (DL-Map) and the Uplink Map (UL-Map). These specify information about the allocations made for each client on Uplink/Downlink. Specifically, these maps contain information about which subcarriers and which time slots are allocated to a given user, in a given frame. The Downlink portion of the frame is followed by the Uplink portion. The horizontal axis denotes time and the vertical denotes subcarriers used in OFDMA (hence this axis denotes frequency). Figure 1 shows a Time Division Duplex (TDD) frame, for an OFDMA PHY, with allocations made for 3 users on the Downlink Subframe. For the remainder of the paper we assume that the system operates over an OFDMA PHY layer. The scheduling problem is to allocate time slots on a subset of the subcarriers available (frequency resource) to meet client demands and maximize system throughput. The time interval $T$ over which these demands must be satisfied can be equal to the frame duration $T_F$ or some other value.

The 802.16 draft specifies that certain channels on the Uplink are designated as *channel quality indication* channels (CQICH). Clients feedback average CINR measures that they perceive on the Downlink using this channel. The Base Station specifies a CQICH allocation for a particular client, in the control portion of the frame, which instructs the client to feedback the average CINR measure using the fast feedback channel to the Base Station. The measurement of channel quality on the downlink is in itself an interesting issue when there are a large number of clients, since channel quality feedback overhead becomes an important factor in the scheduling efficiency. We assume for the purposes of this work that channel quality information is collected for all clients at the same time, and at regular intervals, using either the CQICH or some similar mechanism. Uplink channel conditions can be estimated at the base station every time data is sent out from a client.

### A. Contributions

There is very little literature on scheduling algorithms for wireless networks built around the IEEE 802.16 [2] standard. The draft [2] specifies a number of hooks and features that can be used but does not specify the exact scheduling algorithm, to allow vendors to differentiate their products. Note that in a centralized wireless network, the scheduler is one of the most important components of the system. To the best of the author's knowledge, no published work using the techniques and algorithms described in this paper exist. While a number of papers on the problem of *bit loading* for OFDMA systems exists, these works do not explore in detail the combinatorial nature of the problem (for example refer [5]). In contrast, in this paper, we provide a proof of hardness for the discrete version of the resource allocation problem (described later) and provide a *provably good* algorithm based on solutions to mixed covering and packing LPs. We also present a heuristic based on generalized concurrent flow which performs well in numerical experiments.

### B. Paper Organization

In Section II we describe the system model which is used in the LP formulations. The LP formulations are presented in Section III. Section III-B discusses the solution to a simple case of the problem, when all channel conditions are identical. In Section IV we prove that the discrete version of the resource allocation problem is NP-Hard. Sections VI and VII discuss two different approaches to solve the continuous time relaxation of the discrete version of the problem. Section VII presents a heuristic approach, which is analyzed numerically in Section VIII. Section IX describes heuristics for the online version of the problem. Section X concludes the paper ans presents some interesting directions for future work which this work has thrown open.

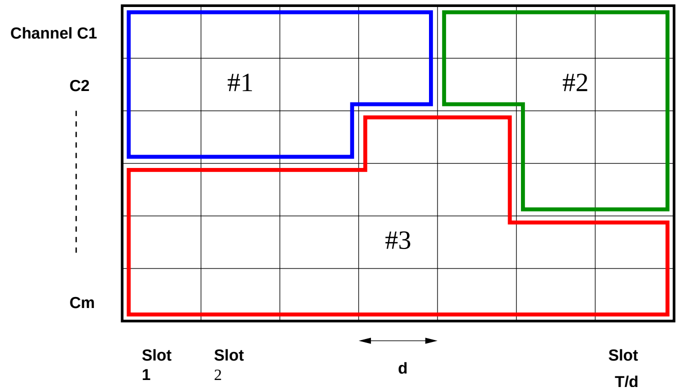


Fig. 2. Representation of Time and Frequency Resources

## II. System Model

The time and frequency resources that must be shared between clients are represented in Figure 2. This serves as an abstraction of the OFDMA PHY used. Unlike in an OFDM system, the OFDMA system provides the added flexibility of allocating a subset of available carriers to a user for some time duration. In Figure 2, the time axis (horizontal) is discretized into slots of length $\Delta$. The vertical axis represents the different subcarriers used in the system.

