RAPPORT DE RECHERCHE

LRI

SCHEDULING IN WIRELESS OFDMA-TDMA NETWORKS USING VARIABLE NEIGHBORHOOD SEARCH METAHEURISTIC

ADASME P / LISSER A / WANG C / SOTO I

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

06/2013

Rapport de Recherche N° 1561

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 650
91405 ORSAY Cedex (France)

Scheduling in Wireless OFDMA-TDMA Networks using Variable Neighborhood Search Metaheuristic

Pablo Adasme · Abdel Lisser · Chen Wang · Ismael Soto

Abstract In this paper, we present a hybrid resource allocation model for OFDMA-TDMA wireless networks and an algorithmic framework using a Variable Neighborhood Search metaheuristic approach for solving the problem. The model is aimed at maximizing the total bandwidth channel capacity of an uplink OFDMA-TDMA network subject to user power and subcarrier assignment constraints while simultaneously scheduling users in time. As such, the model is best suited for non-real time applications where subchannel multiuser diversity can be further exploited simultaneously in frequency and in time domains. The VNS approach is constructed upon a key aspect of the proposed model, namely its decomposition structure. Our numerical results show tight bounds for the proposed algorithm, e.g. less than 2% in most of the instances. Finally, the bounds are obtained at a very low computational cost.

1 Introduction

Orthogonal frequency and time division multiple access (resp. OFDMA, TDMA) are two wireless multi-carrier transmission schemes currently embedded into modern technologies such as Wifi and Wimax [6]. In an OFDMA network, multiple access is achieved by assigning different subsets of subcarriers (subchannels) to different users while maintaining orthogonal frequencies among subcarriers. In theory, this means that interference among subcarriers is completely minimized which allows simultaneous data rate transmissions from/to several users to/from the base station (BS). The transmission direction from the BS to users is known as a downlink process while the opposite is known as an uplink process. The TDMA transmission scheme, on the other hand, has the property of scheduling users in time by assigning all bandwidth channel capacity to only one user within a given time slot in order to transmit signals. Although, these

Pablo Adasme · Ismael Soto
Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile, Avenida Ecuador 3519 Santiago, Chile
E-mail: {pablo.adasme,ismael.soto}@usach.cl

Chen Wang · Abdel Lisser
Laboratoire de Recherche en Informatique, Université Paris-Sud XI, Bâtiment 650, 91405, Orsay Cedex France
E-mail: {chen.wang,abdel.lisser}@lri.fr

transmission schemes work differently, the underlying purpose in both of them is nearly the same, i.e. to make an efficient use of resource allocation of power and bandwidth channel capacity of the network.

In this paper, we propose a hybrid resource allocation model for OFDMA-TDMA wireless networks and an algorithmic framework using a variable neighborhood search metaheuristic approach (VNS for short) for solving the problem [3]. More precisely, we aim at maximizing the total bandwidth channel capacity of an uplink OFDMA-TDMA network subject to user power and subcarrier assignment constraints while simultaneously scheduling users in time. As such, the model is best suited for non-real time applications where signals can be transmitted at different time slots without further restrictions [4]. The latter allows the fact that subchannel multiuser diversity can be further exploited simultaneously in frequency and in time domains. As far as we know, joint OFDMA-TDMA transmission schemes have not been investigated so far. In [8], the authors compare the performance in support of real time multimedia transmission schemes when using separately OFDMA-TDMA and OFDMA networks. Their numerical results show that OFDMA outperforms OFDMA-TDMA in several quality of service metrics for real-time applications. In a similar vein, the authors in [9] consider resource allocation of an OFDM wireless network while mixing real-time and non-realtime traffic patterns. They use a utility based framework to balance efficiency and fairness among users. Thus, they propose a scheduler mechanism which gives in one shot the subcarrier and power allocation plus the transmission scheduling for each time slot. Their numerical results indicate that the proposed method achieves a significant performance in terms of the overall throughput of the system. Another related work is proposed in [10] where an hybrid transmission scheme for non-realtime applications while using simultaneously code division and time division multiple access (CDMA-TDMA) schemes is investigated. The authors use a utility based approach as well, and formulate the optimal downlink resource allocation problem for a non-realtime CDMA-TDMA network. Their numerical results show a significant improvement in the overall throughput of the system due to multi-access-point diversity gain.

