École des Ponts ParisTech

The drawing of the teams in the future Champions League

IMI Project

By

Adle Ben Salem, Thomas Buchholtzer and Mathieu Tanré

This project was supervised by

Julien Guyon and Frédéric Meunier

Contents

1 Introduction

The Union of European Football Associations, also known as UEFA, has decided to implement a new format for the Champions League, the most prestigious competition between football clubs in Europe, for the next season. It aims at increasing the number of participating teams and making it more interesting to watch for supporters.

We focused our work on the first phase of this new format. It consists of a mini-league of **36 teams**. The latter are ranked according to their current UEFA coefficient and divided into **4 pots of 9 teams** named from A to D, with respectively the strongest teams and the weakest teams. What will be later called the constraints of this problem are the following rules:

- Each team plays **8 matches**, exactly **2 matches against two different teams from each pot**, one at home and one away.
- A team **cannot play against a team from the same country**.
- A team **cannot play against more than two teams of each country**.
- The matches must all be played over **8 match days**.

The problem is the following: UEFA, as it has always done, wants to perform the draw of the different matches live, if possible making it visual for the audience, for instance by drawing balls from an urn. At first sight, two natural approaches are possible.

- **Method 1: Schedule-first approach -** We begin by building an 8-day calendar, incorporating placeholders for the teams, noted as X_i where $X \in A, B, C, D$ and $i \in 1, ..., 9$. Then we sequentially assign the identities of the teams to the placeholders, ensuring that the constraints are satisfied. The matches will therefore be known only at the end of the draw.
- **Method 2: Matches-first approach -** We consider each team, one after another, and draw their different opponents according to the constraints, and if they are playing home or away. Then we order the matches to build the schedule.

The structure of this report reflects the evolution of our work, considering the two different methods successively. For the first method, the challenge is to find a schedule and give arguments to explain in what sense the schedule we propose is the "best" and give the limits of this approach, which leads us to examine the second one. For the second method, we question the mathematical feasibility of such a draw and then implement it efficiently.

2 Method 1

To begin with, we focus on developing a calendar template which spans 8 days. This template will incorporate placeholders for teams, noted as X_i where $X \in A, B, C, D$ and $i \in 1, ..., 9$. The core objective is to sequentially assign identities to these team placeholders afterwards while adhering to the constraints.

2.1 Mathematical background

The idea is to express our problem as linear programming. At first, we implement the simple version of the problem. Then, progressively, we try to demand interesting properties to our schedule. All of this was done using the solver Gurobi.

2.1.1 Some notations

In the following, we will assign each team a number $i \in \{1, \dots, 36\}$, with the convention that team i is in pot:

- A if $1 \leq i \leq 9$
- B if $10 \leq i \leq 18$

- C if $19 \leq i \leq 27$
- D if $28 \leq i \leq 36$

Moreover, for all $i \in \{1,\dots,36\}$, $j \in \{1,\dots,36\}$, $t \in \{1,\dots,8\}$, we will introduce the binary variable x_{ijt} such that $x_{\text{ijt}} = 1$ if on day t, i plays against j at home, and $x_{ijt} = 0$ otherwise.

2.1.2 Writing the constraints

Let us turn the constraints into equations:

$$\forall t \in \{1,\dots,8\}, \forall i \in \{1,\dots,36\}, \quad x_{iit} = 0 \tag{1}$$

→ A team cannot play against itself.

$$\forall (i,j) \in \{1,\dots,36\}^2, \quad \sum_{t=1}^{8} (x_{ijt} + x_{jit}) \leq 1 \tag{2}$$

→ A team plays at most once against each other team.

$$\forall t \in \{1,\dots,8\}, \forall i \in \{1,\dots,36\}, \quad \sum_{j=1}^{36} (x_{ijt} + x_{jit}) = 1 \tag{3}$$

→ Each team plays exactly one match per day.

For all $i \in \{1,\dots,36\}$:

$$\begin{aligned}
\sum_{t=1}^{8}\sum_{j=1}^{9} x_{ijt} = 1 \qquad & \sum_{t=1}^{8}\sum_{j=1}^{9} x_{jit} = 1 \\
\sum_{t=1}^{8}\sum_{j=10}^{18} x_{ijt} = 1 \qquad & \sum_{t=1}^{8}\sum_{j=10}^{18} x_{jit} = 1 \\
\sum_{t=1}^{8}\sum_{j=19}^{27} x_{ijt} = 1 \qquad & \sum_{t=1}^{8}\sum_{j=19}^{27} x_{jit} = 1 \\
\sum_{t=1}^{8}\sum_{j=28}^{36} x_{ijt} = 1 \qquad & \sum_{t=1}^{8}\sum_{j=28}^{36} x_{jit} = 1
\end{aligned} \tag{4}$$

→ Each team plays exactly against two teams from each pot, one home match and one away match.

The solver unsurprisingly finds a solution to this problem, giving us a first functional template. We will now try to add constraints to optimize the schedule.

2.1.3 Home-away alternation

We now attempt to add a constraint to alternate home and away matches for each team. It is good for the teams because they are used to alternate between away and home matches, and it is better for them than having for example 3 away matches in a row.

$$\forall t \in \{1,\dots,7\}, \forall i \in \{1,\dots,36\}, \quad \sum_{j=1}^{36} (x_{ijt} + x_{ij(t+1)}) = 1 \tag{5}$$

→ Each team plays once at home and once away, with perfect alternation.

This new constraint is very strong. By integrating it into the solver, we do not get a solution, so no template can fulfill this constraint.

