EE 597 Fall 2015
Project 1

Instructor: Konstantinos Psounis
TAs: Yonglong Zhang
{yonglonz}@usc.edu
Assigned: Sep 23th, 2015
First attempt due: Sep 30th, 2015
TA feedback: Oct 2nd, 2015
Final attempt due: Oct 7th,2015

1 Problem Statement

Recall the discussion of power control over independent channels in class, where SINR (signal to interference and noise ratio) is described:

$$SINR_i = \frac{P_{T_i} g_{ii}}{\sum\limits_{\forall j, j \neq i} P_{T_j} g_{ji} + N} > \theta \tag{1}$$

We noted that the condition for n parallel transmitter(Tx) - receiver(Rx) pairs to be able to simultaneously transmit data is given by

$$G\vec{P} \succcurlyeq \theta N \vec{1} \tag{2}$$

where the matrices G and $\vec{P}$ denote the gains on each link and the power of each transmitter respectively. $\vec{1}$ represents vector of 1's. $\succcurlyeq$ refers to "element-wise greater than or equal to". Specifically,

$$[G]_{ij} = -\theta g_{ji} \quad \forall i \neq j, \; i, j \in [1, 2, \ldots, n] \tag{3}$$

$$[G]_{ii} = g_{ii} \quad \forall i \in [1, 2, \ldots, n] \tag{4}$$

N, θ denote the noise and the SINR threshold respectively. For the system of equations given by (2) to have a solution, we require that matrix G be invertible. In this assignment, the gains on each link are such that G is singular. You are required to come up with an algorithm that will partition the n Tx-Rx pairs into k sets such that the pairs in a set are able to simultaneously transmit data. You may assume that each set participates in a TDMA protocol so that the transmission of one partition does not interfere with those of another. Clearly, the partitioning enables communication between pairs at the expense of throughput. For an algorithm that partitions our Tx-Rx pairs into k sets, we define our throughput to be $1/k$. Let C be a weighting parameter, a known constant bigger than 0. Design a joint scheduling (partitioning) and power allocation algorithm to

$$\max_{k, P_T} \quad \frac{C}{k} - \sum_{i=1}^{n} P_{T_i} \tag{5}$$

In (5), you have to determine both number of partitions (k) and values of power needed to assign to all of the transmitters such that the objective function is maximized.

2 Input Data

Your algorithm will be tested against different values for n, g_{ij}, θ, C and N. They are provided in the following format:

- The file `gains.csv` contains g_{ij} in a matrix form. Specifically, the value at row i and column j corresponds to g_{ij}.
- The values of n, N, θ and C are provided in the file `params.csv`. The file contains a single row whose first, second, third and fourth elements are n, N, θ, and C respectively.

You may assume that $n \leq 100$.

3 Output

Your code should read the parameters correctly from the above values and output the following:

- A file named `pow.csv` that contains a single row whose i-th element is P_{T_i}
- A file named `partition.csv` that contains $k+1$ rows. The first row has a single element, namely k. The next k rows describe the generated partition. For example, if your second row is $1, 2, 3$ this indicates that your first partition contains the Tx-Rx pairs $1, 2, 3$.

4 Output evaluation

Evaluation file (evalP.m) and test cases are provided so that you can use to evaluate your algorithm's performance.

5 Guidelines

You may use C, C++, MATLAB, or Julia for this assignment. If you are using C/C++ make sure to test your code on `aludra.usc.edu`. Your submission must contain a file named `report.pdf` that describes the partitioning algorithm used and its rationale. In addition, the submission must contain a folder named `code` that contains all the files required to compile/run your code. This folder must contain a `README` file that adequately describes how to compile/run your code. In addition, submissions in C/C++ must include a `Makefile`. See http://mrbook.org/tutorials/make/ in case you are not familiar with makefiles.

Note that we will be using a script to parse and evaluate the output of your code. Consequently, it is vital that you adhere to the naming conventions described in this document for your output files. In addition your output files must be a valid CSV file.

Your partitions should also be *valid*. Namely, your partitions must be disjoint and their union should equal the set $\{1, 2, \ldots, n\}$. In addition, your power vector must satisfy (2) for each partition. Failure to meet any of these criteria is equivalent to submitting code that does not run/compile at all.

6 Submission guidelines

You are required to submit the project in a two-step submission procedure through Blackboard:

- The first submission is due on **Sep 30th**,11:59pm. The submission format should be the same as the final submission described in the previous section. You are encouraged to submit your project in order for coarse grained evaluation and feedback in terms of your relative performance with respected to other groups.

- We will provide feedback on **Oct 2nd** (performance comparison among groups)
- The final submission (an improved version of your algorithm) is due on **Oct 7th**,11:59pm.

The final grade is based on the final submission only.