Final Report

SPORT SCHEDULING PROBLEMS WITH ENTERTAINMENT MAXIMIZATION

Dean Bale | CM3203: One Semester Individual Project | 5th June 2019
Supervisor: Dr Richard Booth
Moderator: Víctor Gutierrez Basulto

Acknowledgments

I'd like to express my gratitude to Dr. R. Booth for supervising this project in addition to lectures taught that inspired me to select this project. I'd also like to thank School of Computer Science and Informatics and the High-Performance Program who have allowed me to balance my studies and professional sports career.

Table of Contents

Abstract

A vast majority of sports teams rely on selling broadcasting rights and advertisement alongside merchandise and ticket sales. Therefore, it is important to produce entertaining sport schedules that will attract media companies to purchase these broadcasting rights. These schedules also need to be well balanced and not biased towards a team.

This project explores how Integer Linear Programming can be used to find the most entertaining sport schedules for Double Round Robin Tournaments. In addition to this, it investigates the Travelling Tournament Problem to reduce the distance teams are required to travel during a tournament.

Introduction

Sport scheduling is an inherently difficult problem to solve, trying to satisfy differing constraints whilst optimizing an objective. Though it can be quite trivial when team numbers are kept extremely low, as soon as you start to increase the number of team the problem becomes exceptionally difficult to solve. For example the Major League Baseball season schedule consists of 162 games for each of the 30 teams resulting in 2430 games to be scheduled over a six month period[1]. A brute force approach to find the optimal schedule, involving going through every possible permutation of games, would be infeasible. In the unlikely case the brute force approach manages to find any solution to the problem, it is unlikely to be a desirable schedule.

It is in the interests of not only the sport teams but also media companies to find the optimal schedule. Sport teams and media companies want to find these optimal schedules, that suits their own requirements, as more desirable schedule will allow sports teams to request financial investment from media companies in return for elevated viewing figures. Similarly, these schedules will look to optimize an 'entertainment' factor whilst balancing team requirements and making the schedule as fair as possible for all teams.

This project will examine how Integer Linear Programming can be used to solved sport scheduling problems to optimize an 'entertainment' factor. A schedule will be assumed to be entertaining if teams that are closely seeded together are more likely to compete at the end of the tournament. With the intention of disguising the overall winner and final position of each team for as long as possible.

Entertainment is not the only factor used to determine a tournament schedule; a secondary factor explored in the project is the amount teams must travel during a tournament. With the objective being to minimize the overall distance travelled by all teams. This is to reduce any adverse effect on performance that excessive time away travelling may have on a team. This is often referred to as the Travelling Tournament Problem (TTP)[2]. Whilst there has been plenty of research into the TTP the papers found did not include the formulation of the ILP. Hence this paper explores a novel approach and formulation of the ILP.

Aim

The aims of the project are to produce an application which can specify teams, tournament constraints and how to optimize a sport schedule. Then to use ILP produce these optimal sport schedules. And finally, this application will be used to review the effectiveness of the ILP to produce entertaining and minimal distance schedules.

This report consists of an introduction to Integer Linear Programming, the Travelling Tournament Problem and a brief discussion of alternative approaches. Following this there is an outline of the requirements of the application and user-interface design. Then, there is an explanation of how ILPs can be used to represent sport scheduling problems and the process for determining how 'entertaining' games in a tournament are, including evidence of the function sport scheduling application. Finally, the schedules produced by the application are critically evaluated.

Background

Context

Linear Programming

Operations research (OR) is a discipline that involves mathematical modelling, decision-making, solution optimization and iterative computations. Linear Programming (LP) is one of the most prominent techniques used. It is designed to solve a model of linear equations to optimize a linear objective function[3].

The structure of an LP can be separated into three parts.

- Decision Variable – The variables which we need to find optimal values for
- Objective Function – The linear function that we wish to either maximize or minimize
- Constraints – Linear expressions that must be satisfied

Integer Linear Programming

Whilst there are several variations of this programming technique, such as Dynamic Programming, this project will be focused on ILP. The only difference between LP and ILP is that the decision variable can only take integer values rather than real numbers. This project will use a further subclass of ILP, Binary Integer Programming, where the decision variable can only take the values of 0 or 1.

Travelling Tournament Problem

Sport scheduling involves determining when games should be scheduled in the final tournament schedule. Essentially making it a choice of whether to include or exclude a game between two teams at a certain round. Making a form of Knapsack Problem[4].

The Travelling Tournament Problem is a specific sport scheduling problem that focuses on the minimizing the total distance travelled by all teams when they go on tours. When a team goes on a tour, it involves playing one or more consecutive games away without returning home in between. Effectively combining a simple sport scheduling problem with the Travelling Salesman Problem (TSP)[5]. Although there is no formal proof, it is strongly believed that TTP is *NP*-hard. Certainly, computationally, it appears to be much harder than a regular TSP[2].

Alternative Approaches

Tabu Search

In operations research there are several methods that can be used to find optimal solutions. Local Search with a greedy heuristic is an approach that guides neighborhood search, only allowing changes to the solution that improves it, based on objective we are trying to optimize. Because of this, it is unlikely to find the global optimum solution and only local optimum solutions.

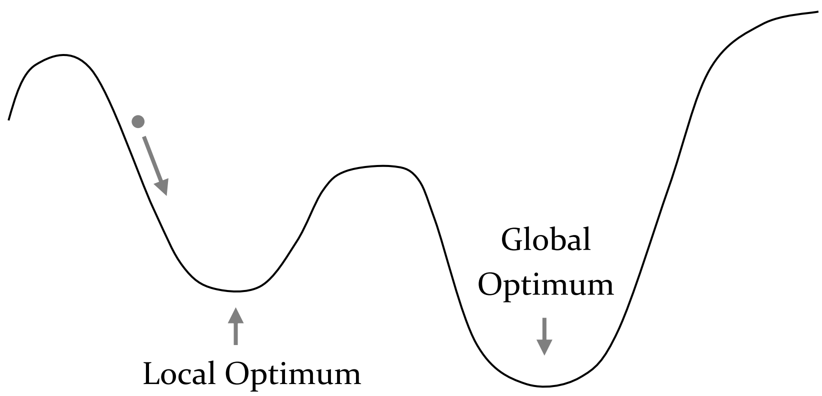

Figure 1 – Local and Global Optimums

Tabu Search (TS) introduces a metaheuristic into this local greedy heuristic search allowing it to search for optimum solutions beyond the current neighborhood[6]. It does this by temporarily moving to worse solutions, often referred to as uphill moves, with the hope that it will lead to a better local optimum. The TS algorithm requires short-term memory to temporarily records a list of disallowed moves, otherwise it will be stuck in a cycle of moving uphill and back down to the local optimum.

Whilst this project will focus solely on ILP, an approach by J.P Hamiez and J. Hao explores a Tabu approach for a version of the Sports League Scheduling Problem[7]. They were able obtain competitive computational times when compared to other OR approaches.

Constraint Satisfaction

While TS involves going from one complete solution to another, for some problems it can be difficult to find any solution let alone the optimum one. Constraint Satisfaction builds a solution in fragments until a final solution produced[8].

Using Constraint Programming, Constraint Satisfaction Problems can be represented as sets of variables, domains (the range of values that each variable can take) and constraints. The constraints restrict the values that can be assigned to variable or relate two variables to each other. Additionally, global constraints can be used to affect several variables at once. Sport Scheduling can be represented as a Constraint Satisfaction Problem and research has investigated how this can be used to produce sport schedules. One such paper by K. Easton1 et. al looked at combining Integer Linear Programming and Constraint Programming to solve the travelling tournament for various numbers of teams[2].

Specialist Libraries

To aid in the creation of an efficient and easy to use application several specialized libraries will be used.

Gurobi

The Gurobi Optimizer is an advanced solver for mathematical programming, such as Linear Programming. The solvers were designed to take advantage of modern architectures and multi-core processors, using the most advanced implementations of the latest algorithms[9].

Windows Presentation Format

Windows Presentation Foundation (WPF) is a UI framework used to create desktop client applications. The WPF development platform supports a broad set of application development features, including an application model, controls, graphics, layout and data binding[10].

DevExpress

DevExpress provides over 120 control and libraries to create a WPF application. These controls and libraries offer greater functionality and with a series of Office-inspired components can improve the usability of the application and decrease the development time spent on the user interface[11].

Approach

Specification

An application with a graphical user interface will allow any user to easily create sport schedules without needing to know anything about the underlying theory behind the application. The application should allow the creation of tournaments based on real-world scenarios.

The English Premier League (EPL) fixtures are an example of a highly constrained schedule. Currently, the scheduling of takes almost six months to complete and has been entrusted to the same individual for nearly thirty years[12]. There are set of essential constraints on the EPL. One being that in any five matches for a team, three must be away fixtures and two at home, or vice versa. Another is the schedule should avoid teams playing more than two home or away games in a row. One other constraint is teams should avoid playing two games at home or away at the start or end of the tournament.

Alongside some of these essential constraints, there are many other more minor constraints that reflect real-world situations. An example would be teams in similar locations often have a partner club[12], so that they avoid playing at home on the same date. This can reduce the demand on the transport network and the amount of policing required for the cities hosting these fixtures. For the same reason other non-sporting events that may occur in the team's location. The schedule also must consider the schedules of another tournament that a team may be involved in.

The application will look to represent a set of constraints that can be used to model some of these real-life circumstances.