Let us attempt to relax this constraint slightly, allowing at most one break per team, meaning a team can have two consecutive home or away days, but only once, and neither in the first two nor the last two days. Formally, it is written:

$$\forall t \in \{2,\dots,6\}, \forall i \in \{1,\dots,36\}, \quad \begin{aligned} \sum_{j=1}^{36} x_{ijt} + x_{ij(t+1)} \leq 1 + b_{it} \\ \sum_{j=1}^{36} x_{jit} + x_{ji(t+1)} \leq 1 + b_{it} \end{aligned} \tag{6}$$

where $b_{it} \in \{0,1\}$ is a binary variable indicating if team i has a break between days t and $t+1$. Moreover, for each team i:

$$\sum_{t=2}^{6} b_{it} \leq 1 \tag{7}$$

which ensures that at most one break is allowed for each team.

The strict constraints for the first two and last two matches remain unchanged. For all $i \in \{1,\dots,36\}$:

$$\sum_{j=1}^{36} x_{ij1} + x_{ij2} = 1 \tag{8}$$

$$\sum_{j=1}^{36} x_{ij7} + x_{ij8} = 1 \tag{9}$$

which ensures a strict home-away alternation for the first two and last two days.

Now, the solver finds a solution. Initially, we wanted to establish an optimal objective function aimed at minimizing the number of breaks, given the constraint that each team was limited to at most one break. However, we encountered significant computational time challenges when attempting to solve with this objective function. Empirically, we discovered a scenario in which only 4 teams, specifically A_1, B_1, C_1, and D_1, experience a single break. This result demonstrates a considerable improvement over our initial constraint, significantly enhancing the schedule's balance and fairness by ensuring that the vast majority of teams enjoy a perfect alternation of home and away matches. We choose to distribute the breaks evenly among the pots, so we add the constraints on the break variables written as:

$$\begin{aligned} \sum_{i=1}^{9} \sum_{t=2}^{6} b_{it} = 1 \\ \sum_{i=10}^{18} \sum_{t=2}^{6} b_{it} = 1 \\ \sum_{i=19}^{27} \sum_{t=2}^{6} b_{it} = 1 \\ \sum_{i=28}^{36} \sum_{t=2}^{6} b_{it} = 1 \end{aligned} \tag{10}$$

2.1.4 Even distribution of matches of the same interest over the match days

To maximize audience engagement over 8 days, UEFA would benefit from not scheduling top teams to play against each other on the same day, and similarly for the lower-ranked teams. Thus, it is advisable to distribute the matches from each pot evenly across the 8 days. For a given pot, there are 9 matches that feature two teams from that pot. Therefore, we set a constraint that on one of the 8 days, 2 such matches will occur, and exactly one match of this type will occur on each of the other 7 days.

Let the binary variables $b_{\mathrm{A}t}$, $b_{\mathrm{B}t}$, $b_{\mathrm{C}t}$, $b_{\mathrm{D}t}$ for each pot (A, B, C, D) and each day t indicate whether that day has 2 matches for the corresponding pot. The constraints are as follows:

- For each pot (A, B, C, D), there is exactly one day with 2 matches:

$$\begin{aligned}\sum_{t=1}^{8} b_{\mathrm{A},t} &= 1\\ \sum_{t=1}^{8} b_{\mathrm{B},t} &= 1\\ \sum_{t=1}^{8} b_{\mathrm{C},t} &= 1\\ \sum_{t=1}^{8} b_{\mathrm{D},t} &= 1\end{aligned} \tag{11}$$

- For each day t and each pot (for example, pot A with teams 1 to 9):

$$\begin{aligned}\sum_{i=1}^{9}\sum_{j=1}^{9} x_{ijt} &= 1 + b_{\mathrm{A},t}\\ \sum_{i=10}^{18}\sum_{j=10}^{18} x_{ijt} &= 1 + b_{\mathrm{B},t}\\ \sum_{i=19}^{27}\sum_{j=19}^{27} x_{ijt} &= 1 + b_{\mathrm{C},t}\\ \sum_{i=28}^{36}\sum_{j=28}^{36} x_{ijt} &= 1 + b_{\mathrm{D},t}\end{aligned} \tag{12}$$

These constraints ensure that matches between teams from the same pot are evenly distributed over the 8 days.

A similar setup can be applied for matches between teams from two different pots. In this scenario, there are 18 matches to be scheduled in total. An ideal distribution would be 2 days with 3 such matches and 6 days with 2 such matches. We present the version for pots A and B. We define the binary variable $b_{\mathrm{AB}t}$, equal to 1 if on day t, there are 3 matches between a team from pot A and a team from pot B. We have:

- Since there are 2 days where 3 matches of A against B take place, the sum equals 2:

$$\sum_{t=1}^{8} b_{\mathrm{AB},t} = 2 \tag{13}$$

- To ensure that every day there are either two matches of a team A against a team B or three such matches:

$$\sum_{i=1}^{9}\sum_{j=10}^{18} x_{ijt} + x_{jit} = 2 + b_{\mathrm{AB},t} \tag{14}$$

Applying this constraint to all pot pairs showed no solution. However, by applying it to the pairs (A, B) and (C, D), the solver was able to find a solution. This decision is made considering these to be respectively the most and least interesting types of matches after (A-A) and (D-D) matches. Nonetheless, we still require that for the other pairs, there be between 1 and 3 such matches per day.

2.1.5 Sequential match ordering within pots

In an effort to streamline the scheduling of inter-pot matches, we adopted a strategy of implementing a single cycle for matches within each pot. This approach dictates that matches are organized in a sequential manner, such as X_1 vs. X_2, X_2 vs. X_3, and so forth, culminating in a match between X_9

and X_1. This method ensures that every team within a pot plays against its immediate predecessor and successor, thereby preventing the formation of mini-groups within the championship phase.

