American Journal of Computational Mathematics, 2024, 14, 401-423
https://www.scirp.org/journal/ajcm
ISSN Online: 2161-1211
ISSN Print: 2161-1203

An Integer Programming Approach for Scheduling a Professional Sports League

Vardges Melkonian

Department of Mathematics, Ohio University, Athens, OH, USA
Email: melkonia@ohio.edu

How to cite this paper: Melkonian, V. (2024) An Integer Programming Approach for Scheduling a Professional Sports League. *American Journal of Computational Mathematics*, **14**, 401-423.
https://doi.org/10.4236/ajcm.2024.144021

Received: December 1, 2024
Accepted: December 24, 2024
Published: December 27, 2024

Abstract

This paper gives integer linear programming (ILP) models for scheduling the League Phase of one of the most popular professional club competitions in the world, UEFA Champion's League. There are 36 teams in the competition, but each team plays only 8 other teams in the League Phase. Thus, the difficulty or ease of a team's opponents, known as strength of schedule (SOS), compared to other teams will be different. Our main ILP model aims to minimize the maximum difference between SOS of any two teams, thus making the schedule as fair as possible. We also give a model for creating a timetable of all the match-ups obtained by the first model. The models were implemented and tested using optimization software AMPL. Our main model obtained a schedule with a difference 0.4 between the highest and the lowest SOS, while that difference is 19 for the actual 2024-2025 competition. Thus, our model returns a schedule that is significantly fairer compared to the actual competition.

Keywords

Sport Scheduling, Optimization Modeling, Integer Linear Programming

1. Introduction

UEFA Champion's League (**CL**) is one of the most popular club soccer competitions in the world. In 2024, **UEFA** (Union of European Football Associations) made significant changes in the format of the competition [1]. The traditional group format was replaced by a single-league format. There are 36 teams participating in the **League Phase** of the competition. Those 36 teams are separated into four pots of 9 teams, based on their UEFA coefficients. Pot 1 includes the teams with the highest 9 coefficients, while Pot 4 includes the teams with the lowest 9 coefficients. Each team plays against two teams from each of the four pots: one at home and one away. After the completion of the League Phase, the top 24 teams advance to the Knockout Phase, with top 8 teams receiving a bye to the round of 16.

Each team plays against 8 other teams (out of 36) in the League Phase, and those teams are determined by a random draw. Since teams will get different opponents, the average coefficients of their opponents will be different. Thus, the teams will have different strength of schedule (**SOS**) in the League Phase. SOS measures the difficulty or ease of a team's opponents compared to other teams and is often used in different professional or college sports leagues. Then for the CL League Phase, it is important to have a fair and balanced schedule of all matchups in a sense that the differences between SOS of the 36 teams are not too big. To achieve that, we develop an integer linear programming (**ILP**) model for finding a schedule that minimizes the difference between the highest and the lowest SOS.

We present three ILP models in this paper. UEFA has requirements about having a valid schedule for the League Phase and about the number of matches played against own and other associations. Our first model, called **Basic model**, creates a matchup schedule for the League Phase that has a valid format and satisfies all the UEFA requirements. The Basic model does not address yet the issue of fairness discussed above. It can be used as a prototype to build other models for the League Phase in case UEFA develops new requirements or if a user of the model wants to obtain a schedule satisfying other criteria (other than fairness). Our second model, called **Fair model**, builds on the Basic model and adds new features to find a valid schedule that minimizes the difference between the highest and the lowest strengths of schedule. The League Phase is played in eight match weeks, with each match week normally having two matchdays. Our third model, called **Timetable model**, takes the schedule of all matchups determined by the Fair (or Basic) model and finds a time schedule for playing all those matches.

Our models were implemented using Optimization Modeling Language AMPL [2] and run on NEOS server using ILP solver Gurobi [3]. We obtained a schedule of all matchups for the League Phase by running the Fair model and a timetable for all the matchups by running the Timetable model. Our schedule significantly reduces the difference between the highest and the lowest SOS when compared to the actual 2024-25 Champion's League competition [4]. That difference is 0.4 for our model, while the difference for the actual competition is 19. The standard deviation for SOS of all 36 teams is 0.13 for our model versus 4.58 for the actual competition. The Fair model returns a matchup schedule within a few minutes, while the Timetable model returns a timetable within a few seconds.

The application of mathematical techniques for scheduling sporting events has been studied extensively. Several surveys of those techniques are given in [5]-[9]. [10] gives an overview of scheduling soccer competitions in Europe. Linear and integer programming have been used for scheduling different sports competitions, including round robin tournaments [11], soccer leagues [12]-[14], National Hockey League (NHL) [15], baseball playoffs [16]. The issue of fairness has been addressed in different contexts, including travel distance fairness [17], fair referee assignments [18] [19]. [20] gives optimization models for fair scheduling of recreational doubles group competitions. This paper applies fair scheduling techniques to a professional sports league, League Phase of UEFA Champion's League. The new format for the League Phase was introduced by UEFA only in the Summer of 2024, and we are not aware of any other mathematical modeling results about creating a fair schedule for the league or for other competitions with a similar format.

The paper is organized as follows. Sections 2, 3, and 4 give the Basic, Fair, and Timetable models correspondingly. Sections 5 gives a discussion of the computational results. Section 6 discusses some future directions. The AMPL implementations of our models along with some outputs are given in the Appendix.

2. The Basic Model for Creating a Valid Schedule for the League Phase

In this section, we present a model that creates a valid schedule for the League Phase, that is, a schedule of all matchups in the League Phase satisfying all the requirements of UEFA. The issue of fairness is not addressed in this model yet. The model does not also give a timetable for playing the matches. Recall that our main goal is to create and analyze the Fair model given in Section 3. But we still present the Basic model separately because one can use the Basic model as a prototype to build other models if they want to explore and implement other ideas of creating a CL schedule.

Input data.

We have the following sets and parameters that form the input for the model.

There are four pots $\boldsymbol{P} = \{1, 2, 3, 4\}$, each with 9 teams.

Let $\boldsymbol{T_1}$, $\boldsymbol{T_2}$, $\boldsymbol{T_3}$, $\boldsymbol{T_4}$ be the sets of teams in pots 1, 2, 3, 4 correspondingly.

