Tournament Schedule

Dailin Cheng 1002149329, Junyu Jiang 1002916561 and Yuejun Lu 1002438016

Mechanical and Industrial Engineering
University of Toronto, 5 King's College Rd. Toronto, ON M5S 3G8
dailin.cheng@mail.utoronto.ca
junyu.jiang@mail.utoronto.ca
yuejun.lu@mail.utoronto.ca

Abstract. Mixed Integer Linear Programming is a good approach for solving scheduling problem. In this project, a scheduling algorithm is investigated to create a tournament schedule for English Premier League. However, we have different rules. The objective is to minimize the total travelling distance for all teams. Mixed Integer Linear Programming with GUROBI Optimizer is used to solve this kind of problem. Due to the complexity of the problem, we focus on the Single Round Robin rather than Double Round Robin. The instances are based on the clubs of English Premier League. In the project, we will try different instances with different club number in order to get their tournament schedules and minimum travelling distance individually. We only discuss the problem that the club number is even, because if the number of participants is odd, we have to add extra games. In this project, we focus on the scheduling problem with even club number. Finally, the instance of six clubs could be run under this model to get an optimal solution within the limit time. Otherwise, the run-time will be beyond the time limit (10 minutes). Our result indicates the tournament schedule for the participants with the minimum travelling distance.

Keywords: Tournament Schedule, Mixed Integer Programming, Constraint Programming.

1 Introduction

1.1 Background

Nowadays, sports are not just leisure activities but have also become a business market with high rewards because of the development of various professional athleticism. Premier League is the kind of league that gains huge benefits from establishing a tournament scheduling model.

Clubs in the Premier League have their own home stadium located in different cities in England. In this project, the rules are different from Premier League. Each team only plays once with each other in the league, where one team plays home game, and the other team plays away game. The purpose of this project is to formulate a mathematical programming model for tournament scheduling of the English Premier League with minimum total travel distance for all teams.

1.2 Problem definition

Premier league clubs

The Premier League [1] 2020-2021 season has **m** (m is even number) clubs homed at different cities or towns in England. Table 1 shows all possible clubs of the 2020-2021 season and their location.

Table 1. clubs of the 2020-2021 season and their location.

Index	Club Name	City
0	Arsenal	London
1	Aston Villa	Birmingham
2	Brighton and Hove Albion	Brighton
3	Burley	Burley
4	Chelsea	London
5	Crystal Palace	London
6	Everton	Liverpool
7	Fulham	London
8	Leeds United	Leeds
9	Leicester City	Leicester
10	Liverpool	Liverpool
11	Manchester City	Manchester
12	Manchester United	Manchester
13	Newcastle United	Newcastle
14	Sheffield United	Sheffield
15	Southampton	Southampton
16	Tottenham Hotspur	London
17	West Bromwich Albion	West Bromwich
18	West Ham United	London
19	Wolverhampton Wanderers	Wolverhampton

Restrictions

Here are some restrictions for scheduling the Premier League tournaments:

- There are m teams in total. (m is always an even integer).
- There are (m-1) rounds of games, and every club has games in each round
- Each club plays (m-1) games in total.
- Each club has only 1 game in one round.
- Each club has games with each other once, one club plays home game and one club play away game.

Let T be the set of all teams and R be the set of all rounds of game. We have m teams and (m-1) rounds of game in total. Index of teams are 0 to m-1. Index of rounds are 0 to m-2. All teams have a home venue. If a team plays a game at its home venue, the game is called home game, while games played at other teams' venue are called away games. The distance matrix is represented by D, and Dij[1] represents the distance between the venue of team i and the venue of team j. There are no restrictions on bounding the number of home games played by a team, because adding this single constraint to the model will make the model spending more than 10 hours to solve a simple instance that considers 6 teams in the tournament. The objective of this project is to minimize the total travel distance for all the teams by using MILP [2].