Such a configuration was chosen to avoid scenarios where teams might end up playing in a small loop, for example, X_1 vs. X_2, X_2 vs. X_3, and X_3 vs. X_1, which could inadvertently create a subgroup effect within the broader tournament structure. By enforcing this sequential match order, we aim to maintain the integrity of the championship's competitive balance and fairness, ensuring that all teams are treated equitably and that the schedule reflects a coherent and logical progression of matches.

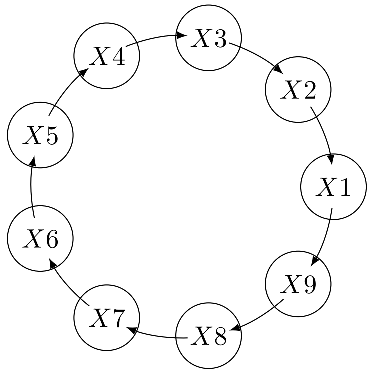

Figure 1: Distribution of intra-pot matches for pot X

2.1.6 Balancing the schedule of matches for each team

We also aim for each team's schedule to be well-balanced over the tournament duration. Specifically, we prioritize distributing encounters with strong teams (from pots A and B) evenly across the 8 match days. This approach ensures that no team faces a concentration of matches against top-tier opponents in a short span, promoting a fairer and more equitable competition structure.

For each team i and for each days $t \in \{2, \ldots, 6\}$:

$$\begin{aligned} &\sum_{j \in \text{Pot A}} \left(x_{ijt} + x_{jit} + x_{ij(t+1)} + x_{ji(t+1)} + x_{ij(t+2)} + x_{ji(t+2)}\right) \leq 1 \\ &\sum_{j \in \text{Pot B}} \left(x_{ijt} + x_{jit} + x_{ij(t+1)} + x_{ji(t+1)} + x_{ij(t+2)} + x_{ji(t+2)}\right) \leq 1 \end{aligned} \tag{15}$$

Further enhancing the balance of the schedule, we ensure diversity in the matchups by regulating the frequency of encounters with teams from pots C and D during critical phases of the tournament. This policy is particularly enforced at the beginning and the end of the schedule to prevent any team from facing two teams from pot C or two teams from pot D in the first two or last two match days. This strategy helps maintain a level playing field and avoids overwhelming any team with consecutive high-stake games against similarly ranked opponents.

For each team i, the constraints for the first two and last two match days are as follows:

$$\begin{aligned} &\sum_{j \in \text{Pot C}} (x_{ij1} + x_{ji1} + x_{ij2} + x_{ji2}) \leq 1, \\ &\sum_{j \in \text{Pot C}} (x_{ij7} + x_{ji7} + x_{ij8} + x_{ji8}) \leq 1, \\ &\sum_{j \in \text{Pot D}} (x_{ij1} + x_{ji1} + x_{ij2} + x_{ji2}) \leq 1, \\ &\sum_{j \in \text{Pot D}} (x_{ij7} + x_{ji7} + x_{ij8} + x_{ji8}) \leq 1. \end{aligned} \tag{16}$$

This arrangement of matches not only diversifies the competition but also strategically distributes the challenge across the tournament, ensuring that all teams face a balanced and fair competitive environment throughout the eight days.

2.2 Explicit formulation of the schedule

Finally, we were able to get a schedule respecting all our properties.

Match day 1

Home	Away
A1	B9
A2	D4
A4	A5
A6	B7
A8	C5
B1	A3
B2	B3
B4	C1
B6	D6
B8	D2
C2	A7
C4	D8
C6	B5
C8	C9
D3	C7
D5	C3
D7	A9
D9	D1

Match day 2

Home	Away
A3	D5
A5	B2
A7	A8
A9	A1
B3	D9
B5	A4
B7	B8
B9	C4
C1	C2
C3	A2
C5	D7
C7	B6
C9	A6
D1	B4
D2	D3
D4	C6
D6	C8
D8	B1

Match day 3

Home	Away
A2	A3
A4	D2
A6	D6
A8	B3
B1	C3
B2	C9
B4	B5
B6	A7
B8	A9
C1	A1
C2	B9
C4	A5
C6	C7
C8	D1
D3	D4
D5	B7
D7	D8
D9	C5

Match day 4

Home	Away
A1	D9
A3	B4
A5	A6
A7	C6
A9	C2
B3	C8
B5	D3
B7	A2
B9	B1
C3	C4
C5	B2
C7	A8
C9	D5
D1	C1
D2	B6
D4	B8
D6	D7
D8	A4

Match day 5

Home	Away
A2	C9
A4	B1
A6	C3
A8	A9
B2	A1
B4	D8
B6	B7
B8	C7
C1	B3
C2	D4
C4	C5
C6	D2
C8	A3
D1	A7
D3	A5
D5	D6
D7	B5
D9	B9

Match day 6

Home	Away
A1	A2
A3	C4
A5	D7
A7	B8
A9	B6
B1	D5
B3	B4
B5	C2
B7	C6
B9	D1
C3	D3
C5	A4
C7	C8
C9	C1
D2	A6
D4	A8
D6	B2
D8	D9

Match day 7

Home	Away
A2	B5
A4	C7
A6	A7
A8	D8
B2	D4
B4	A5
B6	C5
B8	B9
C1	D6
C2	C3
C4	B1
C6	A9
C8	B7
D1	D2
D3	B3
D5	A1
D7	C9
D9	A3

Match day 8

Home	Away
A1	C8
A3	A4
A5	C1
A7	D3
A9	D1
B1	B2
B3	A6
B5	B6
B7	D7
B9	A8
C3	B4
C5	C6
C7	D9
C9	B8
D2	C4
D4	D5
D6	A2
D8	C2

If we sum up, the preceding schedule satisfies all the following properties:

1. A team plays at most once against each other team.
2. Each team plays exactly one match per day.
3. Each team plays exactly against two teams from each pot, one home match and one away match.
4. Perfect home-away alternation for the 32 teams other than A_1, B_1, C_1, and D_1.