Let $\boldsymbol{T}$ be the set of all teams: $T = T_1 \cup T_2 \cup T_3 \cup T_4$.

Let $\boldsymbol{A}$ be the set of associations whose teams participate in the League Phase.

Let binary parameter $\boldsymbol{b(t, a)}$ be 1 if team t represents association a, and 0 otherwise.

Variables.

We have the following set of binary variables.

1) Let $\boldsymbol{match[t, s]}$ ($t \in T$, $s \in T$, $t \neq s$) be a binary variable that equals 1 team t plays with team s as a home game (and it is an away game for team s), and 0 otherwise.

Constraints.

Constraints (C1)-(C3) provide that the selected matchups are consistent with the current format of the League Phase.

(C1) *One home game against each pot*. For any team $t \in T$ and any pot $p \in P$,

$$\sum_{s \in T_p : s \neq t} match[t, s] = 1$$

This constraint provides that each team t plays exactly one home game against a team from each pot p.

(C2) *One away game against each pot*. For any team $t \in T$ and any pot $p \in P$,

$$\sum_{s \in T_p : s \neq t} match[s,t] = 1$$

This constraint provides that each team t plays exactly one away game against a team from each pot p.

(C3) *At most one game between two teams*. For any team $t \in T$, $s \in T$ ($t \neq s$),

$$match[s,t] + match[t,s] \leq 1$$

This constraint provides that any two teams play at most one game with each other. Note that constraints (C1) and (C2) would still allow two games between given two teams, one home game and one away game.

Constraints (C4), (C5) provide that the selected matchups are consistent with the requirements of UEFA associations.

(C4) *No game against own association*. For any team $t \in T$ and association $a \in A$ such that $b(t,a) = 1$ (team t is from association a),

$$\sum_{s \in T : s \neq t, b[s,a]=1} match[t,s] = 0$$

This constraint provides that team t cannot play games against teams from its own association.

(C5) *At most two games against same association*. For any team $t \in T$ and association $a \in A$ such that $b(t,a) = 0$ (team t is not from association a),

$$\sum_{s \in T : s \neq t, b[s,a]=1} \left(match[t,s] + match[t,s] \right) \leq 2$$

This constraint provides that team t can play at most two games against teams from the same association.

No objective function is needed: the objective function can be an arbitrary constant. The goal of this model is to find any feasible schedule satisfying all the UEFA requirements.

The AMPL implementation of the full model is given in the Appendix section A.1.

3. A Fair Model for Creating a League Phase Schedule

In this model, we implement the idea of creating a fair schedule. We define fairness in the following way. First, we compute the average UEFA coefficient of all 36 teams participating in the League Phase. We define a strength of schedule (SOS) of any team t as the average coefficient of all 8 teams that team t plays in the League Phase. Note that SOS is a variable in our model. Then we require that the difference between the average coefficient and SOS of any team is minimized. It automatically implies that the difference between the highest and the lowest SOS is minimized, and thus makes the matchup schedule as fair as possible for all

36 teams.

Input data.

We still have all the data structures introduced for the Basic model and add the following new structures.

Let parameter ***coef*(*t*)** be the UEFA club coefficient of team t just before the draw of the League Phase.

Let parameter ***avg_coef*** be the average coefficient of all 36 teams playing in the League Phase.

$$avg\_coef = \frac{\sum_{t \in T} coef[t]}{36}$$

Variables.

Let ***str_of_sch*[*t*]** be the strength of schedule of team t in the League Phase schedule. It is the average coefficient of the teams that will playing with team t and is determined by the model.

Let **max_dev** be the maximum deviation of *str_of_sch*[*t*] from *avg_coef* for any team t.

$$\text{max\_dev} = \max_{t \in T} \left| str\_of\_sch[t] - avg\_coef \right| \quad (1)$$

Note that ideally, we would like to have the strength of schedule of any team close to *avg_coef*. Then our goal is to create a fair schedule by minimizing max_dev.

Objective function.

The objective function minimizes max_dev, the maximum deviation of *str_of_sch*[*t*] from *avg_coef* for any team t.

Minimize max_dev

The expression for max_dev given in (1) is nonlinear. To linearize it, the objective function is set to minimize just variable max_dev while constraint (C7) below provides that max_dev is equal to the right-hand side expression of (1) in any optimal solution.

Constraints.

The Fair model still has constraints (C1)-(C5) of the Basic model. But there are also new constraints which provide that the team schedules are more balanced and fair.

(C6) *Strength of schedule*. For any team t,

$$str\_of\_sch[t] = \sum_{s \in T: s \neq t} \left(coef[s] * match[t,s] + coef[s] * match[t,s] \right) / 8$$

Strength of schedule of each team t is equal to the average coefficient of the 8 teams that team t plays in the League Phase.

(C7) *Linearizing the expression for max_dev*. We have the following two constraints for every team t,

$$\text{max\_dev} \geq str\_of\_sch[t] - avg\_coef$$

$$\text{max\_dev} \geq avg\_coef - str\_of\_sch[t]$$

These two constraints together imply that $\max\_dev \ge \max_{t \in T} \left| str\_of\_sch[t] - avg\_coef \right|$. But in any optimal solution we will have $\max\_dev = \max_{t \in T} \left| str\_of\_sch[t] - avg\_coef \right|$ since the objective function minimizes max_dev, and max_dev is not in any other constraint.

The AMPL implementation of the full model is given in the Appendix section A.2.

4. Model for Creating a Timetable for the League Phase

The Basic and Fair models create a schedule of all possible matchups in the League Phase. Next, we create a specific timetable for playing all the matches. There are 8 match weeks, and there are two matchdays, normally Tuesday and Wednesday, in each match week. Each team plays exactly one game in each match week. UEFA also has extra requirements for creating a timetable. All these requirements are reflected in our model.

Input data.

We still have the input data structures introduced in the Basic and Fair models and add the following new structures.

A main difference from the Basic and Fair models is that all the matchups are known, since they were determined by the output of the Fair model. Thus, the matchups are defined as binary parameters for the Timetable model.

Let ***match*[*t,s*]** ($t \in T$, $s \in T$, $t \neq s$) be a binary *parameter* that equals 1 if team t plays with team s as a home game (and it is an away game for team s), and 0 otherwise.

Let $\boldsymbol{W}$ be the set of game weeks of the League Phase. In the current Champion's League format, there are eight game weeks: 1, 2, 3, 4, 5, 6, 7, 8.