We propose a simple VNS based metaheuristic approach [3] to compute tight bounds for our hybrid OFDMA-TDMA optimization problem. To this purpose, we randomly partition the set of users into T disjoint subsets of users within each iteration of the VNS approach. By doing so, we must solve T smaller integer linear programming (ILP) subproblems, one for each subset of users assigned to time slot $t \in \mathcal{T} = \{1, ..., T\}$. Note that, in principle, each subproblem could be solved sequentially or in parallel using any algorithmic procedure. As in our case each subproblem is formulated as an ILP problem, so far now, we solve its linear programming (LP) relaxation to compute the bounds. In fact, this is a key aspect in our proposed VNS approach since the LP relaxations of the subproblems are very tight. Since each user must be attended by the BS in only one time slot $t \in \mathcal{T}$, the final solution of the problem can be easily reconstructed for the original problem from the solutions of each time slot $t \in \mathcal{T}$. The decomposition of the problem allows us to apply the VNS procedure in a straightforwardly manner and also to compute tight bounds easily. It turns out that solving the problem to optimality becomes rapidly prohibitive from a computationally point of view when the instances dimensions increase.

The paper is organized as follows. Section 2 briefly introduces the system description and presents the OFDMA-TDMA formulation of the problem. Section 3 presents the VNS algorithmic procedure while section 4 provides preliminary numerical results.

Finally, section 5 gives the main conclusions of the paper and provides some insights for future research.

2 Problem Formulation

We consider a BS surrounded by several mobile users within a single cell area. The BS has to assign a set of $\mathcal{N} = \{1, .., N\}$ subcarriers (or subchannels) to a set of $\mathcal{K} = \{1, .., K\}$ users in different time slots $\mathcal{T} = \{1, .., T\}$ in order to allow users to send signals to the BS. The allocation process is performed by the BS dynamically in time depending on the quality of the channels which are intrinsically stochastic. The latter affects the amount of bandwidth channel capacity needed by users to transmit their signals. Without loss of generality, we assume that the BS can fully and accurately predict the channel state information for each $t \in \mathcal{T}$. This is possible in OFDMA-TDMA networks when using adaptive overlapping pilots in uplink applications [11]. A scheduling formulation for an uplink wireless OFDMA-TDMA network can thus be written as follows

$$\mathcal{P}: \quad \max_{x,\varphi} \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{n=1}^{N} c_{k,n}^{t} x_{k,n}^{t} \tag{1}$$

$$\text{st:} \sum_{n=1}^{N} p_{k,n}^{t} x_{k,n}^{t} \leq P_k \varphi_{k,t}, \quad \forall k, t \tag{2}$$

$$\sum_{t=1}^{T} \varphi_{k,t} = 1, \quad \forall k \tag{3}$$

$$\sum_{k=1}^{K} x_{k,n}^{t} \leq 1, \quad \forall n, t \tag{4}$$

$$x_{k,n}^{t} \in \{0, 1\}; \varphi_{k,t} \in \{0, 1\}, \quad \forall k, n, t \tag{5}$$

where $x_{k,n}^{t}, \forall k, n, t$ and $\varphi_{k,t}, \forall k, t$ are the decision variables. These variables are defined as follows: $x_{k,n}^{t} = 1$ if user k is assigned subcarrier n at time slot t and zero otherwise. Similarly, $\varphi_{k,t} = 1$ if user k is scheduled to be attended in time slot t and zero otherwise. Matrices $(c_{k,n}^{t})$, $(p_{k,n}^{t})$ and (P_k) are input data matrices defined as follows. The entries in $(c_{k,n}^{t})$ denote the capacity achieved by user k using subcarrier n in time slot t while entries in $(p_{k,n}^{t})$ denote the power utilized by user k using subcarrier n in time slot t. Finally, (P_k) denotes the maximum power allowed for each user k to transmit their signals to the BS. The objective function in $\mathcal{P}$ is aimed at maximizing the total bandwidth channel capacity of the network. Constraint (2) is a maximum available power constraint imposed for each user k and for each time slot t to transmit signals to the BS. This is the main constraint which makes the difference between a downlink and an uplink process. In the former, there should be only one power constraint imposed for the BS whereas in the latter, each user is constrained by its own available maximum power $P_k, k \in \mathcal{K}$. Constraint (3) imposes the condition that each user must be attended by the BS in a unique time slot $t \in \mathcal{T}$. This constraint is specifically related to the time domain which is basically the transmission scheme of TDMA wireless networks. Whereas constraint (4) is related to the OFDMA scheme which imposes the condition that each subcarrier should be assigned to at most one user at instant $t \in \mathcal{T}$. Finally, constraint (5) are domain constraints for the decision variables.

