R A P P O R T D E R E C H E R C H E

L R I

A ROBUST SEMIDEFINITE RELAXATION APPROACH FOR DOWNLINK OFDMA RESOURCE ALLOCATION USING ADAPTIVE MODULATION

ADASME P / LISSER A / SOTO I

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

07/2009

Rapport de Recherche N° 1523

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

A Robust Semidefinite Relaxation Approach for Downlink OFDMA Resource Allocation Using Adaptive Modulation

Pablo Adasme[1] Abdel Lisser[1] Ismael Soto[2]

[1] Laboratoire de Recherche en Informatique, Universite Paris-Sud XI
Batiment 490, 91405, Orsay Cedex France

[2] Departamento de Ingenieria Industrial, Universidad de Santiago de Chile
Av. Ecuador 3769, Santiago, Chile

Abstract

This paper proposes two robust binary quadratic formulations for wireless downlink (DL) Orthogonal Frequency Division Multiple Access (OFDMA) networks when using adaptive modulation. The first one is based on a scenario uncertainty approach from Kouvelis and Yu [1] and the second is based on an interval uncertainty approach from Bertsimas and Sim [2]. Both robust models allow to decide what modulations and what sub-carriers are going to be used by a particular user in the system depending on its bits rate requirements. Thus, we derive for each, two semidefinite relaxations and by numerical results, we get a near optimal average tightness of 4.12% under the scenario approach and 1.15% under the interval uncertainty approach when compared to the optimal solution of the problem derived by linearizing the two quadratic models with Fortet linearization method.

Keywords: *Orthogonal Division Multiple Access, resource allocation, adaptive modulation, robust optimization, semidefinite programming.*

1 Introduction

When several users are connected to a Base Station (BS), a large number of signals are using the wireless channel, greater complexity is generated by the negative phenomena of Multiple Access Interference (MAI) and Multi-path distortions [3]. OFDMA is a suitable technology for combating these negative phenomena. It is currently the type of modulation used in wireless multi-user systems such as IEEE 802.11a/g WLAN, in networks of fixed access as IEEE 802.16a and in mobile WiMax deployments [4]. Moreover, it is envisioned as one of the most attractive candidates for 4G uplink and downlink wireless networks since it divides the wireless bandwidth channel into several orthogonal narrow band frequencies forming sub-carriers (or sub-channels) and thus giving access to several users simultaneously. In order to exploit the frequency diversity provided by this technology, the BS has to assign dynamically and efficiently sub-carriers to users depending on their service requirement and on the varying channel quality conditions. Therefore, adaptive modulation appears as an attractive and efficient strategy to increase data transmission throughput with high reliability [14]. However, the sub-carrier allocation problem that emerges is NP-Hard and thus very difficult to solve for the BS. Moreover, if we consider the uncertainty due to stochastic time varying nature in wireless channels, the problem becomes even harder since one solution may not be the right at a particular time. There have been proposed several mathematical models for the problem of resource allocation in DL OFDMA wireless systems [5] and some of them consider the uncertainty of imperfect channel estimations based on probabilities [15, 16]. In this paper, we propose two robust binary quadratic formulations for OFDMA to minimize power subject to sub-carrier and bit rate constraints under uncertainty. The first one is based on a scenario uncertainty approach from Kouvelis and Yu [1] and the second is based on an interval uncertainty approach from Bertsimas and Sim [2]. Thus, we derive two tight Semidefinite programming (SDP) relaxations for both robust formulations. We use SDP and robust optimization since to the best of our knowledge, it has never been applied for OFDMA models. Also, equivalent integer linear formulations are obtained for both quadratic models by applying Fortet linearization method to the quadratic terms [8]. Besides, we also formulate a OFDMA second order cone (SOC) robust model using Aharon Ben-Tal and Arkadi Nemirovski approach [19], but only to compare with Bertsimas approach. The work is organized as follows: in Section 2 we provide the general system description. In Section 3 we describe the main works on SDP and robust optimization related to combinatorial optimization, Section 4 details the robust quadratic models together with their equivalent linear programming formulations. In Section 5, we present and derive tight SDP relaxations for the robust models. In Section 6 we show numerical results for these SDP relaxations when compared to the optimal and relaxed solutions obtained from the equivalent linear formulations. Finally, in Section 7 we give some conclusions of this work.

2 System Description

We consider a DL OFDMA wireless network composed by a single cell with one base station (BS) and several mobile users. The BS consists of a set of N sub-carriers that have to be assigned to a set of K users using modulation sizes of at most M bits in the different sub-carriers. The allocation of sub-carriers to users will depend on the service required by each user. It means that a user might need more sub-carriers using higher modulations sizes if its downlink application is more demanding. Since the quality of a particular sub-channel varies in time and users need different amount of bits per unit of time, then for each user k we may have a function $f(c, BER_k)$ depending on the amount of bits to be transmitted by the channel pair (k, n) considering the BER_k performance for each user. The BER is defined as the ratio of the number of errors over the number of transmitted bits. We can use the following formula for user k using sub-carrier n with c bits.

$$P_{k,n}^c = \frac{f(c_{k,n}, BER_k)}{|\alpha_{k,n}|^2} \tag{2.1}$$

where $\alpha_{k,n}$ represents the time varying channel gain which can be modeled, for example as [11]:

$$\alpha_{k,n} = \sum_{l=1}^{L} \varrho_l \exp^{j(2\pi f_l + \Phi_l)} \tag{2.2}$$

with L, ϱ_l, f_l and Φ_l being the total number of incident waves, the amplitude, the doppler frequency and the initial phase of the incident wave, respectively. We can also use (2.3) for the function in (2.1) since it is the required transmission power for c bits/sub-carrier at a given BER with unity channel gain [23]:

$$f(c_{k,n}, BER_k) = \frac{N_0}{3}\left[Q^{-1}\left(\frac{BER_k}{4}\right)\right]^2 (2^c - 1) \tag{2.3}$$

where $Q^{-1}(x)$ is the inverse function of

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt \tag{2.4}$$

The main idea is to distribute efficiently sub-carriers of the BS while minimizing total power in the system when using adaptive modulation. It means that we should allocate sub-carriers to users, but deciding simultaneously the amount of bits to be transmitted in each sub-carrier. The BS is faced with this NP-hard problem and once the decision is taken, the bits of each user are modulated into an adaptive M-PSK or M-QAM symbol, which are subsequently combined using the inverse fast fourier transform (IFFT) into an OFDMA symbol which is transmitted through a slowly time-varying frequency-selective Rayleigh channel over a bandwidth B.

3 Related work on SDP and Robust Optimization

In this section we briefly explain what are SDP and robust optimization and describe their most important works related to combinatorial optimization. SDP is a subfield of convex optimization concerned with the optimization of a linear objective function subject to the intersection between an affine set and the conic space generated by positive semidefinite matrices. SDP arises as a generalization of linear programming by replacing the vector of variables with a symmetric matrix and replacing the nonnegativity constraints with a positive semidefinite constraint. As in linear programming, SDP has also several important properties: convexity, it has a rich duality theory (although not as strong as linear programming), and admits theoretically efficient solution procedures based on iterating interior point methods. A strong link between SDP and combinatorial optimization has been established in last decades. We mention for example the work of Grötschel, Lovász and Schrijver [27] who investigated in detail both LP and SDP relaxations to combinatorial optimization problems. Later, Lovász and Schrijver [28] showed that SDP problems could provide tighter relaxations for binary programming problems. Other important works are the contributions of Nesterov and Nemirovski [29, 30] and Alizadeh [31] who have shown that interior point methods pioneered by Karmarkar [32] for LP could be extended to SDP. Another recent and important work is the work of Goemans and Williamson [9] which shows that SDP could be highly effective to find a good approximation to the maxcut problem. We use SDP to get tighter bounds due to its proven efficiency in combinatorial optimization. Moreover, actual SDP solvers use interior point algorithms with polynomial time complexity and exploit matrix sparsity [10, 13].

Robust optimization is another important method which extents from linear programming [24]. One usually assumes that the data inputs are known with high exactitude in a linear program, however this assumption ignores the influence and the consequences that parameter uncertainties might have when finding a solution for these problems. For instance, it was shown in [19] that making slightly perturbations in the input data of the constraints of a linear programm, it is possible to violate some of the constraints and to perform poorly from an objective function point of view. In order to find "Robust" solutions which are in some sense inmuned against uncertainties of the input data, there are mainly two robust optimization approaches which do not make explicitly use of probabilities. The first treats the uncertainty by means of scenarios and the second assumes that each uncertain parameter can vary within a bounded interval. Under the scenario approach, the more intuitive method consists in representing the input parameters by a finite number of possible realizations. Each realization corresponds to a possible scenario constructed with exact input data so we can obtain a deterministic problem. Kouvelis and Yu are the main initiators of this approach [1]. In their work a set S of scenarios for the input data is defined and for each $s \in S$ a function f_s has to be minimized. If the feasible set for scenario s is $\mathbb{X}_s$, then the feasible set for the robust scenario problem is $\mathbb{X} = \bigcap_{s \in S} \mathbb{X}_s$. In the case of uncertain data in the objective function, it seems harder to select a unique robust solution, because the objective function will be different for each scenario. It is due to this reason that Kouvelis and Yu proposed three different variants based on a minimax criteria in order to find a robust solution: the absolute robustness, the robust deviation and the relative robustness. In the absolute robustness, the performance measure is applied for evaluating the decision across all scenarios and then, the worst case performance is recorded as the robust solution. For the second case, if we denote the optimal solution for a particular scenario $s \in S$ as x_s^*, then the performance of the decision corresponds to the minimum worst case deviation from x_s^*. Finally, in the third case called relative robustness, the minimum worst observed percentage deviation from the optimality of x_s^* across all scenarios $s \in S$ is registered. These variants assume that the solution is feasible for all scenarios $s \in S$, but this is not true all the time. One method to avoid constraints violations was introduced in [26] which consists in adding some penalties in the objective function, however the approach is hard to state since there is no way to determine the coefficients to apply. In summary, some of the common disadvantages under the scenario approach are: we do not dispose in advance of the appropriate finite number of scenarios, its complexity grows faster when incrementing them and the objective function deteriorates rapidly when using penalties in order to avoid constraint violations. Another representation of uncertainty instead of using a finite number of scenarios is to put the uncertain parameters in a bounded interval. For sake of simplicity, all the works based on interval uncertainty assume, without loss of generality, that the objective function is not subject to uncertainty since it is always possible to put it as another constraint in the model. The first step for an interval approach was taken by Soyster [33]. In this work each uncertain parameter is consider to be in an upper-lower bounded interval. The interval is centered around the nominal value of each parameter which is assumed to have a maximal perturbation value. In order to satisfy each constraint, independent of the values taken by the uncertain parameters, it suffices to take the highest value of them. This approach is considered as a very conservative approach because it is based on a worst case performance criteria since it does not care about the loss in the optimal value in exchange for finding an optimal robust solution. To address the high conservatism level of Soyster approach important and recent works in robust optimization are the works done by Ben-Tal and Nemirovsky [17, 18, 19], El-Ghaoui and Lebret [20], and El-Ghaoui et al., [21]. In these papers, ellipsoidal uncertainties are considered for linear programs which derives in solving the robust counterparts of the nominal problems in the form of conic quadratic problems. In [19], for instance a less conservative conic model based on an interval standard deviation approach is proposed. The authors in this work, also give some probability bounds in order to avoid constraint violation. On the contrary, the work of Bertsimas and Sim [2] considers a nominal linear program that leads to a linear robust counterpart which is more interesting and tractable from a combinatorial point of view, although this work is highly inspired in the work of Ben-Tal and Nemirovsky since uncertainty is treated in the same way, constraint by constraint. However, a more flexible parameter Γ_i, which corresponds to the number of uncertain input parameters in the i_{th} constraint is introduced to control the degree of uncertainty.

In this paper, we use robust optimization to deal with the uncertainty of random wireless channels which directly affect the power we compute when using formula (2.1).

4 The Binary OFDMA Quadratic Formulation

The above problem can be formulated as a binary quadratic integer programming problem. Using the above power formula (2.1) the quadratic model can be formulated as:

$$\text{QIP0}: \quad \min_{\{x_{k,n}, y_{n,c}\}} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} P_{k,n}^{c} \quad x_{k,n} y_{n,c} \tag{4.1}$$

$$\text{st:} \quad \sum_{n=1}^{N} x_{k,n} \left[\sum_{c=1}^{M} c \cdot y_{n,c} \right] = R_k \quad \forall k \tag{4.2}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1 \quad \forall n \tag{4.3}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1 \quad \forall n \tag{4.4}$$

$$x_{k,n}, y_{n,c} \in \{0,1\} \tag{4.5}$$

In this model, (4.1) is the objective function meaning that if a sub-carrier n is assigned to user k using modulation size c, then $P_{k,n}^{c}$ has to be minimized, constraint (4.2) represents the amount of bits needed by user k while (4.3) ensures that each sub-carrier should be assigned to only one user at a time. Constraint (4.4) is a modulation linear constraint used to decide which modulation will be used in each sub-carrier and finally, (4.5) are binary decision variables, meaning that if $x_{k,n} = 1$ and $y_{n,c} = 1$ simultaneously, then user k is using sub-carrier n with a modulation size of c bits. Notice from (4.2) that inequality

$$\sum_{k} R_k \leq MN \tag{4.6}$$

must hold, otherwise the solutions are infeasible in QIP0. In the next we derive two robust optimization models from QIP0. In the first, we adopt a scenario uncertainty approach whereas in the second we adopt interval uncertainty approach.