Let $\boldsymbol{D}$ be the set of gamedays in each game week. In the current format, there are two gamedays in each game week: 1 (Tuesday) and 2 (Wednesday).

Variables.

We have the following set of binary variables.

Let ***time*[*t, s, w, d*]** ($t \in T$, $s \in T$ such that $t \neq s$, $match[t,s] = 1$, $w \in W$, $d \in D$) be a binary *variable* that equals 1 if the match between team t (home team) and team s (away team) is played on gameday d of game week w, and 0 otherwise.

Constraints.

We do not need any of the constraints from the Basic or Fair models to be included in this new model. The Timetable model has the following sets of constraints.

(T1) *Each team one game each week.* For any team $t \in T$ and any game week $w \in W$,

$$\sum_{\substack{s \in T, d \in D: \\ s \neq t \text{ and } match[t,s]=1}} time[t,s,w,d] + \sum_{\substack{s \in T, d \in D: \\ s \neq t \text{ and } match[s,t]=1}} time[s,t,w,d] = 1$$

This constraint provides that each team t plays exactly one game in each game week w. The first summation gives the number of home games of team t in any matchday of game week *w*, and second summation gives the number of away games of team *t* in any matchday of game week *w*.

(T2) *Each matchup exactly once*. For any teams $t \in T$, $s \in T$ such that $t \neq s$ and $match[t,s] = 1$,

$$\sum_{w \in W, d \in D} time[t,s,w,d] = 1$$

This constraint provides that each matchup between teams *t* and *s*, as determined by the Fair model, is played exactly in one game week and gameday.

(T3) *Nine matches in each gameday*. For any $w \in W$, $d \in D$,

$$\sum_{\substack{t \in T, s \in T: \\ s \neq t \text{ and } match[t,s]=1}} time[t,s,w,d] = 9$$

There are 36 teams, and 18 matches are played in each match week. This constraint provides that those 18 matches are equally distributed between the two matchdays of the match week: exactly 9 matches are played on each matchday.

UEFA regulations require that each team should not play more than two home matches or two away matches in a row, and should play one home match and one away match across both the first and last two matchdays. Constraints (T4)-(T7) below achieve that these requirements are satisfied.

(T4) *One home game in first two match weeks*. For any team $t \in T$,

$$\sum_{\substack{s \in T, d \in D: \\ s \neq t \text{ and } match[t,s]=1}} \left(time[t,s,1,d] + time[t,s,2,d] \right) = 1$$

This constraint provides that each team *t* plays exactly one home game in any gameday of the first two match weeks. It also implies that the other game played by team *t* in the first two match weeks is an away game.

(T5) *One home game in last two match weeks*. For any team $t \in T$,

$$\sum_{\substack{s \in T, d \in D: \\ s \neq t \text{ and } match[t,s]=1}} \left(time[t,s,7,d] + time[t,s,8,d] \right) = 1$$

This constraint provides that each team *t* plays exactly one home game in any gameday of the last two match weeks. It also implies that the other game played by team *t* in the last two match weeks is an away game.

(T6) *No more than two home games in a row*. For any team $t \in T$ and $w \in W$ such that $w < 7$,

$$\sum_{\substack{s \in T, d \in D: \\ s \neq t \text{ and } match[t,s]=1}} \left(time[t,s,w,d] + time[t,s,w+1,d] + time[t,s,w+2,d] \right) \leq 2$$

This constraint provides that each team *t* plays no more than two home games in any three consecutive match weeks. It automatically implies that team *t* cannot play more than two home games in a row.

(T7) *No more than two away games in a row*. For any team $t \in T$ and $w \in W$ such that $w < 7$,

$$\sum_{\substack{s \in T, d \in D: \\ s \neq t \text{ and } match[s,t]=1}} \left(time[s,t,w,d] + time[s,t,w+1,d] + time[s,t,w+2,d] \right) \leq 2$$

This constraint provides that each team t plays no more than two away games in any three consecutive match weeks. It automatically implies that team t cannot play more than two away games in a row.

There are also regulations concerning the teams that are based in the same city. Those teams cannot play home games on the same gameday. In 2024-25 tournament, there are two pairs of teams representing the same city: Milan and Inter are based in Milan, Italy, while Real and Atletico are based in Madrid, Spain. Constraints (T8)-(T11) below achieve the requirements for those teams.

(T8) *Milan and Inter cannot both host games same gameday.* For any $w \in W$ and $d \in D$,

$$\sum_{\substack{s \in T: \\ match['Milan',s]=1}} time['Milan',s,w,d] + \sum_{\substack{s \in T: \\ match['Inter',s]=1}} time['Inter',s,w,d] \leq 1$$

This constraint provides that on any given gameday, at most one of the teams Milan and Inter can have a home game.

(T9) *Real and Atletico cannot both host games same gameday.* For any $w \in W$ and $d \in D$,

$$\sum_{\substack{s \in T: \\ match['Real',s]=1}} time['Real',s,w,d] + \sum_{\substack{s \in T: \\ match['Atlet',s]=1}} time['Atlet',s,w,d] \leq 1$$

This constraint provides that on any given gameday, at most one of the teams *Real and Atletico* can have a home game.

(T10) *Milan and Inter cannot both host games in last match week.*

$$\sum_{\substack{s \in T, d \in D: \\ match['Milan',s]=1}} time['Milan',s,8,d] + \sum_{\substack{s \in T, d \in D: \\ match['Inter',s]=1}} time['Inter',s,8,d] \leq 1$$

All the games in the last match week are played at the same time. Thus, this constraint provides that in the last match week, at most one of the teams Milan and Inter can have a home game.

(T11) *Real and Atletico cannot both host games in last match week.*

$$\sum_{\substack{s \in T, d \in D: \\ match['Real',s]=1}} time['Real',s,8,d] + \sum_{\substack{s \in T, d \in D: \\ match['Atlet',s]=1}} time['Atlet',s,8,d] \leq 1$$

All the games in the last match week are played at the same time. Thus, this constraint provides that in the last match week, at most one of the teams Real and Atletico can have a home game.

The AMPL implementation of the full model is given in the Appendix section A.3.