We note that $\mathcal{P}$ is an integer linear programming (ILP) formulation which is NP-Hard and thus difficult to solve directly for medium and large scale instances. Instead, we propose a VNS decomposition approach to compute tight bounds.

3 The VNS Approach

In order to compute tight bounds for $\mathcal{P}$ using a VNS metaheuristic approach, we first note that for any feasible assignment of $\varphi_{k,t} = (\bar{\varphi}_{k,t})$, i.e., such that $\sum_{t=1}^{T} \bar{\varphi}_{k,t} = 1, \forall k$. Problem $\mathcal{P}$ reduces to solving T subproblems of the following form

$$\mathcal{P}(t): \quad \max_{y} \sum_{k \in \mathcal{K}_t} \sum_{n=1}^{N} \hat{c}_{k,n}^{t} y_{k,n}^{t} \tag{6}$$

$$\text{st:} \sum_{n=1}^{N} \hat{p}_{k,n}^{t} y_{k,n}^{t} \leq \hat{P}_k, \quad \forall k \in \mathcal{K}_t \tag{7}$$

$$\sum_{k \in \mathcal{K}_t} y_{k,n}^{t} \leq 1, \quad \forall n \tag{8}$$

$$y_{k,n}^{t} \in \{0,1\}, \quad \forall k \in \mathcal{K}_t, n \in \mathcal{N} \tag{9}$$

where $\bigcup_{t=1}^{T} \mathcal{K}_t = \mathcal{K}$. Variables $y_{k,n}^{t}$ for each $k \in \mathcal{K}_t, n \in \mathcal{N}$ and $t \in \mathcal{T}$ are analogously defined as for $x_{k,n}^{t}$, i.e., $y_{k,n}^{t} = 1$ if user $k \in \mathcal{K}_t \subset \mathcal{K}$ is assigned subcarrier n in time slot t and zero otherwise. Matrices $(\hat{c}_{k,n}^{t})$, $(\hat{p}_{k,n}^{t})$, and $(\hat{P}_k)$ are respectively sub-matrices of $(c_{k,n}^{t})$, $(p_{k,n}^{t})$, and (P_k) we obtain from model $\mathcal{P}$ for each $t \in \mathcal{T}$ according to users in $\mathcal{K}_t$. Note that any solution $x_{k,n}^{t'}$ of $\mathcal{P}$ in a particular time slot $t' \in \mathcal{T}$ can be reconstructed by simply mapping the values of variables $y_{k,n}^{t'}$ $\forall k \in \mathcal{K}_{t'}, n \in \mathcal{N}$ into each user position in $x_{k,n}^{t'}$ $\forall k \in \mathcal{K}_{t'}$. All remaining values in $x_{k,n}^{t'}$ such that $k \notin \mathcal{K}_{t'}$ must be equal to zero. Therefore, for any feasible assignment $\varphi = \tilde{\varphi}$ the optimal solutions $\tilde{x}^{t}$ in $\mathcal{P}$ and optimal solutions $\tilde{y}^{t}$ in $\mathcal{P}(t)$, $\forall t \in \mathcal{T}$, we have

$$\sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{n=1}^{N} c_{k,n}^{t} \tilde{x}_{k,n}^{t} = \sum_{t=1}^{T} \sum_{k \in \mathcal{K}_t} \sum_{n=1}^{N} \hat{c}_{k,n}^{t} \tilde{y}_{k,n}^{t} \tag{10}$$

Note that there are T^K feasible assignments for $\varphi_{k,t} = (\bar{\varphi}_{k,t})$ and each subset $\mathcal{K}_t$ has a cardinality of $\sum_{k \in \mathcal{K}} \bar{\varphi}_{k,t}$ users. In case any subset $\mathcal{K}_{t'} = \emptyset$, it means that no user is scheduled to be attended in time slot $t' \in \mathcal{T}$. Also notice that solving each $\mathcal{P}(t), \forall t \in \mathcal{T}$ such that $\mathcal{K}_t \neq \emptyset$ is an NP-Hard problem as it is equivalent to solve a multiple choice multiple knapsack problem [2].