5. The teams A_1, B_1, C_1, and D_1 will have two successive matches at home or away, which is called a *break*. In particular, it does not not occur during the first two or last two match days.
6. Even distribution over the 8 match days for intra-pot matches (1 day with 2 matches, and 7 days with 1 match).
7. Even distribution over the 8 match days for inter-pot matches AB, AC, and BC (2 days with 3 matches, and 6 days with 2 matches).
8. The number of inter-pot matches AD, BD, and CD per match day ranges between 1 and 3.
9. Inter-pot matches form a cycle of maximum length 9, meaning X_1 hosts X_2, X_2 hosts X_3, ..., X_8 hosts X_9, and X_9 hosts X_1, for any pot X.
10. No team can face teams from pot A or B twice over 3 consecutive match days.
11. No team can face 2 teams from pot C or 2 teams from pot D in the first two or last two matches.

Properties 1-2-3 are basically just saying that the schedule respects the constraints. Properties 4-5 are here for a good home-away alternation. Properties 6-7-8 aim at an even distribution of the matches during the competition. Property 9 aims at avoiding to encounter small championships in the competition. Ultimately, properties 10-11 guarantee each team will have a well-balanced season.

2.3 Drawbacks

Even though we have arguments to say the schedule is "good", it remains highly subjective and reflects some choices of what we thought was more important. Nevertheless, the main problem of this method is that drawing the real teams for each placeholder is not as simple as we might think. Indeed, fixing the schedule of the matches is too restrictive regarding the incredible amount of other material constraints you have to deal with. For instance, teams from Northern Europe are not able to play at home during winter, each stadium may not be available for each match day... As a consequence, taking into account all of this would be a nightmare and would mean to fix *a priori* some team to some placeholder, which would ruin the randomness of the draw.

3 Method 2

To continue, we develop a sequential draw process. Specifically, our aim is to be able to select a team and then identify all possible opponent teams from a given pot that it can play against without leading to a deadlock. From these eligible teams, we will randomly select two opponents. This process will be iteratively applied across all pots and teams.

The advantage of this method is that it allows greater flexibility in scheduling match days once the fixtures are decided. Organizers can thus arrange the matches at their convenience and according to constraints that may be difficult to predict, such as stadium availability. Moreover, the appeal of certain historic matches may not be captured in models. Thus, the random draw of fixtures enables the distribution of highly anticipated matches with a more human-centric approach.

3.1 Feasibility of the method

The second method involves determining the matches *a priori*. For instance, we might get that Real Madrid will host Liverpool and also play away against PSG for the pot A and so forth for the other pots. When throwing a team, we make sure that the nationality constraints are met. However, one potential issue remains: it may not be feasible for the matches to take place within an 8-day period.

3.1.1 A graph theory point of view

The problem can be formulated as the following. Let $G = (V, E)$ be a graph, such that $V = V_1 \sqcup V_2 \sqcup V_3 \sqcup V_4$. The vertices represent the different teams, grouped in the 4 pots. The edges are oriented and an edge uv means team u plays at home against team v. To translate the constraints we will say a graph is (2, 2, 2, 2)-regular if and only if each vertex has exactly 2 neighbors in V_1, 2 neighbors in V_2, 2 neighbors in V_3 and 2 neighbors in V_4. Moreover, when talking about coloration, here we are dealing with a coloration of the edges of the graph. The chromatic index is the number of match days needed to play all the matches. Immediately, we get that the chromatic indice of a (2, 2, 2, 2)-regular graph is at least 8. In what comes next, we want to prove that it is strictly greater than 8.

3.1.2 Cut vertex and useful lemma

Let us say that our graph has k different components. A cut vertex $v \in V$ is a vertex such that $G - v$ has $k + 1$ components.

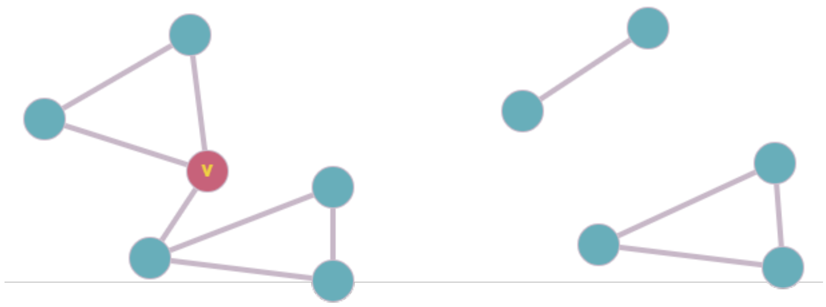

Figure 2: An example of a cut vertex

The lemma retaining our attention is the following. We consider a k-regular graph, namely each of its vertices has exactly k neighbours. If it admits a cut vertex, then its chromatic index is strictly greater than k.

3.1.3 Proving the lemma

Let $G = (V, E)$ be a k-regular graph admitting a cut vertex $v \in V$. By contradiction, let us assume that G admits an edge colouring with k colors. Let us fix such a coloring. Without loss of generality, we will assume that G is connected. As a consequence, removing v, by definition, creates two connected components G_1 and G_2. As the graph is k-regular, v was - at least - connected to some $u_1 \in G_1$ and some $u_2 \in G_2$. To make it concrete, we may say that u_1v was colored red and u_2v was colored blue.