The channel conditions perceived by each station are captured in a channel conditions matrix of dimension $n$x$m$ ($m$ subcarriers and $n$ users). The entries in this matrix are a measure of the rate achievable by user $i$ on subcarrier $j$. For example, these entries may have units of bits/sec which is intuitively useful since the allocations for each user are time durations on channels.

### A. Discrete and Continuous versions of the Problem

We consider two versions of the resource allocation problem, one where the time axis is continuous and another where the time axis is discretized as shown in Figure 2. In the discrete case, the resource allocation problem reduces to the allocation of each of these *cells* to individual stations to achieve certain objectives. In the continuous relaxation, the resource allocation problem is to allocate time chunks to users across the available subcarriers to satisfy demand and maximize throughput.

## III. Problem Formulation

In this section we present LP formulations for the resource allocation problem.

In formulation $1 \cdots 4$ users perceive different conditions on each subcarrier and these values differ across users. Clients modulate multiple subcarriers concurrently. The set of subcarriers allocated to a user is typically a subset of the total number of subcarriers available in the system. Let $\alpha_{ij}$ represent the rate achievable by user $i$ on channel $j$ in bits/sec.

$$Max. \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}x_{ij} \quad (1)$$

$$\sum_{i=1}^{n}x_{ij} \leq T \;\; \forall\, j = 1\cdots m \quad (2)$$

$$\sum_{j=1}^{m}\alpha_{ij}x_{ij} \geq d_i \;\; \forall\, i = 1\cdots n \quad (3)$$

$$x_{ij} \geq 0 \quad (4)$$

In the above formulation, the variables $x_{ij}$ represent the time duration allotted to station $i$ on channel $j$ to transmit data. The exact position of this time chunk is communicated to the user by the base station using control messages that are broadcast to all users, at the start of each frame (these are referred to as Downlink Map and Uplink map in the standard [2] respectively). The objective function seeks to maximize the overall amount of data in bits transmitted. Without any demand constraints, this problem can be solved simply as discussed a little later. The first constraint specifies that the total time allocated across all stations on a channel cannot exceed the duration T. The second constraint is the QoS constraint and specifies that the total data transmitted by a station $i$ in time T must at least equal the demand $d_i$ in bits, in case of Uplink traffic. In the case of downlink traffic, this is the minimum amount of data that must be received by station $i$. Note that while time duration $T$ is a time horizon over which QoS guarantees must be provided. Note that in these situations there is an inherent assumption that channel conditions do not vary significantly over the *update* interval, when channel condition updates are sent from the client to the Base Station, in the case of downlink traffic. In the case of uplink traffic, uplink channel conditions can be measured at the Base Station roughly every $T$ seconds.

The LP (1) is the relaxation of the Integer Program shown below.

$$Max. \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}n_{ij}\Delta \quad (5)$$

$$\sum_{i=1}^{n}n_{ij} \leq \frac{T}{\Delta} \;\; \forall\, j = 1\cdots m \quad (6)$$

$$\sum_{j=1}^{m}\alpha_{ij}n_{ij} \geq \frac{d_i}{\Delta} \;\; \forall\, i = 1\cdots n \quad (7)$$

$$n_{ij} \geq 0 \quad (8)$$

$$n_{ij} \in Z^{+} \quad (9)$$

Here, $n_{ij}$ are the number of slots allocated to station $i$ on channel $j$. It is assumed that the slot length $\Delta$ exactly divides the subframe (Uplink or Downlink) time $T$ (as shown in 2).

### *A. Extensions to Other Scenarios*

The formulation ($1 \cdots 4$) can be easily extended to solve another class of problems. In a number of wireless scenarios it needs to be assumed that a user has only a single wireless interface and hence can talk only on only one frequency (or tone) at a given time. This is typically an assumption in most 802.11 scenarios. However in OFDMA scenarios, users have radios that modulate multiple subcarriers at the same time. The formulations in this paper encompass both these cases depending on the presence or absence of certain constraints in the optimization formulation. Assume that the channel consists of multiple orthogonal frequencies or tones. The Base Station is assumed to be a sophisticated device capable of talking on all tones simultaneously (for example due to the presence of multiple radio interfaces), but clients are simpler devices which can communicate on only a single tone at one time, but are capable of *hopping* between tones dynamically. This is done by adding another set of linear constraints. The complete problem formulation is shown below.