VNS is a recently proposed metaheuristic approach [3] that uses the idea of neighborhood change during the descent toward local optima and to scape from the valleys that contain them. We define only one neighbor structure as $Ngh(\varphi)$ for $\mathcal{P}$ as the set of neighbor solutions φ' in $\mathcal{P}$ at a distance "h" from φ where the distance "h" corresponds to the number of users assigned in solutions φ' and φ. The VNS procedure we propose is depicted in algorithm 3.1. As input receives an instance of problem $\mathcal{P}$

Algorithm 3.1: VNS approach

Data: A problem instance of $\mathcal{P}$
Result: A tight solution $(\bar{x}, \bar{\varphi}, \bar{f})$ for $\mathcal{P}$
$Time \leftarrow 0$; $\mathcal{H} \leftarrow 1$; $count \leftarrow 0$; $\varphi_{k,n} \leftarrow 0, x^t_{k,n} \leftarrow 0, \forall k, n, t$;
foreach $k \in \mathcal{K}$ **do**
    choose randomly $t' \in \mathcal{T}$;
    $\varphi_{k,t'} \leftarrow 1$;
foreach $t \in \mathcal{T}$ **do**
    Solve the linear programming relaxation of $\mathcal{P}(t)$
Let $(\tilde{x}, \tilde{\varphi}, \tilde{f})$ be the initial solution found for $\mathcal{P}$ with objective value function $\tilde{f}$;
while $(Time \leq maxTime)$ **do**
    for $i = 1$ *to* $\mathcal{H}$ **do**
        choose randomly $k' \in \mathcal{K}$ and $t' \in \mathcal{T}$;
        $\varphi_{k',t} \leftarrow 0, \quad \forall t \in \mathcal{T}$;
        $\varphi_{k',t'} \leftarrow 1$;
    foreach $t \in \mathcal{T}$ **do**
        Solve the linear programming relaxation of $\mathcal{P}(t)$
    Let (x^*, φ^*, g^*) be the new found solution for $\mathcal{P}$ with objective value function g^*;
    if $(g^* > \tilde{f})$ **then**
        $\mathcal{H} \leftarrow 1$;
        $(\tilde{x}, \tilde{\varphi}, \tilde{f}) \leftarrow (x^*, \varphi^*, g^*)$;
        $Time \leftarrow 0$; $count \leftarrow 0$;
    else
        Keep previous solution;
        $count \leftarrow count + 1$;
        if $\mathcal{H} \leq K$ *and* $count > \eta$ **then**
            $\mathcal{H} \leftarrow \mathcal{H} + 1$; $count \leftarrow 0$;
$(\bar{x}, \bar{\varphi}, \bar{f}) \leftarrow (\tilde{x}, \tilde{\varphi}, \tilde{f})$;

and provides a tight solution for it. We denote by $(\bar{x}, \bar{\varphi}, \bar{f})$ the final solution obtained with the algorithm where $\bar{f}$ represents the objective value function. The algorithm is simple and works as follows. First, it computes randomly a feasible assignment of $\tilde{\varphi} = (\tilde{\varphi}_{k,t})$ and solve each subproblem $\mathcal{P}(t), \forall t \in \mathcal{T}$ according to $\tilde{\varphi}$. This allows obtaining an initial solution $(\tilde{x}, \tilde{\varphi}, \tilde{f})$ for $\mathcal{P}$ that we keep. Next, the algorithm performs a variable neighborhood search by randomly scheduling $\mathcal{H} \leq K$ users in different time slots. Initially, $\mathcal{H} \leftarrow 1$ while it is increased in one unit when there is no improvement after new "η" solutions have been evaluated. On the other hand, if a new current solution found is better than the best found so far, then $\mathcal{H} \leftarrow 1$, the new solution is recorded and the process continuous. The whole process is repeated until the cpu time variable "$Time$" is less than or equal to the maximum available "$maxTime$". Note we reset "$Time = 0$" when a new better solution is found. This gives the possibility to search other "$maxTime$" units of time with the hope of finding better solutions.

As it can be observed, the VNS approach is constructed upon a key aspect of problem $\mathcal{P}$, namely its decomposition structure. On the other hand, the effectiveness of the algorithm also relies on the fact that the linear programming relaxation of each subproblem $\mathcal{P}(t), \forall t \in \mathcal{T}$ is very tight.