5. Computational Results

In this section, we present our computational results for the Fair and Timetable models presented in Sections 3 and 4. No computations were run for the Basic model. Recall that the Basic model is a prototype for building the Fair model (and possibly other models in the future). Our main goal is to obtain a schedule of all matchups as an output and to compare it to the actual CL schedule in terms of fairness.

The models were implemented and tested using optimization software AMPL. The AMPL models for the integer linear programs were run on the NEOS server using the solver Gurobi.

The section is organized as follows. Subsection 5.1 discusses the computational results for the Fair model, including a complete schedule of all matchups. Subsection 5.2 compares our schedule with the actual CL schedule in terms of fairness. Subsection 5.3 discusses the computational results for the Timetable model, including a timetable of all matchups.

5.1. Computational Results and Analysis for the Fair Model

In this subsection, we present our computational results for the Fair model given in Section 3. The model was implemented and tested using optimization software AMPL. The AMPL model was run on the NEOS server using the ILP solver Gurobi.

Discussion and analysis of the solution process.

The solver could not return an optimal solution for the model after several hours since it exceeded the maximum allotted time for a job. Then we tried to get good feasible solutions using the following approach. The objective function was changed from *minimize max-dev* to *minimize 1*. By minimizing a constant function, the model just tries to find a feasible solution. But to get a fair schedule with relatively small value for max_dev, we added a new constraint **max_dev ≤ *k*** where *k* was chosen to be a relatively small but safe value for getting a good feasible solution.

We ran the model for different values of k, starting from 0.5 (that value is 10.4 for the actual 2024-25 CL League Phase). We were able to obtain feasible solutions for $k = 0.5$, $k = 0.25$, $k = 0.2$. The results of those computations are summarized in Table 1.

Table 1. Highest and lowest SOS for different values for *k*.

k	$\min\{t \in T\}$ $str\_of\_sch[t]$	$\max\{t \in T\}$ $str\_of\_sch[t]$	$\max\{t \in T\}$ $str\_of\_sch[t]$ − $\min\{t \in T\}$ $str\_of\_sch[t]$	Solve times (seconds)
0.5	63.92	64.9155	0.9955	18.2
0.25	64.17	64.6526	0.4826	377.9
0.2	**64.228**	**64.6153**	**0.38725**	**1785.96**

As we can see from the table, for $k = 0.2$ the model returned a fairly good schedule with the difference between the highest and lowest SOS only 0.38725. But the running time also increased quickly when moving to lower values of *k*. Next, we ran the model for $k = 0.15$. Gurobi did not return an output after several hours and was terminated because it exceeded the maximum allotted time for a job.

In an attempt to obtain a better solution, we tried the following approach. The solution for $k = 0.2$ (namely, the values of matchup variables $match[t,s]$) was given to the model as default values for $match[t,s]$ when the model was run for smaller values of k. Having an initial solution with already a small value for max_dev significantly accelerates the process of finding a new, even better solution. The model was run for $k = 0.18$, $k = 0.19$, $k = 0.195$ in that order. In all three cases, the time limit was set to 3600 seconds. But no feasible solution was found in that time period, while the final values of variables match[t,s] after 3600 seconds were the same as the initial default values. We also ran the model by having the $k = 0.2$ solution as input and going back to the original objective function minimize max_dev (without any extra constraints on max_dev); after 3600 seconds and millions of simplex iterations, the solver was not able to obtain a better solution than the input.

Based on the discussion above, our conjecture is that the solution found for $k = 0.2$ is either the optimal solution (most likely scenario) or very close to the optimal. It is possible that the ILP model is highly degenerate with multiple optimal (or near optimal) basic solutions which causes cycling problems when the solution with $k = 0.2$ is given as an input.

Thus, the solution for $k = 0.2$ is the best possible solution found by our computations. It is presented below and is given as input to the Timetable model of Section 5.

The description of the best solution found.

The schedule of all matchups for the League Phase is given in Table 2. The last two columns give SOS for all 36 teams for (i) the output of our model, (ii) the actual 2024-25 CL League Phase. In our schedule, club Salzburg has the highest SOS, 64.62, and club Brest has the lowest SOS, 64.23. In the actual competition, club Feyenoord has the highest SOS, 74.8, and club Young Boys has the lowest SOS, 55.8.

Table 2. League Phase opponents and SOS for each club.

Club	Pot 1 opponents		Pot 2 opponents		Pot 3 opponents		Pot 4 opponents		Strength of schedule	
	Home	Away	Home	Away	Home	Away	Home	Away	our model	actual
Real	MC	Dortm	Lever	Milan	Young	Lille	Sturm	Monac	64.25	59.8
MC	Leipz	Real	Atlet	Juv	Celti	Salzb	Brest	Stutt	64.34	65.7
Bayer	Barc	Liver	Benf	Atal	Salzb	DinZ	Slova	Aston	64.55	63.4
PSG	Liver	Inter	Arsen	Atlet	DinZ	Sport	Bolog	Giron	64.56	74
Liver	Bayer	PSG	Milan	Shakh	PSV	Crven	Stutt	Spart	64.48	64.9
Inter	PSG	Leipz	Brugg	Lever	Lille	PSV	Giron	Slova	64.55	66
Dortm	Real	Barc	Atal	Arsen	Crven	Feyen	Spart	Sturm	64.25	58.6
Leipz	Inter	MC	Juv	Brugg	Sport	Young	Aston	Brest	64.53	63.2