4.1 Absolute Robust Scenario Approach

If we assume that we can have several prediction power matrices at a given time, then we can associate each matrix as an element of a set S of possible scenarios, so we can write the objective function (4.1) of QIP0 as

$$\min_{\{x_{k,n},y_{n,c}\}} \max_{\{s\in S\}} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} \left[P_{k,n}^{c}\right]^{s} x_{k,n} y_{n,c} \tag{4.7}$$

From which an equivalent quadratic absolute robust formulation can be stated according to [1]. Using (4.7) together with constraints of QIP0, we can minimize power overall possible scenarios in order to get a robust solution to the problem. We refer to it as RoQIP1:

$$\text{RoQIP1}: \quad \min_{\{x_{k,n},y_{n,c},t\}} \quad t \tag{4.8}$$

$$\text{st:} \quad \sum_{k=1}^{K} \sum_{n=1}^{N} x_{k,n} \left[\sum_{c=1}^{M} \left[P_{k,n}^{c}\right]^{s} y_{n,c}\right] \leq t \quad \forall s \in S \tag{4.9}$$

$$\sum_{n=1}^{N} x_{k,n} \left[\sum_{c=1}^{M} c \cdot y_{n,c}\right] = R_k \quad \forall k \tag{4.10}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1 \quad \forall n \tag{4.11}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1 \quad \forall n \tag{4.12}$$

$$x_{k,n}, y_{n,c} \in \{0,1\}, \quad t \in \mathbb{R} \tag{4.13}$$

Notice from RoQIP1 that if we consider only one scenario, the problem reduces to QIP0. In order to compare RoQIP1 with some semidefinite relaxation, we introduce linearization variables $\varphi_{k,n}^{c} = x_{k,n} y_{n,c}$ [8] to get an equivalent integer linear programming model which we denote by RoIP1:

$$\text{RoIP1}: \quad \min_{\{x_{k,n}, y_{n,c}, \varphi^c_{k,n}, t\}} \quad t \tag{4.14}$$

$$\text{st:} \quad \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{c=1}^{M} \left[P^c_{k,n}\right]^s \cdot \varphi^c_{k,n} \leq t \quad \forall s \in S \tag{4.15}$$

$$\sum_{n=1}^{N}\sum_{c=1}^{M} c \cdot \varphi^c_{k,n} = R_k \quad \forall k \tag{4.16}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1 \quad \forall n \tag{4.17}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1 \quad \forall n \tag{4.18}$$

$$x_{k,n} \geq \varphi^c_{k,n} \quad \forall k, n, c \tag{4.19}$$

$$y_{n,c} \geq \varphi^c_{k,n} \quad \forall k, n, c \tag{4.20}$$

$$\varphi^c_{k,n} \geq x_{k,n} + y_{n,c} - 1 \quad \forall k, n, c \tag{4.21}$$

$$\varphi^c_{k,n}, x_{k,n}, y_{n,c} \in \{0,1\}, \quad t \in \mathbb{R} \tag{4.22}$$

where (4.19)-(4.21) are the linearization Fortet constraints implying that if $x_{k,n} = 1$ and $y_{n,c} = 1$ simultaneously then $\varphi^c_{k,n} = 1$. Next we present another important robust optimization approach which is more suitable when considering small variations in the input random powers.

4.2 Interval Uncertain Robust Approach

In order to find an interval robust approach for the OFDMA problem, we linearize the quadratic formulation of QIP0 and derive from it two robust models; one using Bertsimas and Sim approach [2] and other one using Aharon Ben-Tal and Arkadi Nemirovski [19] approach. Since the main idea of these approaches is to model the uncertainty input parameters in a symmetric and bounded interval, we put the power $p^c_{k,n} \in [p^c_{k,n} - \widehat{p}^c_{k,n}, p^c_{k,n} + \widehat{p}^c_{k,n}]$ where $\widehat{p}^c_{k,n} = \varepsilon p^c_{k,n}$ for some $\varepsilon > 0$.

4.2.1 OFDMA Bertsimas Robust Approach

To derive a robust model under this approach, we define the sets: $J = \{(k,n,c) \in K \times N \times M / P^c_{k,n}$is subject to uncertainty$\}$ and $\Upsilon = \{\Omega \cup \{(k',n',c')\}, \Omega \subseteq J, \Omega = \lfloor \Gamma \rfloor, (k',n',c') \in J \backslash \Omega\}$, where Γ is an input parameter which corresponds to the number of uncertain input parameters. In our problem $\Gamma \in [0, KNM]$. Using the Fortet linearized model of QIP0, we relax variables $\varphi^c_{k,n}$ to be between zero and one to state the following robust non-linear model. We refer to it as RoBSim:

$$\text{RoBSim} \quad \min_{\{x_{k,n}, y_{n,c}, \varphi^c_{k,n}, t\}} \quad t \tag{4.23}$$

$$\text{st:} \quad \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{c=1}^{M} P^c_{k,n}\varphi^c_{k,n} + \max_{\Upsilon}\left\{\sum_{(k,n,c)\in\Omega} \widehat{P}^c_{k,n}\varphi^c_{k,n} + (\Gamma - \lfloor\Gamma\rfloor)\widehat{P}^{c'}_{k',n'}\varphi^{c'}_{k',n'}\right\} \leq t \tag{4.24}$$

$$\sum_{n=1}^{N}\sum_{c=1}^{M} c \cdot \varphi^c_{k,n} = R_k, \quad \forall k \tag{4.25}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1, \quad \forall n \tag{4.26}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1, \quad \forall n \tag{4.27}$$

$$x_{k,n} \geq \varphi^c_{k,n} \quad \forall k,n,c \tag{4.28}$$

$$y_{n,c} \geq \varphi^c_{k,n} \quad \forall k,n,c \tag{4.29}$$

$$\varphi^c_{k,n} \geq x_{k,n} + y_{n,c} - 1 \quad \forall k,n,c \tag{4.30}$$

$$x_{k,n}, y_{n,c}, \varphi^c_{k,n} \in [0,1], t \in \mathbb{R} \tag{4.31}$$

Notice from RoBSim, that we do not need variables to bound $|\varphi^c_{k,n}|$ since these variables are nonnegative. In order to derive a linear robust counterpart for RoBSim we take the Max term from constraint (4.24) and evaluate it in its optimal value $|\varphi^c_{k,n}|^*$ to get

$$\beta(\varphi^*, \Gamma) = \max_{\Upsilon} \left\{ \sum_{(k,n,c)\in\Omega} \widehat{P}^c_{k,n} |\varphi^c_{k,n}|^* + (\Gamma - \lfloor \Gamma \rfloor) \widehat{P}^{c'}_{k',n'} |\varphi^{c'}_{k',n'}|^* \right\} \tag{4.32}$$

but this problem is equivalent to solving the following linear programming model [2]:

$$\beta(\varphi^*, \Gamma) = \max_{\phi^c_{k,n}} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} \widehat{P}^c_{k,n} |\varphi^c_{k,n}|^* \phi^c_{k,n} \tag{4.33}$$

$$\text{st:} \quad \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{c=1}^{M} \phi^c_{k,n} \leq \Gamma \tag{4.34}$$

$$0 \leq \phi^c_{k,n} \leq 1 \quad \forall k,n,c \tag{4.35}$$

which has the following dual minimization problem

$$\beta(\varphi^*, \Gamma) = \min_{w^c_{k,n}, \rho} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} w^c_{k,n} + \Gamma \rho \tag{4.36}$$

$$\text{st:} \quad \rho + w^c_{k,n} \geq \widehat{P}^c_{k,n} |\varphi^c_{k,n}|^* \quad \forall k,n,c \tag{4.37}$$

$$w^c_{k,n} \geq 0 \quad \forall k,n,c \tag{4.38}$$

$$\rho \geq 0 \tag{4.39}$$

Then, using strong duality theory from linear programming, we get the following linear robust model which we call RoLP2:

$$\text{RoLP2}: \min_{\{\varphi^c_{k,n}, x_{k,n}, y_{n,c}, t, w^c_{k,n}, \rho\}} t \tag{4.40}$$

$$\text{st:} \quad \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} P^c_{k,n} \varphi^c_{k,n} + \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} w^c_{k,n} + \Gamma \rho \leq t \tag{4.41}$$

$$\rho + w^c_{k,n} \geq \widehat{P}^c_{k,n} \varphi^c_{k,n} \quad \forall k,n,c \tag{4.42}$$

$$\sum_{n=1}^{N} \sum_{c=1}^{M} c \cdot \varphi^c_{k,n} = R_k, \quad \forall k \tag{4.43}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1, \quad \forall n \tag{4.44}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1, \quad \forall n \tag{4.45}$$

$$x_{k,n} \geq \varphi_{k,n}^{c} \quad \forall k,n,c \tag{4.46}$$

$$y_{n,c} \geq \varphi_{k,n}^{c} \quad \forall k,n,c \tag{4.47}$$

$$\varphi_{k,n}^{c} \geq x_{k,n} + y_{n,c} - 1 \quad \forall k,n,c \tag{4.48}$$

$$w_{k,n}^{c} \geq 0 \quad \forall k,n,c \tag{4.49}$$

$$\rho \geq 0 \tag{4.50}$$

$$\varphi_{k,n}^{c}, x_{k,n}, y_{n,c} \in [0,1], \quad t \in \mathbb{R} \tag{4.51}$$

We can easily get an integer linear programming formulation just changing variables $\varphi_{k,n}^{c}, x_{k,n}, y_{n,c} \in [0,1]$ to be in $\{0,1\}$. We denote this integer linear model as RoIP2. Now, if we drop Fortet constraints (4.46)-(4.48), we get the following robust quadratic formulation RoQIP2:

$$\text{RoQIP2}: \quad \min_{\{x_{k,n}, y_{n,c}, t, w_{k,n}^{c}, \rho\}} t \tag{4.52}$$

$$\text{st:} \quad \sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n}\left[\sum_{c=1}^{M} P_{k,n}^{c} y_{n,c}\right] + \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{c=1}^{M} w_{k,n}^{c} + \Gamma\rho \leq t \tag{4.53}$$

$$\rho + w_{k,n}^{c} \geq \widehat{P}_{k,n}^{c} x_{k,n} y_{n,c} \quad \forall k,n,c \tag{4.54}$$

$$\sum_{n=1}^{N} x_{k,n}\left[\sum_{c=1}^{M} c \cdot y_{n,c}\right] = R_k, \quad \forall k \tag{4.55}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1, \quad \forall n \tag{4.56}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1, \quad \forall n \tag{4.57}$$

$$w_{k,n}^{c} \geq 0 \quad \forall k,n,c \tag{4.58}$$

$$\rho \geq 0 \tag{4.59}$$

$$x_{k,n}, y_{n,c} \in \{0,1\}, \quad t \in \mathbb{R} \tag{4.60}$$

With RoQIP1, RoQIP2 and these equivalent linear integer programming models RoIP1, RoIP2, now we want to derive tight robust semidefinite relaxations to compare with the optimal solutions.

An interesting observation that emerges from these two robust quadratic models RoQIP1 and RoQIP2 is that, even when the scenario approach is considered to be more conservative in the sense that we lose optimality in the objective function in exchange for a more robust solution is that we can also get weaker robust solutions by using the scenario approach instead of using the interval if we wanted. This is better shown by means of the following proposition.

Proposition 1. *Let $\mathcal{P}_{k,n}^{c}$ be a nominal sample power matrix and $\varepsilon > 0$ be the input parameter used in model RoQIP2. If we generate each scenario matrix in RoQIP1 with values equal to $\mathcal{P}_{k,n}^{c} + \varepsilon'_{k,n}\mathcal{P}_{k,n}^{c}$, for any $\varepsilon'_{k,n} < \varepsilon$, and if we also set $\Gamma = KNM$, then the robust solutions obtained by RoQIP1 are less conservative; ie, the optimal values of RoQIP1 are lower than the optimal values obtained with RoQIP2, independent of the number* *of scenarios considered for RoQIP1.*

Proof: *We simply take the sum over all constraints in (4.54) of RoQIP2 to obtain:*

$\rho(KNM) + \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} w_{k,n}^{c} \geq \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} \widehat{P}_{k,n}^{c} x_{k,n} y_{n,c} > \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{c=1}^{M} (\varepsilon'_{k,n} \mathcal{P}_{k,n}^{c}) x_{k,n} y_{n,c}$

We can use the right term of the second strict inequality to put it in place of the left term of the first inequality in RoQIP2, so we can drop constraints (4.54), (4.58), (4.59) to get RoQIP1, but only for one scenario constraint. If we add more matrix scenario constraints we will always get lower optimal values than RoQIP2 since the robust scenario approach optimizes over the worst case so any other scenario constraint will be feasible. ■

4.2.2 OFDMA Nemirovsky Robust Approach

Another approach, also interesting is due to Aharon Ben-Tal and Arkadi Nemirovski [19]. In this approach, the robust counter part of a linear programming model is derived as a conic convex program, more precisely as a second order cone programm (SOCP). It is called this way due to the Euclidean Norm term that appears in some of the inequality constraints of a linear programm [24]. To derive an OFDMA robust model under this approach, similarly as we did for RoLP1, again we use the Fortet linearized model from QIP0 to state the following SOCP programm which we call RoNem.

$$\text{RoNem}: \quad \min_{\{\varphi_{k,n}^{c}, x_{k,n}, y_{n,c}, \phi_{k,n}^{c}, \psi_{k,n}^{c}, t\}} \quad t \tag{4.61}$$

$$\text{st:} \quad \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{c=1}^{M} P_{k,n}^{c}\varphi_{k,n}^{c} + \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{c=1}^{M} |\widehat{P}_{k,n}^{c}|\phi_{k,n}^{c} + \vartheta\sqrt{\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{c=1}^{M} (\widehat{P}_{k,n}^{c}\psi_{k,n}^{c})^{2}} \leq t \tag{4.62}$$

$$\sum_{n=1}^{N}\sum_{c=1}^{M} c\cdot\varphi_{k,n}^{c} = R_k, \quad \forall k \tag{4.63}$$

$$-\phi_{k,n}^{c} \leq \varphi_{k,n}^{c} - \psi_{k,n}^{c} \leq \phi_{k,n}^{c} \quad \forall k,n,c \tag{4.64}$$

$$\sum_{k=1}^{K} x_{k,n} \leq 1, \quad \forall n \tag{4.65}$$

$$\sum_{c=1}^{M} y_{n,c} \leq 1, \quad \forall n \tag{4.66}$$

$$x_{k,n} \geq \varphi_{k,n}^{c} \quad \forall k,n,c \tag{4.67}$$

$$y_{n,c} \geq \varphi_{k,n}^{c} \quad \forall k,n,c \tag{4.68}$$

$$\varphi_{k,n}^{c} \geq x_{k,n} + y_{n,c} - 1 \quad \forall k,n,c \tag{4.69}$$

$$\varphi_{k,n}^{c}, x_{k,n}, y_{n,c} \in [0,1], \phi_{k,n}^{c} \geq 0, \psi_{k,n}^{c} \geq 0, \quad t \in \mathbb{R} \tag{4.70}$$

In this model the Euclidean Norm term added corresponds to the standard deviation due to uncertainty. The main difference of RoNem with RoLP1 is that RoNem is a convex nonlinear model which is much harder to solve than the linear programm RoLP2, on the other hand, both RoNem and RoLP2 use a perturbation parameter ε and a similar way to manage the uncertainty level, in RoNem it is by means of parameter ϑ. We use RoNem in this paper only to compare the range of uncertainty offered by each one of them when varying ϑ and Γ parameters to their extreme values.