2 Solution Approaches

2.1 Model Forming

Optimal Objective

Our goal is to minimize the total travel, so we set our optimal objective to be the sum of travel distances of all teams between all rounds of games. Notation for this objective is L.

Model Selection

We selected mixed integer linear programming (MILP) model to solve our scheduling problem. The single optimal objective is a sum of variables. It can be easily calculated by summation. All the game restrictions can be interpreted by variables with linear relationships. Distances between venues can be rounded to integers for simplicity without affecting the output and result expected. Therefore, MILP model is chosen for the problem.

Constraint Forming

First, we need to set constraints for the restrictions we have for the problem. The first restriction is that each team only plays with every other team once in all rounds. We introduced a decision variable to help interpret the restriction better, Z_{ijp}. Z_{ijp} equals to 1 if team i and team j plays together in a game in round p; equals to zero if they do not play game together in round p. Z_{ijp} should always eqauals to Z_{jip}.Therefore, to restrict every team only plays once with every other team, the sum of all Z_{ijp} of all rounds should be 1. Our first constraint is

$$\sum_{p \in R} Zijp = 1, \forall i, j \in T, i \neq j \tag{1}$$

The second restriction is that each team only plays once in each round. So for a certain team i, the sum of Z_{ijp} of all teams and rounds should be 1. The constraint is

$$\sum_{i \in T} Zijp = 1, \forall j \in T, i \neq j, \forall p \in R \tag{2}$$

---

[1] Dij is a symmetric matrix

The next restriction is that number of total home games played in each round should be m/2. We introduced another decision variable X_{ip} to help interpret the restriction. X_{ip} is 1 when team i plays home game in round p; equals to 0 if plays away game in round p. The restriction is

$$\sum_{i \in T} Xip = m/2, \ \forall p \in R \tag{3}$$

We need to restrict that when two teams play a game together, there should be one team playing home game and the other game playing away game. This is to restrict the sum of X_{ip} and X_{jp} to be 1 when Z_{ijp} is 1. First, set the constraint to restrict the sum of X_{ip} and X_{jp} to be less than or equal to 1 when Z_{ijp} is 1. Meanwhile, the constraint restricts the sum of X_{ip} and X_{jp} to be less than or equal to 2 when Z_{ijp} is 0. However, the value of X is 0 or 1, and the maximum of sum of X_{ip} and X_{jp} is 2, thus the constraint does not restrict the value of sum of X_{ip} and X_{jp} when when Z_{ijp} is 0. The constraint is

$$Z_{ijp} + X_{ip} + X_{jp} \leq 2, \ \forall i, j \in T, i \neq j, \ \forall p \in R \tag{4}$$

Next, we will restrict the sum of X_{ip} and X_{jp} to be greater than or equal to 1 when Z_{ijp} is 1. Meanwhile, the constraint restricts the sum of X_{ip} and X_{jp} to be greater than or equal to 0 when Z_{ijp} is 0. However, the value of X is 0 or 1, and the minimum of sum of X_{ip} and X_{jp} is 0, thus the constraint does not restrict the value of sum of X_{ip} and X_{jp} when when Z_{ijp} is 0. With constraints (4) and (5), the sum of X_{ip} and X_{jp} has to be 1 when Z_{ijp} is 1. This constraint is

$$X_{ip} + X_{jp} \geq Z_{ijp}, \ \forall i, j \in T, i \neq j, \ \forall p \in R \tag{5}$$