$$Max. \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}x_{ij} \quad (10)$$

$$\sum_{i=1}^{n}x_{ij} \leq T \;\; \forall\, j = 1\cdots m \quad (11)$$

$$\sum_{j=1}^{m}x_{ij} \leq T \;\; \forall\, i = 1\cdots n \quad (12)$$

$$\sum_{j=1}^{m}\alpha_{ij}x_{ij} \geq d_i \;\; \forall\, i = 1\cdots n \quad (13)$$

$$x_{ij} \geq 0 \quad (14)$$

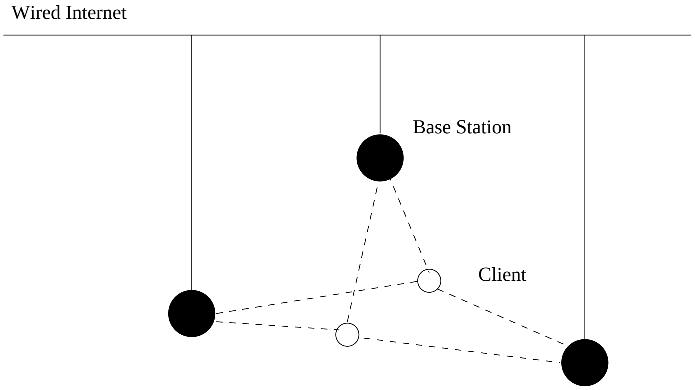


Fig. 3. Multi-Homed Clients

The added set of constraints imply that the total allocated time to any given station across all channels cannot exceed $T$. The ideas used in the formulations presented thus far can be combined together to form an LP for another interesting scenario. In a network scenario, with a number of base stations each client is typically in the communication range of some number of base stations, say $k$. Hence there are $k$ channel condition matrices associated with each client, one for each base station. In general some fraction of demand for a given client can be satisfied by one base station, and the remaining, by the other. Note that it is assumed that adjacent base stations use the same frequency band. In the formulation below, the additional index $k$ denotes a base station. Scenarios such as the one just described are represented in Figure 3. Note that in general, some clients may *see* less than $k$ base stations, causing some $\alpha_{ijk}$ to equal zero.

$$\text{Max} \sum_k \sum_i \sum_j \alpha_{ijk} x_{ijk} \quad (15)$$

$$\sum_k \sum_j \alpha_{ijk} x_{ijk} \geq d_i \;\; \forall i \quad (16)$$

$$\sum_i x_{ijk} \leq T \, \forall j \quad \text{for each } k \quad (17)$$

$$\sum_k x_{ijk} \leq T \quad \text{for each } i \quad \text{for each } j \quad (18)$$

$$x_{ijk} \geq 0 \quad (19)$$

### B. Identical channel conditions

We first address the simple case where all users perceive identical channel conditions. Consider the discrete version of the resource allocation problem where cells need to be assigned to users so that demand is satisfied and throughput is maximized. In the following algorithm, let the total number of available slots be $S$. Note that since all users see identical channel conditions, we are not constrained to look at the problem as slots in two dimensions, but can reduce it to a single dimensional problem. In algorithm 1, the $satisfied\_demand[]$ array stores the current amount of demand satisfied for user $i$, and is initialized to zero. The variables $s_j$ keep track of which user is allocated a particular slot $s_j$. The variable $total\_demand$ keeps track of the total demand satisfied across all users. We note that the performance of this algorithm (and indeed any slot allocation algorithm) depends on the granularity $\Delta$ (slot length).

Note that when the demand of all users is satisfied, the algorithm terminates, with some slots left unallocated. These can be allocated arbitrarily. It is easy to see that in the case where the demands of the stations are not satisfiable, the algorithm returns an allocation which is max-min fair.

**Algorithm 1** Identical Channel Conditions

```
j ← 1
i ← 1
satisfied_demand[] ← 0
total_demand ← 0
while j ≤ S do
  if satisfied_demand[i] ≥ demand[i] then
    i ← i + 1
    if i > n then
      i = 1
    end if
    continue
  end if
  allocate slot j to station i
  s_j ← i
  increment = slotlength * α_ij
  satisfied_demand[i]+ = increment
  total_demand+ = increment
  if total_demand ≥ Σ_i d_i then
    break
  end if
  i ← i + 1
  j ← j + 1
  if i > n then
    i = 1
  end if
end while
```

## IV. Hardness Result

In this section we prove that the demand constrained discrete version of the problem is NP-Hard. In the discrete version of the problem, slots on each channel must be assigned to clients (a slot can be assigned only to a single client) so that demands are satisfied and throughput is maximized.