#	Instances Dimensions			Linear programs				VNS Approach			Gaps	
	K	N	T	$Opt.\mathcal{P}$	$\mathcal{LP}$	$Time\mathcal{P}$	$Time\mathcal{LP}$	$Ini.Sol.$	VNS	$Time$	$\mathcal{LP}$	VNS
1	8	32	10	2092	3488.1608	6.62	0.56	1622.7385	2118.9987	5.79	66.73	1.29
2	10	32	10	2771	3741.1867	13.28	0.62	2050.6937	2819.2678	7.49	35.01	1.74
3	12	32	10	3049	3967.9855	6.90	0.65	1786.4225	2972.4251	1.85	30.14	2.51
4	14	32	10	3109	4061.5510	8.79	0.71	2250.4192	3051.9827	1.71	30.63	1.83
5	20	32	10	3408	4137.6329	11.23	0.89	2246.0866	3409.5674	12.98	21.40	0.04
6	25	32	10	3591	4242.6719	28.84	1.00	2391.5610	3605.4858	11.20	18.14	0.40
7	30	32	10	3587	4208.7842	4.48	1.21	3039.4362	3592.3845	3.06	17.33	0.15
8	8	32	20	2351	6588.8126	11.48	0.84	1591.9860	2372.0500	1.45	180.25	0.89
9	10	32	20	2875	6548.8915	27.96	1.03	1505.5937	2879.8230	2.96	127.78	0.16
10	12	32	20	3281	7383.7370	51.51	1.11	2191.0147	3281.7037	1.60	125.04	0.02
11	14	32	20	4025	7801.8038	312.28	1.20	3077.3167	4025.6315	9.92	93.83	0.01
12	20	32	20	5965	8202.6180	74.56	1.51	3537.3049	5714.2134	58.03	37.51	4.20
13	25	32	20	6164	8195.6053	72.51	2.01	4022.6847	6186.9960	99.39	32.95	0.37
14	30	32	20	6548	8246.2234	105.09	2.28	4414.5224	6466.0896	115.84	25.93	1.25
15	8	64	10	3954	6754.0846	21.76	0.78	2829.6211	3970.8774	13.23	70.81	0.42
16	10	64	10	5606	7952.9545	35.12	0.87	3099.0138	5609.6752	6.56	41.86	0.06
17	12	64	10	5637	7780.5791	33.48	1.09	3950.4271	5541.4436	1.87	38.02	1.69
18	14	64	10	6334	7877.6160	47.54	1.20	5561.6601	6348.4896	3.87	24.37	0.22
19	20	64	10	6538	8225.0918	55.17	1.51	5386.4010	6553.5962	4.03	25.80	0.23
20	25	64	10	6941	8482.0337	77.23	1.78	6052.6174	6947.6831	5.95	22.20	0.09
21	30	64	10	7326	8496.0615	81.28	2.20	6282.7184	7326.0244	12.95	15.97	3e-4
22	8	64	20	4586	12789.0347	67.64	1.50	3200.9471	4544.7142	15.92	178.87	0.90
23	10	64	20	5753	14772.2571	167.57	1.60	4341.6136	5797.1178	10.15	156.77	0.76
24	12	64	20	6751	13449.2497	257.04	2.23	3891.1626	6781.0690	25.71	99.21	0.44
25	14	64	20	7692	14758.2530	576.20	2.37	4934.2025	7725.3751	18.18	91.86	0.43
26	20	64	20	11520	16342.8073	949.50	2.95	7692.8888	10795.0753	39.93	41.86	6.29
27	25	64	20	12297	16036.8844	536.17	3.86	8692.1874	12314.8432	110.20	30.41	0.14
28	30	64	20	12981	16873.0624	624.00	4.36	9327.2974	13017.9855	46.52	29.98	0.28
29	8	128	10	9292	15469.9953	167.86	1.31	6093.5991	9008.2856	32.44	66.49	3.05
30	10	128	10	10416	14341.4803	409.72	1.69	5150.2308	10590.8256	7.83	37.69	1.68
31	12	128	10	12248	16081.9795	728.13	1.77	8734.9364	12332.6018	13.91	31.30	0.69
32	14	128	10	12454	16002.4214	273.94	2.11	9538.7185	12510.7866	35.45	28.49	0.46
33	20	128	10	13441	16831.7606	387.81	2.69	9426.4508	13525.7884	9.31	25.23	0.63
34	25	128	10	14211	17059.0616	89.80	3.36	11492.2275	14236.2739	24.95	20.04	0.18
35	30	128	10	14546	17237.8062	519.84	4.39	12087.5565	14628.2521	8.53	18.51	0.57
36	8	128	20	9485	26344.7369	288.05	2.73	6802.1687	9547.7500	45.30	177.75	0.66
37	10	128	20	10993	25420.2375	479.06	3.84	8000.8647	11244.5429	6.77	131.24	2.29
38	12	128	20	13252	29327.4069	1577.70	4.05	8246.3013	13440.7492	23.98	121.31	1.42
39	14	128	20	14344	29151.5630	2239.27	5.73	8248.1641	14349.1162	33.30	103.23	0.04
40	20	128	20	23355	33356.0498	8416.50	5.64	16501.7620	22609.7154	34.83	42.82	3.19
41	25	128	20	24769	32729.1363	4964.91	7.48	16515.5073	24252.8271	96.31	32.14	2.08
42	30	128	20	25475	33352.2393	6170.50	11.44	17457.1910	25512.9978	162.50	30.92	0.15
Minimum values				2092	3488.2	4.48	0.56	1505.6	2119	1.45	15.97	3e-4
Maximum values				25475	33356	8416.5	11.44	17457	25513	162.50	180.25	6.29
Average values				8690.8	13431	737.57	2.43	6077.8	8656.2	28.18	61.37	1.04