Let us now consider $H_1 \subseteq G_1$ the subgraph containing only the blue and red edges. In G_1, u_1 has only one incident edge, as long as the other one is u_1v. However, all the other vertices in H_1 have two incident edges. Indeed, the graph was k-regular at the beginning and removing v did not remove any other edges incident to H_1.

To conclude, we need to use the handshake lemma, stating than in any - finite - graph,

$$\sum_{v \in V} \deg(v) = 2|E|$$

. In particular, the sum of the degrees of each vertex has to be even. Therefore, a k-regular graph does not admit an edge coloring with k colours. Obviously, it cannot take less than k color. That is why the chromatic index is strictly greater than k.

3.1.4 Back to our problem

Let us apply it in our case. The draw of the matches will give us a (2, 2, 2, 2)-regular graph, which is a particular case of a 8-regular graph. The existence of a cut vertex would make it impossible to color it with 8 colors, which means it would be impossible to fit the matches in 8 days. The following question is can we find a draw which would lead to the existence of a cut vertex.

3.1.5 Building a counter-example

At first, we had to find a simple way to manually create a (2, 2, 2, 2)-regular graph. For this purpose, a simple approach is to replicate the following pattern, based on cycles of size 3.

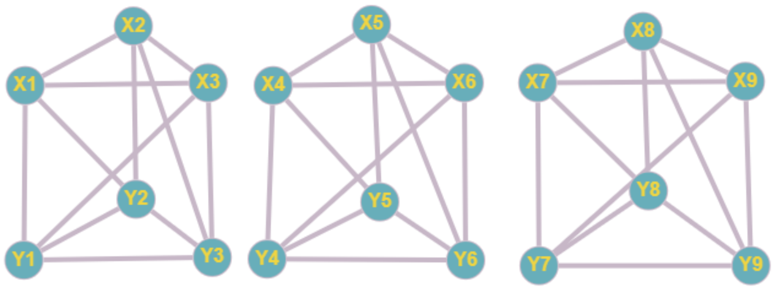

Figure 3: A useful pattern for the matches between pot X and pot Y

The principle is the following, repeat the same construction between each pair (X, Y) of pots. We can easily convince ourselves that it creates indeed a (2, 2, 2, 2)-regular graph.

In order to create a cut vertex, we will need to modify it a little bit. The idea is to create dissymmetry by dividing the inner-pot cycles into one cycle of size 4 and one of size 5. For instance, we will modify the pattern for the matches between pot A and pot B, all the other parts of the graph remaining the same.

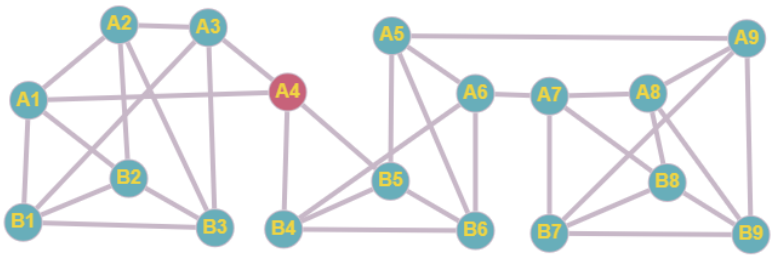

Figure 4: Modification of the pattern between pot A and pot B

It still satisfies the constraints of (2, 2, 2, 2)-regularity. And if we take a look at the vertex A_4, it is clear that it is a cut vertex.

3.1.6 Conclusion

Drawing the matches successively focusing only about the constraints of regularity and nationality could eventually fail and make the matches impossible to fit in a 8-day calendar. To illustrate this, we even simulated it with some real teams.

