ISYE 4133: Advanced Optimization
Swiss Chess Project
Name: Leonardo Calizaya

Introduction:

A Swiss-system chess tournament is a type of tournament different than traditional tournaments styles. It is characterized by a set of rules that will determine the pairings of each of the players, their game color, the results, etc.

Using data from previous rounds and the set of rules that define the Swiss-system chess tournaments, we can create an integer program formulation to obtain the pairings for the next round as well as the color each player is assigned. The objective of this system is to maximize the number of pairings, it may have other minor objectives as well. The set of rules can be transformed into constraints that the objective needs to meet.

For instance, some constraints are: Two players shall not play against each other more than once, players are paired with others with the same running score, no player shall receive the same color three times in a row, etc.

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.

Decision Variables:

- $"X_{ij} \in \{0,1\}\ where\ X_{ij} = 1\ if\ a\ participant\ i\ will\ play\ in\ slot\ j, otherwise, 0."$

$$i \in I, j \in J, I = \{x \in \mathbb{N} \mid x \leq 94\}, J = \{x \in \mathbb{N} \mid x \leq 48\}$$

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.

- $"Y_j \in \{0,1\}\ where\ Y_j = 1\ if\ slot\ "\mathrm{i}"\ is\ full\ with\ 2\ players, otherwise, 0."$

- "$C_{ik} \in \{0,1\}$ *where* $C_{ik} =$
  1 *if participant i* is assigned color k, *otherwise*, 0."

$k \in \{W, B, N\}$, where "W" stands for white, "B" for black and N for "Not Assigned"

- "$Z_p \in \{0,1\}$ *where* Z_p is a decision variable required for a set of "if constraints""
  (See constraint down)

Coefficient Variables

- "$U_i \in \{0,1\}$ *where* $U_i =$
  1 *if a participant i was unpaired in the previous round*, *otherwise*, 0"
- "$P_{mn} \in \{0,1\}$ *where* $P_{mn} =$
  1 *if participant* m played with participant n, in previous rounds , *otherwise*, 0."
- "$S_i \in \{x \in \mathbb{R} \mid x > 0\}$ *where* S_i is the running score of participant i"
- "$W_i \in \{x \in \mathbb{N}\}$ *where* W_i is the total number of white games played by participant i"
- "$B_i \in \{x \in \mathbb{N}\}$ *where* B_i is the total number of black games played by participant i"
- "$R_{ik} \in \{0,1\}$ *where* $R_{ik} =$
  0 if participant i received color k two times in a row previously, otherwise, 1"
- "$T_{ik} \in \{0,1\}$ *where* $T_{ik} =$
  0 if participant i had color k previously, otherwise, 1"

Constraints:

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

  $$\forall j \in J, \sum_{i \in I} (X_{ij}) - 2 = -2(1 - Y_j)$$

  We introduce the variable Y_j so that the sum can only be 2 or 0.
  If Y_j = 1, then:

  $$\sum_{i \in I} (X_{ij}) = 2$$

  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:

$$\sum_{i \in I} (X_{ij}) = 0$$

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:

$$\forall i \in I, \sum_{j \in J} (X_{ij}) \leq 1$$

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:

$$\forall j \in J, \ P_{mn} * \sum_{l \in m,n} X_{lj} \leq 1$$

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:

$$X_{mj} + X_{nj} \leq 1$$

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:

$$0 \leq 1$$

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.

$$\forall i \in \{(a, b) \in I \ x \ I | a \neq b\}, \forall j \in J$$

$$\min\ (|S_aX_{aj} - S_bX_{bj}|)$$

which is equivalently as saying:

$$\forall (a,b) \in I\ x\ I,\ \ \forall j \in J$$
$$min\ H_z$$
$$H_z \geq S_aX_{aj} - S_bX_{bj}$$
$$H_z \geq -(S_aX_{aj} - S_bX_{bj})$$

We will introduce all H_z created to the objective function

- No player shall receive the same color three times in a row.

$$\forall i\ \in\ I, (\sum_{k \in \{W,B\}} (R_{ik} * C_{ik}))\ +\ C_{iN}\ =\ 1$$

We create a constraint per each player, so that the sum of the R_iW, C_iB and C_iN must be 1. A player “i” can only be assigned white, black, or “Not assigned color”. Since each of this decision variables are binary, there can only be a “1” from the three of them. Moreover, we introduce R_ik so that if a player has played with a color k two times previously, R_ik = 0. Then we eliminate one possibility for that player. Let’s say, if R_iB = 0, then C_iW+C_iN = 1.