Requirements

There are many specific requirements for the Sport Scheduling Application to produce an entertaining sport schedule. These include:

- A feature to add teams and an associated rating of the team
- A feature to choose what type of tournament should be scheduled. Either,
    - A Double Round Robin (DRR) Schedule, such that all teams have played all other teams in the first half of the tournament
    - A Mirrored DRR Tournament Schedule, such that if a team plays another team at home (away) in the first half of the tournament, they will play the same team away (home) in the corresponding round in the second half of the tournament
- A feature to add numerous constraints on the tournament, including:

  - Prohibiting a game between two teams on a certain round
  - Specifying a game between two teams to occur on a certain round
  - Prohibiting a team playing away in a certain round
  - Specifying a team to play away in a certain round
  - Prohibiting a team to play at home in a certain round
  - Specifying a team to play home in a certain round
  - Set how many away games or home games teams can play in a row
  - Setting two teams as pairs. So that these teams cannot play at home at the same time
- A feature to choose the entertainment heuristic that determines how exciting each potential game can be
- Creating DRR Tournament Schedules based on the constraints listed above to create the most entertaining tournament possible
- Reviewing the created schedule in either a tabular or a list format
- Feature to save and open previous tournament designs

Travelling Tournament Requirements

To produce schedules that minimize the distance travelled by teams' further requirements are needed. These include a feature to add the distances between teams, the ability to choose the maximum tour length for any team and to display the total distance travelled by each team.

Optional Requirements

There are many ways that the application could be extended. For example, a feature that would allow the selection of a priority games. This would involve:

- The ability to choose how many games per round are selected as priority games
- Being able to select a minimum number of times all teams must appear in the selected priority games

Similarly, another way to extend this application would be to dynamically reschedule a tournament. To do this it would require:

- The ability to select the number of rounds to reschedule
- A feature to adjust team ratings/rankings, based on performances in the already scheduled games before creating more of the schedule

Design

Integer Linear Programming

Two of the core requirements and one of the optional requirements involve using ILPs to find the optimal solution. Each requirement needs to be translated into decision variables, an objective function and a series of linear constraints.

Entertaining Sport Schedules

ILPs can be easily used to model sport scheduling problems with lots of different constraints. The sports tournaments that are focused on in this project are DRR tournaments, so we can assume there will be n even number of teams, meaning there will be $2(n-1)$ rounds. Each potential game will be given an entertainment rating.

$$e_{ijr} = entertainment\ value\ if\ team\ \boldsymbol{i}\ plays\ team\ \boldsymbol{j}\ in\ round\ \boldsymbol{r}\ at\ home$$

Equation 1 - Entertainment coefficient for the scheduling decision variables

The algorithm to calculate the entertainment ratings will be discussed later in the Design Section.

Decision Variables

$$x_{ijr} = \begin{cases} 1, & if\ team\ \boldsymbol{i}\ plays\ team\ \boldsymbol{j}\ in\ round\ \boldsymbol{r}\ at\ \text{home} \\ 0, & otherwise \end{cases}$$

$$\forall\ i = 1,\ \ldots, n\ ,\ j = 1,\ \ldots, n\ ,\ r = 1,\ \ldots,\ 2(\ n\ -1)$$

Equation 2 - Sport scheduling decision variables

Objective Function

$$\boldsymbol{max}: \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{2(n-1)} e_{ijr}x_{ijr}$$

Equation 3 - Sport scheduling objective function

Essential Tournament Constraints

The ILP requires a series of constraints to ensure the basic structure of the tournament.

- A team cannot play themselves

$$x_{iir} = 0, \quad \forall i = 1,\ \ldots, n\ , \mathrm{r} = 1, \ldots, 2(n\ -1)$$

Equation 4 - Prohibit teams playing themselves constraint

- Teams must play every team at home only once

$$\sum_{r=1}^{2(n-1)} x_{ijr} = 1, \qquad \forall i = 1, \dots, n, \ j = 1, \dots, n \ , \ i \neq j$$

Equation 5 - Enforce teams to play everyone constraint

- One game per team per round

$$\sum_{j=1}^{n} x_{jir} + x_{ijr} = 1, \qquad \forall i = 1, \dots, n, r = 1, \dots, 2(n-1)$$

Equation 6 - Enforce teams play only once a round constraint

There can be several variations of the DRR tournaments of how teams must play in the first half compared to the second half. Two possible variations include:

- All teams must play every team once in the first half

$$\sum_{r=1}^{n-1} x_{ijr} + x_{jir} = 1, \qquad \forall i = 1, \dots, n, j = 1, \dots, n, \qquad j \neq i$$

Equation 7 - Double Round Robin constraint

- Mirrored Tournament. i.e. If team i play team j at home (away) in the first round, then team i play team j must play away (home) in the first round of the second half of the tournament

$$x_{ijr} - x_{ji\,r+(n-1)} = 0, \qquad \forall i = 1, \dots, n, j = 1, \dots, n, r = 1, \dots, n-1$$

Equation 8 - Mirrored Double Round Robin constraint

Only one of the two constraints described by Equation 7 and Equation 8 can be included in the final ILP.

Additional Constraints

In addition to the fundament constraints of the sport tournament, a combination of extra constraints can be added.

- Team i must play team j in round r

$$x_{ijr} = 1$$

Equation 9 - Fixed game constraint

- Team i cannot play team j in round r

$$x_{ijr} = 0$$

Equation 10 - Prohibited game constraint

- Team i must play at home in round r

$$\sum_{j=1}^{n} x_{ijr} = 1$$

Equation 11 - Home game constraint

- Team i must play away in round r

$$\sum_{j=1}^{n} x_{jir} = 1$$

Equation 12 - Away game constraint

- Team i and team j cannot both play at home in the same round

$$\sum_{k=1}^{n} x_{ikr} + x_{jkr} \leq 1, \qquad \forall r = 1, \ \dots, \ 2(n \ -1)$$

Equation 13 - Team pairs constraint

- Teams cannot play more than A away games in a row

$$\sum_{a=0}^{A} \sum_{j=1}^{n} x_{ij(r+a)} \leq A\,, \qquad \forall j = 1, \ \dots, \ n, \ \ r = 1, \dots, 2(n-1) - A$$

Equation 14 - Consecutive away games constraint

- Teams cannot play more than H home games in a row

$$\sum_{h=0}^{H} \sum_{j=1}^{n} x_{ij(r+h)} \leq H\,, \qquad \forall j = 1, \ \dots, \ n, \ \ r = 1, \dots, 2(n-1) - H$$

Equation 15 - Consecutive home games constraint

Travelling Tournament

The TTP is a significantly more constrained problem compared to solving a double round robin tournament schedule. Along with most of the constraints presented above, an additional decision variable is required to allow selection of certain tours. Again, it can be assumed that there will be n even number of teams, meaning there will be $2(n-1)$ rounds.

For each possible tour a team can take there will be a total distance for that tour.

$$d_{it} = \textit{distance to travel for team } \boldsymbol{i} \textit{ to each team in the tour } \boldsymbol{t} \textit{ and return home}$$

Equation 16 - Tour decision variable distance coefficient

TTPs are often constrained with a maximum tour length l, being the number of away games in a row a team can play before returning home. As only half of the games played by team i will be away games, the maximum value that l could be is $n-1$.

To create the decision variable required for the ILP the total number of potential tours for each team is required. This involves calculating the number of permutations of teams excluding the team that is travelling, thus the number of teams to select permutations of is $n-1$.

For a four-team tournament involving teams A, B, C and D the potential tours for team A are shown in Table 1.

Tour Length	Tour Permutations
1	$(B), (C), (D)$
2	$(B, C), (B, D), (C, B), (C, D), (D, B), (D, C)$
3	$(B, C, D), (B, D, C), (C, B, D), (C, D, B), (D, B, C), (D, C, B)$

Table 1 - Tour permutations for a four-team tournament

These permutations can then be used at different rounds throughout the r rounds of the tournament, demonstrated by Figure 2.

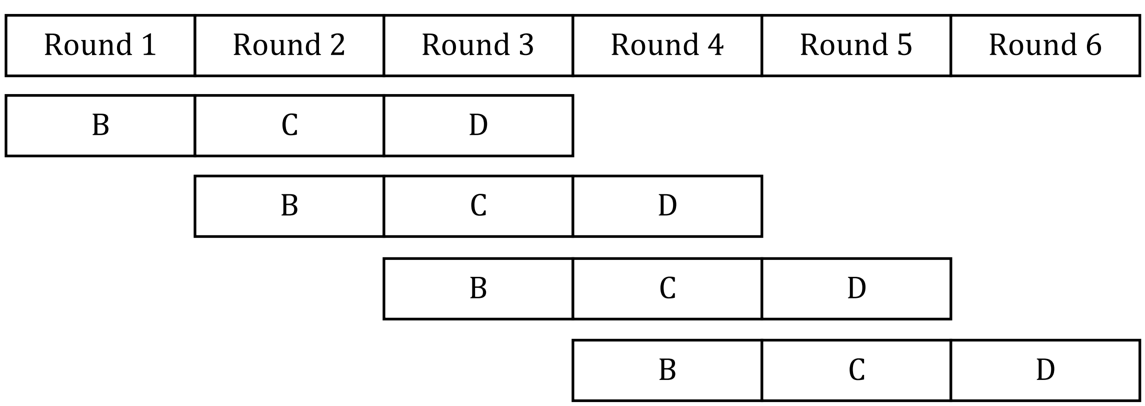

Figure 2 - Tour permutation at different rounds

The number of variations of a tour of length k when placed at each round is $r + 1 - k$. Using all the elements described above, it is possible to form an equation to calculate the number of potential tours for a single team.

$$T = \sum_{k=0}^{l} (r + 1 - k) \times P(n-1, k) = \sum_{k=0}^{l} \frac{(r + 1 - k) \times (n-1)!}{(n-1-k)!}$$

Equation 17 - Total number of potential tours per team

Decision Variables

The decision variable declaration in Equation 2 will also be required for the Travelling Tournament ILP in addition to the tour decision variables.

$$y_{it} = \begin{cases} 1, & if\ tour\ \boldsymbol{t}\ for\ team\ \boldsymbol{i}\ is\ included\ in\ the\ tournament\ schedule \\ 0, & otherwise \end{cases}$$

$$\forall\, i = 1, \ \dots, n\,,\ t = 1, \ \dots, T$$

Equation 18 - Tour decision variables

Objective Function

$$\boldsymbol{min}\colon \sum_{i=1}^{n} \sum_{t=1}^{T} d_{it} y_{it}$$

Equation 19 - Travelling Tournament objective function

Constraints

All the constraints from the Entertaining Sport Schedule ILP above are also required for this Travelling Tournament ILP. But to allow the algorithm to minimize the total tournament length a constraint that links the x and y decision variables to each other is needed.