Having constraits interpreting the game rules, we need to set constraints for the travel distance. C_{ip} is another decision variable we implemented, representing the travel distance of team i between round p and round p+1. Moreover, p only goes from 0 to m-3 in C_{ip}. C_{ip} should always be non-negative. These constraints set the lower bound for C_{ip}. When team i plays home game in round p and away game in round p+1, assuming team i plays with team j in round p and plays with team k in round p+1, X_{ip}, $X_{k(p+1)}$, Z_{ijp} and $Z_{ik(p+1)}$ are all 1, then C_{ip} is greater than or equal to D_{ik}, which is the travel distance for team i between round p and round p+1.When team i does not play games under at least one condition as mentioned before, at least one of X_{ip}, $X_{k(p+1)}$, Z_{ijp} and $Z_{ik(p+1)}$ would be zero, and C_{ip} is restricted to be greater than or equal to a non-positive number, thus it will not affect the value of C_{ip} since it is bound to be non-negative. There is no constraint on whether team j should be different than team k. P only goes from 0 to m-3 in this constraint. The constraint is

$$C_{ip} \geq (Z_{ijp} + X_{ip} + Z_{ik(p+1)} + X_{k(p+1)} - 3) * D_{ik}, \ \forall i, j, k \in T, i \neq j, i \neq k, \ \forall p \in [0, m\text{-}3] \tag{6}$$

When team i plays away game in round p and home game in round p+1, assuming team i plays with team k in round p and plays with team j in round p+1, X_{kp}, $X_{i(p+1)}$, Z_{ikp} and $Z_{ij(p+1)}$ are all 1, then C_{ip} is greater than or equal to D_{ik}, which is the travel distance for team i between round p and round p+1. This constraint also does not affect the value of C_{ip} when team i does not play under at least one condition as mentioned. There is no constraint on whether team j should be different than team k. P only goes from 0 to m-3 in this constraint. The constraint is

$$C_{ip} \geq (Z_{ikp} + X_{kp} + Z_{ij(p+1)} + X_{i(p+1)} - 3) * D_{ik}, \ \forall i, j, k \in T, \ i \neq j, i \neq k, \ \forall p \in [0, m\text{-}3] \quad (7)$$

When team i plays away game in round p and away game in round p+1, assuming team i plays with team j in round p and plays with team k in round p+1, X_{jp}, $X_{k(p+1)}$, Z_{ijp} and $Z_{ik(p+1)}$ are all 1, then C_{ip} is greater than or equal to D_{jk}, which is the travel distance for team i between round p and round p+1. This constraint also does not affect the value of C_{ip} when team i does not play under at least one condition as mentioned. There is no constraint on whether team j should be different than team k. P only goes from 0 to m-3 in this constraint. The constraint is

$$C_{ip} \geq (Z_{ijp} + X_{jp} + Z_{ik(p+1)} + X_{k(p+1)} - 3) * D_{jk}, \ \forall i, j, k \in T, \ i \neq j, i \neq k, \ \forall p \in [0, m\text{-}3] \quad (8)$$

There is no needed for a constraint to restrict C_{ip} when team i plays home game in round p and home game in round p+1, because C_{ip} is zero in this condition. C_{ip} is bound to non-negative. When minimizing the objective L, which is the sum of all C_{ip}'s, C_{ip} would be set to minimum, which is zero, when i plays home game in round p and home game in round p+1.

2.2 Mathematical Model

Indices

- i, j, k: indices for teams
- p: indices for rounds

Sets

- T: set of all teams [0, m-1]
- R: set of all rounds [0, m-2]

Decision Variables

- X_{ip}: if team i plays home game in round p; 1 for yes, 0 otherwise
- Z_{ijp}: if team i and team j play game together in round p; 1 for yes, 0 otherwise
- C_{ip}: travel distance of team i between round p and round p+1