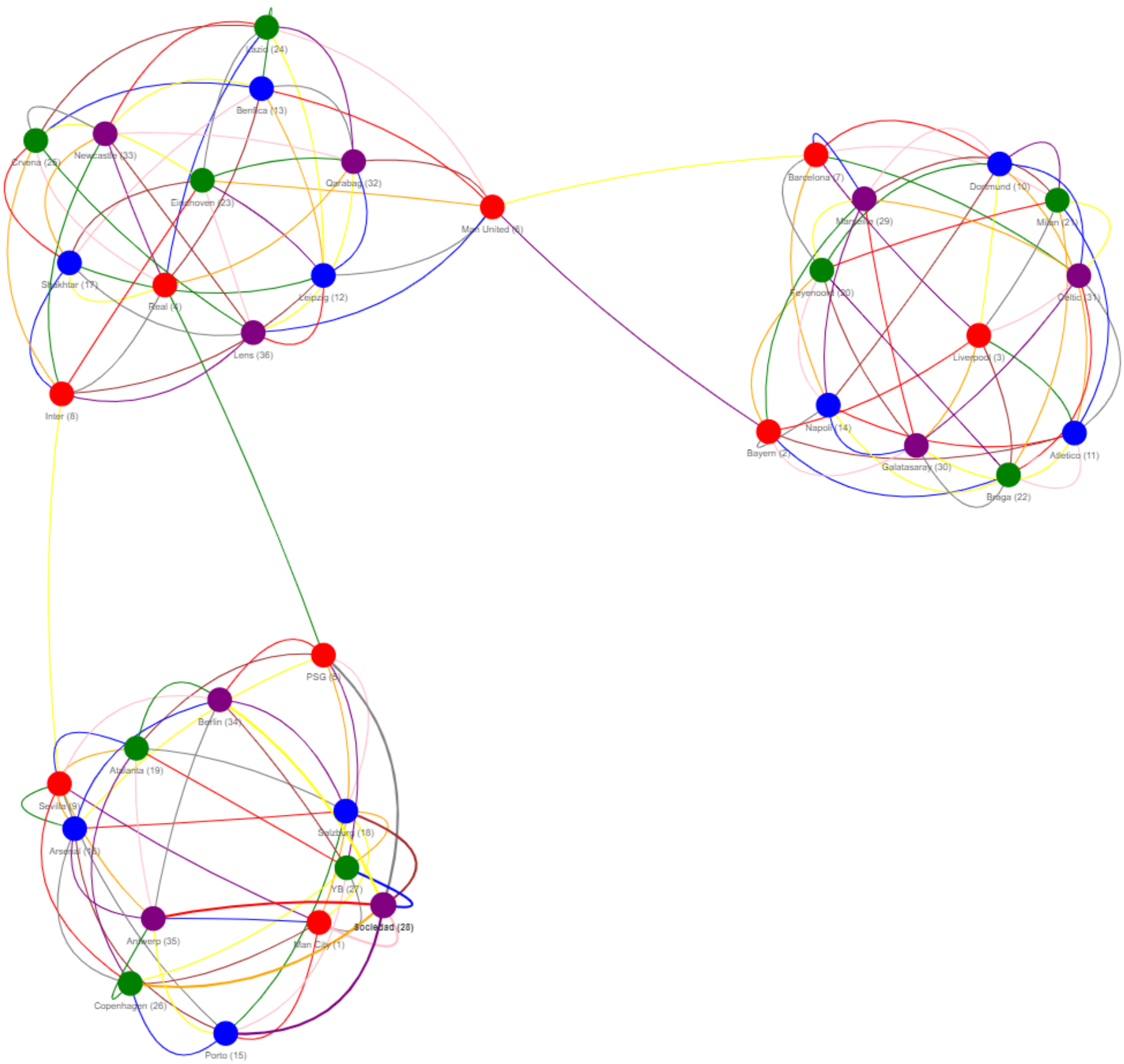

Figure 5: Example of an impossible draw

Manchester United is the famous cut vertex. Nevertheless, we can be notice that such cases seem pathological and that there is a very small chance for such event to happen. But it is possible and it is now impossible not to take account of that. We understand that after Julien Guyon discussed this matter with UEFA, the requirement that all matches be played in 8 matchdays has been taken into account.

3.2 Explanation of the procedure

1. **Initialization:**
    - The nationalities of the teams are extracted and stored for later use in the match validation process. A set of constraints is initialized and will contain the current state of the drawing.
    - A text file (`tirage_au_sort.txt`) is opened in write mode to record the results of the draw.
2. **The drawing process:**
    - The process iterates over the 4 pots of teams. For each pot:
        (a) The indices of the teams within the pot are randomly shuffled.
        (b) Each team is sequentially selected from the shuffled list to participate in the draw. Following selection, we determine the opponents this team will face using the Gurobi

optimizer along with a set of constraints. These constraints reflect the current state of the draw, incorporating the matches that have already been selected.

- For the selected team and the current pot, we identify eligible matches with teams from this pot while adhering to the predefined constraints.
- We display the list of admissible match pairs (home, away) and randomly select one of them.

3. **Updating constraints:**
   - Constraints are updated for each determined match to reflect the matchups already drawn.
   - These constraints are then passed as an argument to the function that determines the list of admissible matches.

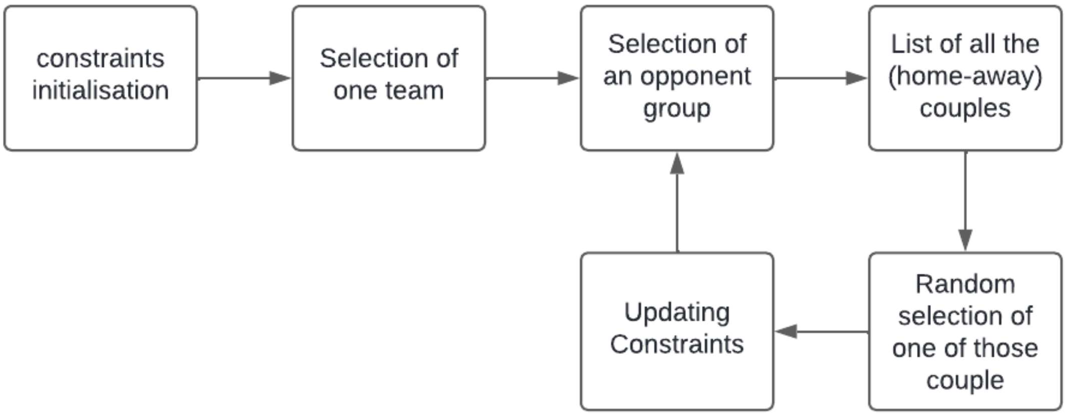

Figure 6: Process of the sequential drawing

3.3 Team representation and constraint management

3.3.1 Team representation

Teams are represented as a list of dictionaries in Julia, where each dictionary contains key-value pairs representing the team's name under the key `"club"` and its nationality under the key `"nationality"`. This structured format facilitates easy access and manipulation of team attributes.

Structure The teams are organized into four separate lists, each corresponding to a different pot of teams. Each list contains multiple dictionaries, each representing a team as shown below:

```
teams = [
    [Dict("club" => "ManCity", "nationality" => "England"), ...],
    ...
]
```

3.3.2 Constraint management

Constraints are critical to ensuring that the draw respects specific rules, such as preventing teams from the same country playing against each other too early in a tournament.

