INF421 PI: USER SCHEDULING IN 5G

MARCEAU COUPECHOUX

1. Introduction

In 5G, an antenna transmits data packets to smartphones (or users) through a wireless medium, which is divided into a set of frequency channels. Figure 1 is an example of simultaneous transmission towards three users using three channels.

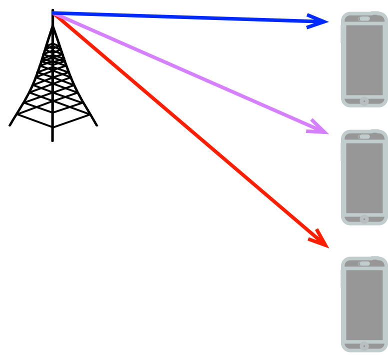

Figure 1. Wireless packet scheduler transmitting data packets simultaneously to three users over three (frequency) channels.

The higher the power dedicated to a user, the higher the data rate it can experience. The exact dependence between power and data rate is however user and channel specific. With the same transmit power, a user close to the antenna will enjoy for example a higher data rate than a user far away. A wireless packet scheduler is thus responsible to allocated channels to users and to divide the total power budget of the antenna among the available channels. The goal of this project is to design optimal packet schedulers in this context.

This subject as well as test files are available on this page:
`https://marceaucoupechoux.wp.imt.fr/enseignement/english-inf421-pi/`

2. Problem Formulation

We consider a system with a set $\mathcal{K}$ of K users to be served over a set $\mathcal{N}$ of N channels. Every channel shall be used to serve a single user and cannot be left unallocated for efficiency reasons; a user can be served using several channels; however some users may not be served. The scheduler chooses a transmit power $p_{k,n}$, $k \in \mathcal{K}$, $n \in \mathcal{N}$ to serve user k on channel n. If user k is not served on channel n, we have $p_{k,n} = 0$. When user k is served over channel n with power $p_{k,n}$, its

Date: 29 Nov. 2019.

data rate is $r_{k,n} = u_{k,n}(p_{k,n})$, where the function $u_{k,n}$ is called the *rate utility function* of user k on channel n. This utility function is assumed to be known by the scheduler. In practical systems, $u_{k,n}$ is a non-decreasing step function that takes a finite number of non-zero values, say M for all k and n and such that $u_{k,n}(0) = 0$ for all k and n. To fix notations, we thus define:

$$u_{k,n}(p_{k,n}) \quad = \quad \begin{cases} 0 & \text{if } p_{k,n} < p_{k,1,n} \\ r_{k,1,n} & \text{if } p_{k,1,n} \leq p_{k,n} < p_{k,2,n} \\ \cdots & \cdots \\ r_{k,M,n} & \text{if } p_{k,1,M} < p_{k,n} \end{cases} \tag{1}$$

The task of the scheduler is to allocate channels to users and transmit powers to users so as to maximize the sum data rate of the system under the constraint of a total transmit power budget p and the constraint of having exactly one user served per channel. For simplicity, we assume that all coefficients $r_{k,m,n}$, $p_{k,m,n}$ and p are non-negative integers.

Question 1. Consider the binary variables $x_{k,m,n} \in \{0, 1\}$, $k \in \mathcal{K}$, $n \in \mathcal{N}$, $m = 1, ..., M$ and formulate the problem as an integer linear program (ILP).

This ILP is known to be NP hard. By relaxing the integrality constraint on $x_{k,m,n}$, we obtain a linear program (LP), which provides an upper bound for our problem. If the solution to the LP is integer, then we have a solution for the ILP. In the following sections, we first make some preprocessing to reduce problem instance size, then solve the LP and the IP, and at last consider an online version of the problem.

3. Preprocessing

Question 2. We want to make a quick preprocessing to check if an instance has obviously no solution and to remove triplets (k, m, n) that, if chosen, obviously prevent any solution to be feasible. Implement this preprocessing step given an instance of our problem.

Lemma 1. *For a given channel n, if $p_{k,m,n} \leq p_{k',m',n}$ and $r_{k,m,n} \geq r_{k',m',n}$ then there is an optimal solution of the ILP such that $x_{k',m',n} = 0$. We say that (k', m', n) is IP-dominated.*

Question 3. Based on Lemma 1, propose an algorithm to remove IP-dominated terms of an instance of the IP problem. Provide the pseudo-code, implement and show that a complexity of $O(NKM \log(KM))$ can be achieved.

Lemma 2. *For a given channel n, if $p_{k,m,n} < p_{k',m',n} < p_{k",m",n}$ and $r_{k,m,n} < r_{k',m',n} < r_{k",m",n}$ satisfy:*

$$\frac{r_{k",m",n} - r_{k',m',n}}{p_{k",m",n} - p_{k',m',n}} \geq \frac{r_{k',m',n} - r_{k,m,n}}{p_{k',m',n} - p_{k,m,n}} \tag{2}$$

then there is an optimal solution of the LP such that $x_{k',m',n} = 0$. We say that (k', m', n) is LP-dominated.

Question 4. Based on Lemma 2, propose an algorithm to remove LP-dominated terms of an instance of the LP problem. Provide the pseudo-code, implement and show that a complexity of $O(NKM \log(KM))$ can be achieved.