Each y variable is related to an ordered set of x variables where the first x variable is the first game in the tour and the last x variable is the last game in the tour. This possible tour will be represented as

$$p_{it} = (x_{jir} \mid x_{jir}\ is\ a\ game\ in\ the\ tour)$$

Equation 20 - Ordered set of games for a possible tour

$$P = \{p_{it} | p_{it}\ is\ a\ possible\ tour\}$$

Equation 21 - Set of all possible tours

Three subsets can be extracted from set of all set possible tours. A set of tours that begins at round 1 of the tournament, a set of tours were the last games is in round r and a set that includes all the other possible tours. Categorization of these tours is required to determine whether the constraint needs to indicate a home game before and/or after the tour.

Without this the ILP could schedule consecutive tours without the team returning home, resulting in tours of length greater than maximum tour length l.

Tour at Start of a Tournament

Tours that begin at the start of a tournament requires the summation of games: a summation of all possible home games after the tour and an association to the relevant tour decision variable.

$$P_{Start} = \{p_{it} \mid p_{it} \in P \text{ and } p_{it} \text{ begins at the start of the tournament}\}$$

Equation 22 - Subset of possible tours that begin at the start of a tournament

$$\sum_{x_{jir} \in p_{it}} x_{jir} + \sum_{j=1}^{n} x_{ij(MaximumRound(p_{it})-1)} - y_{it} = |p_{it}|, \qquad \forall\, p_{it} \in P_{Start}$$

Equation 23 - Game and tour association constraint for tours at the beginning of a tournament

Tours at the End of the Tournament

Instead of a summation of home games after the tour, this set of tours requires this summation of home games before the tour.

$$P_{End} = \{p_{it} \mid p_{it} \in P \text{ and } p_{it} \text{ ends in the final round of the tournament}\}$$

Equation 24 - Subset of possible tours that end at the end in the final round of a tournament

$$\sum_{j=1}^{n} x_{ij(MinimumRound(p_{it})-1)} + \sum_{x_{jir} \in p_{it}} x_{jir} - y_{it} = |p_{it}|, \qquad \forall\, p_{it} \in P_{End}$$

Equation 25 - Game and tour association for tours at the end of a tournament

All other Tours

All other possible tours require both summations of possible home games before and after the tour to enforces a home game occurs before and after team i goes on the tour.

$$\sum_{j=1}^{n} x_{ij(MinimumRound(p_{it})-1)} + \sum_{x_{jir} \in p_{it}} x_{jir} + \sum_{j=1}^{n} x_{ij(MaximumRound(p_{it})-1)} - y_{it} = |p_{it}|,$$

$$\forall p_{it} \in P - P_{start} - P_{End}$$

Equation 26 - Game and tour association for tours not at the beginning of a tournament

Priority Match Selection

During sports tournaments broadcasters may want to prioritize which games to show on TV if there is a select number of TV slots available. Although it doesn't use a double round robin tournament style, the tennis tournament at Wimbledon exemplifies this scenario.

Broadcasters will prefer matches held on centre court and so there is a requirement to ensure priority games are played there. Given a tournament schedule the following ILP can be used to select priority games in a double round robin tournament. The games to be selected from the tournament will continue to use the previously calculated entertainment value used when creating schedule.

$$e_{ijr} = excitement\ value\ if\ team\ i\ plays\ team\ j\ in\ round\ r\ at\ \text{home}$$

Equation 27 - Entertainment coefficient for selected game decision variables

The ILP will also only need to deal with selected games for the final tournament schedule and not all the potential games.

$$F(tournament) = \{(\mathrm{i,j,r}) \mid \begin{matrix} team\ \boldsymbol{i}\ plays\ team\ \boldsymbol{j}\ in\ round\ \boldsymbol{r}\ at\ home \\ is\ game\ in\ the\ tournament\ schedule \end{matrix}\}$$

Equation 28 - Set of games in the final tournament schedule

Decision Variable

$$z_{ijr} = \begin{cases} 1, & \begin{matrix} if\ team\ \boldsymbol{i}\ plays\ team\ \boldsymbol{j}\ in\ round\ \boldsymbol{r}\ at\ \text{home} \\ is\ selected\ as\ a\ priority\ game \end{matrix} \\ 0, & otherwise \end{cases} \qquad \forall\ (\mathrm{i,j,r}) \in G$$

Equation 29 - Priority game decision variables

Objective Function

$$\boldsymbol{max}: \sum_{(\mathrm{i,j,r}) \in \mathrm{G}} e_{ijr} z_{ijr}$$

Equation 30 - Priority games objective function

Constraints

There will not be enough priority slots for every single game. Instead there will be p number of priority games per round. In addition to this, teams will also require a minimal representation over all the priority games.

- For each round fill p priority slots with a scheduled game

$$\sum_{(\mathrm{i,j,r}) \in \mathrm{G}} z_{ijr} = p, \qquad \forall\ r = 1, \dots, 2(n-1)$$

Equation 31 - Priority slots constraint

- For each team make sure there is a minimal representation of m

$$\sum_{x_{ijr} \in \mathrm{G}} x_{ijr} + \sum_{x_{jir} \in \mathrm{G}} x_{jir} \geq m\,, \qquad \forall\ \mathrm{i} = 1, \dots, n$$

Equation 32 - Minimum representation constraint

Entertainment

In order to create entertaining sport schedules there needs to be a measure of how entertaining a potential game in a tournament. Allowing the ILP to select the games with the greatest entertainment values. As stated previously in Equation 1, each game is assigned a value e_{ijr}.

This project focusses on creating tournaments such that teams should hopefully have something to play for right up to the end of the tournament. It is desirable to suppress the ability to predict early on in a tournament which teams will finish at the top of the leaderboard. With the hope being that everything hangs on the result of the final round. This effectively requires teams with similar ranking to avoid competing against each other at the start of the tournament and more likely to be competing at the end of the tournament.

One possible way to work out the e_{ijr} values could be to focus on the ranking of each team. With the purpose of producing higher e_{ijr} for games scheduled later in the tournament with teams that are ranked more closely.

An alternative approach would be to use the rating of teams and attempt to predict potential outcomes of games. Then using these predictions, try to schedule games with certain outcomes in different rounds in a tournament.

Ranking

Teams that are ranked next to each other ideally should be scheduled for a game in the final round and teams that are ranked furthest away should be scheduled for a game at the start of the tournament.

A Gaussian function would allow entertainment values to be distributed to potential games based on the round of the tournament and the difference in ranking.

$$f(x) = a\, e^{\left(\frac{-(b-x)^2}{2\,c^2}\right)}$$

Equation 33 - Gaussian Function

Using this function where x is the difference in ranking between the two teams in the potential game the constant values can be set as follows:

- a – Maximum entertainment any game can have
- b – Offset of entertainment values based on rounds number
    - For the last round, a maximum value is sought after for teams ranked one position away and for the first-round teams ranked furthest away need to have the lowest entertainment value. The function below produces suitable values for b based on the number of teams that satisfies the requirements.

$$g(x) = -\frac{n-2}{2(n-1)}x + n - 1 + \frac{n-2}{2(n-1)-1}$$

Equation 34 - Entertainment offset value

- c –This constant affects distribution of the entertainment as the difference in ranking increases
  - After experimentation, a value of a quarter of the number of teams results zero entertainment for games between the teams ranked furthest away at the end of the tournament.

