ISYE 4133: Advanced Optimization

Swiss Chess Project

Name: Leonardo Calizaya

Introduction:

Model:

To model our integer program, we have to determine our variables that will help us model our constraints and objectives. In this case we are going to divide our variables in two: decision variables, and coefficient variables. Decision Variables are the ones that will be determined by optimizing our integer program. In our program, all of our decision variables are integer and most of them are binary. Coefficient variables are predetermined variables, in this case, data from previous swiss chess rounds. The combination of our two types of variables will allow us to optimize our integer program.

1. Decision Variables:

I is the set of all swiss chess players since there are 94 participants and J is the set of all possible number of slots that there can be with 94 participants, in this case, there could be a maximum of 47 slots. In each slot there can be 2 players or none.

(See constraint XX)

* Coefficient Variables

1. Constraints:

* For each pair of players there can only be 2 or 0 players:

We introduce the variable Y\_j so that the sum can only be 2 or 0.

If Y\_j = 1, then:

Knowing that each X\_ij can only have a value of 0 or 1, there should be only two players whose value is 1 for each pair.

If Y\_j = 0, then:

In this case, the pair is totally empty because the sum all X\_ij for a specific j will be 0. We introduce the above constraint to avoid having pairs with only 1 player, which is not possible. Using Y\_j also allows us to count how many completed pairs there are.

* Each participant can only be in one pair:

We do a similar thing now for each participant. We sum all the participation each participant has in all pairs. The sum can be 0 if the participant is not assigned to any pair, or it could be 1 if the participant is assigned to one pair, but no more.

* Two players shall not play against each other more than once:

If P\_mn = 1, that means player m has played with player n previously, so they cannot play again. For a specific j, if P\_mn = 1, then:

This means that the sum of X\_mj and X\_nj cannot be equal to 2, so they cannot be in the same pair. This constraint will be for all J for the pair (m, n).

If P\_mn = 0, then player m has not played with player n previously, then:

This means there is no restriction on any of the mentioned variables.

* In general, players with the same running score are paired.

This is a not totally mandatory constraint, since we should still pair players with similar score, but not necessarily the same score. Therefore, we should connect this constraint with the objective function as we want to minimize the difference between the running score of players in the same slot. We display a minimizing function and its constraints. We will introduce the whole objective function later.