Parameters

- D_{ij}: distance between venue of team i and team j

MILP Model

$$Minimize\ L \tag{9}$$

Subject to:

$$\sum_{p \in R} Zijp = 1, \forall i, j \in T, i \neq j \tag{10}$$

$$\sum_{i \in T} Zijp = 1, \forall j \in T, i \neq j, \forall p \in R \tag{11}$$

$$Z_{ijp} = Z_{jip}, , \forall i, j \in T, i \neq j, \forall p \in R \tag{12}$$

$$\sum_{i \in T} Xip = m/2, \ \forall p \in R \tag{13}$$

$$Z_{ijp} + X_{ip} + X_{jp} \leq 2, \ \forall i, j \in T, \ i \neq j, \ \forall p \in R \tag{14}$$

$$X_{ip} + X_{jp} \geq Z_{ijp}, \ \forall i, j \in T, \ i \neq j, \ \forall p \in R \tag{15}$$

$$C_{ip} \geq 0, \ \forall i \in T, \ \forall p \in [0, m\text{-}3] \tag{16}$$

$$C_{ip} \geq (Z_{ijp} + X_{ip} + Z_{ik(p+1)} + X_{k(p+1)} - 3) * D_{ik}, \ \forall i, j, k \in T, \ i \neq j, i \neq k, \ \forall p \in [0, m\text{-}3] \tag{17}$$

$$C_{ip} \geq (Z_{ikp} + X_{kp} + Z_{ij(p+1)} + X_{i(p+1)} - 3) * D_{ik}, \ \forall i, j, k \in T, \ i \neq j, i \neq k, \ \forall p \in [0, m\text{-}3] \tag{18}$$

$$C_{ip} \geq (Z_{ijp} + X_{jp} + Z_{ik(p+1)} + X_{k(p+1)} - 3) * D_{jk}, \ \forall i, j, k \in T, \ i \neq j, i \neq k, \ \forall p \in [0, m\text{-}3] \tag{19}$$

$$L = \sum_{i \in T} \sum_{p=0}^{p=m-3} Cip \tag{20}$$

$$X_{ip} \in \{0, 1\}, \forall i \in T, p \in R \tag{21}$$

$$Z_{ijp} \in \{0, 1\}, \forall i, j \in T, i \neq j, \forall p \in R \tag{22}$$

3 Result and Discussion

3.1 Solver and Hardware

The Solver chosen was Gurobi Optimizer, a commercial optimization solver for linear programming (LP), quadratic programming (QP), quadratically constrained programming (QCP), mixed integer linear programming (MILP), mixed integer quadratic programming (MIQP), and mixed-integer quadratically constrained programming (MIQCP).It supports a variety of programming and modeling languages including C, Python, MATLAB and R, which is a suitable tool for solving the MILP problem in this project. GUROBI Optimizer has been used under an academic license in the version of 9.0.3.

The project has been performed on a quad-core system featuring an intel Core i7-6700k processor. Each of the two cores runs at a maximum speed of 4.20GHz and has an 8MB cache. The system has 16GB of RAM and runs a 64-bit Windows operating system. The model has been written in Python and the computations have been performed using Python in version 3.6.5.

3.2 Result

Since solving the problem with 20 teams is a very hard problem, we tried a series of smaller problems -- 4 teams, 6 teams and 8 teams as problem instances, to test the model. Different Dij were created by finding the distance between each city of the team via Google Map. Table 2 shows the resulting optimal schedule for each instance. Table 3 shows the result of L for all of the instances and their corresponding run time. A time limit of 10 minutes was designed that if the run-time was over 10 minutes, it would be terminated.