Continued

Barc	Dortm	Bayer	Shakh	Benf	Feyen	Celti	Monac	Bolog	64.26	64.1
Lever	Inter	Real	Benf	Arsen	Celti	PSV	Bolog	Spart	64.32	63.2
Atlet	PSG	MC	Brugg	Milan	Crven	Feyen	Sturm	Bolog	64.57	66.5
Atal	Bayer	Dortm	Shakh	Brugg	Feyen	Lille	Spart	Aston	64.42	57.5
Juv	MC	Leipz	Arsen	Shakh	DinZ	Sport	Brest	Giron	64.47	65.9
Benf	Barc	Bayer	Milan	Lever	PSV	Young	Giron	Monac	64.3	67.9
Arsen	Dortm	PSG	Lever	Juv	Sport	Crven	Monac	Sturm	64.5	63.4
Brugg	Leipz	Inter	Atal	Atlet	Lille	Salzb	Aston	Slova	64.55	63.2
Shakh	Liver	Barc	Juv	Atal	Salzb	DinZ	Slova	Stutt	64.23	64.2
Milan	Real	Liver	Atlet	Benf	Young	Celti	Stutt	Brest	64.4	67.8
Feyen	Dortm	Barc	Atlet	Atal	Sport	Lille	Slova	Monac	64.25	**74.8**
Sport	PSG	Leipz	Juv	Arsen	PSV	Feyen	Spart	Giron	64.55	64.3
PSV	Inter	Liver	Lever	Benf	Crven	Sport	Stutt	Aston	64.59	62.3
DinZ	Bayer	PSG	Shakh	Juv	Celtic	Young	Sturm	Slova	64.31	63.6
Salzb	MC	Bayer	Brugg	Shakh	Young	Celti	Bolog	Brest	**64.62**	71.7
Lille	Real	Inter	Atal	Brugg	Feyen	Crven	Giron	Stutt	64.28	70.4
Crven	Liver	Dortm	Arsen	Atlet	Lille	PSV	Monac	Bolog	64.38	57.5
Young	Leipz	Real	Benf	Milan	DinZ	Salzb	Aston	Spart	64.3	**55.8**
Celtic	Barc	MC	Milan	Lever	Salzb	DinZ	Brest	Sturm	64.49	59.4
Slova	Inter	Bayer	Brugg	Shakh	DinZ	Feyen	Spart	Brest	64.36	69.7
Monac	Real	Barc	Benf	Arsen	Feyen	Crven	Aston	Bolog	64.24	59
Spart	Liver	Dortm	Lever	Atal	Young	Sport	Sturm	Slova	64.5	70.7
Aston	Bayer	Leipz	Atal	Brugg	PSV	Young	Giron	Monac	64.55	61.7
Bolog	Barc	PSG	Atlet	Lever	Crven	Salzb	Monac	Sturm	64.31	62.4
Giron	PSG	Inter	Juv	Benf	Sport	Lille	Stutt	Aston	64.46	64.6
Stutt	MC	Liver	Shakh	Milan	Lille	PSV	Brest	Giron	64.53	67.6
Sturm	Dortm	Real	Arsen	Atlet	Celti	DinZ	Bolog	Spart	64.57	59
Brest	Leipz	MC	Milan	Juv	Salzb	Celti	Slova	Stutt	**64.23**	65.1

The model has 1215 variables, 1178 of them binary, and 1834 constraints. The Gurobi solve time to find the solution was 1785.96 seconds. The actual time of getting the output from NEOS server was about 8 minutes. The solution was found after 307801 branching nodes and more than 33 million simplex iterations.

The full output from the solver is given in the Appendix section A.4.

5.2. Comparison of Our Solution with the Actual CL Schedule

Table 3 gives a comparison of our schedule with the actual schedule of 2024-25 CL League Phase. We give different statistics about strengths of schedules. The lowest and highest SOS of actual CL schedule are 55.8 (Young Boys) and 74.8 (Feyenoord), with a range 74.8 - 55.8 = 19. Thus, the opponents of Feyenoord on average are significantly stronger than the opponents of Young Boys. On the other hand, the lowest and highest SOS of our schedule are 64.2 (Brest) and 64.6 (Salzburg), with a range 64.6 - 64.2 = 0.4. Thus, the difference between SOS is much smaller in our schedule. The standard deviation of all SOS for actual competition is 4.58, while it is only 0.13 for our schedule. Based on all those numbers, our schedule is significantly more balanced and fairer (in terms of SOS) than the actual CL schedule.

Table 3. Comparison of strengths of schedules

	Actual CL competition	Output of our model
Minimum	64.2	55.8
Maximum	64.6	74.8
Range	0.4	19
Standard deviation	0.13	4.58
Coefficient of variance	0.002	0.071

5.3. Computational Results for the Timetable Model

In this subsection, we give the timetable of playing all the matches as returned by our Timetable model of Section 4. The schedule of all matchups of Table 2 was given as an input to the Timetable model. The model has 2304 binary variables and 986 constraints. The ILP solver needed 1741 branching nodes and 193708 simplex iterations to obtain the solution. The Gurobi solve time was 31.11 seconds.

The timetable found by the solver is given in Table 4.

Table 4. The timetable found by our model.

(a)			
Matchweek 1			
Matchday 1		Matchday 2	
Bayer	Benf	Real	Young
Barc	Monac	MC	Leipz
Brugg	Atal	PSG	DinZ
Shakh	Slova	Inter	Lille
Milan	Atlet	Lever	Bolog
Sport	Juv	Feyen	Dortm
Aston	PSV	Crven	Arsen
Giron	Stutt	Spart	Liver
Brest	Salzb	Sturm	Celti

Continued

(b)

Matchweek 2			
Matchday 1		Matchday 2	
Arsen	Lever	Liver	Bayer
PSV	Inter	Dortm	Real
DinZ	Sturm	Leipz	Sport
Salzb	Brugg	Atlet	PSG
Lille	Giron	Atal	Shakh
Celtic	Barc	Juv	Brest
Slova	Spart	Benf	Milan
Bolog	Crven	Young	Aston
Stutt	MC	Monac	Feyen

(c)

Matchweek 3			
Matchday 1		Matchday 2	
Real	Sturm	PSG	Arsen
MC	Celti	Benf	PSV
Bayer	Salzb	Crven	Lille
Liver	Milan	Young	DinZ
Inter	Brugg	Monac	Aston
Dortm	Atal	Spart	Lever
Leipz	Juv	Bolog	Barc
Feyen	Atlet	Giron	Sport
Brest	Slova	Stutt	Shakh

(d)

Matchweek 4			
Matchday 1		Matchday 2	
Barc	Dortm	Inter	PSG
Atal	Bayer	Lever	Benf
Arsen	Monac	Atlet	Crven
Brugg	Leipz	Juv	MC
Shakh	Liver	Sport	Spart
Milan	Real	PSV	Stutt
Salzb	Young	Lille	Feyen
Aston	Giron	Celti	Brest
Sturm	Bolog	Slova	DinZ