- For each pair of players that will play against each other, one plays with white and the other with black.

  Lets say, if x_11 = 1 and x_31 =1, then C_1W+C_1B+C_3W + C_3B >= 2, because player 1 and player 3 in pair 1, so they play against each other. Therefore, one needs to be assigned white and the other black, “Not assigned” is not an option in this case.

  Doing some linear transformations, we get the following constraints:

$$\forall (a,b) \in IxI, \forall j \in J$$
$$\sum_{k \in W,B} C_{ak} + (\sum_{k \in W,B} (C_{bk}))\ - 2\ \geq\ -3(1 - Z_p)$$

We create a binary decision variable Z_p for each of these constraints. This variable results after linearizing logical expressions.

- The difference between the number of black and the number of white games shall not be greater than 2 or less than -2.

$$\forall i\ \in\ I$$
$$W_i - B_i + C_{iW} - C_{iB} \leq 2$$
$$W_i - B_i + C_{iW} - C_{iB} \geq -2$$

- If colors are already balanced, then, in general, the player is given the color that alternates from the last one with which they played.

$$\forall i \in I, \; If \; \mathrm{W_i} = \mathrm{B_i}, then$$

$$\mathrm{T_{iw}C_{iw}} + \mathrm{T_{iB}C_{iB}} + \mathrm{C_{iN}} = 1$$

- A strong colour preference for black occurs when a player's colour difference is +1

$$\forall i \in I, \; If \; \mathrm{W_i} - \mathrm{B_i} = 1, then$$

$$\mathrm{W_i} - \mathrm{B_i} + 2\mathrm{C_{iW}} - \mathrm{C_{iB}} \leq 2$$
$$\mathrm{W_i} - \mathrm{B_i} + 2\mathrm{C_{iW}} - \mathrm{C_{iB}} \geq -2$$

In this case we add the coefficient 2 to C_iW so that it gives preference to C_iB be the option selected be our program. C_iW will have more probability to be out of bounds, then less likely to happen. In this case the preference is for color black. Note that W_i and B_i are not decision variables, those are data that can be obtained from previous rounds.

- A strong colour preference for white occurs when a player's colour difference is -1

$$\forall i \in I, \; If \; \mathrm{W_i} - \mathrm{B_i} = -1, then$$

$$\mathrm{W_i} - \mathrm{B_i} + \mathrm{C_{iW}} - 2\mathrm{C_{iB}} \leq 2$$
$$\mathrm{W_i} - \mathrm{B_i} + \mathrm{C_{iW}} - 2\mathrm{C_{iB}} \geq -2$$

In this case we add the coefficient 2 to C_iB so that it gives preference to C_iW be the option selected by our program. C_iB will have more probability to be out of bounds, then less likely to happen. In this case the preference is for color white. Note that W_i and B_i are not decision variables, those are data that can be obtained from previous rounds.

For each i in I:

- If W_i – B_i + C_iW – C_iB = 2, then L_i = 1
- If W_i – B_i + C_iW – C_iB = -2, then M_i = 1

OBJECTIVE FUNCTION:

$$\text{MAX } 50^* \sum_{j \in J} Y_j + 30^* \sum_{i \in I} \sum_{j \in J} U_i X_{ij} + 20^* \sum_{i \in I} L_i + \sum_{i \in I} M_i \text{ - } 5^* \sum_{i \in I} H_i$$

Where L_i is the number of players that have a color difference of 2 and M_i is the number of players that have a color difference of -2. We introduce a different weight to each of the sums to give more importance to some objectives than other. In the case of the substraction, we want to minimize the difference between the running scores of players assigned to each other.