5 The Semidefinite Relaxations

In this section we derive two robust SDP relaxations, one for RoQIP1 and another one for RoQIP2. To do so, let's recall a few mathematical concepts. The set $\mathcal{S}_n = \{Z \in \mathbb{M}_n, Z = Z^T\}$ is the set of all n square symmetric matrices and $\mathcal{S}_n^+ = \{Z \in \mathcal{S}_n, a \in \mathbb{R}^n, a^T Z a \geq 0\}$ is the set of symmetric matrices satisfying the condition of positive semidefiniteness [12]. A set $\mathcal{C}$ is an affine space if the line through any two distinct points in $\mathcal{C}$ lies in $\mathcal{C}$, i.e. if for any two points $p_1, p_2 \in \mathcal{C}$ and $\theta \in \mathbb{R}$, we have $\theta p_1 + (1-\theta)p_2 \in \mathcal{C}$ [24].

5.1 SDP Relaxation for RoQIP1

In order to propose a SDP relaxation for RoQIP1, we define vector z as:

$$z^T = \begin{pmatrix} x_{1,1} & \cdots & x_{1,N} & \cdots & x_{K,1} & \cdots & x_{K,N} & y_{1,1} & \cdots & y_{1,M} & \cdots & y_{N,1} & \cdots & y_{N,M} \end{pmatrix}$$

Then, let matrix Z be a symmetric positive semidefinite matrix defined as:

$$Z = \begin{pmatrix} D & z & 0 \\ z^T & 1 & \vdots \\ 0 & \cdots & t \end{pmatrix} \succeq 0$$

where $D = zz^T$ is a semidefinite matrix with $\text{rank}(D) = 1$. Let's also define matrix A with the same order of matrix Z as:

$$A = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \succeq 0$$

and for each scenario $s \in S$ matrix Λ_s as sparse power symmetric matrices with half the coefficients taken from constraints (4.9). Using the trace operator $\text{Tr}\{\cdot\}$, the objective function of RoQIP1 can be written as:

$$\sum_i \sum_j A_{i,j} Z_{i,j} = \text{Tr}\{AZ\} = t \tag{5.1}$$

and constraints in (4.9) as

$$\sum_{k=1}^{K} \sum_{n=1}^{N} x_{k,n} \left[\sum_{c=1}^{M} \left[P_{k,n}^c\right]^s y_{n,c} \right] = z^T \Lambda_s z = \text{Tr}\{\Lambda_s D\} \leq \text{Tr}\{AZ\}, \quad \forall s \in S \tag{5.2}$$

Analogously to Λ_s we can construct sparse symmetric matrices U_k for each k in (4.10). We can also write constraints in (4.13) as $diag(D) = z$. This last constraint is a relaxation constraint for the condition of $z_i^2 = z_i$ for all i. Now, let's define for each constraint in (4.11), (4.12) the coefficient vectors ex_n, ey_n such that each element represents the coefficient of a variable in vector z, then using the fact that $diag(D) = z$ we can write these constraints as $\text{Tr}\{Diag(ex_n)D\} \leq 1$ and $\text{Tr}\{Diag(ey_n)D\} \leq 1$ for each n. So we use the following proposition which is valid for all ex_n, ey_n vectors.

Proposition 2. *For all n, constraints $Tr\{[ex_n][ex_n]^T D\} \leq 1$ are Tighter than $Tr\{Diag(ex_n)D\} \leq 1$.*
Proof: *(See lemma 2.1 in [12]). To see this, we just focus in one variable, say $x_{k,n}$. For a particular n, we use ex_n to write: $\sum_{k=1}^{K} x_{k,n} \leq 1$ as $[ex_n]^T z \leq 1$, then $([ex_n]^T z)(z^T[ex_n]) \leq 1^2$ and $Tr([ex_n][ex_n]^T D) \leq 1$ ($D = zz^T$) since $[ex_n]^T z \geq -1$. Now, for some matrix $Q = D - zz^T \succeq 0$, we have that $Tr\{[ex_n][ex_n]^T[Q + zz^T]\} \leq 1$ which is equivalent to $[ex_n]^T Q[ex_n] + ([ex_n]^T z)^2 \leq 1$ where the proof follows since $[ex_n]^T Q[ex_n] \geq 0$ due to the positive semidefiniteness of Q.* ■

Next, adding one zero row-column vector to the rank one positive semidefinite matrices $[ex_n][ex_n]^T$ and $[ey_n][ey_n]^T$ we use this proposition to re-write constraints (4.11) and (4.12) from RoQIP1 and to derive a tight semidefinite relaxation which we call RoSDP1:

$$
\begin{aligned}
\text{RoSDP1}: \quad & \min_{Z} && \text{Tr}\{AZ\} && (5.3)\\
& \text{st:} && \text{Tr}\{\Lambda_s Z\} \leq \text{Tr}\{AZ\} \quad \forall s \in S && (5.4)\\
& && \text{Tr}\{U_k Z\} = R_k \quad \forall k && (5.5)\\
& && \text{Tr}\{[ex_n][ex_n]^T Z\} \leq 1, \quad \forall n && (5.6)\\
& && \text{Tr}\{[ey_n][ey_n]^T Z\} \leq 1, \quad \forall n && (5.7)\\
& && \text{Tr}\{\zeta_i Z\} \geq 0 \quad \forall i \in \xi && (5.8)\\
& && diag(D) = z && (5.9)\\
& && Z \succeq 0 && (5.10)
\end{aligned}
$$

where ξ represents the set of positions of matrix D with nonzero entries and thus ζ_i are symmetric matrices used to keep positive all elements $i \in \xi$, i.e. in all positions where $P^c_{k,n} > 0$. Finally constraint (5.10) imposes the condition of matrix Z to be positive semidefinite.

5.2 SDP Relaxation for RoQIP2

In the case of RoQIP2 we use the same vector z as defined above to propose a semidefinite relaxation. We also define $\varpi = \text{Diag}(w^1_{1,1}, w^2_{1,1}, \ldots, w^M_{1,1}, w^1_{1,2}, \ldots, w^c_{k,n}, \ldots, w^M_{K,N})$ as a diagonal matrix of appropriate size where the entries are nonnegative and continuous real variables ($w^c_{k,n} \geq 0$). Then, let Σ be a symmetric positive semidefinite matrix defined as:

$$
\Sigma = \begin{pmatrix} D & z & & & \\ z^T & 1 & & & \\ & & \varpi & & \\ & & & \rho & 0 \\ & & & 0 & t \end{pmatrix} \succeq 0
$$

where ρ, t are continuous real variables according to model RoQIP2. Obviously this matrix is bigger than matrix Z, as the name suggests in [2], this is *"The Price of Robustness"*, since we are introducing a higher number of variables, nevertheless we recall that sparsity structure is efficiently exploited in SDP solvers [10]. We construct, analogously as we did for RoSDP1, sparse symmetric matrices: $\wp$ for the objective function in RoQIP2, $\digamma$ for (4.53), Π_j for all constraints in (4.54), V_k for all all constraints in (4.55) and $\Im_i \quad \forall i$ to keep positive elements in the positions where $p^c_{k,n} > 0$. Besides, we can also use the same semidefinite rank one matrices $[ex_n][ex_n]^T$ and $[ey_n][ey_n]^T$ used in RoSDP1 but now adding as many as necessary zero row-columns to get the appropriate size of matrix Σ, so we state the following robust SDP relaxation for RoQIP2 which we call RoSDP2:

$$
\begin{aligned}
\text{RoSDP2}: \quad & \min_{\Sigma} && \text{Tr}\{\wp\Sigma\} && (5.11)\\
& \text{st:} && \text{Tr}\{\digamma\Sigma\} \leq \text{Tr}\{\wp\Sigma\} && (5.12)\\
& && \text{Tr}\{\Pi_j \Sigma\} \leq 0 \quad \forall j \in J && (5.13)\\
& && \text{Tr}\{V_k \Sigma\} = R_k \quad \forall k && (5.14)\\
& && \text{Tr}\{[ex_n][ex_n]^T \Sigma\} \leq 1, \quad \forall n && (5.15)\\
& && \text{Tr}\{[ey_n][ey_n]^T \Sigma\} \leq 1, \quad \forall n && (5.16)
\end{aligned}
$$

$$\text{Tr}\{\Im_i \Sigma\} \geq 0 \quad \forall i \in \xi \tag{5.17}$$

$$diag(D) = z \tag{5.18}$$

$$\Sigma \succeq 0 \tag{5.19}$$

Notice that we can omit constraints (4.58) and (4.59) from RoQIP2 thanks to the semidefiniteness of matrix Σ.

6 Simulation Results

In order to get numerical results, we solve RoIP1, RoIP2 which correspond to the equivalent integer linear programs of RoQIP1 and RoQIP2, the linear programming relaxations RoLP1, RoLP2 and the proposed semidefinite relaxations RoSDP1 and RoSDP2. We also solve RoNem, but just to compare with RoLP2. We calculate power using formulas (2.1), (2.2) and (2.3) from section 2. Without loss of generality, we set the total number of incident waves to $L = 200$ as done in [16]. The amplitude vector $(\varrho_1, \ldots, \varrho_L)$ is assumed to be identically and Normally distributed for each component $\varrho_l \sim N(\mu = 0, \sigma^2 = 1)$. The initial phase of the l^{th} incident wave is calculated as $\Phi_l = 2\pi\lambda(l)/\max_{\{l\}}\{\lambda(l)\}$ for each l where $\lambda(l) \sim N(0,1)$ is also assumed to be identically and Normally distributed. The Doppler frequency is assumed to be $f_l = 30Hz$ for each l and the BER_k is set to 10^{-3} for each user. Finally, we set the power spectral density to $N_0 = 1/N$ dBW in each sub-carrier. A Matlab program is developed using Cplex 9.1, Csdp [13] and Cvx [25] software for solving RoIP1, RoIP2, RoLP1, RoLP2, RoSDP1, RoSDP2 and RoNem. The numerical experiments have been carried out on a Pentium IV, 1.9GHz with 2 GoBytes of RAM under windows.

6.1 Robust OFDMA under Kouvelis and Yu Approach

For RoSDP1, RoLP1 and RoIP1, we simulate one random sample power varying the number of users from 4 to 14 for different fixed number of sub-carriers $N = 32, 64, 128$. Only one channel sample is used due to the high computational effort when computing integer solutions. These values are realistic in OFDMA systems since the bandwidth of a single channel can span from 1.25MHz to 20MHz and is closely linked to the number of sub-carriers to be used in the Discrete Fast Fourier Transform (DFFT) which commonly take values of 32, 64, 128, 512, 1024 [5]. Besides, the relation between the number of users and sub-carriers usually satisfy $K << N$ [16]. We vary the modulation sizes from $M = 2$ up to $M = 4$ bits since they are also common values when using M-PSK or M-QAM modulations in OFDMA systems [14]. The number of constraints and variables of the instances we simulate, for different scenarios $S = 5, 10, 15$, are shown in table 1. These scenario power matrices are generated using different sample channels which we assume are independent among them. We calculate the number of constraints and variables from RoLP1 and RoSDP1 respectively. As RoLP1 corresponds to the relaxation of RoIP1, here we change constraint (4.22) by relaxing variables $\varphi^c_{k,n}, x_{k,n}, y_{n,c} \in [0,1]$. The number of variables in RoSDP1 is calculated according to the size of matrix Z, if the order of this matrix is F then the number of variables is $F(F+1)/2$ [10]. From a theoretically point of view, this is the correct number of variables in an SDP formulation, although only the non-zero entries are used and the solvers usually exploit the sparsity structure of the matrix. As an observation from this table we can say that RoLP1 has more constraints than RoSDP1, but we have the opposite for the number of variables. Also notice that the number of constraints, in each column of RoLP1 and RoSDP1, is augmented depending on the number of scenarios. On the contrary, the number of variables remains constant for different number of scenarios. Tables 2, 3 and 4 show results for $S = 5, 10$ and 15 scenarios respectively. These three tables show the optimum solution of RoIP1, feasible integer solutions when approximated with a simple greedy heuristic, the lower bounds for the linear programming RoLP1 and RoSDP1 relaxations, and the cpu time in seconds for the relaxations. It also shows the gaps for RoLP1

Table 1: Instances for Kouvelis Robust Approach

Size				RoLP1 Relaxation				RoSDP1 Relaxation			
				S=5	S=10	S=15	S=5,10,15	S=5	S=10	S=15	S=5,10,15
#	n	k	c	# Const.	# Const.	# Const.	# Var.	# Const.	# Const.	# Const.	# Var.
1	32	4	2	1737	1742	1747	449	522	527	532	18915
2	32	6	2	2507	2512	2517	641	716	721	726	33411
3	32	8	2	3277	3282	3287	833	910	915	920	52003
4	32	10	2	4047	4052	4057	1025	1104	1109	1114	74691
5	32	12	2	4817	4822	4827	1217	1298	1303	1308	101475
6	32	14	2	5587	5592	5597	1409	1492	1497	1502	132355
7	32	4	3	2441	2446	2451	609	682	687	692	25651
8	32	6	3	3531	3536	3541	865	940	945	950	42195
9	32	8	3	4621	4626	4631	1121	1198	1203	1208	62835
10	32	10	3	5711	5716	5721	1377	1456	1461	1466	87571
11	32	12	3	6801	6806	6811	1633	1714	1719	1724	116403
12	32	14	3	7891	7896	7901	1889	1972	1977	1982	149331
13	32	4	4	3145	3150	3155	769	842	847	852	33411
14	32	6	4	4555	4560	4565	1089	1164	1169	1174	52003
15	32	8	4	5965	5970	5975	1409	1486	1491	1496	74691
16	32	10	4	7375	7380	7385	1729	1808	1813	1818	101475
17	32	12	4	8785	8790	8795	2049	2130	2135	2140	132355
18	32	14	4	10195	10200	10205	2369	2452	2457	2462	167331
19	64	4	2	3465	3470	3475	897	1034	1039	1044	74691
20	64	6	2	5003	5008	5013	1281	1420	1425	1430	132355
21	64	8	2	6541	6546	6551	1665	1806	1811	1816	206403
22	64	10	2	8079	8084	8089	2049	2192	2197	2202	296835
23	64	12	2	9617	9622	9627	2433	2578	2583	2588	403651
24	64	14	2	11155	11160	11165	2817	2964	2969	2974	526851
25	64	4	3	4873	4878	4883	1217	1354	1359	1364	101475
26	64	6	3	7051	7056	7061	1729	1868	1873	1878	167331
27	64	8	3	9229	9234	9239	2241	2382	2387	2392	249571
28	64	10	3	11407	11412	11417	2753	2896	2901	2906	348195
29	64	12	3	13585	13590	13595	3265	3410	3415	3420	463203
30	64	14	3	15763	15768	15773	3777	3924	3929	3934	594595
31	64	4	4	6281	6286	6291	1537	1674	1679	1684	132355
32	64	6	4	9099	9104	9109	2177	2316	2321	2326	206403
33	64	8	4	11917	11922	11927	2817	2958	2963	2968	296835
34	64	10	4	14735	14740	14745	3457	3600	3605	3610	403651
35	64	12	4	17553	17558	17563	4097	4242	4247	4252	526851
36	64	14	4	20371	20376	20381	4737	4884	4889	4894	666435
37	128	4	2	6921	6926	6931	1793	2058	2063	2068	296835
38	128	6	2	9995	10000	10005	2561	2828	2833	2838	526851
39	128	8	2	13069	13074	13079	3329	3598	3603	3608	822403
40	128	10	2	16143	16148	16153	4097	4368	4373	4378	1183491
41	128	12	2	19217	19222	19227	4865	5138	5143	5148	1610115
42	128	14	2	22291	22296	22301	5633	5908	5913	5918	2102275
43	128	4	3	9737	9742	9747	2433	2698	2703	2708	403651
44	128	6	3	14091	14096	14101	3457	3724	3729	3734	666435
45	128	8	3	18445	18450	18455	4481	4750	4755	4760	994755
46	128	10	3	22799	22804	22809	5505	5776	5781	5786	1388611
47	128	12	3	27153	27158	27163	6529	6802	6807	6812	1848003
48	128	14	3	31507	31512	31517	7553	7828	7833	7838	2372931
49	128	4	4	12553	12558	12563	3073	3338	3343	3348	526851
50	128	6	4	18187	18192	18197	4353	4620	4625	4630	822403
51	128	8	4	23821	23826	23831	5633	5902	5907	5912	1183491
52	128	10	4	29455	29460	29465	6913	7184	7189	7194	1610115
53	128	12	4	35089	35094	35099	8193	8466	8471	8476	2102275
54	128	14	4	40723	40728	40733	9473	9748	9753	9758	2659971