Continued

(e)			
Matchweek 5			
Matchday 1		Matchday 2	
Bayer	Barc	Real	MC
Benf	Giron	Liver	PSV
Brugg	Aston	Lever	Inter
Feyen	Slova	Juv	Arsen
DinZ	Celti	Shakh	Salzb
Lille	Atal	Milan	Stutt
Crven	Monac	Sport	PSG
Spart	Young	Sturm	Dortm
Bolog	Atlet	Brest	Leipz
(f)			
Matchweek 6			
Matchday 1		Matchday 2	
PSG	Liver	MC	Brest
Barc	Feyen	Dortm	Crven
Atal	Spart	Leipz	Inter
Arsen	Sport	Atlet	Sturm
DinZ	Shakh	PSV	Lever
Young	Benf	Salzb	Bolog
Slova	Brugg	Celti	Milan
Giron	Juv	Monac	Real
Stutt	Lille	Aston	Bayer
(g)			
Matchweek 7			
Matchday 1		Matchday 2	
Real	Lever	MC	Atlet
Liver	Stutt	Bayer	Slova
Inter	Giron	PSG	Bolog
Arsen	Dortm	Barc	Shakh
PSV	Crven	Juv	DinZ
Celti	Salzb	Brugg	Lille
Monac	Benf	Feyen	Sport
Aston	Atal	Young	Leipz
Brest	Milan	Spart	Sturm

Continued

(h)			
Matchweek 8			
Matchday 1		Matchday 1 (cont.)	
Dortm	Spart	Leipz	Aston
Lever	Celti	Shakh	Juv
Atlet	Brugg	Salzb	MC
Atal	Feyen	Lille	Real
Benf	Barc	Crven	Liver
Milan	Young	Slova	Inter
Sport	PSV	Bolog	Monac
DinZ	Bayer	Stutt	Brest
Giron	PSG	Sturm	Arsen

6. Conclusions and Future Directions

We created an ILP model for fair scheduling of the UEFA CL League Phase, one of the most popular club competitions in the world. Minimizing the maximum difference in strength of schedule is used as a fairness criterion. The models were implemented, tested and demonstrated significant improvements in fairness compared to the actual 2024-25 schedule. We also created a model for getting a timetable for the matchups determined by the fairness model.

Below we give some future directions grouped in three categories.

Other possible variations of the models

- Another way to obtain a fair schedule would be the following. In the Fair model, one could minimize the standard deviation instead of the maximum deviation from the average coefficient. Our expectation is that the output of that modified model will not be too different from the schedule we obtained. But that model will be nonlinear and computationally more expensive to solve. There might also be other criteria for making the schedule fairer and more balanced.
- One could use any of our models as a prototype to build other models if they want to explore and implement other ideas of creating a League Phase schedule.

Extending the models to other sports

The models can be extended to other sports leagues that have the following feature: any team plays not with every other team, but only with selected number of teams in the tournament.

Our models with slight modifications can be certainly used for other UEFA club competitions (Europa League, Conference League) that have roughly the same format as the Champion's League.

Examples of other competitions for which our modeling ideas can be applied are National Football League (NFL), and College Football (NCAAF). NFL has a total of 32 teams. But every team plays only with 17 teams (which includes playing twice with 3 teams in its own division). NCAAF includes many conferences with a large number of teams, for example, Big Ten, and SEC. Not every pair of teams play with each other within a conference. For example, Big Ten has 18 teams, but each team plays only against 9 of them in the regular season. Thus, the concept of fairness can be applied to creating matchup schedules for these two leagues too.

Computational issues

We were able to find a close-to-optimal solution for our Fair model, and that solution had a significantly better maximum SOS deviation compared to the actual Phase League competition. But the solver could not find an optimal solution after the maximum allotted time on NEOS server. Improving the running time and finding an optimal solution for the Fair model is an open question. This might be achieved by adding cutting planes, reducing the number of binary variables, using other solvers, etc.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] UEFA Website: New Format for Champions League Post-2024: Everything You Need to Know. https://www.uefa.com/uefachampionsleague/news/0268-12157d69ce2d-9f011c70f6fa-1000--new-format-for-champions-league-post-2024-everything-you-ne/

[2] The AMPL Website. http://www.ampl.com/

[3] The Gurobi Website. https://neos-server.org/neos/solvers/milp:Gurobi/AMPL.html

[4] 2024-25 UEFA CL League Phase. https://en.wikipedia.org/wiki/2024%E2%80%9325_UEFA_Champions_League_league_phase

[5] Dinitz, J.H., Froncek, D., Lamken, E.R. and Wallis, W.D. (2006) Scheduling a Tournament. In: Colbourn, C.J. and Dinitz, J.H., Eds., *Handbook of Combinatorial Designs*, CRC Press, 617-631.

[6] Easton, K., Nemhauser, G. and Trick, M. (2004) Sports Scheduling. In: Leung, J.T., Ed., *Handbook of Scheduling*, CRC Press, 1-19.

[7] Kendall, G., Knust, S., Ribeiro, C.C. and Urrutia, S. (2010) Scheduling in Sports: An Annotated Bibliography. *Computers & Operations Research*, **37**, 1-19. https://doi.org/10.1016/j.cor.2009.05.013

[8] Ribeiro, C.C. (2012) Sports Scheduling: Problems and Applications. *International Transactions in Operational Research*, **19**, 201-226. https://doi.org/10.1111/j.1475-3995.2011.00819.x

[9] Wright, M.B. (2009) 50 Years of or in Sport. *Journal of the Operational Research Society*, **60**, S161-S168. https://doi.org/10.1057/jors.2008.170

[10] Goossens, D.R. and Spieksma, F.C.R. (2011) Soccer Schedules in Europe: An Overview. *Journal of Scheduling*, **15**, 641-651. https://doi.org/10.1007/s10951-011-0238-9

[11] Briskorn, D. and Drexl, A. (2009) IP Models for Round Robin Tournaments. *Computers & Operations Research*, **36**, 837-852. https://doi.org/10.1016/j.cor.2007.11.002

[12] Alarcón, F., Durán, G. and Guajardo, M. (2013) Referee Assignment in the Chilean Football League Using Integer Programming and Patterns. *International Transactions in Operational Research*, **21**, 415-438. https://doi.org/10.1111/itor.12049

[13] Della Croce, F. and Oliveri, D. (2006) Scheduling the Italian Football League: An ILP-Based Approach. *Computers & Operations Research*, **33**, 1963-1974. https://doi.org/10.1016/j.cor.2004.09.025