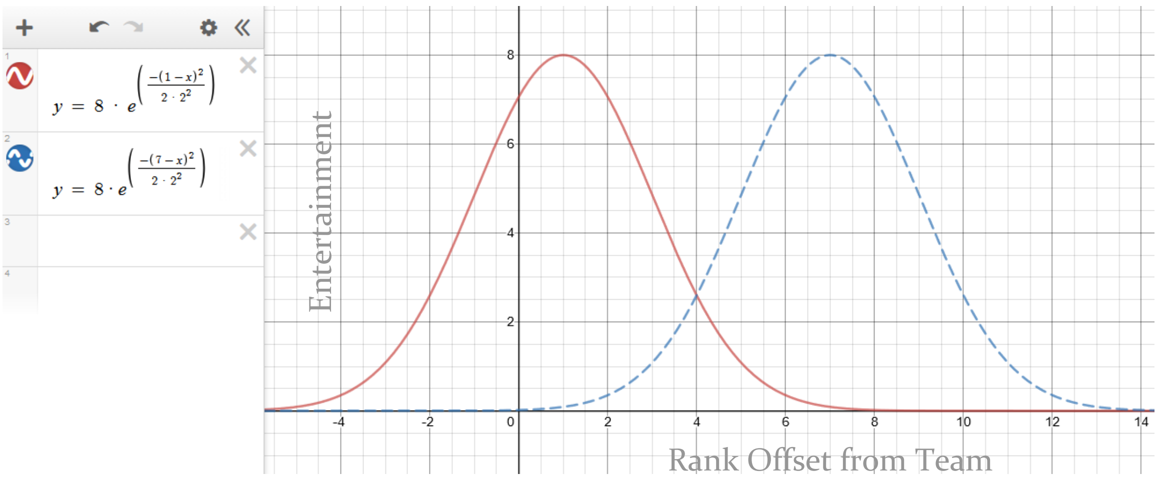

Figure 3 - Graphical representation of the entertainment function

Figure 3Figure 3 shows a graphical representation of the distribution of entertainment values when the round number and difference in ranking changes. The first equation (red) represents the entertainment values for games in the final round of an eight-team tournament. Teams ranked one place away getting the maximum entertainment value. The second equation (blue) represents the entertainment values for games in the first round of the tournament. Games where teams are ranked furthest away, in this case seven ranks away, will receive maximum entertainment and games with teams ranked closest getting minimal entertainment.

An alternative approach could have been to use a linear function to distribute the entertainment values. The likelihood that a final schedule will include all the games where teams are ranked next to each other in the final round, when additional constraints are applied, is quite unlikely. For that reason, the normal distribution provided by the Gaussian function doesn't penalize teams who are still ranked relatively closely as harshly.

Pseudocode for Calculating Entertainment based on Ranking

An algorithm is required to systematically calculate the entertainment based on the number of teams, number of rounds, round number and the difference in ranking between two teams.

$teams \leftarrow Lit\ of\ team\ ID's\ with\ the\ firt\ being\ the\ best\ team\ and\ last\ the\ worst\ team$

$numberOfTeams\ \leftarrow Number\ of\ ID's\ in\ Teams$

$numberOfRounds \leftarrow 2 \times (NumberOfTeams - 1)$

$entertainments\ \leftarrow array\ of\ size\ numberOfTeams \times NumberOfTeams \times NumberOfRounds$

$round \leftarrow 1$

$While\ (round \leq numberOfRounds)$

$rankingIndex \leftarrow 0$

$While\ (rankingIndex < numberOfTeams)$

$teamsAboveCurrentTeam \leftarrow numberOfTeams - rankingIndex - 1$

$teamsBelowCurrentTeam \leftarrow rankingIndex$

$If\ (teamsAboveCurrentTeam > teamsBelowCurrentTeam)$

$maxTeamOffset \leftarrow\ teamsAboveCurrentTeam$

$Else$

$maxTeamOffset\ \leftarrow teamsBelowCurentTeam$

$EndIf$

$teamRankOffset\ \leftarrow 1$

$While\ (teamRankOffset \leq maxTeamOffset)$

$entertainmentValue\ \leftarrow GetEntertainmentValue(numberOfTeams, numberOfRounds, round, teamRankOffset)$

$If\ (rankingIndex - teamRankOffset \geq 0)$

$entertainments[teams[rankingIndex], teams[rankingIndex - teamRankOffset],\ round - 1] \leftarrow entertainmentValue$

$EndIf$

$If\ (rankingIndex + teamRankOffset \leq numberOfTeams - 1)$

$entertainments[teams[rankingIndex], teams[rankingIndex + teamRankOffset],\ round - 1] \leftarrow entertainmentValue$

$EndIf$

$EndWhile$

$EndWhile$

$EndWhile$

$Return\ excitments$

Pseudocode for the Entertainment of a Single Game

The main algorithm calls a function to compute the exact entertainment value for each individual game based on the round number and difference in ranking of a team. This function implements the Gaussian function and offset of the function shown in Equation 34.

$Function\ GetEntertainmentValue(numberOfTeams, numberOfRounds, round, teamRankOffset)$

$maximumEntertainment\ \leftarrow numberOfTeams$

$spreadValue\ \leftarrow numberOfTeams\ \div 2$

$a\ \leftarrow maximumEntertainment$

$b\ \leftarrow\ -\frac{numberOfTeams - 2}{numberOfRounds - 1} \times roundNumber + \frac{numberOfRounds}{2} + \frac{numberOfTeams-2}{numberOfRounds - 1}$

$c\ \leftarrow spreadValue$

$x\ \leftarrow teamRankOffset$

$entertainment\ \leftarrow a\ \times\ \frac{-(x-b)^2}{2 \times c^2}$

$return\ entertainment$

$EndFunction$

Rating

On a given scale, each team is given a rating. This rating can stem from many factors such as previous performances of the teams, the players in the team and expert advice. For the purpose of this project, all ratings will be between 0 – 100.

Using the ratings given to teams, it is possible to make a naive prediction of the result of each game. If the difference between the rating of two teams is greater than a certain threshold it is assumed that there will be a clear winner. Lower than this threshold and the game is assumed to result in a draw. With these predictions, a schedule could try to concentrate as many games as possible that are predicted to be a draw near the end of a tournament. In theory, this would result in many of games that have teams of similar abilities competing against each other near the end of the tournament.

Pseudocode for Calculating Entertainment based on Rating

The entertainment values for each game are calculated to increase the chance that the games which potentially could result in a draw, are at the end or near the start. The following algorithm expresses the sequence of step required to predict the outcome of matches and then produce entertainment values for potential games. The threshold used determine the outcome of a match is set to a value based on the highest and lowest teams rating. Experimenting with this value may lead to better and worse entertaining schedules.

$teams \leftarrow List\ of\ team\ ID's\ with\ the\ first\ being\ the\ best\ team\ and\ last\ the\ worst\ team$

$teamRatings \leftarrow List\ of\ team\ ratings\ in\ the\ same\ order\ as\ teams\ list$

$numberOfTeams \leftarrow Number\ of\ ID's\ in\ Teams$

$numberOfRounds \leftarrow 2 \times (NumberOfTeams - 1)$

$maxRatingValue \leftarrow Max(teamRatings)$

$minRatingValue \leftarrow Min(teamRatings)$

$$winThreshold \leftarrow \frac{(maxRatingValue - MinRatingValue)}{4}$$

$predictedMatchResults \leftarrow PredictMatches(teamRatings, winThreshold, numberOfTeams)$

$entertainments \leftarrow EntertainmentForDrawsAtTheEnd(predictedMatchResults, numberOfRounds, numberOfTeams)$

$Return\ entertainments$

Pseudocode for Match Predictions

The function used in previous pseudocode to predict the match results between teams in the tournament, simply works out the difference in team ratings and added the match prediction to a list.

$Function\ PredictedMatches(\ teamRatings, numberOfTeams, winThreshold)$

$\quad matches \leftarrow empty\ list\ for\ match\ predictions$

$\quad teamI \leftarrow 0$

$\quad While\ (teamI < numberOfTeams)$

$\quad\quad teamJ \leftarrow 0$

$\quad\quad While(teamJ < numberOfTeams)$

$\quad\quad\quad newMatchPrediction.TeamI \leftarrow teamI$

$newMatchPrediction.TeamJ \leftarrow teamJ$

$teamRatingDifference \leftarrow teamRatings[teamI] - teamRatings[j]$

$If(teamRatingDifference < winThreshold)$

$newMatchPrediction.Result \leftarrow Draw$

$Else\ If\ (teamRatings[teamI] > teamRatings[teamJ])$

$newMatchPrediction.Result \leftarrow TeamIWins$

$Else$

$newMatchPrediction.Result \leftarrow TeamJWins$

$EndIf$

$matches.Add(newMatchPrediction)$

$End\ While$

$EndWhile$

$EndFunction$

Pseudocode for Entertainment Values Based on Match Predictions

To create a tendency for games that result in a draw to be scheduled near the end of the tournament, higher entertainment values are given to these games the later in the tournament they are. Conversely, games that do not result in a draw should receive higher entertainment values at the start of a tournament.

Two linear functions can be used to calculate the entertainment value depending on the match results where m is the maximum entertainment and r is the number of rounds.

If the match results in a draw

$$E_{Draw}(x) = m \times \frac{x}{r} - \frac{m}{4}$$

If the match does not result in a draw

$$E_{NotDraw}(x) = m \times \frac{r-x}{r} - \frac{m}{4}$$

The addition of $-\frac{m}{4}$ in allows the function to produce a negative value at either the start or end of the tournament dependent on the match result. The penalizes the entertainment of the whole schedule if any addition of games furthest away from their desired position are included in the final schedule. Figure 4 - Linear functions for entertainmentFigure 4 demonstrates the two linear functions for an eight-team tournament where m is equal to the number of teams and r is $2(n-1)$ representing a DRR tournament.