Table 2: Results for Robust Approach with S=5

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	0.3010	0.3303	0.2505	0.3290	0.3010	0.3008	20.25	0.0007	0.2018
2	0.5000	0.6210	0.4273	0.4220	0.5090	0.4993	37.75	0.0015	0.1701
3	0.2044	0.2343	0.1661	0.5620	0.2255	0.2013	51.42	0.0155	0.2311
4	0.3019	0.3582	0.2472	0.7810	0.3527	0.2920	71.98	0.0339	0.2210
5	0.1507	0.1687	0.1261	0.6870	0.1748	0.1450	113.98	0.0393	0.1954
6	0.1212	0.1654	0.1028	0.9850	0.1367	0.1201	175.12	0.0088	0.1793
7	0.2787	0.4235	0.2124	0.4060	0.3680	0.2642	15.21	0.0549	0.3121
8	0.2598	0.4961	0.1912	0.5940	0.4235	0.2399	35.45	0.0830	0.3587
9	0.2369	0.3331	0.1525	0.7970	0.2989	0.2334	109.61	0.0152	0.5532
10	0.5201	0.6765	0.3400	0.7340	0.6237	0.5068	123.06	0.0262	0.5298
11	0.4436	0.6366	0.2863	1.2500	0.6291	0.4245	206.07	0.0449	0.5492
12	0.3865	0.4712	0.2531	2.0790	0.4459	0.3570	281.54	0.0828	0.5270
13	0.9144	1.9363	0.5982	0.7190	1.3474	0.8905	26.50	0.0268	0.5285
14	0.8224	1.2849	0.4055	0.7970	0.8922	0.8100	67.06	0.0154	1.0283
15	1.0580	1.4541	0.4768	1.4060	1.1967	0.9998	135.42	0.0582	1.2190
16	0.8390	1.4965	0.3854	1.7810	0.9960	0.7624	308.23	0.1004	1.1770
17	0.7173	1.8225	0.3221	1.8440	0.7744	0.6716	262.20	0.0679	1.2270
18	0.5626	0.8902	0.2831	2.5940	0.6982	0.5010	400.35	0.1230	0.9874
19	0.4402	0.5713	0.4232	0.5000	0.4502	0.4397	76.26	0.0010	0.0402
20	0.2624	0.3542	0.2051	0.6720	0.3879	0.2545	144.71	0.0309	0.2792
21	0.2970	0.4896	0.2281	1.1720	0.5072	0.2920	322.56	0.0171	0.3023
22	0.1811	0.2839	0.1482	1.4060	0.2318	0.1725	524.59	0.0500	0.2223
23	0.1730	0.2491	0.1493	1.8750	0.1796	0.1725	804.95	0.0027	0.1587
24	0.1969	0.2628	0.1556	3.2190	0.2196	0.1920	1325.80	0.0251	0.2649
25	0.9723	1.5457	0.8070	5.8750	1.0848	0.9678	144.26	0.0046	0.2048
26	0.2956	0.4689	0.2008	1.1560	0.3702	0.2743	307.15	0.0778	0.4725
27	0.3365	0.7066	0.2358	1.8750	0.3488	0.3349	639.42	0.0047	0.4269
28	0.5423	0.8573	0.4300	3.2350	0.8336	0.4980	808.31	0.0889	0.2612
29	0.3218	0.4898	0.2467	5.1410	0.4465	0.3182	1349.10	0.0115	0.3043
30	0.2917	0.4253	0.2091	8.1400	0.3723	0.2832	2013.60	0.0299	0.3950
31	1.7401	2.7039	0.8789	1.5470	1.7401	1.7375	345.67	0.0015	0.9799
32	0.3053	0.5637	0.2052	1.5310	0.3256	0.3036	456.73	0.0057	0.4878
33	0.6651	1.0334	0.3530	3.2810	0.6769	0.6602	895.84	0.0075	0.8842
34	0.8186	1.6094	0.5517	5.7660	0.8322	0.8122	1670.20	0.0079	0.4838
35	0.6711	1.1059	0.3295	7.7190	0.8702	0.6447	2035.70	0.0410	1.0367
36	0.5366	0.8839	0.2874	7.8750	0.7867	0.5217	2806.50	0.0285	0.8667
37	0.3618	0.5059	0.3438	1.0320	0.3672	0.3612	546.14	0.0017	0.0524
38	0.2625	0.3342	0.2347	3.5470	0.2869	0.2620	927.95	0.0020	0.1186
39	0.1616	0.2309	0.1330	2.6410	0.1675	0.1607	2155.00	0.0060	0.2155
40	0.1561	0.1742	0.1246	7.5620	0.1699	0.1515	3603.00	0.0303	0.2528
41	0.1509	0.4380	0.1307	6.6720	0.1509	0.1509	5808.00	0	0.1552
42	0.1366	0.1946	0.1132	10.4690	0.1525	0.1361	9526.00	0.0039	0.2066
43	0.1900	0.3446	0.1492	1.6720	0.2047	0.1893	858.29	0.0033	0.2735
44	0.6525	1.4273	0.4515	3.6720	0.7680	0.6378	2322.00	0.0231	0.4451
45	0.3284	5.7482	0.2298	5.1090	0.4209	0.3161	5187.00	0.0390	0.4292
46	0.2717	0.3540	0.1836	11.3750	0.2903	0.2696	6450.00	0.0076	0.4796
47	0.3791	0.8449	0.2571	13.6710	0.3957	0.3773	10065.00	0.0047	0.4746
48	0.4408	0.5391	0.2885	30.4220	0.4847	0.4350	14745.00	0.0135	0.5281
49	0.6501	1.2821	0.3716	2.9060	0.8695	0.6439	1532.60	0.0095	0.7495
50	0.8224	1.1917	0.3934	3.7660	0.9619	0.7519	3953.50	0.0937	1.0902
51	1.2506	2.9358	0.9346	7.3750	1.3190	1.2431	9507.50	0.0060	0.3381
52	0.6285	1.1024	0.3830	25.5780	0.6812	0.6224	12221.00	0.0098	0.6410
53	0.8795	2.3105	0.6696	12.6880	1.2849	0.8683	13810.00	0.0129	0.3134
54	0.4828	0.8895	0.3064	14.2030	0.7296	0.4760	21909.00	0.0142	0.5758

Table 3: Results for Robust Approach with S=10

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	0.2494	0.3676	0.1981	0.3600	0.2903	0.2260	12.93	0.1036	0.2588
2	0.2659	0.5004	0.2421	0.5780	0.5079	0.2645	24.95	0.0054	0.0983
3	0.1983	0.3104	0.1526	0.5780	0.2193	0.1811	55.60	0.0952	0.2999
4	0.1970	0.2951	0.1655	0.8120	0.2734	0.1882	78.25	0.0464	0.1899
5	0.1770	0.2215	0.1483	0.9060	0.2038	0.1706	130.93	0.0376	0.1933
6	0.2166	0.2584	0.1803	1.2180	0.2660	0.2043	236.93	0.0603	0.2011
7	0.4281	0.7564	0.3046	0.6880	0.5970	0.3943	18.60	0.0857	0.4056
8	0.4019	0.7197	0.2918	0.6720	0.5561	0.3778	45.51	0.0639	0.3775
9	0.5153	0.7494	0.3753	0.9850	0.6147	0.4884	110.06	0.0550	0.3730
10	0.3162	0.7691	0.2281	1.1720	0.4868	0.2982	142.96	0.0604	0.3863
11	0.4478	0.6549	0.3145	1.3750	0.5844	0.4336	248.32	0.0327	0.4238
12	0.3666	0.6353	0.2800	4.9690	0.5498	0.3490	347.79	0.0504	0.3093
13	0.6222	1.5894	0.3569	0.6410	0.9806	0.6037	36.98	0.0307	0.7433
14	0.8315	1.6858	0.5542	1.3430	1.3572	0.7750	73.64	0.0729	0.5003
15	1.1287	2.0457	0.6267	1.9220	1.7763	1.0457	151.76	0.0794	0.8010
16	0.9116	3.5234	0.5525	1.6720	1.5488	0.8175	267.43	0.1151	0.6500
17	0.6698	1.4522	0.3731	2.4530	1.1916	0.6123	444.64	0.0939	0.7953
18	0.9176	1.8987	0.4951	3.3120	1.5752	0.7942	488.39	0.1555	0.8536
19	0.3769	0.5996	0.2488	0.6410	0.3769	0.3663	161.90	0.0288	0.5150
20	0.2113	0.3932	0.1914	0.7970	0.2626	0.2099	190.11	0.0065	0.1039
21	0.3017	0.4032	0.2460	1.3440	0.3700	0.2827	396.18	0.0672	0.2262
22	0.2902	0.5144	0.2501	1.6100	0.4019	0.2869	655.62	0.0117	0.1604
23	0.2724	0.5774	0.2162	3.4840	0.3372	0.2652	1306.50	0.0274	0.2603
24	0.1598	0.1917	0.1322	4.6710	0.1795	0.1524	1761.40	0.0484	0.2089
25	0.7056	5.4029	0.4851	0.8440	0.8518	0.6909	183.03	0.0213	0.4545
26	0.4735	1.0506	0.4205	1.8120	0.8156	0.4623	379.03	0.0242	0.1259
27	0.3332	0.5620	0.2619	1.8900	0.5309	0.3217	692.15	0.0358	0.2722
28	0.6710	1.1555	0.4247	4.2340	0.7782	0.6478	1298.20	0.0358	0.5800
29	0.3176	0.4380	0.2019	5.5000	0.4013	0.2963	1843.20	0.0720	0.5734
30	0.4857	1.1657	0.3308	6.0930	0.6295	0.4687	2731.80	0.0364	0.4684
31	0.6873	1.5651	0.4767	1.0940	1.1914	0.6779	200.60	0.0140	0.4417
32	0.8903	3.2594	0.6116	2.6560	1.6481	0.8656	596.68	0.0286	0.4558
33	0.6781	2.6603	0.4825	4.2500	1.3182	0.6557	1061.70	0.0342	0.4055
34	1.1307	2.1771	0.5999	6.0940	1.5504	1.0540	1997.10	0.0728	0.8848
35	0.2975	0.7905	0.1849	6.3440	0.6248	0.2759	2140.00	0.0781	0.6091
36	0.7122	1.2222	0.3758	15.0930	1.1833	0.6430	3457.00	0.1077	0.8951
37	0.3248	0.5285	0.2879	1.2030	0.3946	0.3209	794.95	0.0120	0.1282
38	0.3387	0.5350	0.2817	2.1410	0.4549	0.3204	1404.70	0.0574	0.2026
39	0.2062	0.5335	0.1787	5.2180	0.2634	0.2004	3070.00	0.0291	0.1539
40	0.1838	0.2775	0.1533	7.7500	0.2089	0.1790	5356.70	0.0273	0.1994
41	0.1575	0.2063	0.1338	14.2500	0.2276	0.1544	8603.10	0.0201	0.1767
42	0.1632	0.2213	0.1384	15.6880	0.1866	0.1615	12815.00	0.0105	0.1798
43	1.4555	2.9882	1.2708	3.2960	1.5969	1.4505	1270.50	0.0035	0.1454
44	0.5263	0.8205	0.3522	4.4370	0.5921	0.5202	4311.20	0.0117	0.4943
45	0.5295	0.8508	0.3732	9.6410	0.5992	0.5253	7659.60	0.0079	0.4187
46	0.3504	0.6947	0.2554	9.6560	0.3857	0.3487	7351.60	0.0047	0.3718
47	0.5635	1.3113	0.4702	8.3280	0.8342	0.5455	13410.00	0.0330	0.1985
48	0.4603	0.6684	0.2990	19.6250	0.5778	0.4509	20440.00	0.0209	0.5392
49	1.2198	2.6128	0.8931	7.1410	1.8619	1.2124	1602.20	0.0062	0.3658
50	0.6538	1.8558	0.4074	7.3440	1.7809	0.6377	4131.80	0.0252	0.6048
51	0.6167	1.6796	0.3279	9.3130	0.9983	0.5773	9623.10	0.0683	0.8811
52	0.8073	1.7436	0.4858	17.6560	1.4350	0.7805	14818.00	0.0342	0.6618
53	1.0664	1.8629	0.5370	28.5780	1.2505	1.0423	15425.00	0.0231	0.9859
54	0.9428	2.0687	0.4439	76.6720	1.3376	0.9282	25724.00	0.0157	1.1239