Table 1 Upper and Lower bounds for $\mathcal{P}$

4 Numerical Results

In this section, we present preliminary numerical results for problem $\mathcal{P}$ using the proposed VNS algorithm. We generate realistic power data using a wireless channel from [7] while we set the capacities $c_{k,n}^t = \mathcal{M}_{k,n}^t, \forall k, n, t$ where $\mathcal{M}_{k,n}^t$ represents an integer number of bits randomly and uniformly generated between $\{1, .., 10\}$. These number of bits are required in higher order $\mathcal{M}$-PSK or $\mathcal{M}$-QAM modulation transmission schemes [1]. Specially for multimedia applications where the users bit rate demands are significantly higher. So far, we assume that the bit rate demands are uniformly distributed. In a larger version of this work, we will also consider other distribution types. Finally, we set $P_k = 0.4 \cdot \sum_{n\in\mathcal{N}} p_{k,n}^1, \forall k \in \mathcal{K}$ and $\eta = 500$. A Matlab program is implemented using CPLEX 12 to solve problem $\mathcal{P}$ while we use MOSEK solver [5] to solve its linear programming relaxation we denote hereafter by $\mathcal{LP}$ and each linear programming relaxation $\mathcal{P}(t), \forall t \in \mathcal{T}$ within each iteration of the VNS algorithm. The numerical experiments have been carried out on a Pentium IV, 1 GHz with 2 GoBytes of RAM under windows XP. In table 1, column 1 gives the instance number and columns 2-4 give the instances dimensions. In columns 5-8, we provide the optimal solutions of $\mathcal{P}$, $\mathcal{LP}$, and the cpu time in seconds CPLEX needs to solve $\mathcal{P}$ and $\mathcal{LP}$, respectively.

Similarly, in columns 9-11, we present the initial solutions found with the algorithm 3.1, its best solution found and the cpu time in seconds the algorithm needs to reach that solution. Notice that this cpu time considers all the time spent when solving all the subproblems involved in the algorithm sequentially and not in parallel as it could be improved. In all our tests we set the maximum time available to $maxTime = 50$ seconds. We also mention that whenever the variable $Time$ reached this amount, it means the algorithm did not find any better solution within 50 seconds, therefore we subtract this amount to the complete registered time. The latter provides the exact cpu time the VNS approach needs to find the best solution found so far. Finally, in columns 12 and 13 we provide gaps we compute as $\frac{\mathcal{LP}-Opt.\mathcal{P}}{Opt.\mathcal{P}} * 100$ and $\frac{|VNS-Opt.\mathcal{P}|}{Opt.\mathcal{P}} * 100$, respectively. Additionally, the last three rows in table 1 provide minimum, maximum and average values for columns 5-13, respectively. From table 1, we mainly observe that the bounds obtained with the VNS approach are very tight when compared to those obtained with $\mathcal{LP}$. For example, the gaps are less than 1 % in about 66.6 % and less 2 % in about 83.3 % of the instances when using the VNS approach. This is confirmed by the total average gap which is 1.04 %. Whereas the gaps obtained with the $\mathcal{LP}$ are not tight when compared to the optimal solutions in all cases. Another