Initialization of constraints Constraints are initialized and stored in a dictionary where each team's name is a key, and the value is another dictionary detailing various constraints related to that team:

```
constraints = Dict("ManCity" => {"played-home" => Set(), "played-ext" => Set(),
                                 "nationalities" => Dict("England" => 2, ...)})
```

Detailed breakdown

- **Nationalities:** A dictionary is used to track the number of times teams have played against other nationalities. This is initialized to zero for all potential opponent nationalities and set to two for the team's own nationality to forbid a match against two teams of a same country.
- **Played-Home and Played-Ext:** Sets are used to keep track of which teams have played at home and which have played as visitors, ensuring no team plays at one venue too often.

3.4 Determination of admissible matches

The optimization process then evaluates whether the proposed match pairings can lead to a viable scheduling solution, using the Gurobi Optimizer to assess feasibility.

3.4.1 Pre-admissibility selection

Before solving the match scheduling problem, certain pre-admissibility checks are performed to filter out ineligible matches based on prior games and nationality constraints.

Filtering based on previous home games This check identifies if the selected team has already hosted a game against any team in the opponent pot. If such a match has occurred, the home team involved is identified and excluded from being scheduled again, preventing repeat home games.

Filtering based on previous away Games This process checks for previous away games that the selected team has played against teams in the opponent pot. It aims to avoid scheduling repeat matches that could disrupt the balance of the tournament.

Nationality check We check if the selected team has already played against two teams of the same nationality as one of the two teams in the couple (home, away). We also check that there are not from the same country as the selected team.

Application in admissible match selection This step uses the results from the filtering processes to compile a list of potential matches. This is this list of potential matches (home-away) that are passed consecutively as an argument in the solver function.

3.4.2 Initialization of the optimization model

The function begins by initializing a mathematical model with Gurobi, setting specific attributes to control solver output and functionality. The model uses binary variables, `match_vars[i, j, t]`, indicating whether team i plays against team j on day t as we did in section 1. This way, we also assure a chromatic index equals to 8.

- **No self-play:** Ensures that no team is scheduled to play against itself.
- **Unique matches:** Prevents any two teams from playing more than once.
- **Pot-specific constraints** Each team must play exactly one game home and one game away against two different teams in every pot.
- **No matches within the same nationality**: This constraint ensures that teams of the same nationality across different pots do not play against each other.
- **Match inclusion** We add one of the possible couple (home-away) and add the matchups in the constraint in order to see if this couple is admissible.

$$\forall i, j \in \{1, 2, \ldots, 36\}, \text{ if Nationality}(team_i) = \text{Nationality}(team_j) : \sum_{t=1}^{T} (x_{ijt} + x_{jit}) = 0 \tag{17}$$

- **Constraint to limit matches among teams of the same nationality under 2**: This constraint ensures that any team plays against teams of the same nationality no more than twice throughout the tournament. If we denote Nationality(team$_i$) as N(team$_i$), then the following condition holds:

$$\forall \text{ nationality}, \forall i \in \{1, 2, \ldots, 36\} : \sum_{\substack{j=1 \\ N(\text{team}_j)=\text{nationality}}}^{36} \sum_{t=1}^{8} (x_{ijt} + x_{jit}) \leq 2 \tag{18}$$

- **Previously played matches:** Finally, we add in our model the matches that are stored in our set of constraints.

3.4.3 Solution and verification

The solver is run to optimize the model. The function checks if the solution status returned by Gurobi indicates an optimal solution, confirming the feasibility of the proposed match schedule:

```
optimize!(model)
return termination_status(model) == MOI.OPTIMAL
```

3.5 Hypothesis for further optimization

The enhancements implemented in the draw process have significantly streamlined operations, reducing the total time required for a complete draw to approximately three minutes. This represents a substantial improvement in efficiency, allowing for quicker and more reliable scheduling outcomes.

It may be possible to achieve even greater efficiency by optimizing the use of the Gurobi solver. Currently, the environment is set up anew for each draw, which includes redefining constraints. A potential improvement could involve initializing the Gurobi environment once and then updating constraints within this persistent environment for subsequent draws. This approach could reduce overhead associated with repeatedly setting up the solver, thereby decreasing the total computation time further. Implementing this modification requires careful management of the solver's state to ensure that all constraints are accurately updated without residual effects from previous draws.

3.6 Result

We put to the test our algorithm with the potential 36 teams selected for the Champions League next year. Basically, the top teams from each country's league earn a spot. Additionally, some countries have multiple spots depending on their league's strength and also, some clubs can qualify through winning previous' year Champions League or Europa League.

The following tables are written in the order of the draw. The first column represents the team drawn. Then, each column represents its opponents in each pot. The first name always represents the match played at home and the second one the away match.

Sevilla	Inter|ManUnited	Shakthar|Benfica	Atalanta|YB	Lens|Marseille
ManUnited	Sevilla|PSG	Napoli|Salzburg	Copenhagen|Milan	Copenhagen|Milan
Bayern	Real|ManCity	Salzburg|Atletico	Braga|Copenhagen	Antwerp|Qarabag
Inter	ManCity|Sevilla	Benfica|Shakhtar	Crvena|Eindhoven	Newcastle|Lens
Liverpool	Barcelona|Real	Porto|Napoli	YB|Braga	Qarabag|Galatasaray
Barcelona	PSG|Liverpool	Leipzig|Dortmund	Milan|Crvena	Celtic|Newcastle
PSG	ManUnited|Barcelona	Arsenal|Leipzig	Eindhoven|Feyenoord	Berlin|Antwerp
ManCity	Bayern|Inter	Atletico|Porto	Feyenoord|Lazio	Sociedad|Berlin
Real	Liverpool|Bayern	Dortmund|Arsenal	Lazio|Atalanta	Galatasaray|Celtic