Table 4: Results for Robust Approach with S=15

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	0.2702	0.5522	0.2316	0.3910	0.5430	0.2631	21.46	0.0271	0.1668
2	0.8385	1.6449	0.6374	0.7660	0.9352	0.7519	36.00	0.1151	0.3154
3	0.2431	0.2874	0.1912	0.8910	0.2835	0.2325	93.21	0.0456	0.2717
4	0.2535	0.3671	0.1914	0.9220	0.2957	0.2341	122.29	0.0828	0.3242
5	0.4173	0.5177	0.3241	1.4530	0.4889	0.3780	192.42	0.1040	0.2876
6	0.2639	0.4253	0.1961	1.5310	0.3383	0.2520	295.01	0.0473	0.3461
7	1.0380	2.1193	0.9640	0.6570	1.2606	1.0003	29.76	0.0377	0.0768
8	1.0535	1.5044	0.6954	0.9530	1.2767	0.9684	59.48	0.0878	0.5150
9	0.7196	2.1544	0.4397	1.1880	0.7781	0.6629	132.60	0.0856	0.6366
10	0.5107	1.0691	0.3391	1.9220	0.6219	0.4625	191.11	0.1041	0.5060
11	0.5991	0.8952	0.4070	2.5160	0.8929	0.5684	299.39	0.0541	0.4719
12	0.3006	0.4488	0.2102	2.1250	0.4308	0.2746	374.51	0.0946	0.4297
13	2.4513	6.7529	1.9312	0.6410	4.7120	2.4026	39.06	0.0203	0.2693
14	1.5297	2.6864	0.8914	1.3290	2.6316	1.3653	92.78	0.1204	0.7160
15	0.9375	2.3865	0.5521	1.4220	1.5358	0.8884	150.95	0.0553	0.6982
16	0.5670	1.1578	0.3561	2.1570	0.9793	0.5296	265.81	0.0705	0.5923
17	1.1531	2.4034	0.5640	3.4850	1.5243	1.0350	382.62	0.1141	1.0444
18	0.8774	1.5069	0.4588	3.3130	1.2710	0.8074	533.64	0.0867	0.9126
19	0.5641	1.4920	0.4875	1.9840	0.6081	0.5409	121.31	0.0429	0.1571
20	0.3537	0.6085	0.2903	1.2650	0.7185	0.3357	238.67	0.0535	0.2182
21	0.3895	0.7518	0.3069	2.4850	0.4041	0.3819	704.46	0.0199	0.2690
22	0.2636	0.3864	0.2319	2.3900	0.3318	0.2529	907.28	0.0425	0.1370
23	0.2051	0.2438	0.1643	2.9840	0.2366	0.1921	1350.90	0.0674	0.2479
24	0.2281	0.2715	0.1895	7.8590	0.2595	0.2185	2202.50	0.0437	0.2036
25	0.7536	1.3329	0.6213	1.0620	1.1144	0.7424	168.79	0.0151	0.2129
26	0.4428	0.8247	0.3571	1.8280	0.7243	0.4225	452.54	0.0480	0.2398
27	0.9099	2.2182	0.7281	3.0620	1.1770	0.9000	1064.50	0.0110	0.2497
28	0.4624	0.7891	0.2873	4.6560	0.5925	0.4213	1356.20	0.0975	0.6095
29	0.5082	1.1917	0.3567	7.1720	0.7010	0.4894	2235.70	0.0384	0.4246
30	0.4717	0.5602	0.2808	8.0470	0.5539	0.4395	3074.70	0.0731	0.6796
31	0.9636	2.1469	0.5438	1.5780	1.3451	0.9544	275.54	0.0097	0.7721
32	1.0309	2.1342	0.5818	3.4690	1.4864	0.9590	776.60	0.0749	0.7720
33	1.5135	3.1254	0.8546	4.0940	2.2119	1.3986	1515.00	0.0821	0.7711
34	0.9658	3.0176	0.6245	8.7340	2.2006	0.8764	2308.30	0.1020	0.5465
35	1.0776	14.6670	0.4988	8.7030	1.6360	1.0220	3367.30	0.0544	1.1604
36	0.8060	1.1961	0.4101	15.9060	1.1367	0.7608	4649.00	0.0594	0.9653
37	0.4244	0.7096	0.3527	1.5000	0.5099	0.4052	972.95	0.0475	0.2035
38	0.3318	0.5637	0.2478	3.7500	0.3479	0.3155	2099.00	0.0518	0.3390
39	0.5642	0.7069	0.5085	9.1250	0.6505	0.5552	3707.00	0.0162	0.1094
40	0.2748	0.5691	0.2554	5.3750	0.3678	0.2715	5671.60	0.0120	0.0758
41	0.3233	0.5080	0.3048	8.2030	0.4420	0.3186	9332.00	0.0150	0.0609
42	0.2749	0.5045	0.2198	20.1400	0.2899	0.2651	17447.00	0.0369	0.2508
43	3.8904	11.3699	2.6122	4.7810	3.8904	3.7890	2634.60	0.0268	0.4894
44	1.0097	1.5105	0.6322	6.1870	1.1879	1.0027	4319.50	0.0069	0.5970
45	0.6654	1.6615	0.5984	8.0940	0.9856	0.6623	6376.80	0.0047	0.1121
46	0.5451	0.8690	0.3743	16.2970	0.6470	0.5321	8919.80	0.0243	0.4561
47	0.5671	3.3929	0.3733	16.6880	0.7082	0.5400	13787.00	0.0502	0.5191
48	0.4618	1.2139	0.3054	40.4690	0.5892	0.4549	21165.00	0.0152	0.5123
49	2.7115	3.7895	1.3779	4.7970	2.9881	2.6613	2009.80	0.0189	0.9679
50	1.7706	5.4858	1.4722	16.0780	1.9706	1.7619	6188.00	0.0049	0.2027
51	1.9429	6.7339	1.6646	4.3440	2.2159	1.9366	10545.00	0.0032	0.1672
52	0.8469	2.8122	0.6033	22.6410	1.0288	0.8438	16180.00	0.0037	0.4038
53	1.0879	1.8467	0.5791	41.7650	1.5318	0.9835	17744.00	0.1061	0.8787
54	0.8293	1.6929	0.4831	38.0470	1.2954	0.8150	33892.00	0.0176	0.7166

Table 5: Results for Robust Approach with S=5 and $\sum_k R_k = MN$

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	0.5438	0.8431	0.4926	0.3280	0.5438	0.5436	24.25	3.1e-4	0.1039
2	0.3061	0.3464	0.2533	0.4380	0.3061	0.3028	50.90	0.0109	0.2084
3	0.1774	0.1884	0.1410	0.5000	0.1774	0.1735	74.08	0.0222	0.2580
4	0.2866	0.3797	0.1975	0.6250	0.2995	0.2582	146.21	0.1100	0.4514
5	0.2991	0.6818	0.2139	0.8120	0.3025	0.2694	205.97	0.1104	0.3981
6	0.2939	0.2993	0.2047	0.9840	0.2939	0.2670	247.12	0.1007	0.4359
7	0.7438	1.0514	0.4367	0.4060	0.7438	0.7332	24.43	0.0145	0.7034
8	0.4620	0.4968	0.2616	0.5940	0.4636	0.4336	52.91	0.0655	0.7661
9	1.0670	6.6151	0.7054	0.8130	1.0670	1.0594	154.83	0.0072	0.5125
10	0.5184	0.6175	0.2963	1.0470	0.5184	0.5064	208.08	0.0236	0.7491
11	0.6528	2.3378	0.3840	1.4060	0.6528	0.6528	343.16	0	0.6999
12	0.4674	0.4924	0.2684	1.4060	0.4686	0.4553	420.63	0.0265	0.7411
13	1.4761	2.5905	0.7445	0.4680	1.4761	1.3605	62.70	0.0850	0.9828
14	0.9681	1.1076	0.4079	0.8120	1.0454	0.9232	109.10	0.0486	1.3736
15	1.0306	1.4837	0.4422	1.0310	1.0306	1.0047	190.09	0.0257	1.3303
16	1.6395	2.4370	0.7124	1.3440	1.9036	1.6336	283.65	0.0036	1.3012
17	1.1768	1.3900	0.4505	2.6870	1.1768	1.1410	433.19	0.0314	1.6119
18	1.2978	3.3316	0.5306	1.9840	1.2978	1.2328	499.50	0.0528	1.4459
19	0.4598	0.5599	0.3615	0.4850	0.4598	0.4373	99.14	0.0513	0.2719
20	0.2732	0.4975	0.2180	0.8280	0.3008	0.2724	284.94	0.0028	0.2531
21	0.3499	0.8137	0.2708	1.2660	0.3499	0.3386	639.74	0.0334	0.2920
22	0.1992	0.2107	0.1631	1.7500	0.1992	0.1991	945.05	5.2e-4	0.2218
23	0.1681	0.2499	0.1312	2.8750	0.1705	0.1659	1546.90	0.0134	0.2810
24	0.2594	1.0220	0.2070	2.9380	0.2640	0.2569	2310.90	0.0097	0.2533
25	0.7542	7.5841	0.4428	1.2970	0.7542	0.7509	254.43	0.0044	0.7033
26	0.5161	0.6345	0.3050	1.7030	0.5161	0.5153	498.35	0.0016	0.6919
27	0.4385	0.6185	0.2603	3.0310	0.4385	0.4271	989.20	0.0268	0.6850
28	0.4961	0.6974	0.2893	1.7970	0.4961	0.4888	1445.90	0.0151	0.7150
29	0.4855	0.5997	0.2887	4.2810	0.4855	0.4829	2374.00	0.0055	0.6818
30	0.6105	3.3295	0.3744	8.8280	0.6105	0.6018	3998.70	0.0144	0.6304
31	1.5101	4.7885	0.8829	1.1720	1.5101	1.5101	343.86	0	0.7104
32	1.6205	3.2693	0.7332	2.9380	1.6205	1.5946	787.03	0.0162	1.2102
33	1.7496	2.4992	0.7721	3.9530	1.7496	1.6928	1488.60	0.0336	1.2660
34	1.4989	2.1820	0.6335	4.6720	1.4989	1.4273	2944.30	0.0502	1.3662
35	0.9841	1.3876	0.4061	8.2660	1.0012	0.9754	1793.20	0.0089	1.4234
36	0.9041	1.1859	0.3804	6.4060	0.9041	0.8990	3500.50	0.0058	1.3765

Figure 1: Greedy Heuristic

Input: $x_{Sdp}, y_{Sdp}; x_{Lp}, y_{Lp}$; R_k Output: Feasible Solutions for RoQP1 or RoQIP2

```
for each position
   Pick a uniform Random number r ∈ [0,1]
   if x_Sdp, x_Lp; y_Sdp, y_Lp ≥ r then
      x_Sdp, x_Lp; y_Sdp, y_Lp ← 1
   else
      x_Sdp, x_Lp; y_Sdp, y_Lp ← 0
   end if
end for
if(Not Feasible x_Sdp and y_Sdp; x_Lp and y_Lp due to (4.10),(4.11),(4.12))
      Randomly adjust y_Sdp, y_Lp
      add or erase sub-carriers if necessary
end if
```

Table 6: Instances for Bertsimas Robust Approach

Instances				RoLP2		RoSDP2	
#	n	k	c	# Constraints	# Variables	# Constraints	# Variables
1	32	4	2	2246	706	775	101926
2	32	6	2	3272	1026	1097	207046
3	32	8	2	4298	1346	1419	349030
4	32	10	2	5324	1666	1741	527878
5	32	12	2	6350	1986	2063	743590
6	32	14	2	7376	2306	2385	996166
7	32	4	3	3206	994	1063	186966
8	32	6	3	4680	1442	1513	376278
9	32	8	3	6154	1890	1963	631126
10	32	10	3	7628	2338	2413	951510
11	32	12	3	9102	2786	2863	1337430
12	32	14	3	10576	3234	3313	1788886
13	32	4	4	4166	1282	1351	297606
14	32	6	4	6088	1858	1929	595686
15	32	8	4	8010	2434	2507	996166
16	32	10	4	9932	3010	3085	1499046
17	32	12	4	11854	3586	3663	2104326
18	32	14	4	13776	4162	4241	2812006
19	64	4	2	4486	1410	1543	404550
20	64	6	2	6536	2050	2185	823686
21	64	8	2	8586	2690	2827	1390278
22	64	10	2	10636	3330	3469	2104326
23	64	12	2	12686	3970	4111	2965830
24	64	14	2	14736	4610	4753	3974790
25	64	4	3	6406	1986	2119	743590
26	64	6	3	9352	2882	3017	1499046
27	64	8	3	12298	3778	3915	2516646
28	64	10	3	15244	4674	4813	3796390
29	64	12	3	18190	5570	5711	5338278
30	64	14	3	21136	6466	6609	7142310
31	64	4	4	8326	2562	2695	1185030
32	64	6	4	12168	3714	3849	2375110
33	64	8	4	16010	4866	5003	3974790
34	64	10	4	19852	6018	6157	5984070
35	64	12	4	23694	7170	7311	8402950
36	64	14	4	27536	8322	8465	11231430

and RoSDP1 which are calculated as $Gap_{LP} = \left[\frac{RoIP1-RoLP1}{RoLP1}\right]$ and $Gap_{SDP} = \left[\frac{RoIP1-RoSDP1}{RoSDP1}\right]$. The simple greedy heuristic is shown in figure 1 and it is intended only to find a feasible solution in a fair manner rather than to find the optimal solution of RoQIP1. The algorithm for this heuristic simply takes as input the R_k bits needed by each user, the sub-carrier allocation matrices x_{Sdp}, y_{Sdp} from RoSDP1 and x_{Lp}, y_{Lp} from RoLP1, then it does a one randomized rounding iteration on its elements and corrects if there is no feasible solution. To generate these feasible solutions we assign randomly N_k sub-carriers for each user such that $\sum_k N_k = N$. Then, we put $R_k = N_k T$ where T is a uniform integer random number generated to be between $1 \leq T \leq M$. Here, we recall that inequality (4.6) must be satisfied in order to have feasible solutions for RoIP1. Now, if we compute from tables 2, 3, 4, the averages gaps from the optimal solutions for SDP and LP relaxations over the 54 instances, we get a tightness of 2.81% and 47.42 % for $S = 5$, 4.46 % and 43.21 % for $S = 10$ and 5.09% and 44.96% for $S = 15$. We can say, from this instances that, the more scenarios we consider the less tightness we can ensure with RoSDP1. However, in the three cases clearly we get near optimal solutions for SDP and very low quality solutions for LP. Similarly, if we compute for the three tables, the averages for the gaps of SDP for $N = 32$, $N = 68$ and $N = 128$ respectively, we get 4.44 %, 2.42%, 1.56% for $S = 5$, 6.91%, 4.17%, 2.28% for $S = 10$ and 7.52%, 5.2%, 2.57% for $S = 15$. These last averages again show the detrimental while incrementing the number of scenarios, however they also show that while the bigger the ratio N/K is, the better the gain that can be achieved. The other positive observation extracted from these tables is that when using a simple greedy heuristic just to approximate integer feasible solutions, we get better results for RoSDP1 than for RoLP1. For example, if we compute for each instance of tables 2, 3, 4 the ratio $\left[\frac{feasible-optimum}{optimum}\right] * 100$ and calculate the average over all instances, for RoLP1 and RoSDP1 respectively, we get 91.4% and 19.6% for $S = 5$; 107.1% and 43.2% for $S = 10$; and 126.3%, 34.9% for $S = 15$, which confirm SDP tightness. The only drawback for RoSDP1 compared to RoLP1 is that Csdp requires more time than Cplex to get the solutions. Taking the averages over all the instances, for SDP and LP respectively, the cpu time in seconds are 2671.1 and 4.5391 for $S = 5$, 3366.3 and 6.3341 for $S = 10$, and 3945.8, 7.0967 for $S = 15$. We also present some results for $S = 5$ scenarios in table 5, but now for the particular case in which we have that $\sum_k^K R_k = MN$. This forces all users to use the maximum modulation size in their allotted sub-carriers. We do this by multiplying each N_k by M, and then all variables $y_{n,M} = 1$ for RoSDP1 and RoLP1 respectively. We just show these results for $S = 5$ since it is in this case where we have a better tightness for RoSDP1. Naturally, in this case, the greedy heuristic of figure 1 does the randomized rounding only in the matrices x_{Sdp} and x_{Lp} obtained from RoSDP1 and RoLP1. We see from table 5 that almost all the feasible solutions obtained with the greedy heuristic, which is not intended to find the optimal solution, are equal to the optimal value of RoIP1. Moreover, if again we compute for each instance of table 5 the ratio $\left[\frac{feasible-optimum}{optimum}\right] * 100$ and calculate the average over all instances, for RoLP1 and RoSDP1 respectively, we get 110,8% and 1.26%.