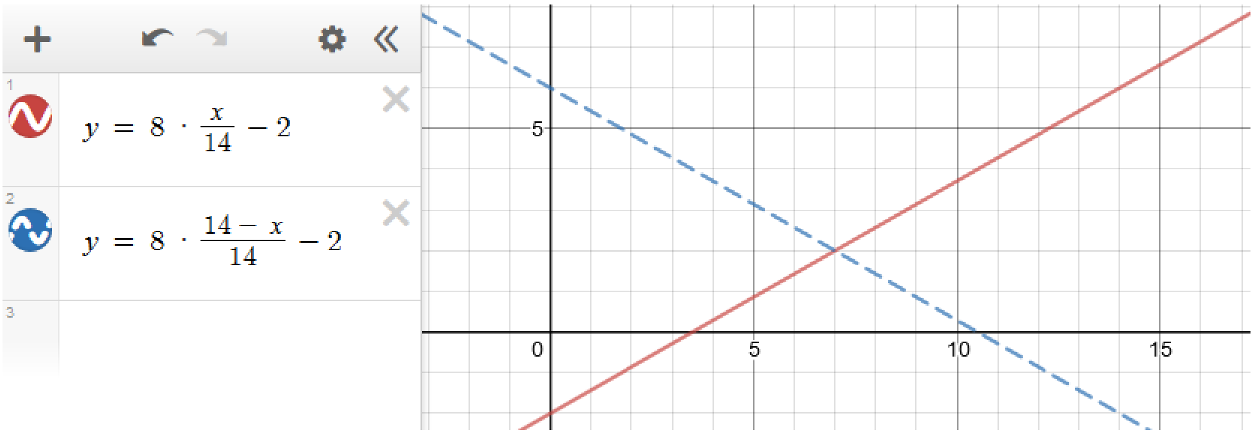

Figure 4 - Linear functions for entertainment

The main algorithm above calls the function expressed in the following pseudocode.

$Function\ EntertainmentForDrawsAtTheEnd(predictedMatchResults, numberOfRounds, numberOfTeams)$

$entertainments \leftarrow array\ of\ size\ numberOfTeams \times NumberOfTeams \times NumberOfRounds$

$maximumEntertainment \leftarrow numberOfTeams$

$round \leftarrow 0$

$While\ (round < numberOfRounds)$

$ForEach(predictedMatch\ In\ predictedMatchResults)$

$If(predcitedMatch.Result = Draw)$

$eValue \leftarrow maximumEntertainment \times \frac{round}{numberOfRounds} - \frac{maximumEntertainment}{4}$

$entertainments[predictedMatch.TeamI, predictedMatch.TeamJ, round] \leftarrow eValue$

$entertainments[predictedMatch.TeamJ, predictedMatch.TeamI, round] \leftarrow eValue$

$Else$

$$eValue \leftarrow maximumEntertainment \times \frac{numberOfRounds - round}{numberOfRounds} - \frac{maximumEntertainment}{4}$$

$$entertainments[predictedMatch.TeamI, predictedMatch.TeamJ, round] \leftarrow eValue$$

$$entertainments[predictedMatch.TeamJ, predictedMatch.TeamI, round] \leftarrow eValue$$

$EndIf$

$EndForEach$

$EndWhile$

$return\ excitements$

$EndFunction$

A variation on this function would be to swap $If(predictedMatch.Result = Draw)$ to $If(predcitedMatch.Result \neq Draw)$. This would result in all the games predicted to result in a draw to occur near the start of the tournament.

User-Interface (UI)

Whilst the purpose of the application is to aid this project in evaluating the use of ILP to solve sport scheduling problem and not as a commercial product, it is still beneficial to produce clear UI designs to guide the implementation. To improve usability of the application the ten usability heuristics for user interface design created by The Nielsen Norman Group have been considered in each of the designs[13].

The application requires the user to perform several actions and modify several settings to set up the application to create a sport schedule. For that reason, a Ribbon User Interface (RUI) lends itself to reducing the complexity of the application. The ribbon in a RUI has limited hierarchical structure, which improves the discoverability of options and user commands. This allows most commands relevant to a task to be only a single click away[14].

In Figure 5, the ribbon includes many commands the user will need to create a schedule. Each section of user commands is presented in a logical order from left to right to make the flow of steps intuitive. The user will need to add teams followed by selecting the tournament type, the type of entertainment heuristic and any travelling tournament options. Then the user can run the application to solve the schedule.

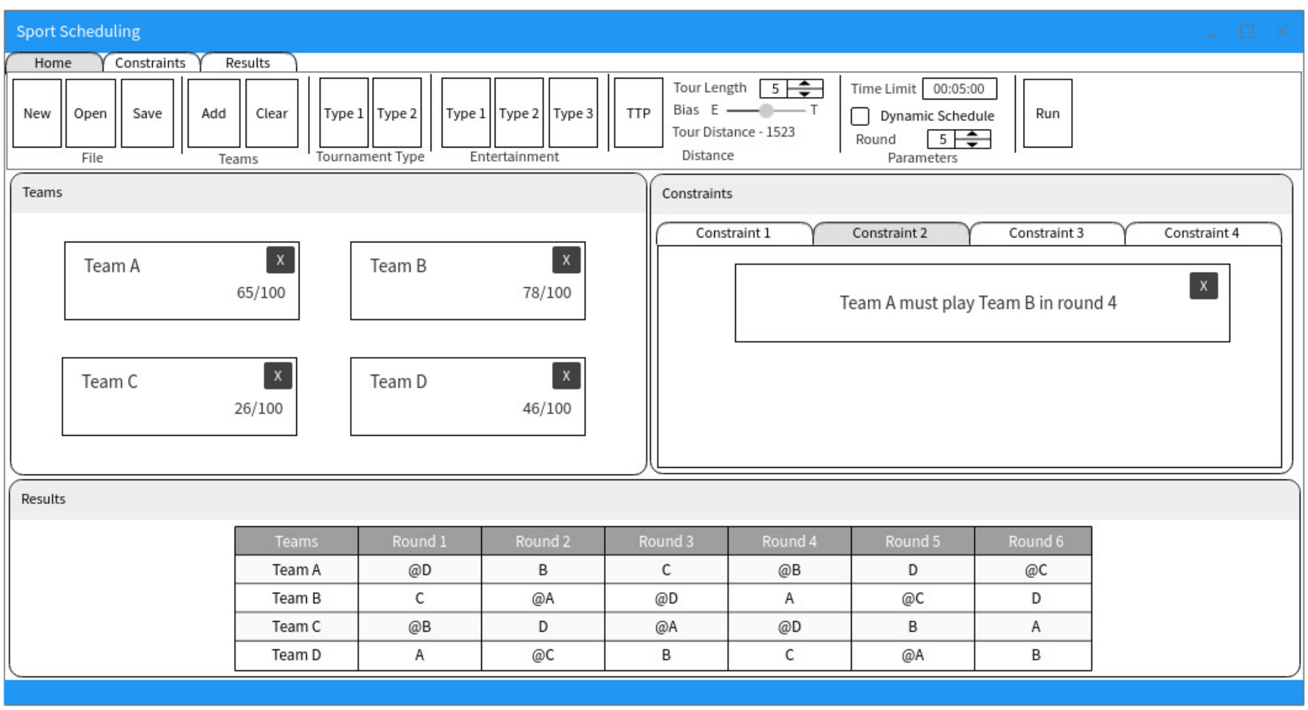

Figure 5 - Main window application mock-up

The rest of the window displays two panels that allow the user to easily see the relevant information about setup of the current schedule to be solved and a third panel displays the results, if any. This lets the user to easily recognize the status of the application with no need to recall previous actions of adding teams and constraints. Each team and constraint is listed in a card style list rather than a table to make it easy to distinguish each item. With each entity represented by a single card, it makes it instinctive to delete any item.

As the distances between teams is not a crucial requirement to create schedules unless the option has been selected, the table to input these distances is concealed on a collapsed panel. This panel will be easily accessible from a section of collapsed panels stored on the left of the application and perform a flyout action when hovered over. The expanded panel can be seen in Figure 6.

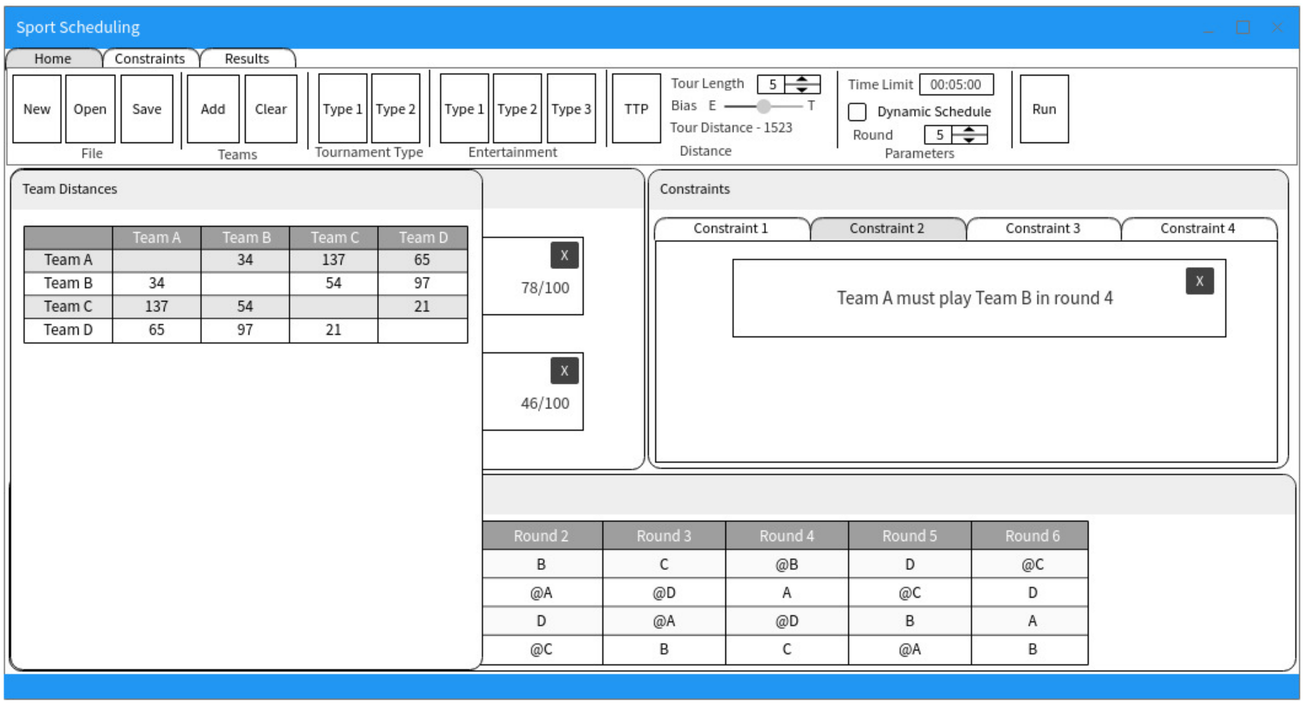

Figure 6 - Distance panel application mock-up

As constraints are not essential to creating schedules but will often be required, they are hidden away in a second tabbed ribbon. This helps the application be less cluttered and overwhelming to the user while still being easily accessible as the ribbon is only one click away. Figure 7 shows the structure of the constraint ribbon. Each user command on this ribbon will create a modal pop-up window, providing a minimal set of options to add the relevant constraint.