The proof is by reduction from MAXIMUM CONSTRAINED PARTITION (henceforth referred to only as PARTITION), which is a well known NP-Complete problem, for example refer [11], or [12]. For completeness, we state the problem here. An instance of PARTITION is a finite set $A$ and a size $s(a) \in Z^+$ for each $a \in A$. A solution to an instance of PARTITION, is a partition of A, a subset $A' \subseteq A$, so that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a) \tag{20}$$

(The optimization version of this problem seeks to maximize the number of elements from $S$ on the same side as a given element $a_0$.) Now consider the following version of the discrete scheduling problem. There are some number of subcarriers, $m$, and each subcarrier has only one time slot associated with it. There are only two clients both of which see exactly the same channel conditions on the given set of channels (assuming that the channel conditions seen by the two clients can be represented as integers). Hence we have a set A consisting of $m$ elements, each having some value $a_i$, $i = 1...m$. Let each user have a demand $d = \sum_i \frac{a_i}{2}$. Since each element of the set can be assigned to only one client and not more, we see that we can solve this problem iff we can solve the PARTITION problem. Therefore, even this simplified version of the discrete scheduling problem is NP-Complete. Hence we can say that the general discrete scheduling problem described earlier (for throughput maximization) is NP-Hard.

## V. Mixed Covering and packing Linear Programs

We note that the LP ($1 \cdots 4$) can be solved optimally using the simplex algorithm. While it is known that simplex is exponential in complexity for certain classes of problems, there are few results on the performance of simplex for other classes of problems. Other approaches like the ellipsoidal algorithm or interior point approaches are polynomial in the size of the LP, but they do not in practice perform as well as simplex. Hence, we are interested in *provably efficient* approaches to solve Linear Programs of the type 1.

There is significant literature on the topic of *approximately* solving LPs for feasibility and/or optimality using simple and fast algorithms. We note that LP (1) can be modeled as a flow problem on a graph. However, this is not a vanilla flow problem. The demand constraints impose lower bound constraints on some edges. Further, the flow on some edges is associated with *multipliers* (channel conditions). Hence, we have a generalized flow problem at hand.

There is some literature on approximating generalized flow problems, for example [7] presents fast and simple fully polynomial approximations (FPTAS) for generalized versions of maximum flow, multicommodity flow and minimum cost maximum flow. This paper extends the work of [3] to a more general setting. Also of interest, given the formulation of the problem here are covering and packing linear problems. A packing LP is of the form

$$\max\ C^T x \text{ Subj. to: } \ Ax \leq b; \ \ x \geq 0 \tag{21}$$

A covering LP is of the form

$$\min\ C^T x \text{ Subj. to: } \ Ax \geq b; \ \ x \geq 0 \tag{22}$$

Note that all matrix entries in the above formulations are positive, real numbers. A *mixed covering and packing problem* (MCPP) contains both types of constraints from the formulations above.

$$\min\ C^T x \tag{23}$$
$$Ax \geq b \tag{24}$$
$$Px \leq u \tag{25}$$
$$x \geq 0 \tag{26}$$

MCPPs can be solved exactly using standard LP solvers. However, these solvers are typically slow for larger instances of the problem. There is significant literature using Frank-Wolfe methods to solve MCPPs in the *feasibility* sense. That is, all these approaches find solutions that are only approximately feasible. That is the schemes find a solution that satisfies either $Ax \geq b$ and $Px \leq (1+\epsilon)u$ or satisfies $Ax \geq b/(1+\epsilon)$ and $Px \leq u$ (refer [13]). Since, $x$ does not satisfy all the constraints exactly, the solution can be arbitrarily far off from the optimal. We note that there is no known work to the knowledge of the author that approximates the optimal value of the MCPP with a strictly feasible solution. However, some simplified versions of this problem have been solved, for example, refer [13] or [14]. These solve a restricted problem which is formulated as:

$$\min\ C^T x \text{ Subj. to: } \ Ax \geq b; \ \ x \leq u; \ \ x \geq 0 \tag{27}$$

The problems described above, especially the MCPP is of interest, since the type of constraints are similar to those in LP1 (1). We note that an efficient algorithm that can solve the MCPP can be used to solve all the LP formulations presented earlier in this paper.