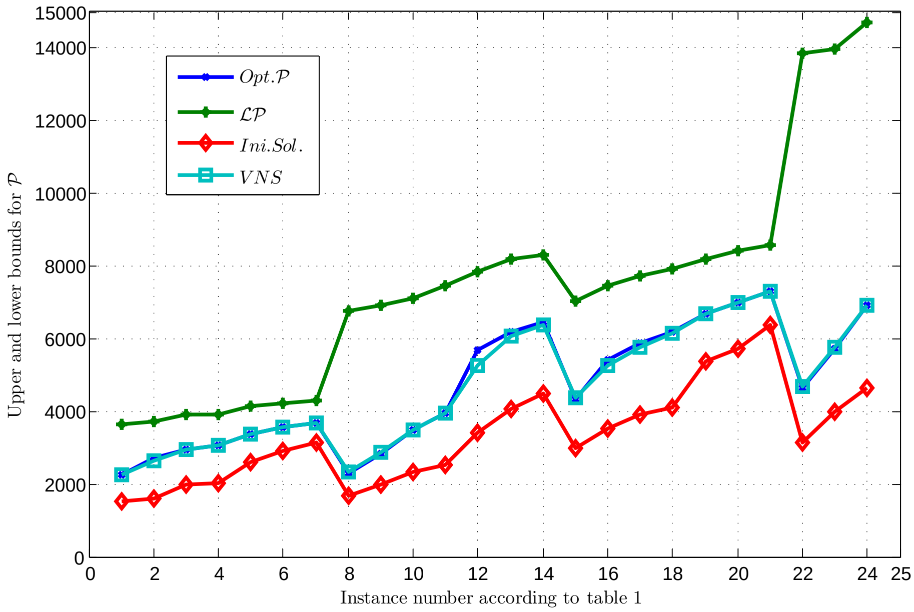

Fig. 1 Average bounds for instances 1-24 in table 1.

observation is that the average best solution found by the VNS algorithm improves in approximately 43% from the initial solution found by the algorithm which confirms its effectiveness. Moreover, when computing the difference for the average cpu time needed to solve problem $\mathcal{P}$ between the VNS approach and CPLEX, we obtain an improvement of 97,16 %. Finally, we observe that the cpu time required by CPLEX to compute an optimal solution of a particular instance grows rapidly while increasing its dimensions. So far, the averages presented in table 1 are computed using only one sample for the input data of instances 1-42. In order to provide more insight about these numerical results, in figures 1 and 2 we plot average results for instances 1-24 of table 1. We do not present averages for instances 25-42 since their cpu times become highly prohibitive as shown in table 1. For this purpose, we generate 25 samples for the input data of these instances. We use plots in this case to appreciate easily the trends of the average numerical results. In figure 1, the instance number appears in the horizontal axis while the vertical axis gives the averages we compute for the optimal solution of $\mathcal{P}$, for the linear programming relaxation of $\mathcal{P}$ ($\mathcal{LP}$), for the initial solution ($Ini.Sol.$) found with the VNS algorithm 3.1, and for the VNS approach respectively. Here, the trends of the curves mainly confirm the numerical results of table 1. We observe that VNS provides very tight near optimal solutions. By computing the average differences between VNS and the optimal solutions of $\mathcal{P}$ we obtain a 1.06 % of tightness which is similar to the average obtained in table 1. We also observe that the initial solutions are substantially improved by the VNS approach. In this case, we compute an average difference of 42.35% between the initials and best solutions of the VNS approach. Finally, we confirm that $\mathcal{LP}$ relaxation is not tight at all. In figure 2, the instance

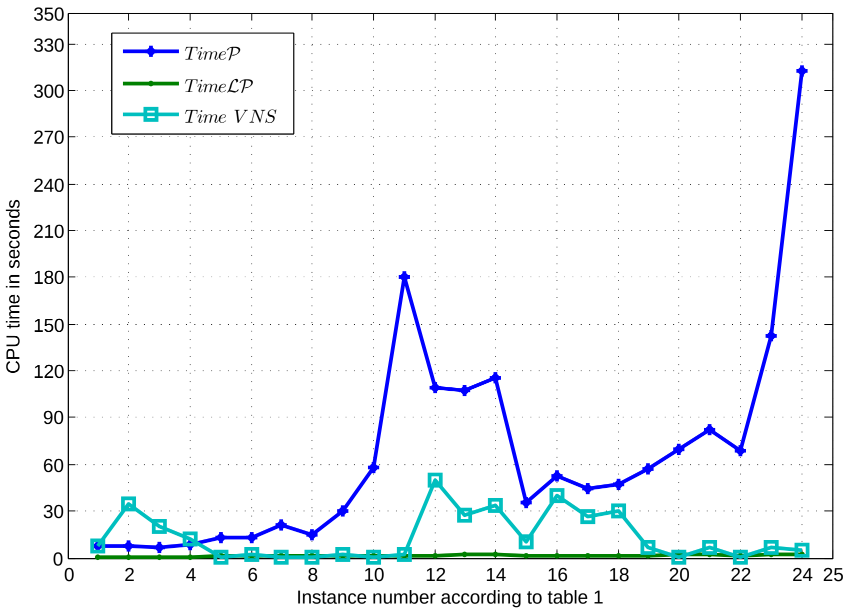

Fig. 2 Average CPU times in seconds for instances in table 1.