[14] Durán, G., Guajardo, M. and Sauré, D. (2017) Scheduling the South American Qualifiers to the 2018 FIFA World Cup by Integer Programming. *European Journal of Operational Research*, **262**, 1109-1115. https://doi.org/10.1016/j.ejor.2017.04.043

[15] Fleurent, C. and Ferland, J.A. (1993) Allocating Games for the NHL Using Integer Programming. *Operations Research*, **41**, 649-654. https://doi.org/10.1287/opre.41.4.649

[16] Robinson, L.W. (1991) Baseball Playoff Eliminations: An Application of Linear Programming. *Operations Research Letters*, **10**, 67-74. https://doi.org/10.1016/0167-6377(91)90089-8

[17] Durán, G.A., Guajardo, M., López, A.F., Marenco, J. and Zamorano, G.A. (2021) Scheduling Multiple Sports Leagues with Travel Distance Fairness: An Application to Argentinean Youth Football. *INFORMS Journal on Applied Analytics*, **51**, 136-149. https://doi.org/10.1287/inte.2020.1048

[18] Mancini, S. and Isabello, A. (2014) Fair Referee Assignment for the Italian Soccer Seriea. *Journal of Quantitative Analysis in Sports*, **10**, 153-160. https://doi.org/10.1515/jqas-2013-0108

[19] Yavuz, M., İnan, U.H. and Fığlalı, A. (2008) Fair Referee Assignments for Professional Football Leagues. *Computers & Operations Research*, **35**, 2937-2951. https://doi.org/10.1016/j.cor.2007.01.004

[20] Melkonian, V. (2021) Fair Scheduling Models for Doubles Group Competitions. *American Journal of Operations Research*, **11**, 338-356. https://doi.org/10.4236/ajor.2021.116021

Appendix A. AMPL Programs and Outputs for Our Models

A.1. AMPL Program for the Basic Model

```
set pot_n;
# set of pot numbers

set pot{n in pot_n} ordered;
# set of pots

set all_teams := union{n in pot_n} pot[n] ordered;

param coefficient{all_teams};

set countries;

param team_country{t in all_teams, c in countries} binary;

var matches{t1 in all_teams,t2 in all_teams: t1 != t2} binary default 0;
# is 1 if team t1 plays team t2 as a home game
#let matches['Real_M','Bayern'] := 1;

minimize something: 1;

subject to one_home_game_against_each_pot {t1 in all_teams, n in pot_n}:
    sum{t2 in pot[n]: t2 != t1} matches[t1,t2] = 1;
# each team plays exactly one home game against a team from each pot

subject to one_away_game_against_each_pot {t1 in all_teams, n in pot_n}:
    sum{t2 in pot[n]: t2 != t1} matches[t2,t1] = 1;
# each team plays exactly one away game against a team from each pot

subject to at_most_one_match_between_two_teams {t1 in all_teams, t2 in
all_teams : t2 != t1}:
    matches[t1,t2] + matches[t2,t1] <= 1;
# any two teams play at most one game against each other

subject to no_game_against_same_association{t1 in all_teams, c in countries:
team_country[t1,c]=1}:
    sum{t2 in all_teams: t2 != t1 and team_country[t2,c]=1} matches[t1,t2] = 0;
# no games are played against the teams of the same association

subject to no_more_than_two_games_against_other_associations
      {t1 in all_teams, c in countries: team_country[t1,c]=0}:
```

```
        sum{t2 in all_teams: t2 != t1 and team_country[t2,c]=1}
(matches[t1,t2]+matches[t2,t1]) <= 2;
# no more than two games against the teams of any other association
```

A.2. AMPL Program for the Fair Model

```
set pot_n;
# set of pot numbers

set pot{n in pot_n} ordered;
# set of pots

set all_teams := union{n in pot_n} pot[n] ordered;

param coefficient{all_teams};

param avgS := sum{t in all_teams} coefficient[t] / card(all_teams);
# average expected strength of schedule

set countries;

param team_country{t in all_teams, c in countries} binary;

var matches{t1 in all_teams,t2 in all_teams: t1 != t2} binary default 0;
# is 1 if team t1 plays team t2 as a home game

var str_of_sch{t in all_teams};
# strength of schedule of team t

var max_dev;
# maximum deviation from average strength of schedule

minimize Maximum_Deviation: max_dev;

subject to strength_of_schedule{t1 in all_teams}:
    str_of_sch[t1] =
     sum{t2 in all_teams: t2 != t1}
        (coefficient[t2]*matches[t1,t2]+coefficient[t2]*matches[t2,t1])/8;

subject to deviation1{t in all_teams}:
    str_of_sch[t] - avgS <= max_dev;

subject to deviation2{t in all_teams}:
    avgS - str_of_sch[t] <= max_dev;
```

```
#subject to max_dev_limit1: max_dev <= .2;

subject to one_home_game_against_each_pot {t1 in all_teams, n in pot_n}:
    sum{t2 in pot[n]: t2 != t1} matches[t1,t2] = 1;
# each team plays exactly one home game against a team from each pot

subject to one_away_game_against_each_pot {t1 in all_teams, n in pot_n}:
    sum{t2 in pot[n]: t2 != t1} matches[t2,t1] = 1;
# each team plays exactly one away game against a team from each pot

subject to at_most_one_match_between_two_teams {t1 in all_teams, t2 in
all_teams : t2 != t1}:
    matches[t1,t2] + matches[t2,t1] <= 1;
# any two teams play at most one game against each other

subject to no_game_against_same_association{t1 in all_teams, c in countries:
team_country[t1,c]=1}:
    sum{t2 in all_teams: t2 != t1 and team_country[t2,c]=1} matches[t1,t2] = 0;
# no games are played against the teams of the same association

subject to no_more_than_two_games_against_other_associations
        {t1 in all_teams, c in countries: team_country[t1,c]=0}:
            sum{t2 in all_teams: t2 != t1 and team_country[t2,c]=1}
(matches[t1,t2]+matches[t2,t1]) <= 2;
# no more than two games against the teams of any other association
```