## VI. An Input Dependent Approximation Algorithm for LP (1)

In this section we present an input dependent approximation algorithm for LP ($1 \cdots 4$) based on results for approximating mixed covering and packing linear programs. In [9] the authors describe efficient sequential algorithms to solve the feasibility problem approximately. Specifically, the algorithm returns a solution satisfying all constraints within a $1 \pm \epsilon$ factor in $O(Mp \log M/\epsilon^2)$ time where $M$ is the number of constraints and $p$ is the maximum number of constraints any variable occurs in.

It is possible to use the efficient feasibility algorithms as a subroutine to calculate the optimal solution by using a bisection search on the range of the optimal solution. Assuming we know the maximum data rate achievable across all channels in the system (denoted as $W$), we can compute an approximately optimal solution in $O(k \log mWT)$ time where $k$ is the time complexity for a single call to the approximate feasibility subroutine.

In the case of the LP (1), we note that $M = m + n + mn$ (supply, demand and non-negativity constraints respectively), and $p$ can have a value of at most 5 for a given iteration. This can be seen as follows: the demand constraint for a given client $i = 1 \cdots n$ contains one occurrence of $x_{ij}$ for some $j = 1 \cdots m$. Similarly, the supply constraint for channel $j$ contains one occurrence of $x_{ij}$. When performing a bisection search, two constraints are added for a given range of the objective function, each containing one occurrence of the variable $x_{ij}$ (the fifth occurrence is due to the non-negativity constraint). In reality, since the number of orthogonal carriers (or subcarriers in OFDMA) for a given system is typically fixed, as $n$ grows large, the complexity looks roughly like $O(n \log(n)/\epsilon^2)$.

Note that the upper bound on the system throughput can be improved in the following way: If the subcarriers are numbered $1 \cdots m$, denote $s_i$ as the station with the best data rate on subcarrier $i$. Let $W_{s_i}$ be the maximum rate achievable across all users on subcarrier $i$. Therefore the running time of the algorithm can be improved to $O(k \log(\sum_{i=1}^{m} W_{s_i} T)$.

## VII. A Heuristic Approach Based on Maximum Concurrent Flow

In this section we present a heuristic for the LP ($1 \cdots 4$) which makes use of the maximum concurrent flow interpretation of ($1 \cdots 4$). The advantage of this heuristic is that it's time complexity does not depend on the value of the maximum data rate achievable on a given channel. An alternate formulation of the relaxed resource allocation problem ($1 \cdots 4$) can be to maximize a common multiple of satisfied demand across all users, that is, some $\lambda$ so that at least $\lambda d_i$ is satisfied for all $i$. However, this is not a traditional concurrent flow problem. There are multipliers associated some of the variables in the formulation. This can be posed as a *generalized flow* problem. Efficient techniques to approximate generalized concurrent flow are presented in [15]. The path formulation of the generalized concurrent flow (refer [15]) problem is:

$$\max \lambda \tag{28}$$

$$\sum_{P:e \in P} \gamma_P(e) x(P) \leq u(e) \;\; \forall e \tag{29}$$

$$\sum_{P \in P_j} x(P) - \lambda d_j \geq 0 \;\; \forall j \tag{30}$$

$$x(P) \geq 0 \;\; \forall P \tag{31}$$

$$\lambda \geq 0 \tag{32}$$

Given an s-t (source-destination) path $P = e_1, \cdots, e_r$, $\gamma_P(e_q) = 1/\Pi_{i=q}^{r} \gamma(e_i)$. This is the amount of flow sent into arc $e_q$ to deliver one unit of flow at $t$ using path P. For the formulation ($1 \cdots 4$) under consideration, we can construct a graph with $mn$ paths with $m$ paths from one source of data through $m$ channels connected to a sink. The variables $y_{ij}$ can be interpreted as bits. Note that only the edges corresponding to the $m$ channels have a capacity of $T$ associated with them, the rest of the edges are uncapacitated. The resulting formulation is shown in ($33 \cdots 37$).

$$\max \lambda \tag{33}$$

$$\sum_{i=1}^{n} \frac{y_{ij}}{\alpha_{ij}} \leq T \;\; \forall\, j = 1..m \tag{34}$$

$$\sum_{j=1}^{j=m} y_{ij} - \lambda d_j \geq 0 \;\; \forall\, i = 1..n \tag{35}$$

$$y_{ij} \geq 0 \;\; \forall\, i, j \tag{36}$$

$$\lambda \geq 0 \tag{37}$$

Once the concurrent flow problem is solved, the optimal $\lambda$ will either be larger or smaller than 1. In case of the latter, we are dealing with an infeasible program, and the resulting allocations are in fact a good solution to the resource allocation problem. In case of the latter, the program is feasible but the resulting allocation is in general not throughput optimal. In this case, scale the solution back by the objective function value in the generalized concurrent flow program. Next, allocate the remainder of the frame in a throughput optimal manner, by allocating the remaining time on each subcarrier to the client with the best data rate on that subcarrier. From