Question 5. Show how instances of test files 'test1.txt', 'test2.txt', 'test3.txt', 'test4.txt', and 'test5.txt' are reduced after the initial pre-processing (Question 2), after removing IP-dominated (Question 3) and then LP-dominated terms (Question 4).

4. Linear Program and Greedy Algorithm

Now that the instance size has been reduced, we study a greedy algorithm for the LP problem. For a given channel n, all possible pairs $(p_{k,m,n}, r_{k,m,n})$ are sorted in ascending order of $p_{k,m,n}$ and reindexed with the set $\mathcal{L} = \{1, ..., L\}$, $L = KM$. We define the incremental efficiency of choosing pair $\ell > 1$ instead of pair $\ell - 1$ for the transmission on channel n as follows:

$$e_{\ell n} \quad = \quad \frac{r_{\ell,n} - r_{\ell-1,n}}{p_{\ell,n} - p_{\ell-1,n}}. \tag{3}$$

Question 6. Based on this notion, propose a greedy algorithm to provide a solution to the LP problem. You may allow at most two variables $x_{k,m,n}$ to be fractional and if there are fractional variables, they should have the same n. Provide the pseudo-code, implement the algorithm and give its complexity.

Question 7. Run the greedy algorithm on test files 'test1.txt', 'test2.txt', 'test3.txt', 'test4.txt', 'test5.txt' and give for every scenario the optimal rate and the corresponding used power. Use a LP solver to compare your greedy algorithm with the exact solution. Compare the CPU runtime on 'test4.txt' and 'test5.txt'.

5. Algorithms for solving the ILP

As the relaxed solution cannot be implemented in practice (this would require the possibility to schedule two users on the same channel), we tackle in this section the ILP problem (when the power budget is an integer). Unfortunately, the greedy algorithm can be arbitrarily bad in this case. Although there is an improved greedy algorithm for this problem, this is only a 1/2-approximation. We are looking here for an optimal solution by first relying on Dynamic Programming (DP).

Question 8. Propose a DP algorithm to solve the IP based on subproblems with power budget less than p. Give the DP equations, the pseudo-code, implement the algorithm and derive its complexity. What is the space requirement?

An alternative DP approach is to consider subproblems of finding minimal power allocations providing a given sum data rate r less than some upper bound U for the objective function.

Question 9. Based on this idea, propose an alternative DP algorithm. Give the DP equations, the pseudo-code, implement the algorithm and derive its complexity. What is the space requirement?

Another classical way of solving IPs is called Branch-and-Bound (BB). The principle of BB is as follows. We construct a tree of subproblems, whose root corresponds to the initial problem. Each vertex v corresponds to a subproblem, which is generated from its parent in the tree by adding an additional constraint. At node v (a *branch* of the tree), the relaxed subproblem is solved in order to get an upper bound $\bar{z}_v$ of the optimal value for the subproblem. This branch is not further explored if (1) $\bar{z}_v$ is less than a current feasible solution we have already or (2) $\bar{z}_v$ is associated to an integer solution, in which case we can update the current feasible solution or (3) the relaxed subproblem is infeasible.

Question 10. Propose a BB algorithm for solving the ILP. Explain how you map the tree nodes to subproblems. Provide the pseudo-code and implement your solution. Motivate the data structures you chose. Provide the time complexity.

Question 11. Run the Dynamic Programming and Branch and Bound algorithm on test files 'test1.txt', 'test2.txt', 'test3.txt', 'test4.txt' and 'test5.txt' and give for every scenario the optimal rate and the corresponding used power. Compare the CPU runtime on 'test4.txt' and 'test5.txt'.

6. Stochastic Online Scheduling

In this section, users arrive sequentially in the system. The scheduler is no longer aware of the whole instance before taking a decision and has to take a decision each time a new user is coming. To be more precise, we assume that the scheduler is aware of the number of users K that will arrive in the system. At time $t = k$, user k arrives in the system providing to the scheduler all the pairs $(p_{k,m,n}, r_{k,m,n})$, $m = 1, ..., M$, $n = 1, ..., N$. All pairs indexed by $k' > k$ are unknown. At this time instant, the scheduler must assign the variables $x_{k,m,n}$, $m = 1, ..., M$, $n = 1, ..., N$ without being able to modify them in the sequel, i.e., at $t > k$. We assume that powers are independent and identically distributed with uniform discrete distribution on the set $\{1, 2, ..., p^{max}\}$; rates are independent and identically distributed with uniform discrete distribution on the set $\{1, 2, ..., r^{max}\}$, where p^{max} and r^{max} are positive integers. These distributions are supposed to be known by the scheduler.

Question 12. Propose an online algorithm for this problem. Give the pseudo-code and implement it. Motivate your design choices. Hint: you might be inspired by the (offline) Greedy algorithm.

Question 13. Assume $p = 100$, $p^{max} = 50$, $r^{max} = 100$, $M = 2$, $N = 4$ and $K = 10$. Run your online algorithm as well as an offline solution to the corresponding ILP over a large number of problem instances and compute the average competitive ratio. Provide the average power budget used by your online algorithm and compare to the average power budget used by the offline optimal algorithm.