number appears in the horizontal axis while the vertical axis provides the average cpu time needed by CPLEX to solve problem $\mathcal{P}$, the average cpu time for $\mathcal{LP}$, and for the VNS approach as well. Here, we mainly observe that the cpu times required by VNS approach are significantly lower than CPLEX. In particular, we notice that for larger instances these cpu times remain below 10 seconds which is an interesting result.

5 Conclusions

In this paper, we proposed a hybrid resource allocation model for OFDMA-TDMA wireless networks and a VNS metaheuristic approach for solving the problem. The model is aimed at maximizing the total bandwidth channel capacity of an uplink OFDMA-TDMA network subject to user power and subcarrier assignment constraints while simultaneously scheduling users in time. As such, the model is best suited for non-real time applications where subchannel multiuser diversity can be further exploited in frequency and in time domains, simultaneously. The effectiveness of the proposed VNS approach relies on the decomposition structure of the problem which allowed solving a set of smaller integer linear programming subproblems within each iteration of the VNS approach. It turned out that the linear programming relaxations of these subproblems were very tight. Our numerical results showed tight bounds for the proposed algorithm, e.g. less than 2% in most of the instances. Besides, the bounds were obtained at a very low computational cost.

Future research will be focussed on developing other algorithmic approaches for solving each subproblem while considering other variants of the proposed model such as minimizing power subject to capacity constraints for uplink and downlink applications.

Acknowledgements The authors Pablo Adasme and Ismael Soto would like to thank to the projects: Conicyt/Insertion project Number 79100020, "Center for multidisciplinary research on signal processing" (project Conicyt/ACT1120) and project USACH/Dicyt Number 061117S) for their partial financial support.

References

1. Air Interface for Fixed Broadband Wireless Access Systems, MAC and Additional PHY Specifications for 2-11 GHz, IEEE Std. 802.16a, 2003.
2. Akbara Md Mostofa, M. Sohel Rahman, M. Kaykobad, E.G. Manning and G.C. Shoja, Solving the Multidimensional Multiple-choice Knapsack Problem by constructing convex hulls, Computers & Operations Research **33**, 12591273, (2006).
3. Hansen Pierre and Nenad Mladenovic, Variable neighborhood search: Principles and applications, European Journal of Operational Research **130**, 449-467 (2001).
4. Keivan Navaie and Halim Yanikomeroglu, Optimal Downlink Resource Allocation for Non-realtime Traffic in Cellular CDMA/TDMA Networks, IEEE Communications Letters, **10**, (2006).
5. MOSEK is an optimization software designed to solve large-scale mathematical optimization problems http://www.mosek.com/.
6. Sternad M., T. Svensson, T. Ottosson, A. Ahlen, A. Svensson and Anna Brunstrom, Towards Systems Beyond 3G Based on Adaptive OFDMA Transmission, Proceedings of the IEEE, 95, (2007).
7. Zukang Shen, Jeffrey G. Andrews and Brian L. Evans, Short Range Wireless Channel Prediction Using Local Information, Signals, Systems and Computers, In Conference Record of the Thirty-Seventh Asilomar Conference on, **1**, 1147-1151, (2003).
8. Yu-Jung Chang, Feng-Tsun Chien, and C. Kuo, Cross-layer QoS Analysis of Opportunistic OFDM-TDMA and OFDMA Networks, IEEE Journal on Selected Areas in Communications, Vol. 25, Issue 4, (2007), pp. 657-666.
9. M. Katoozian, Navaie, K., and Yanikomeroglu, H., Optimal Utility-Based Resource Allocation for OFDM Networks with Multiple Types of Traffic, In IEEE Vehicular Technology Conference, 2008. VTC Spring 2008, pp. 2223-2227.
10. K. Navaie and H. Yanikomeroglu, Optimal Downlink Resource Allocation for Non-realtime Traffic in Cellular CDMA/TDMA Networks, IEEE Communications Letters, vol. 10, pp. 278280, April 2006.

11. Mikael Sternad and Daniel Aronsson, Channel estimation and prediction for adaptive OFDMA/TDMA uplinks based on overlapping pilots, In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005, Vol. 3, pp. 861–864.