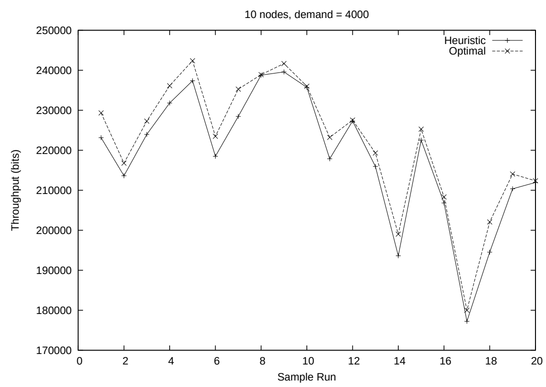


Fig. 4. Performance of Heuristic, 10 nodes, demand = 4000

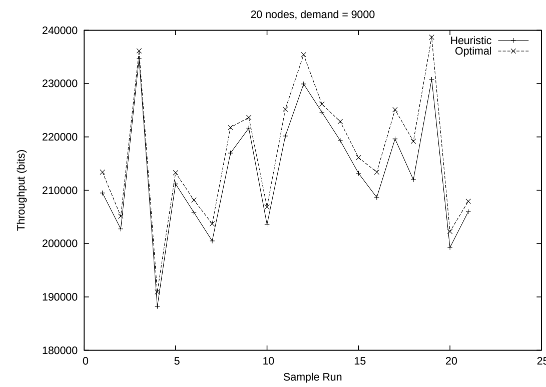


Fig. 5. Performance of Heuristic, 20 nodes, demand = 9000

the results in [4], the time complexity of the concurrent flow heuristic is $O(\epsilon^{-2}(k1 + k_2 \log k_2)(k_1 + n))$ where $k_1 = 2mn + m$ is the number of edges, and $k_2 = 2(m + n)$ is the number of nodes in the graph on which the concurrent flow is computed. We note that the advantage of this formulation is that the algorithm is not dependent on the values in the channel conditions matrix. The solution provided by the heuristic is such that some portion of the frame is used for the concurrent flow based allocation, that is, when the optimal solution is scaled back, the total time allocated on each subcarrier is the same. This is because the optimal solution to the concurrent flow problem will find allocations $x_{ij}$ so that $\sum_i x_{ij} = T \ \forall j$. This is true since any leftover space would imply a larger value of $\lambda$.

## VIII. Numerical Results

In this section we present simulation results for the heuristic outlined in Section VII. We assume an 802.16 system with frame time 5msec. All clients have the same demand in these experiments. The channel conditions for each client are chosen randomly between 1 and 10Mbps. The system operates over 5 subcarriers. $\epsilon$ is chosen to be small, $\epsilon = 0.01$, so that the concurrent flow problem is almost optimally solved. The optimal solution is computed using CPLEX to solve the Linear Program. We note that the heuristic performs very well for the problem instances considered, and closely approximates the optimal solution, from Figures 4 and 5.

## IX. Heuristics for the Online version of the Scheduling Problem

We consider the following instances of the problem when an online version is more applicable. It is assumed that changes do not occur in batches, they happen one at a time. The objective remains the same, that is we seek to maximize throughput.

1) The number of clients which require bandwidth resources at the Base Station decreases by one.
2) The number of clients requiring bandwidth resources at the Base Station increases by one. It is assumed that required demand and associated channel conditions for the stations are provided as input.
3) The channel conditions associated with one of the clients currently being served changes. We identify two subcases:

   a) The change in channel conditions leads to the current solution being feasible but not optimal.
   b) The change in channel conditions leads to the current solution being infeasible (change in channel conditions leads to demand for corresponding client not being satisfied).