Table 2. Result of tournemant schedule for each instance

Instance No.	Teams index from Table 1.	Resulting schedule
1	7,8,9,10	(441, {0: [(1, 3), (2, 0)], 1: [(1, 2), (3, 0)], 2: [(1, 0), (3, 2)]})
2	8,9,10,11	(249, {0: [(2, 0), (3, 1)], 1: [(2, 1), (3, 0)], 2: [(0, 1), (3, 2)]})
3	0,1,2,3,4,5	(459, {0: [(0, 3), (2, 1), (5, 4)], 1: [(0, 5), (2, 3), (4, 1)], 2: [(0, 1), (4, 2), (5, 3)], 3: [(0, 2), (4, 3), (5, 1)], 4: [(0, 4), (3, 1), (5, 2)]})
4	6,7,8,9,10,11	(663, {0: [(0, 5), (2, 1), (4, 3)], 1: [(0, 1), (2, 3), (4, 5)], 2: [(0, 2), (4, 1), (5, 3)], 3: [(0, 3), (4, 2), (5, 1)], 4: [(0, 4), (3, 1), (5, 2)]})
5	12,13,14,15,16,17	(1407, {0: [(0, 2), (4, 3), (5, 1)], 1: [(0, 1), (2, 3), (5, 4)], 2: [(0, 4), (2, 1), (5, 3)], 3: [(0, 5), (1, 3), (2, 4)], 4: [(0, 3), (1, 4), (2, 5)]})
6	1,2,3,4,5,6	(1167, {0: [(1, 2), (3, 0), (4, 5)], 1: [(1, 5), (3, 2), (4, 0)], 2: [(0, 5), (3, 1), (4, 2)], 3: [(0, 2), (3, 5), (4, 1)],

		4: [(0, 1), (3, 4), (5, 2)]})
7	2,3,4,5,6,7	(658, {0: [(3, 0), (4, 1), (5, 2)], 1: [(2, 0), (3, 1), (5, 4)], 2: [(2, 1), (3, 4), (5, 0)], 3: [(0, 1), (2, 4), (5, 3)], 4: [(0, 4), (2, 3), (5, 1)]})
8	3,4,5,6,7,8	(1222, {0: [(2, 3), (4, 1), (5, 0)], 1: [(1, 0), (2, 4), (5, 3)], 2: [(1, 3), (2, 5), (4, 0)], 3: [(1, 5), (2, 0), (4, 3)], 4: [(2, 1), (3, 0), (4, 5)]})
9	4,5,6,7,8,9	(792, {0: [(1, 5), (3, 0), (4, 2)], 1: [(0, 2), (1, 4), (3, 5)], 2: [(0, 5), (1, 2), (3, 4)], 3: [(0, 4), (1, 3), (5, 2)], 4: [(0, 1), (3, 2), (5, 4)]})
10	5,6,7,8,9,10	(1116, {0: [(1, 5), (2, 0), (4, 3)], 1: [(1, 0), (4, 2), (5, 3)], 2: [(1, 3), (4, 0), (5, 2)], 3: [(1, 2), (3, 0), (5, 4)], 4: [(1, 4), (3, 2), (5, 0)]})
11	7,8,9,10,11,12	(588, {0: [(1, 3), (2, 0), (4, 5)], 1: [(1, 0), (4, 2), (5, 3)], 2: [(1, 2), (4, 3), (5, 0)], 3: [(3, 2), (4, 0), (5, 1)], 4: [(3, 0), (4, 1), (5, 2)]})
12	8,9,10,11,12,13	(588, {0: [(0, 2), (1, 5), (4, 3)], 1: [(0, 5), (3, 1), (4, 2)], 2: [(0, 1), (3, 2), (4, 5)], 3: [(2, 1), (3, 5), (4, 0)], 4: [(2, 5), (3, 0), (4, 1)]})
13	0,1,2,3,4,5,6,7	N/A
14	13,14,15,16,17,18,19 ,20	N/A

Table 3. Result of the problem instances

Instance No.	Team Size	File Name	Teams index from Table 1.	L (miles)	Run-time (s)
1	4	Dij4(1).txt	7,8,9,10	441	0.09
2	4	Dij4(2).txt	8,9,10,11	249	0.10
3	6	Dij6(1).txt	0,1,2,3,4,5	459	1.75
4	6	Dij6(2).txt	6,7,8,9,10,11	663	6.25
5	6	Dij6(3).txt	12,13,14,15,16,17	1407	8.52
6	6	Dij6(4).txt	1,2,3,4,5,6	1167	2.53
7	6	Dij6(5).txt	2,3,4,5,6,7	658	1.32
8	6	Dij6(6).txt	3,4,5,6,7,8	1222	2.51
9	6	Dij6(7).txt	4,5,6,7,8,9	792	1.72
10	6	Dij6(8).txt	5,6,7,8,9,10	1116	9.44
11	6	Dij6(9).txt	7,8,9,10,11,12	588	2.78
12	6	Dij6(10).txt	8,9,10,11,12,13	588	2.40
13	8	Dij8(1).txt	0,1,2,3,4,5,6,7	1082 (not optimal)	881 (not completed)
14	8	Dij8(2).txt	13,14,15,16,17,18,19,20	1580 (not optimal)	1209 (not completed)