Table 1: Draw for the pot A

Porto	ManCity|Liverpool	Salzburg|Atletico	Crvena|Copenhagen	Berlin|Antwerp
Shakhtar	Inter|Sevilla	Arsenal|Benfica	Copenhagen|Crvena	Lens|Newcastle
Salzburg	ManUnited|Bayern	Dortmund|Porto	Lazio|Milan	Marseille|Galatasaray
Napoli	Liverpool|ManUnited	Atletico|Leipzig	YB|Braga	Celtic|Lens
Dortmund	Barcelona|Real	Benfica|Salzburg	Braga|Eindhoven	Antwerp|Marseille
Leipzig	PSG|Barcelona	Napoli|Arsenal	Feyenoord|Atalanta	Newcastle|Sociedad
Benfica	Sevilla|Inter	Shakhtar|Dortmund	Shakhtar|Dortmund	Sociedad|Qarabag
Atletico	Bayern|ManCity	Porto|Napoli	Milan|YB	Galatasaray|Berlin
Arsenal	Real|PSG	Leipzig|Shakhtar	Atalanta|Lazio	Qarabag|Celtic

Table 2: Draw for the pot B

YB	Sevilla|Liverpool	Atletico|Napoli	Eindhoven|Feyenoord	Qarabag|Lens
Lazio	ManCity|Real	Arsenal|Salzburg	Feyenoord|Copenhagen	Antwerp|Berlin
Feyenoord	PSG|ManCity	Benfica|Leipzig	YB|Lazio	Lens|Galatasaray
Milan	ManUnited|Barcelona	Salzburg|Atletico	Crvena|Eindhoven	Celtic|Antwerp
Atalanta	Real|Sevilla	Leipzig|Arsenal	Copenhagen|Braga	Berlin|Marseille
Eindhoven	Inter|PSG	Dortmund|Benfica	Milan|YB	Marseille|Celtic
Braga	Liverpool|Bayern	Napoli|Dortmund	Atalanta|Crvena	Newcastle|Sociedad
Copenhagen	Bayern|ManUnited	Porto|Shakhtar	Lazio|Atalanta	Sociedad|Newcastle
Crvena	Barcelona|Inter	Shakhtar|Porto	Braga|Milan	Galatasaray|Qarabag

Table 3: Draw for the pot C

Galatasaray	Liverpool|Real	Salzburg|Atletico	Feyenoord|Crvena	Berlin|Celtic
Berlin	ManCity|PSG	Atletico|Porto	Lazio|Atalanta	Newcastle|Galatasaray
Celtic	Real|Barcelona	Arsenal|Napoli	Eindhoven|Milan	Galatasaray|Antwerp
Lens	Inter|Sevilla	Napoli|Shakhtar	YB|Feyenoord	Antwerp|Sociedad
Marseille	Sevilla|ManUnited	Dortmund|Salzburg	Atalanta|Eindhoven	Qarabag|Newcastle
Sociedad	ManUnited|ManCity	Leipzig|Benfica	Braga|Copenhagen	Lens|Qarabag
Antwerp	PSG|Bayern	Porto|Dortmund	Milan|Lazio	Celtic|Lens
Qarabag	Bayern|Liverpool	Benfica|Arsenal	Crvena|YB	Sociedad|Marseille
Newcastle	Barcelona|Inter	Shakhtar|Leipzig	Copenhagen|Braga	Marseille|Berlin

Table 4: Draw for the pot D

Normally, all the abbreviations are straightforward, except maybe YB which stand for BSC Young Boys.

4 Conclusion

In this project, we explored and developed two distinct methodologies for scheduling the matches in the new format of the UEFA Champions League. The goal was to create a balanced and feasible schedule for 36 teams divided into 4 pots, while adhering to a set of constraints designed to ensure fairness and competitive integrity.

4.1 Method 1: Schedule-First Approach

The first method, the schedule-first approach, involved creating a pre-defined 8-day calendar with placeholders for the teams. This method required us to develop an optimal template schedule and then assign teams to these placeholders while respecting all constraints. Through linear programming and the use of the Gurobi solver, we were able to generate a schedule that met all the necessary criteria:

- Each team plays exactly 8 matches, 2 against each pot.
- No team plays against more than two teams from the same country.

- Matches are evenly distributed over the 8 match days.
- Home-away alternation is maintained for all but four teams, minimizing breaks.

Despite the success in meeting these criteria, the schedule-first approach revealed significant limitations. The rigid structure made it challenging to accommodate additional constraints such as stadium availability and weather conditions. This inflexibility highlighted the need for a more adaptable method.

4.2 Method 2: Matches-First Approach

In response to the limitations of the first method, we developed the matches-first approach. This method focuses on determining the matches *a priori*, ensuring that each draw adheres to the constraints before scheduling the match days. By using a sequential draw process and incorporating the Gurobi optimizer, we ensured that:

- Each team's opponents are chosen in a manner that prevents deadlocks and infeasibility.
- Constraints related to nationality and the number of matches against teams from the same country are respected.
- The resulting matches can be scheduled flexibly, accommodating external constraints more effectively.

This approach demonstrated greater flexibility and adaptability, crucial for handling real-world constraints that arise after the initial draw. However, it also introduced the potential for infeasibility if not carefully managed, as shown in our theoretical analysis using graph theory.

4.3 Key Findings

Our exploration has led to several key findings:

1. **Feasibility and Flexibility**: While the schedule-first approach guarantees a structured schedule, the matches-first approach offers the necessary flexibility to adapt to unforeseen constraints, making it more practical for real-world implementation.
2. **Graph Theory Insights**: The application of graph theory provided valuable insights into the feasibility of scheduling matches within the constraints, highlighting potential pitfalls and ensuring robustness in our methodology.
3. **Optimization Techniques**: Leveraging optimization techniques such as linear programming and the Gurobi solver proved essential in handling the complex constraints involved in the scheduling process.