### A. Case 1: Lesser Demand

If the number of clients is decreased by one, then the associated constraints and variables are no longer applicable to this problem. The allocation made for the removed client on a given channel are allocated to the client from the remaining set of clients with the best channel conditions for the channel being considered

### B. Case 2: Increased Demand

Since the number of clients increases by one, allocate time in a greedy manner for the new client. That is, find the channel on which the new client has the best response, and decrease all allocations to other clients on that channel without violating their demand feasibility constraints. Allocate this free portion to the new client. If the demand for the new client is not satisfied after iterating through all channels, then reject the new client.

### C. Case 3a: Feasible Channel Change

Let the client in question be $r$. This is a simpler case since the current solution is already feasible. Find the channel on which some client has the best channel conditions across all channels. This corresponds to finding the largest $\alpha_{ij}$, across all $i$, $j$. If this $\alpha_{ij}$ is such that $r \neq i$, then decrease the allocation of $r$ on that particular channel till it's demand is not violated and allocate the free portion to $i$. This process iterates over the next best $\alpha_{ij}$ and so on.

### D. Case 3b: Infeasible Channel Change

Again, let the client in question be $r$. Due to changed channel conditions, let the demand for $r$ that is satisfied by the current allocation be $d_{new}$. Let $\delta d = d_r - d_{new}$, be the remaining demand for $r$ that needs to be satisfied yet. The problem reduces to adding a new client with demand $\delta d$ with the new channel conditions.

## X. Conclusions and Future Work

In this paper we have presented Linear Programming based formulations for the demand constrained maximum throughput problem applied to IEEE 802.16 based wireless networks. We prove that the discrete version of the problem is NP-Hard in general. We present an algorithm to find the maximum throughput, based on ideas from Mixed Covering and Packing LPs. Due to the dependence of the runtime of the algorithm on the best achievable data rate on any subchannel in the system, we also present a heuristic based on an interpretation as a generalized concurrent flow problem. The heuristic closely tracks the optimal value in numerical experiments. There are some interesting questions yet to be answered. The question of how well the solution to the LP approximates the discrete version of the problem which is NP-Hard is open. Also of interest are algorithms and heuristics for the online version of the throughput maximization problem. While initial directions have been presented in this paper, their detailed analysis and study needs work. We are currently addressing these questions.

## References


[1] The IEEE 802.16 Working Group on Broadband Wireless Access Standards, *http://grouper.ieee.org/groups/802/16/*

[2] Draft IEEE standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems

[3] N. Garg, J. Konemann *Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Algorithms*, Proc. IEEE Symposium on Foundations of Computer Science (FOCS), 1998

[4] L. Fleischer, K. Wayne *Fast and Simple Approximation Schemes for Generalized Flow*, in Math. Programming 91 (2002), no. 2, pp. 215-238.

[5] G. Kulkarni, S. Adlakha, M. Srivastava emphSubcarrier Allocation and Bit Loading Algorithms for OFDMA based Wireless Networks, IEEE Transactions on Mobile Computing, Vol. 04, no. 6, pp. 652-662, Nov. 2005.

[6] Christos Papdimitriou, Kenneth Steiglitz, *Combinatorial optimization: Algorithms and Complexity*

[7] Lisa Fleischer, Kevin Wayne, *Fast and Simple Approximation Schemes for Generalized Flow*, Math. Program. Ser. A (2001)

[8] Murty Katta G. *Linear Programming*.

[9] N. Young *Sequential and Parallel Algorithms for Mixed Covering and Packing*, Proc. FOCS 2001, pp: 538.

[10] A.K. Ahuja, T. Magnanti, J. Orlin *Network Flows- Theory, Algorithms and Applications*, prentice Hall, 1993.

[11] M. Garey, D. Johnson *Computers and Intractability* W.H. Freeman, 1979.

[12] P. Crescenzi, V. Kann (Editors) *A Compendium of NP Optimization Problems*, available online at *http://www.nada.kth.se/ viggo/problemlist/compendium.html*.

[13] L. Fleischer *Fast Approximation Algorithms for Fractional Covering Problems with Box Constraints*, proc. 36th ACM SIAM Symposium on Discrete Algorithms (SODA), January 2004.

[14] N. Garg, R. Khandekar *Fractional Covering with Upper Bounds on the Variables: Solving LPs with Negativve Entries*, in proc. 12th European Symposium on Algorithms (ESA), LNCS 3321, pp: 371-382, 2004.

[15] L. Fleischer, K. Wayne, *Faster Approximation Algorithms for Generalized Network Flow*, proc. ACM/SIAM Symposium on Discrete Algorithms, 1999.