6.2 Comparison between RoNem and RoLP2

In order to see which model, if RoNem or RoLP2, offers a higher range of uncertainty, we take a few instances and run some simulations. To do this, we use one sample power matrix for four different size instances where we set $\varepsilon = 5\%$. We compute RoIP2, RoLP2, RoNem and the cpu time in seconds for both relaxations just varying the parameters ϑ (for RoNem) and Γ (for RoLP2). We put ϑ in the extreme values 0 and 5 and Γ in the extreme values 0 and KNM so we can see the range of variation offered by both relaxations. Results are shown in table 7. We observe from this table that both approaches offers exactly the same range of variation for uncertainty and that RoNem needs more time to be solved than RoLP2, so from now on we just focus on RoLP2.

Table 7: Comparison Between RoNem and RoLP2 Relaxations

Instances	K	N	M	ϑ	Γ	RoIP2	RoLP2	Cpu time	RoNem	Cpu time
1	6	32	4	0	0	0.2351	0.1447	0.6400	0.1446	26
1	6	32	4	5	KNM	0.2469	0.1519	0.7190	0.1519	22
2	10	32	4	0	0	0.2196	0.1164	0.7970	0.1164	56
2	10	32	4	5	KNM	0.2306	0.1223	0.9220	0.1222	53
3	6	64	4	0	0	0.4061	0.2948	0.9690	0.2948	91
3	6	64	4	5	KNM	0.4264	0.3096	1.4530	0.3095	62
4	10	64	4	0	0	0.2218	0.1171	1.2810	0.1170	305
4	10	64	4	5	KNM	0.2329	0.1229	1.5930	0.1229	254

6.3 Robust OFDMA under Bertsimas and Sim Approach

In the case of Bertsimas and Sim approach for RoIP2, RoLP2 and RoSDP2, we also simulate one random sample power varying the number of users from 4 to 14, but now for only two fixed number of sub-carriers $N = 32$ and $N = 64$, since the number of variables is bigger. In table 6, we show the sizes of the instances we simulate, the number of constraints and variables from RoLP2 and RoSDP2 respectively. The number of variables in RoSDP2 are calculated depending on the size of matrix Σ. Obviously this matrix is larger than matrix Z used in the Kouvelis and Yu approach. We observe from table 6, that RoLP2 has more constraints than RoSDP2, however the number of variables in RoSDP2 grows faster compared to augmentation of variables for RoLP2 in table 1. Here, we recall that we are using only the non-zero entries of matrix Σ and that the sparsity structure is efficiently exploited by SDP solvers. As in this case we do not have scenarios, we can play with two parameters to have a certain level of protection against uncertainty, the $\varepsilon > 0$ and $\Gamma \geq 0$ parameters. The first is used to provide a wider interval for uncertainty while the second is used to vary the uncertainty level depending on the number of input random parameters. In our problem, $0 \leq \Gamma \leq KNM$. Figure 2 shows three graphs for a small, medium and big size instances using $\varepsilon = 5\%$ and varying Γ parameter from 1 to 100 with increments of 5 units. We do not show higher values of Γ since we observe that in the three cases the curves remains constant for values of $\Gamma \geq 100$. From figure 2, we see that the three graphs behave near optimal for RoSDP2 and far from optimal in the case of RoLP2. Besides, we can freely use any value for $0 \leq \Gamma \leq KNM$ contrarily to [2], since we do not have any violation probability. This is because the robust constraints in these models are all bounded by variable t which is obviously not fixed. The fact that we can vary $\Gamma \in [0, KNM]$ without having violation probabilities is positive because the major variation occurs for small values of Γ according to figure 2. Tables 8, 9 and 10 show results for $\varepsilon = 1\%$, $\varepsilon = 5\%$ and $\varepsilon = 10\%$ respectively for a fixed value of $\Gamma = 0.3KNM$. We set Γ to this value since we know from figure 2 that for these values of Γ the optimal and lower bounds remains constant and so we just vary ε now. These three tables show the optimum solution of RoIP2, feasible integer solutions when using the same greedy heuristic shown in figure 1, the lower bounds for the linear programming RoLP2 and RoSDP2 relaxations, and the cpu time in seconds for both relaxations. It also shows the gaps for RoLP2 and RoSDP2 which are calculated as $Gap_{SDP} = \left[\frac{RoIP2-RoSDP2}{RoSDP2}\right]$ and $Gap_{LP} = \left[\frac{RoIP2-RoLP2}{RoLP2}\right]$. Notice here that constraints in (4.54) are formed with variables in both sides, so it is not hard to adjust them suitably to get feasible solutions within the greedy heuristic. To generate feasible solutions we again assign randomly N_k sub-carriers for each user such that $\sum_k N_k = N$ and we put $R_k = N_k T$ where T is randomly generated to be between $1 \leq T \leq M$. If we compute, from tables 8, 9 and 10 the averages gaps from the optimal solutions for RoSDP2 and RoLP2 relaxations over the 36 instances, we get a tightness of 1.33% and 34.89% for $\varepsilon = 1\%$, 1.02%, 33.08% for $\varepsilon = 5\%$ and 1.12%, 34.61% for $\varepsilon = 10\%$. This clearly shows that lower bounds for RoSDP2 are almost optimal and for RoLP2 are far from being good. Now, if we compute, for the three tables, the averages for the gaps of SDP for $N = 32$ and $N = 68$ respectively, we get 2.05%, 0.61% for

Figure 2: OFDMA Bertsimas Robust Approach varying Γ

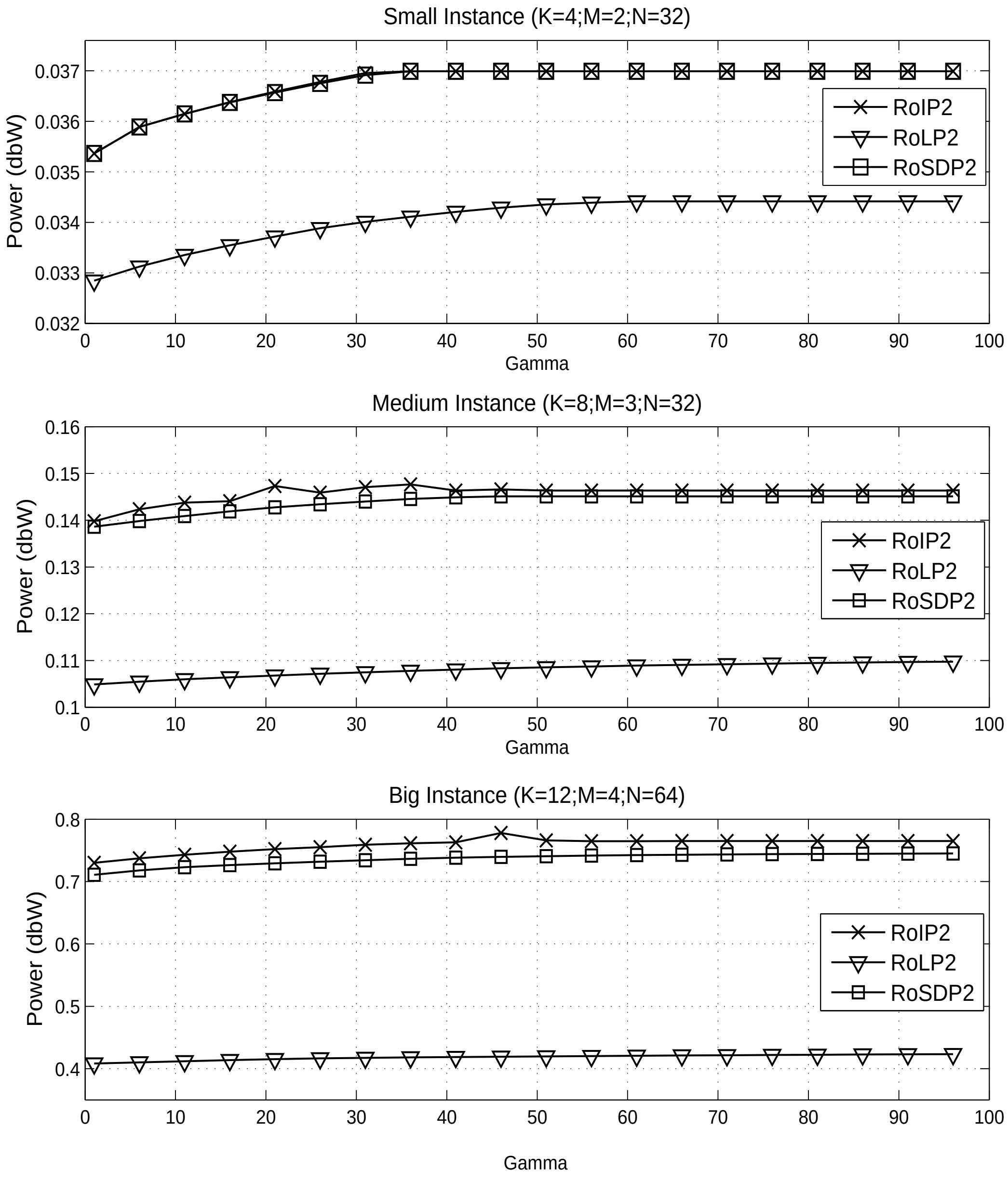

Table 8: Results for Bertsimas Approach with $\varepsilon = 1\%$ and $\Gamma = 0.3KNM$

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	0.0442	0.0507	0.0386	0.2660	0.0442	0.0442	28.39	0	0.1459
2	0.1469	0.1659	0.1266	0.3130	0.1595	0.1462	82.84	0.0044	0.1598
3	0.1647	0.2232	0.1376	0.3600	0.1647	0.1647	185.47	0	0.1966
4	0.1358	0.1888	0.1164	0.4060	0.1358	0.1357	381.28	4.6e-5	0.1658
5	0.0956	0.1178	0.0860	0.4220	0.1108	0.0950	573.27	0.0057	0.1116
6	0.0787	0.2724	0.0658	0.5000	0.0884	0.0782	921.58	0.0066	0.1968
7	0.1909	0.2372	0.1404	0.3280	0.2083	0.1898	69.99	0.0058	0.3596
8	0.2228	0.3112	0.1631	0.3910	0.2340	0.2208	178.48	0.0091	0.3664
9	0.1278	0.1995	0.0996	0.4220	0.1798	0.1254	347.91	0.0193	0.2828
10	0.0949	0.1252	0.0672	0.5000	0.1198	0.0912	804.46	0.0404	0.4119
11	0.2087	0.2278	0.1298	0.6560	0.2217	0.1988	2019.00	0.0498	0.6084
12	0.1403	0.4997	0.1050	0.6560	0.1958	0.1373	2221.00	0.0221	0.3362
13	0.1100	0.2229	0.0906	0.3440	0.1100	0.1098	116.23	0.0019	0.2136
14	0.9802	1.7197	0.6287	0.4850	1.3831	0.9247	468.32	0.0600	0.5591
15	0.3481	0.7536	0.2640	0.5620	0.3859	0.3414	720.26	0.0197	0.3185
16	0.5161	0.8337	0.2742	0.7180	0.6594	0.5044	1591.20	0.0234	0.8825
17	0.4349	0.7951	0.2547	0.7340	0.6409	0.4078	3445.70	0.0666	0.7074
18	0.1775	0.2971	0.1171	0.8590	0.2384	0.1716	4305.30	0.0344	0.5160
19	0.1666	0.2720	0.1401	0.4060	0.1998	0.1654	227.86	0.0078	0.1894
20	0.2611	0.3484	0.2480	0.5160	0.2611	0.2611	472.43	0	0.0531
21	0.0694	0.0827	0.0626	0.5470	0.0694	0.0694	1869.50	0	0.1091
22	0.1525	0.1917	0.1393	0.6250	0.1772	0.1515	1807.40	0.0066	0.0951
23	0.1332	0.1500	0.1129	0.7810	0.1448	0.1315	4463.10	0.0127	0.1802
24	0.0514	0.0668	0.0452	0.8910	0.0514	0.0513	4640.80	8.4e-4	0.1372
25	0.3332	0.4487	0.2373	0.5780	0.3632	0.3288	490.38	0.0132	0.4037
26	0.1391	0.3208	0.1235	0.5930	0.1391	0.1391	973.73	0	0.1262
27	0.1334	0.1921	0.1010	0.6720	0.1603	0.1324	1980.10	0.0074	0.3206
28	0.1839	0.2968	0.1364	0.9070	0.2284	0.1826	4536.70	0.0071	0.3482
29	0.1397	0.3157	0.1044	1.0630	0.1923	0.1385	10561.00	0.0081	0.3371
30	0.1413	0.2382	0.1072	2.7820	0.1536	0.1403	13341.00	0.0071	0.3182
31	0.7804	1.5851	0.6144	0.6560	0.7804	0.7804	999.43	0	0.2701
32	0.5544	0.9574	0.3144	0.9380	0.6772	0.5534	3241.60	0.0018	0.7631
33	0.2584	0.4334	0.1556	0.9850	0.2635	0.2565	6378.90	0.0074	0.6610
34	0.2792	0.5171	0.1809	1.3120	0.2890	0.2775	15243.00	0.0061	0.5440
35	0.2508	0.4274	0.1532	1.3120	0.2598	0.2500	15250.00	0.0033	0.6371
36	0.1746	0.3095	0.1142	1.6250	0.1945	0.1709	23884.00	0.0221	0.5295