A.3. AMPL Program for the Timetable Model

```
set pot_n;
# set of pot numbers

set pot{n in pot_n} ordered;
# set of pots

set all_teams := union{n in pot_n} pot[n] ordered;

set countries;

param team_country{t in all_teams, c in countries} binary;

param matches{t1 in all_teams,t2 in all_teams: t1 != t2} binary;
# is 1 if team t1 plays team t2 as a home game

set gameweek ordered;
```

```
set gameday ordered;

var time{t1 in all_teams,t2 in all_teams, w in gameweek, d in gameday:
    t1 != t2 and matches[t1,t2]==1} binary;

minimize something: 1;

subject to one_game_each_gameweek{t1 in all_teams, w in gameweek}:
sum{t2 in all_teams, d in gameday: t1 != t2 and matches[t1,t2]==1}time[t1,t2,w,d]
+
sum{t2 in all_teams, d in gameday: t1 != t2 and matches[t2,t1]==1}time[t2,t1,w,d]
= 1;
# each team plays exactly one game in each gameweek

subject to each_matchup_exactly_once
    {t1 in all_teams,t2 in all_teams: t1 != t2 and matches[t1,t2]==1}:
        sum{w in gameweek, d in gameday}time[t1,t2,w,d] = 1;
# each matchup between teams t1 and t2 must be played exactly once

subject to nine_games_each_gameday{w in gameweek, d in gameday}:
    sum{t1 in all_teams,t2 in all_teams: t1 != t2 and
matches[t1,t2]==1}time[t1,t2,w,d] = 9;
# total number of games each gameday should be exactly 9

subject to one_home_game_in_first_two_rounds{t1 in all_teams}:
    sum{t2 in all_teams, d in gameday: t1 != t2 and matches[t1,t2]==1}
        (time[t1,t2,1,d]+time[t1,t2,2,d]) = 1;
# each team plays exactly one home game in the first two rounds

subject to one_home_game_in_last_two_rounds{t1 in all_teams}:
    sum{t2 in all_teams, d in gameday: t1 != t2 and matches[t1,t2]==1}
        (time[t1,t2,7,d]+time[t1,t2,8,d]) = 1;
# each team plays exactly one home game in the last two rounds

subject to no_more_than_two_home_matches_in_a_row{t1 in all_teams, w in
gameweek: w<=6}:
    sum{t2 in all_teams, d in gameday: t1 != t2 and matches[t1,t2]==1}
        (time[t1,t2,w,d]+time[t1,t2,w+1,d]+time[t1,t2,w+2,d]) <= 2;
# each team can play no more than two home games in a row

subject to no_more_than_two_away_matches_in_a_row{t1 in all_teams, w in
gameweek: w<=6}:
    sum{t2 in all_teams, d in gameday: t1 != t2 and matches[t2,t1]==1}
```

```
(time[t2,t1,w,d]+time[t2,t1,w+1,d]+time[t2,t1,w+2,d]) <= 2;
# each team can play no more than two away games in a row

subject to Milan_teams{w in gameweek, d in gameday}:
    sum{t in all_teams: t!='Milan' and matches['Milan',t]==1}time['Milan',t,w,d]
+
    sum{t in all_teams: t!='Inter' and matches['Inter',t]==1}time['Inter',t,w,d] <=
1;
# Milan and Inter (both teams are based in city of Milan)
# cannot both host games on the same game day

subject to Milan_teams_last_week:
    sum{t in all_teams,d in gameday: t!='Milan' and
matches['Milan',t]==1}time['Milan',t,8,d] +
    sum{t in all_teams,d in gameday: t!='Inter' and
matches['Inter',t]==1}time['Inter',t,8,d] <= 1;
# Milan and Inter (both teams are based in city of Milan)
# cannot both host games in the last game week

subject to Madrid_teams{w in gameweek, d in gameday}:
    sum{t in all_teams: t!='Real_M' and
matches['Real_M',t]==1}time['Real_M',t,w,d] +
    sum{t in all_teams: t!='Atl_M' and
matches['Atl_M',t]==1}time['Atl_M',t,w,d] <= 1;
# Real_M and Atl_M (both teams are based in city of Madrid)
# cannot both host games on the same game day

subject to Madrid_teams_last_week:
sum{t in all_teams,d in gameday: t!='Real_M' and
matches['Real_M',t]==1}time['Real_M',t,8,d] + sum{t in all_teams,d in gameday:
t!='Atl_M' and matches['Atl_M',t]==1}time['Atl_M',t,8,d] <= 1;
# Real_M and Atl_M (both teams are based in city of Madrid)
# cannot both host games in the last game week
```

A.4. The AMPL Output of the Fair Model for k = 0.2

```
You are using the solver gurobi_ampl.
Checking ampl.mod for gurobi_options...
Checking ampl.com for gurobi_options...
Executing AMPL.
processing data.
processing commands.
Executing on prod-exec-1.neos-server.org

Presolve eliminates 399 constraints and 82 variables.
```

Adjusted problem:
1215 variables:
1178 binary variables
37 linear variables
1834 constraints, all linear; 9044 nonzeros
324 equality constraints
1510 inequality constraints
1 linear objective; 0 nonzeros.

Gurobi 11.0.3: tech:threads = 4
Gurobi 11.0.3: optimal solution; objective 1
3.323e+07 simplex iterations
307801 branching nodes

_total_solve_time = 1785.96

avgS = 64.4167

str_of_sch [*] :=

Real_M 64.25	Leverk 64.3195	Feyen 64.25	Slovan_B 64.3582
MC 64.3362	Atl_M 64.5695	Sporting 64.5496	Monaco 64.2395
Bayern 64.545	Atal 64.42	PSV 64.5855	Sparta_P 64.5
PSG 64.5566	Juv 64.4704	Din_Z 64.3125	Aston_V 64.5496
Liverp 64.478	Benf 64.2996	**Salzb 64.6153**	Bologna 64.3125
Inter 64.5496	Arsenal 64.5	Lille 64.2776	Girona 64.4605
Dortm 64.25	Brugge 64.545	Crvena 64.382	Stuttg 64.5329
Leipzig 64.5282	Shakhtar 64.228	Young_B 64.295	Sturm 64.5695
Barc 64.257	Milan 64.3988	Celtic 64.4832	**Brest 64.228**

;

max_dev = 0.2

min{t in all_teams} str_of_sch[t] = 64.228

max{t in all_teams} str_of_sch[t] = 64.6153

max{t in all_teams} str_of_sch[t] - min{t in all_teams} str_of_sch[t] = 0.38725