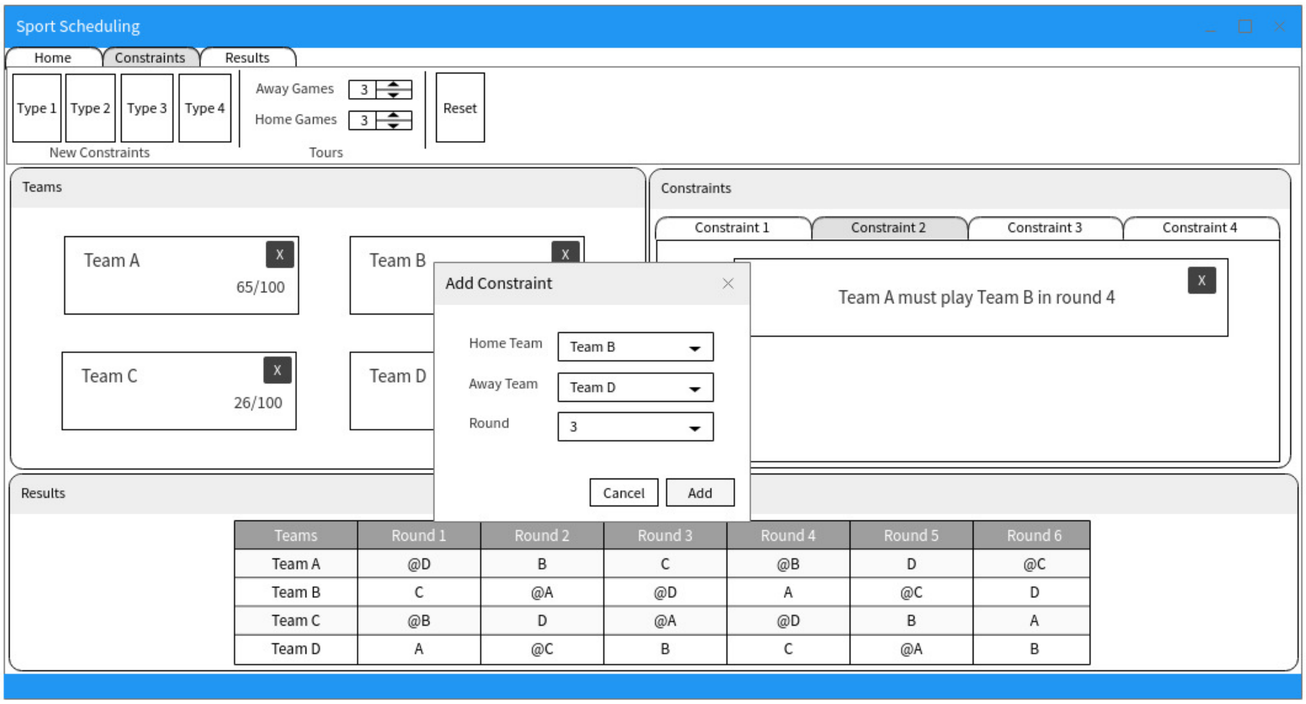

Figure 7 - Constraints application mock-up

The final application design mock-up in Figure 8 shows the results ribbon bar with user commands which are relevant after a schedule has been obtained. It provides the option for the user to determine the best way to view the results, either a table seen in the results panel or a list view like the cards in the priority game panel. The table view allows a quick overview of the whole schedule. Whilst the list of cards is more human readable but takes up more space, resulting in only being able to see a few rounds at a time.

After a tournament is scheduled the user will have the option to select the priority games of the tournament through the results ribbon. The result of this will be presented on a panel that is initially hidden, as it is only required at the end of the process of making the schedule and would clutter the screen if permanently visible.

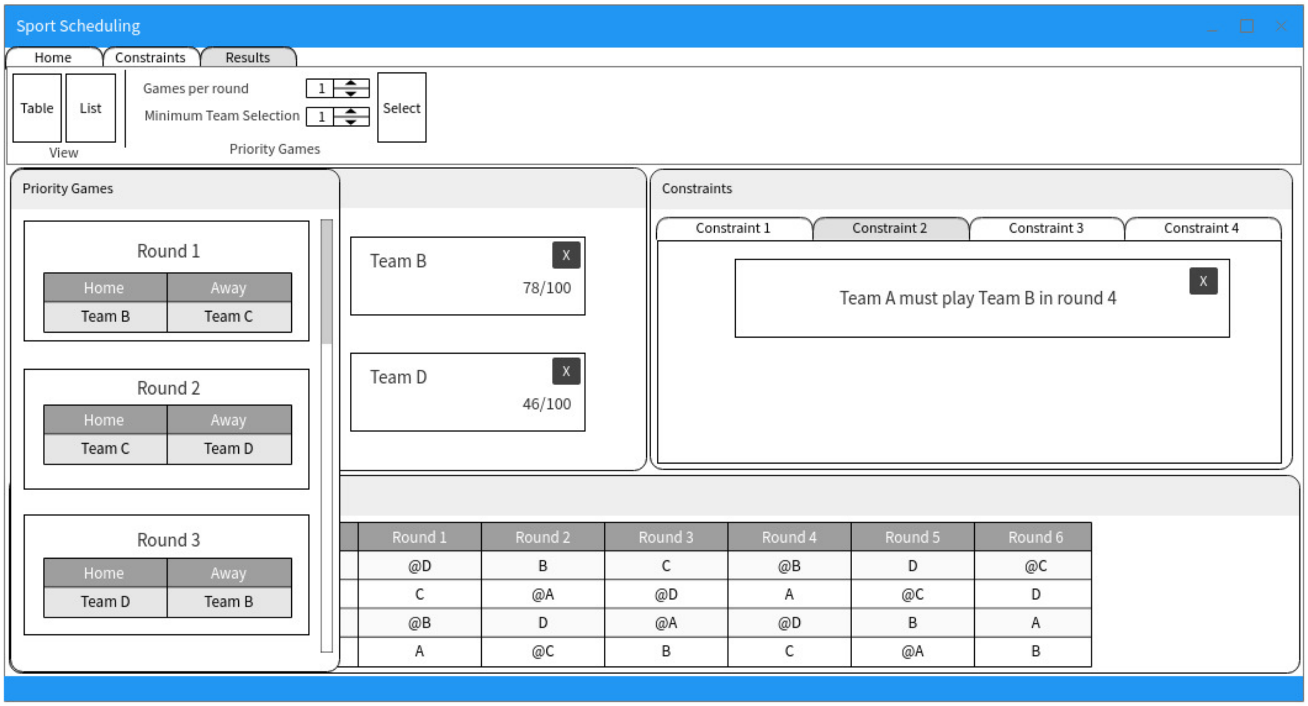

Figure 8 - Results application mock-up

Implementation

The scheduling application has been developed using WPF and the .NET framework using a Model-view-viewmodel (MVVM) design pattern. Additional libraries have also been used to aid the development of the application. Gurobi has been used to efficiently solve the ILP model. While the DevExpress controls and libraries have been used to replace the standard WPF controls to improve the look and feel of the GUI and speed up development.

Limitations

The Gurobi Optimizer is a commercial product by Gurobi Optimization which fully exploits parallelism of a computer to solve the mathematical programs. Fortunately, Gurobi Optimization provides an academic licence, but this only allows the optimizer to run on the computer that has said licence. Consequently, the performance of the optimizer is limited to the hardware specification on the computer. The application can solve the scheduling problems but can take an excessive amount of time for greater number of teams.

If a full software license was possible to acquire, it would allow access to Gurobi Instant Cloud, shifting all the extensive optimization work to the cloud where it can access the resources needed to process the problem faster.

The Application

The development of the application has taken a considerable amount of time to create and contains over 80 source code files. Rather than including code snippets in this section, the pseudocode for algorithms that relate to creating the ILPs have been included in the previous section. A few screenshots of the application with brief descriptions of the demonstrated functionality have been included.