Table 9: Results for Bertsimas Approach with $\varepsilon = 5\%$ and $\Gamma = 0.3KNM$

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	0.0756	0.0862	0.0679	0.2650	0.0756	0.0756	24.03	0	0.1136
2	0.0702	0.0810	0.0585	0.3130	0.0702	0.0702	76.91	0	0.1993
3	0.1598	0.1982	0.1435	0.3750	0.1618	0.1587	131.42	0.0073	0.1137
4	0.1030	0.1269	0.0925	0.4380	0.1030	0.1030	278.39	0	0.1132
5	0.1324	0.1594	0.1141	0.4370	0.1414	0.1296	613.61	0.0210	0.1600
6	0.1310	0.1622	0.1097	0.5000	0.1424	0.1283	652.98	0.0208	0.1937
7	0.3104	0.5502	0.2701	0.3280	0.3104	0.3104	45.25	0	0.1493
8	0.1584	0.2201	0.1264	0.3750	0.2030	0.1567	189.28	0.0107	0.2528
9	0.2988	0.4451	0.2421	0.5000	0.3206	0.2974	358.29	0.0047	0.2343
10	0.1578	0.2883	0.1203	0.5940	0.1739	0.1570	636.03	0.0046	0.3119
11	0.2125	0.3685	0.1638	0.6400	0.2798	0.2033	1486.59	0.0451	0.2974
12	0.1653	0.2325	0.1071	0.6100	0.1855	0.1580	2415.30	0.0463	0.5440
13	0.4634	1.1995	0.3351	0.3430	0.4634	0.4634	93.83	0	0.3831
14	0.2279	0.4283	0.1655	0.4690	0.2415	0.2258	384.71	0.0094	0.3773
15	0.2052	0.4471	0.1424	0.5310	0.2339	0.2025	546.79	0.0135	0.4411
16	0.6224	1.0015	0.3777	0.7500	0.8803	0.6001	1425.60	0.0372	0.6478
17	0.2132	0.3410	0.1244	0.7660	0.3163	0.2076	1878.40	0.0272	0.7141
18	0.2775	0.4197	0.1594	0.8910	0.3294	0.2719	3443.90	0.0206	0.7410
19	0.0941	0.1058	0.0888	0.3910	0.0941	0.0940	155.91	0.0011	0.0586
20	0.1218	0.1440	0.1114	0.4540	0.1218	0.1218	467.69	0	0.0935
21	0.1270	0.1517	0.1147	0.5940	0.1270	0.1270	995.29	0	0.1070
22	0.1599	0.1926	0.1435	0.7030	0.1599	0.1598	1936.00	7.1e-4	0.1142
23	0.1321	0.1612	0.1161	0.7970	0.1321	0.1321	4351.80	0	0.1374
24	0.1130	0.1355	0.0981	1.0940	0.1130	0.1126	7342.40	0.0033	0.1524
25	0.6340	1.0030	0.5650	0.4850	0.6459	0.6330	660.91	0.0016	0.1221
26	0.1242	0.1896	0.0973	0.5630	0.1355	0.1238	1369.80	0.0029	0.2761
27	0.1644	0.2329	0.1204	0.7040	0.1721	0.1638	3186.10	0.0036	0.3655
28	0.3199	0.5007	0.2434	1.1870	0.3516	0.3147	5321.70	0.0167	0.3147
29	0.1695	0.2514	0.1289	1.0930	0.2082	0.1678	7584.10	0.0096	0.3147
30	0.1469	0.2476	0.1114	1.2970	0.1624	0.1458	13485.00	0.0071	0.3182
31	0.8113	1.6479	0.6388	0.6720	0.8113	0.8113	890.13	0	0.2701
32	0.4561	0.7232	0.2485	1.0000	0.4561	0.4561	2619.50	0	0.8356
33	0.3227	0.5403	0.1963	1.1410	0.3749	0.3191	6878.60	0.0111	0.6441
34	0.5607	0.9884	0.3435	1.3290	0.6537	0.5511	20934.00	0.0174	0.6325
35	0.2608	0.4444	0.1593	1.4370	0.2860	0.2599	18737.00	0.0033	0.6370
36	0.1816	0.3218	0.1187	1.7190	0.2566	0.1776	23060.00	0.0221	0.5295

Table 10: Results for Bertsimas Approach with $\varepsilon = 10\%$ and $\Gamma = 0.3KNM$

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
1	1.1137	1.5705	0.9960	0.2820	1.2008	1.0187	25.96	0.0933	0.1182
2	0.0985	0.1219	0.0863	0.3120	0.0985	0.0980	61.88	0.0048	0.1409
3	0.2177	0.2754	0.2067	0.3590	0.2177	0.2175	174.74	8.1e-4	0.0533
4	0.1003	0.1302	0.0831	0.4070	0.1063	0.0992	270.66	0.0103	0.2064
5	0.0930	0.1112	0.0830	0.4530	0.0930	0.0929	543.77	3.1e-4	0.1203
6	0.0750	0.0800	0.0611	0.5780	0.0750	0.0750	733.90	0	0.2281
7	0.2187	0.4440	0.1834	0.3290	0.2187	0.2187	62.39	0	0.1925
8	0.1070	0.1942	0.0957	0.3590	0.1070	0.1068	154.50	0.0021	0.1184
9	0.2100	0.3291	0.1590	0.4690	0.2248	0.2091	413.40	0.0041	0.3208
10	0.3087	0.3683	0.1996	0.5470	0.3476	0.3060	787.60	0.0087	0.5469
11	0.2026	0.2790	0.1438	0.7500	0.2579	0.1954	1151.50	0.0366	0.4087
12	0.2010	0.2522	0.1324	0.8590	0.2381	0.1994	2142.40	0.0077	0.5178
13	0.6247	1.6657	0.5853	0.3750	0.6247	0.6247	107.42	0	0.0673
14	0.2411	0.4265	0.1468	0.4530	0.2620	0.2400	317.38	0.0042	0.6420
15	0.4412	0.6027	0.2727	0.6720	0.4676	0.4379	575.32	0.0077	0.6181
16	0.3086	0.5254	0.1932	0.7180	0.3142	0.3018	1177.60	0.0224	0.5972
17	0.1907	0.3585	0.1293	0.7500	0.2446	0.1869	1797.00	0.0204	0.4750
18	0.3732	0.6086	0.2078	1.0310	0.4373	0.3442	4202.70	0.0840	0.7960
19	0.1214	0.1485	0.1129	0.3750	0.1214	0.1214	207.80	0	0.0752
20	0.1654	0.2084	0.1550	0.5160	0.1654	0.1654	510.63	0	0.0674
21	0.0942	0.1096	0.0827	0.6090	0.0942	0.0942	1230.30	0	0.1387
22	0.0958	0.1147	0.0838	0.7660	0.0958	0.0956	2114.30	0.0023	0.1428
23	0.0984	0.1854	0.0821	1.0310	0.1044	0.0981	3289.20	0.0039	0.1983
24	0.0708	0.0872	0.0619	0.9530	0.0817	0.0698	4494.00	0.0151	0.1448
25	0.1666	0.2754	0.1512	0.4370	0.1802	0.1650	396.85	0.0098	0.1017
26	0.1509	0.2605	0.1305	0.6400	0.1584	0.1506	872.86	0.0020	0.1563
27	0.1019	0.1542	0.0780	0.8280	0.1019	0.1019	3172.50	0	0.3061
28	0.1607	0.2397	0.1241	1.3280	0.1607	0.1607	6435.00	0	0.2946
29	0.2071	0.2827	0.1612	1.3120	0.2071	0.2069	7815.50	0.0012	0.2849
30	0.1659	0.2411	0.1231	1.5310	0.2243	0.1641	13188.00	0.0115	0.3479
31	0.2833	0.6412	0.1996	0.6250	0.3253	0.2829	736.64	0.0014	0.4192
32	0.3788	0.5551	0.2109	1.4370	0.3788	0.3784	2014.70	8.2e-4	0.7956
33	0.1189	0.2018	0.0807	1.1090	0.1189	0.1186	4627.70	0.0030	0.4734
34	0.3244	0.4350	0.2140	1.7030	0.3649	0.3202	12885.00	0.0132	0.5164
35	0.2423	0.3403	0.1279	2.3280	0.2676	0.2392	18143.00	0.0129	0.8949
36	0.3616	0.4886	0.1870	2.3750	0.3892	0.3539	30298.00	0.0220	0.9343

$\varepsilon = 1\%$; 1.49%, 0.5584% for $\varepsilon = 5\%$ and 0.85%, 0.27% for $\varepsilon = 5\%$. We observe that while the bigger the ratio N/K is, the better the gain that can be reached. On the other hand, when using the same greedy heuristic just to approximate integer feasible solutions, we get better results for RoSDP2 than for RoLP2. Now, if we compute for each instance of tables 8, 9, 10 the ratio $\left[\frac{feasible-optimum}{optimum}\right] * 100$ and calculate the average over all instances, for RoSDP2 and RoLP2 respectively, we get 14.22%, 67.24% for $\varepsilon = 1\%$, 10.48%, 53.08% for $\varepsilon = 5\%$ and 7.12%, 51.13% for $\varepsilon = 10\%$. Unfortunately for RoSDP2 and RoLP2, it is harder to get the optimal solution with the greedy heuristic by simple randomizing as we did for Kouvelis and Yu approach for the special case when $\sum_k^K R_k = MN$, so we omit these results. The drawback for RoSDP2 compared to RoLP2 is that Csdp requires much more time than Cplex to get the solutions. For example, if we take the averages over all the instances of the three tables 8, 9, 10, for SDP and LP respectively, the cpu time in seconds are 3603.4 and 0.7546. We do not take an average for each table in this case, because the number of constraints and variables are the same.

Table 11: Conservatism level v/s optimality for both Robust approaches

Uncertainty Degree		Scenario Approach			Interval Approach			Approach with Higher Conservatism
S	Γ	RoIP1	RoSDP1	RoLP1	RoIP2	RoSDP2	RoLP2	
Instance 1 using values $\mathcal{P}^c_{k,n} + \varepsilon \mathcal{P}^c_{k,n} \cdot (1 + 0.15\triangle_{k,n})$ for the scenarios								
	5				0.2465	0.2448	0.1487	Scenario
5	50	0.2572	0.2555	0.1560	0.2561	0.2544	0.1527	Scenario
	100				0.2561	0.2544	0.1546	Scenario
	5				0.2465	0.2448	0.1487	Scenario
10	50	0.2572	0.2555	0.1560	0.2561	0.2544	0.1527	Scenario
	100				0.2561	0.2544	0.1546	Scenario
	5				0.2465	0.2448	0.1487	Scenario
50	50	0.2573	0.2556	0.1560	0.2561	0.2544	0.1527	Scenario
	100				0.2561	0.2544	0.1546	Scenario
Instance 2 using values $\mathcal{P}^c_{k,n} + \varepsilon \mathcal{P}^c_{k,n} \cdot (1 + 0.01\triangle_{k,n})$ for the scenarios								
	5				0.9674	0.9350	0.5345	Scenario
5	50	1.0028	0.9700	0.5583	1.0026	0.9697	0.5498	Scenario
	100				1.0026	0.9697	0.5563	Scenario
	5				0.9674	0.9350	0.5345	Scenario
10	50	1.0029	0.9700	0.5583	1.0026	0.9697	0.5498	Scenario
	100				1.0026	0.9697	0.5563	Scenario
	5				0.9674	0.9350	0.5345	Scenario
50	50	1.0029	0.9700	0.5583	1.0026	0.9697	0.5498	Scenario
	100				1.0026	0.9697	0.5563	Scenario
Instance 3 using values $\mathcal{P}^c_{k,n} + \varepsilon \mathcal{P}^c_{k,n}$ for the scenarios								
	5				0.9674	0.9350	0.5345	Scenario
5	50	1.0026	0.9697	0.5582	1.0026	0.9697	0.5498	Equal except for RoLP2
	100				1.0026	0.9697	0.5563	Equal except for RoLP2
	5				0.9674	0.9350	0.5345	Scenario
10	50	1.0026	0.9697	0.5582	1.0026	0.9697	0.5498	Equal except for RoLP2
	100				1.0026	0.9697	0.5563	Equal except for RoLP2
	5				0.9674	0.9350	0.5345	Scenario
50	50	1.0026	0.9697	0.5582	1.0026	0.9697	0.5498	Equal except for RoLP2
	100				1.0026	0.9697	0.5563	Equal except for RoLP2
Instance 4 using values $\mathcal{P}^c_{k,n} + \varepsilon \mathcal{P}^c_{k,n} \cdot (\triangle_{k,n})$ for the scenarios								
	5				0.2735	0.2680	0.1544	Scenario
5	50	0.2763	0.2705	0.1568	0.2833	0.2776	0.1580	Interval
	100				0.2833	0.2776	0.1596	Interval
	5				0.2735	0.2680	0.1544	Scenario
10	50	0.2774	0.2721	0.1573	0.2833	0.2776	0.1580	Interval
	100				0.2833	0.2776	0.1596	Interval
	5				0.2735	0.2680	0.1544	Scenario
50	50	0.2782	0.2727	0.1575	0.2833	0.2776	0.1580	Interval
	100				0.2833	0.2776	0.1596	Interval