The solution shown in Table 2 contains the resulting optimal solution of each problem instance. The first element in the tuple is L, which is the minimum total distance travelled by all the team selected in the instance; the second element represent the specific schedule for each round. For example, in instance NO.2, the number 249 is the minimum travelling distance of all the teams in the instance; 0: [(2, 0), (3, 1)] means in the first round, team with index 2 of the instance, which is team 10, plays game against the team with index 0 of the instance, which is team 8; also, team with index 3 of the instance, which is team 11, plays game against the team with index 1 of the instance, which is team 9; the rest can be done in the same manner.

3.3 Discussion

Table 2 shows the feasible solutions for the problem instances. As a result, all the instances can find optimal solutions except the ones with 8 teams. An increasing number of teams in the instance leads to a massive increase in run-time. The two instances with 8 teams cannot get optimal solutions within the time limit, which is 10 minutes.

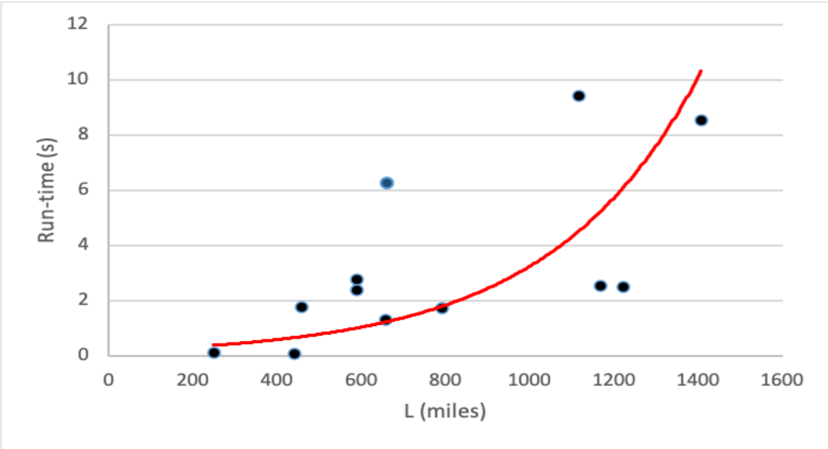

Fig. 1. Scatter plot of 12 instances (with optimal solutions) with the exponential trend line. The x-axis is the total travelling distance, and the y-axis is the overall run-time.

Since the run-time for different team size has different orders of magnitude, it can be considered that the number of operations has an exponential increase with the increase in team size. If a prediction is made on Fig.1 that the instances of 8 teams have optimal solution, it will fit the trend line since the run-time of them have a tremendous large value and Ls have similar values compared to the instances with 6 teams.

4 Conclusion

In this paper, tournament scheduling problem for finding the minimum total travelling distances of teams in premier league was solved by MIP with the use of GRUOBI Optimizer under Python. As a result, the model created in this program can get optimal solution only when the team size of the league is lower than or equal to 6 teams. Otherwise, the run-time will exceed to designated time limit of 10 minutes. Tournament scheduling is a complicated problem. MIP may not be a suitable programming method for solving the problem with large team size.

References

1. Premier League Homepage, http://www.premierleague.com
2. Wen-Yang Ku, J. Christopher Beck: Mixed Integer Programming Models for Job Shop Scheduling. A computational Analysis (2016).
3. LNCS Homepage, http://www.springer.com/lncs.