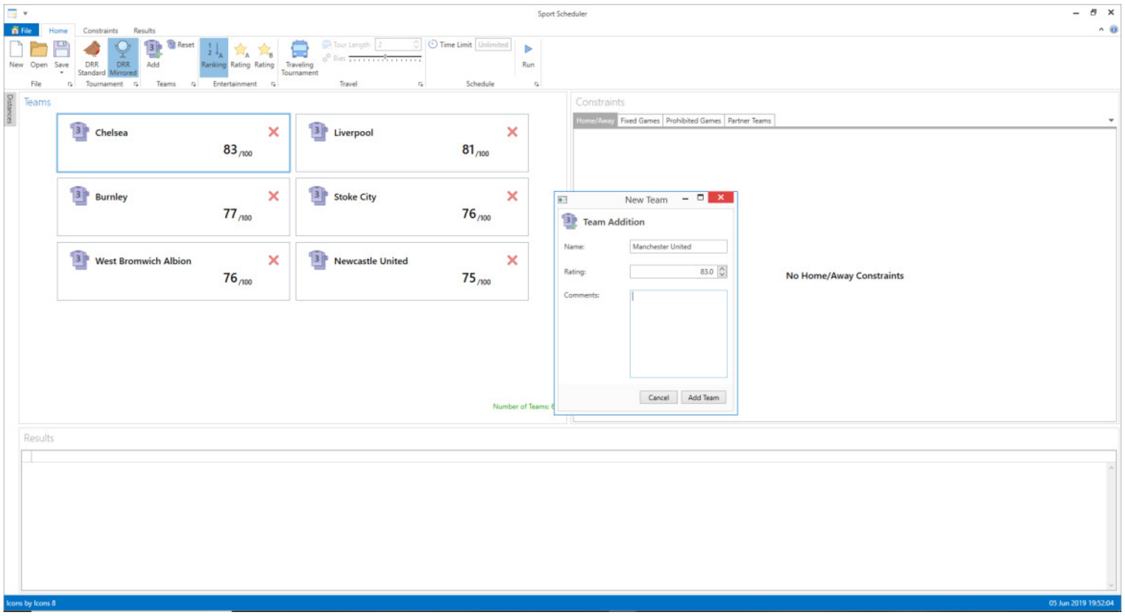

Figure 9 - Team additions

Figure 9 shows the main window of the application, clearly showing the ribbon toolbar with all the user commands. Teams can be added through the team addition button, which brings up a small modal window.

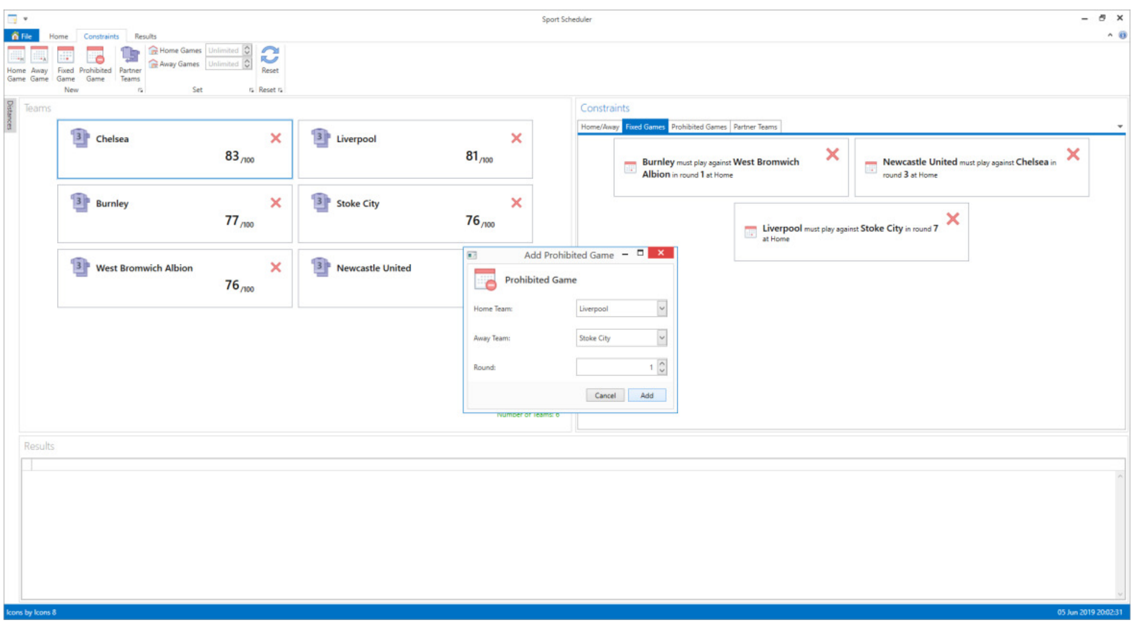

Figure 10 - Constraint addition

Figure 10 shows the constraints tab on the ribbon toolbar, along with how constraints are displayed in the application. Similarly, to the team addition, a modal pop up window appears to add the new constraint.

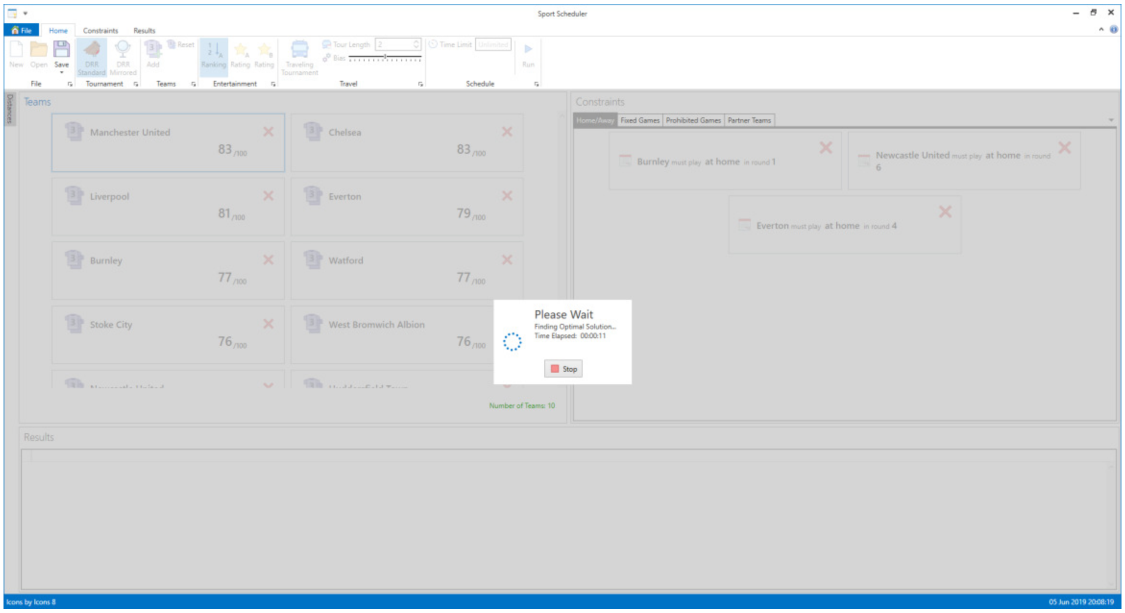

Figure 12 - Schedule optimization

Figure 12 shows the wait message that appear when the application is run. Providing this message with a timer and progress indicator clearly shows that application is working and not unresponsive.

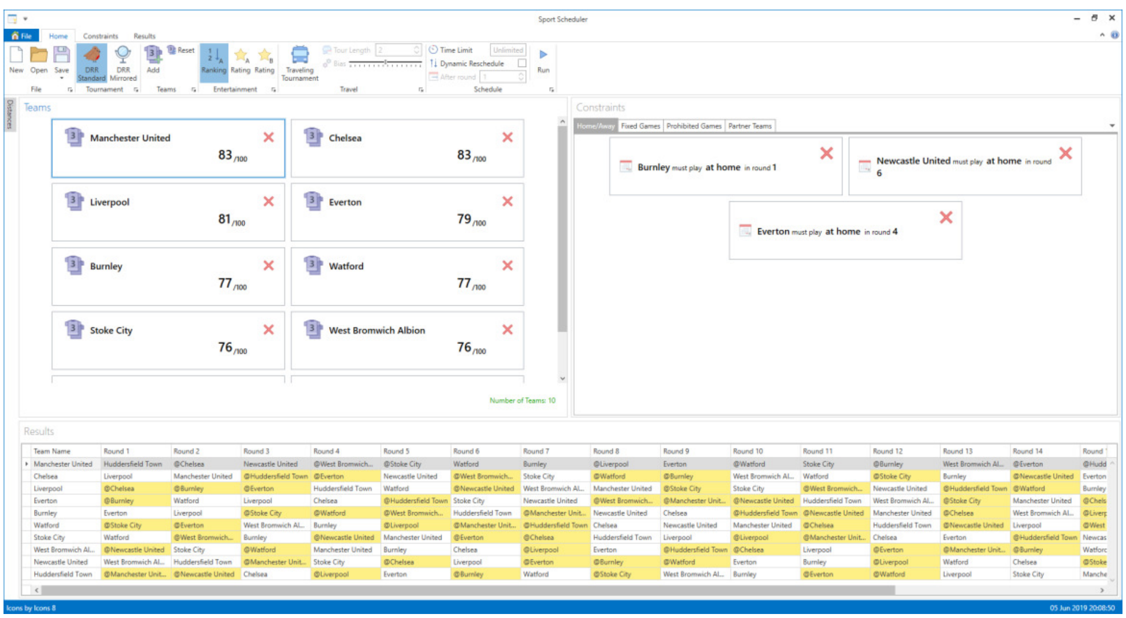

Figure 11 - Application results table

Figure 11 demonstrate the schedule produced by the application in a table style format. Away games in the table contain an @ symbol and are highlighted to make it clear at a first glance.