Table 12: Average Results for OFDMA Kouvelis and Yu Robust Approach

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
S=5									
1	0.2134	0.2785	0.1842	0.3546	0.2411	0.2115	10.64	0.0086	0.1518
4	0.1864	0.2279	0.1498	0.6751	0.2057	0.1774	73.67	0.0510	0.2453
7	0.4993	0.9315	0.3679	0.4718	0.5860	0.4878	21.02	0.0260	0.3619
8	0.4801	0.8796	0.3644	0.5811	0.6975	0.4682	40.17	0.0267	0.3578
9	0.4878	0.6871	0.3317	0.8156	0.5775	0.4750	85.72	0.0365	0.4663
13	1.1187	2.6614	0.7096	0.5781	1.5562	1.0604	26.41	0.0403	0.5880
14	0.9079	1.8517	0.5793	0.7843	1.3095	0.8591	60.95	0.0473	0.6314
22	0.1747	0.2347	0.1455	1.6578	0.2076	0.1696	550.62	0.0280	0.2021
S=10									
1	0.3604	0.6805	0.3004	0.4124	0.3873	0.3421	19.84	0.0474	0.1993
4	0.2272	0.3373	0.1881	0.8173	0.2884	0.2153	97.14	0.0522	0.2060
7	0.8213	1.8327	0.6291	0.5015	0.9735	0.8053	29.72	0.0254	0.3221
8	0.6938	1.2534	0.5137	0.7846	0.9827	0.6546	48.19	0.0635	0.3939
9	0.4788	0.9194	0.3242	1.0078	0.6430	0.4562	112.11	0.0524	0.4612
13	1.0098	2.8810	0.6231	0.6346	1.6526	0.9624	35.99	0.0561	0.5989
14	0.9339	2.5242	0.5778	1.0264	1.6186	0.8572	69.66	0.0854	0.6088
22	0.2481	0.3637	0.2096	2.0905	0.3000	0.2402	737.44	0.0315	0.1806
S=15									
1	0.4454	0.7269	0.3539	0.4063	0.4900	0.4175	22.98	0.0695	0.2711
4	0.3253	0.4902	0.2559	0.9732	0.4121	0.2997	124.46	0.0827	0.2679
7	1.0362	2.0306	0.7060	0.5547	1.3475	0.9890	36.01	0.0545	0.4619
8	0.9425	1.8301	0.6338	0.8782	1.1966	0.8732	60.66	0.0720	0.5134
9	0.6785	1.5643	0.4778	1.2172	0.9139	0.6369	128.34	0.0657	0.4221
13	2.0374	5.2355	1.3003	0.7516	2.8142	1.9073	42.62	0.0749	0.5723
14	1.4590	3.1972	0.7711	1.2985	2.0893	1.3376	99.60	0.0853	0.9084
22	0.2783	0.5617	0.2279	3.2282	0.3431	0.2654	960.04	0.0474	0.2244

Table 13: Average Results for OFDMA Bertsimas and Sim Robust Approach

Instance	LP				SDP			Gaps	
	optimum	feasible	lower bound	time	feasible	lower bound	time	SDP	LP
$\varepsilon = 1\%$									
1	0.0745	0.0966	0.0700	0.2685	0.0745	0.0745	28.24	0	0.0687
8	0.2229	0.3616	0.1729	0.3955	0.2933	0.2214	197.02	0.0062	0.3375
9	0.1478	0.2019	0.1062	0.4547	0.1891	0.1451	437.16	0.0155	0.3976
10	0.1624	0.2273	0.1190	0.6094	0.2085	0.1603	940.17	0.0139	0.3859
11	0.2070	0.2816	0.1466	0.6563	0.2608	0.2022	1436.10	0.0240	0.4205
12	0.1769	0.2541	0.1211	0.7219	0.2157	0.1713	2365.30	0.0348	0.4694
14	0.4134	0.6525	0.2359	1.2766	0.5620	0.4043	399.94	0.0197	0.7790
$\varepsilon = 5\%$									
1	0.0772	0.1409	0.0697	0.2796	0.0941	0.0761	28.71	0.0109	0.1063
8	0.1928	0.2795	0.1476	0.4092	0.2551	0.1905	199.40	0.0159	0.3281
9	0.1905	0.2616	0.1370	0.5311	0.2396	0.1870	453.75	0.0169	0.3979
10	0.1943	0.2732	0.1391	0.5720	0.2494	0.1923	858.82	0.0112	0.4046
11	0.1630	0.2439	0.1157	0.6782	0.2082	0.1584	1448.80	0.0279	0.4113
12	0.2198	0.3019	0.1505	0.7843	0.2698	0.2076	2409.60	0.0426	0.4485
14	0.4781	0.7889	0.2719	0.5140	0.6236	0.4722	375.29	0.0130	0.7559
$\varepsilon = 10\%$									
1	0.0812	0.1053	0.0762	0.2845	0.0897	0.0811	28.31	1.6e-4	0.0687
8	0.1988	0.2862	0.1461	0.4328	0.2504	0.1974	186.19	0.0080	0.3571
9	0.2301	0.3513	0.1736	0.5187	0.2993	0.2279	463.45	0.0121	0.3558
10	0.1781	0.2835	0.1265	0.5453	0.2241	0.1719	763.17	0.0330	0.4060
11	0.2382	0.3776	0.1739	0.7014	0.3150	0.2328	1543.10	0.0236	0.3917
12	0.1986	0.2575	0.1382	1.3109	0.2436	0.1913	2333.60	0.0319	0.4299
14	0.4364	0.6891	0.2501	0.5406	0.5770	0.4310	396.13	0.0140	0.7452

6.4 Some Comparison between the two Robust Approaches

In order to compare both approaches, models RoIP1, RoSDP1, RoLP1 with models RoIP2, RoSDP2 and RoLP2, we run four small instances of size N=32, K=8, M=4 using one sample power matrix $\mathcal{P}^c_{k,n}$. We set $\varepsilon = 5\%$ and generate randomly $S = 5, 10, 50$ scenarios from the same matrix $\mathcal{P}^c_{k,n}$. For the first two instances, we generate scenario matrices with elements equal to $\mathcal{P}^c_{k,n} + \varepsilon\mathcal{P}^c_{k,n} \cdot (1+\delta\triangle_{k,n})$, where δ is a fixed percentage and $\triangle_{k,n}$ is a random number uniformly distributed between zero and one, then we begin to decrease these values, so for the third instance we use $\mathcal{P}^c_{k,n} + \varepsilon\mathcal{P}^c_{k,n}$ and for the last instance we put elements equal to $\mathcal{P}^c_{k,n} + \varepsilon\mathcal{P}^c_{k,n} \cdot (\triangle_{k,n})$. We vary Γ in three levels 5, 50 and 100 since we know from Figure 1 that after these values the solutions remains constant, besides the greater variation occurs for small values of Γ. Table 11 shows the results for these four instances. We observe for instances 1 and 2 that the scenario approach is more conservative which means that the optimal values are greater than the optimal values obtained with the interval approach. Instance 3 is a special case since all scenarios are equal and the problem could be reduced to use only one scenario. Besides, for the third instance we start having less conservative optimal values compare to the interval approach. Finally instance 4 shows that it is possible to have lower optimal values than the interval approach and this is a consequence of proposition 1 since for values of $\Gamma \geq 100$ the solutions are constantly the same and all the scenarios are generated using values of $\varepsilon'_{k,n} = \varepsilon \cdot \triangle_{k,n} < \varepsilon$. Even though, we can get less conservative solutions, we can not ensure in any case, that we are decreasing the optimal input parameters as the interval Bertsimas approach does.

We also compare some average results for the two robust approaches, so we choose some few small size instances from tables 1 and 6 and calculate the averages over 25 independent sample power matrices. Results are shown in tables 12 and 13. Taking the average gaps for RoLP1, RoSDP1, we get 37.55%, 3.3% for $S = 5$; 37.13%, 5.17% for $S = 10$ and 45.51%, 6.9% for $S = 15$, respectively. On the other hand, if we do the same for RoLP2 and RoSDP2 from table 13, we get 40.83%, 1.63% for $\varepsilon = 1\%$; 40.75%, 1.97% for $\varepsilon = 5\%$ and 39.34%, 1.75% for $\varepsilon = 10\%$, respectively. So we say the tightness behavior is better for RoSDP2 model than for RoSDP1 when incrementing the uncertainty level since we observe the same detrimental for RoSDP1 that we see in the averages of the above tables.

7 Conclusions

We have proposed in this paper, two robust binary quadratic formulations for wireless DL OFDMA networks when using adaptive modulation. One which is based on a scenario uncertainty approach from Kouvelis and Yu, and another one based on an interval uncertainty approach from Bertsimas and Sim. We have derived for both models two semidefinite relaxations and by numerical results we have obtained a near optimal average tightness of 4.12% under the scenario approach and 1.15% under the interval uncertainty approach compared to the optimal solution of the problem. Besides, we can say from the experimental results that the tightness performance is better for the interval approach than for the scenarios approach which degrades when incrementing the number of scenarios. We have also compared both approaches for some same instances and we conclude that the scenario approach is in general more conservative than the interval approach, however under certain conditions, it is possible to get less conservative robust solutions using the scenario approach. Finally we conclude that while the bigger the ratio N/K is, the bigger the gain that can be reached under both approaches.

References

[1] Kouvelis and G. Yu, *“Robust discrete optimization and its applications”*, Kluwer Academic Publishers, 1997.

[2] Dimitris Bertsimas, Melvyn Sim, *"The Price of Robustness"*, Operations Research, Vol. 52, No. 1, JanuaryFebruary 2004, pp. 3553.

[3] IEEE 802.16 WG, *"Air Interface for Fixed Broadband Wireless Access Systems"*, IEEE Standard for Local and Metropolitan Area Networks, December 2001.

[4] Hassan Yaghoobi, *"Scalable OFDMA Physical Layer in IEEE 802.16 WirelessMAN"*, Intel Technology Journal, Volume 8, Issue 3, 2004.

[5] Amzallag, D. Armarnik, T. Livschitz, M. Raz, D., *"Multi-Cell Slots Allocation in OFDMA Systems"*, Mobile and Wireless Communications Summit, 2007. 16th IST, ISBN: 963-8111-66-6, 2007.

[6] D. Kivanc, G. Li, and H. Liu, *"Computationally efficient bandwidth allocation and power control for an OFDMA"*, IEEE Transactions on Wireless Communications, vol. 2, no. 6, pp. 11501158, 2003.

[7] Sanam Sadr, Alagan Anpalagan and Kaamran Raahemifar, *"A Novel Subcarrier Allocation Algorithm for Multiuser OFDM System With Fairness: User's Perspective"*, Proceedings of the 66th IEEE Vehicular Technology Conference, VTC Fall 2007.

[8] R. Fortet, *"Applications de lalgebre de boole en recherche operationelle"*, Revue Francaise de Recherche Operationelle, 4:1726, 1960.

[9] Goemans, M. X. and Williamson, D.P. , *"Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming "*, J. ACM.,42:1115-1145, 1995.

[10] Helmberg, C., *"Semidefinite Programming for Combinatorial Optimization"*, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin, October 2000.

[11] Zukang Shen, Jeffrey G. Andrews, and Brian L. Evans, *"Short Range Wireless Channel Prediction Using Local Information"*, Signals, Systems and Computers, Conference Record of the Thirty-Seventh Asilomar Conference on, On page(s): 1147- 1151 Vol.1, ISBN: 0-7803-8104-1, 2003.

[12] Helmberg, C., Rendl, F., and Weismantel, R. *"A Semidefinite Programming Approach to the Quadratic Knapsack Problem"*, Journal of combinatorial optimization, ISSN 1382-6905, vol. 4, no2, pp. 197-215 (1 p.1/4) ,2000.

[13] CSDP is a Brian Borchers C code project of COIN-OR for SDP. It is a library of routines that implements a predictor corrector variant of the SDP algorithm of Helmberg, Rendl, Vanderbei, and Wolkowicz. https://projects.coin-or.org/Csdp/.

[14] Chadi Tarhini, Tijani Chahed , *On capacity of OFDMA-based IEEE802.16 WiMAX including Adaptive Modulation and Coding (AMC) and inter-cell interference*, Local and Metropolitan Area Networks, 15th IEEE Workshop on, (2007), pp. 139-144.

[15] Yingwei Yao and Georgios B. Giannakis, *"Rate-Maximizing Power Allocation in OFDM Based on Partial Channel Knowledge"*, IEEE Transactions on Wireless Communications, Vol. 4, No. 3, MAY 2005.

[16] Ian C. Wong and Brian L. Evans, *"Optimal OFDMA Subcarrier, Rate, and Power Allocation for Ergodic Rates Maximixation with Imperfect Channel Knowledge"*, Acoustics, Speech and Signal Processing, ICASSP, IEEE International Conference on, ISBN: 1-4244-0727-3, 2007.

[17] Aharon Ben-Tal and Arkadi Nemirovski, *"Robust convex optimization"*, Math. Oper. Res., 23 769805, 1998.

[18] Aharon Ben-Tal and Arkadi Nemirovski, *"Robust solutions to uncertain programs"*,Oper. Res. Lett. 25 113, 1999.

[19] Aharon Ben-Tal and Arkadi Nemirovski, *"Robust solutions of Linear Programming problems contaminated with uncertain data"*, Mathematical Programming, Springer Berlin-Heidelberg, Volume 88, Number 3-septembre 2000, Pages 411-424.

[20] El-Ghaoui, L., H. Lebret., *"Robust solutions to least-square problems to uncertain data matrices"*, SIAM J. Matrix Anal. Appl. 18 10351064, 1997.

[21] El-Ghaoui, L., F. Oustry, H. Lebret, *"Robust solutions to uncertain semidefinite programs"*, SIAM J. Optim. 9 3352, 1998.

[22] P. Vincke, *"Robust solutions and Methods in decision-aid"*, Journal of Multicriteria Decision Analysis, Vol. 8, 1999, Pages 181-187.

[23] M. Ergen, S. Coleri, and P. Varaiya, *QoS aware adaptive resource allocation techniques for fair scheduling in OFDMA based broadband wireless access systems*, IEEE Transactions on Broadcasting, vol. 49, no. 4, (2003).

[24] Stephen Boyd, Lieven Vandenberghe, *"Convex Optimization"*, Cambridge University Press, ISBN 0-521-83378-7, 2008.

[25] CVX was designed by Michael Grant and Stephen Boyd, with input from Yinyu Ye. It is a Matlab-based modeling system for convex optimization, http://www.stanford.edu/ boyd/cvx/.

[26] J. Mulvey, R. Vanderbei and S. Zenios, *"Robust Optimization of Large Scale Systems"*, Operations Research, Vol. 43, no. 2, 1995.

[27] Grötschel, Lovász and Schrijver, *"Geometric Algorithms and Combinatorial Optimization"*, Springer-Verlag, Berlin, 1988.

[28] L. Lovász and A. Schrijver, *"Cones of matrices and set functions and 0-1 optimization"*, SIAM J. Optim., 1:166-190, 1991.

[29] Yu. E. Nesterov, and A. S. Nemirovsky, *"Conic formulation of a convex programming problem and duality"*, Optim. Methods Softw, 1:95-115, 1992.

[30] Yu. E. Nesterov, and A. S. Nemirovsky, *"Interior Point Polynomial Algorithms in Convex Programming"*, SIAM Publications, SIAM Philadelphia, USA, 1994.

[31] F. Alizadeh, *"Interior point methods in semidefinite programming with applications to combinatorial optimization"*, SIAM J. Optim., 5:13-51, 1995.

[32] N. K. Karmarkar, *"A new polynomial-time algorithm for linear programming"*, Combinatorica, 4:373-395, 1984.

[33] Soyster, A. L., *"Convex programming with set-inclusive constraints and applications to inexact linear programming"*, Operations Research, 21-5, pp. 11541157, 1973.