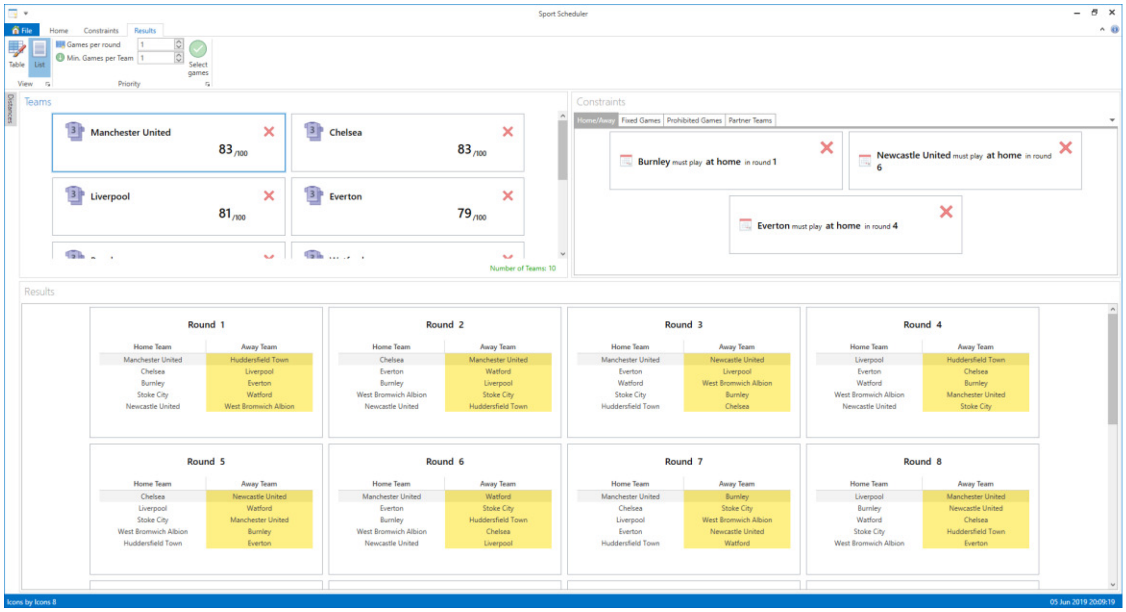

Figure 14 – Alternate schedule view

Figure 14 show the alternative viewing option for the schedule

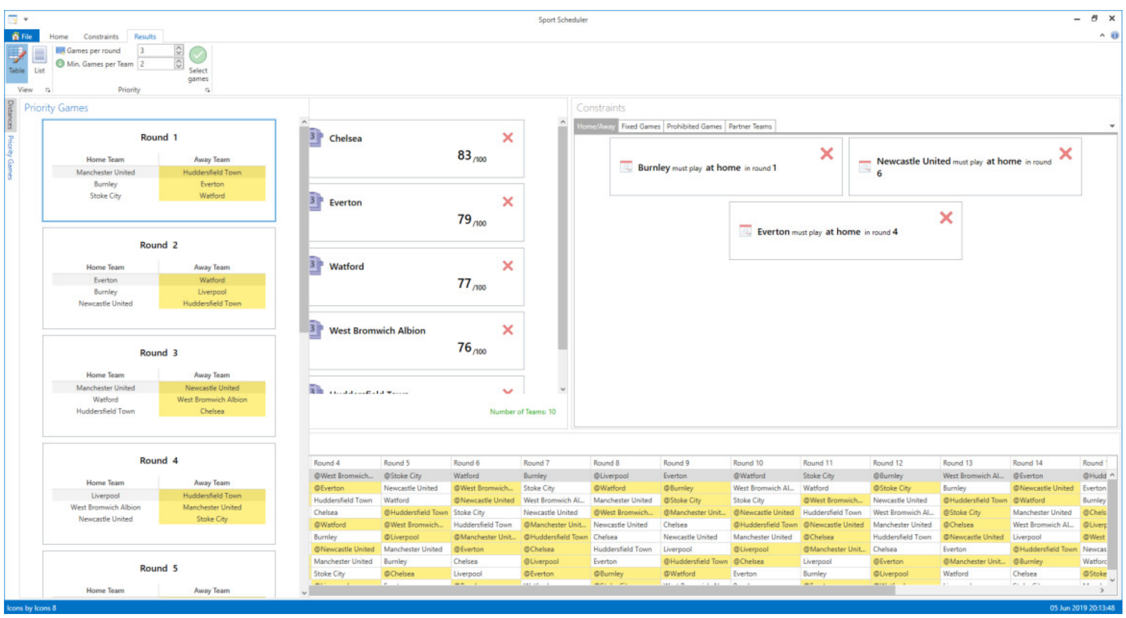

Figure 13 – Priority game selection

Figure 13 demonstrated the view tab on the ribbon toolbar which contains the options to select priority games from a schedule. These priority games selected are then display on a hidden panel.

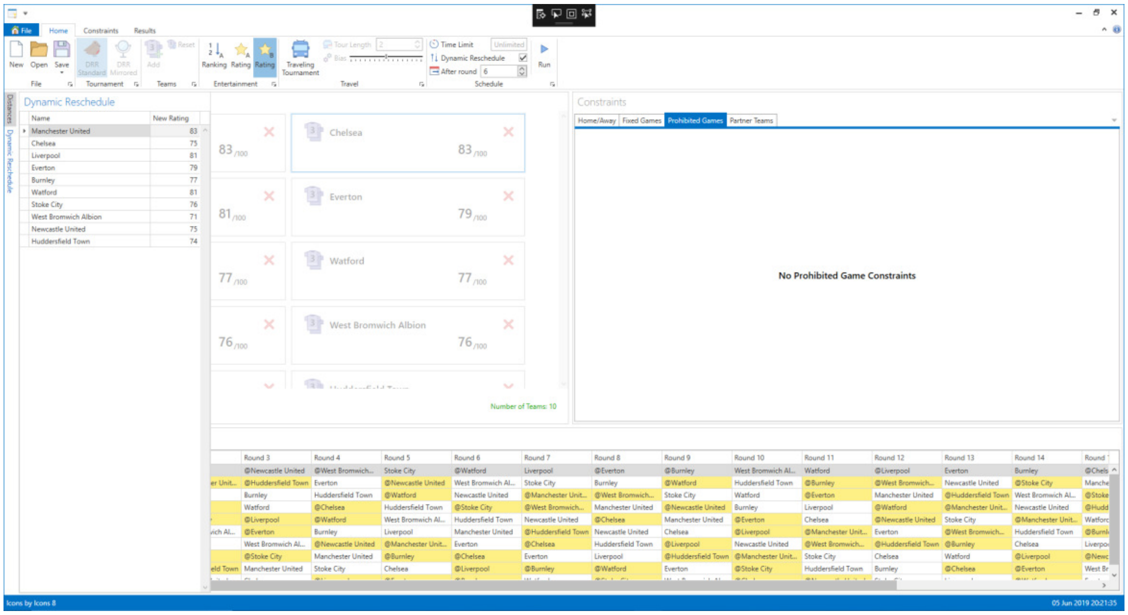

Figure 15 - Dynamic rescheduling

Figure 15 shows the options to dynamically reschedule the tournament. An option to select how many rounds should stay fixed is given alongside a hidden panel that can be used to give teams new rating values. During the rescheduling different optimization heuristics can also be selected. For example, initially the application was focused on create a minimal distance schedule, but during the reschedule an entertainment heuristic can be applied instead.

Results & Evaluation

It is evident that the application can produce valid sport schedules using Gurobi to solve the ILPs. However, there are different entertainment heuristics to compare and evaluate how well it meets the aim of obscuring the overall winner and final position for each team for as long as possible. Similarly, it will be useful to compare how these 'Entertaining' schedules compare to a schedule focused on minimizing distance.

Entertaining Schedules

The application implements two main heuristic to calculate the entertainment of a potential game: by the ranking of teams and by the rating given to the teams. To compare these two methods with respect to the aim for an entertaining schedule, a dataset of teams and match results will be required. Producing a league table at each round of the tournament will demonstrate how the teams are progressing and for how long teams still have something to play for.

Dataset

The English Premier League (EPL) has readily available information on previous year's results, so is an ideal tournament to select teams from with the actual results from their games. Furthermore, football is internationally governed by Fédération Internationale de Football Association (FIFA). FIFA produce and store information on many national and international teams including an associated rating[15]. With these two sources of data, a list of teams can be formed and the league table at each round can be calculated.

With the performance of the application limited by the hardware it is run on, the number of teams in the tournament will need to be reduced so that it can find a schedule in a reasonable amount of time. The EPL typically consists of 20 teams but 10 teams will be extracted and used to evaluate the application. Table 2 show the teams that have been selected.

Team	FIFA Rating
Manchester United	83
Chelsea	83
Liverpool	81
Everton	79
Burnley	77
Watford	77
Stoke City	76
West Bromwich Albion	76
Newcastle United	75
Huddersfield Town	74

Table 2 - Teams and FIFA rating

Home/Away	MUN	CHE	L IV	EVE	BUR	WAT	STK	WBA	NEW	HUD
Manchester United	-	W	W	W	D	W	W	L	W	W
Chelsea	W	-	W	W	L	W	W	W	W	D
Liverpool	D	D	-	D	D	W	D	D	W	W
Everton	L	D	D	-	L	W	W	D	W	W
Burnley	L	L	L	W	-	W	W	L	W	D
Watford	L	W	D	W	L	-	L	W	W	L
Stoke City	D	L	L	L	D	D	-	W	L	W
West Bromwich Albion	L	L	D	D	L	D	D	-	D	L
Newcastle United	W	W	D	L	D	L	W	L	-	W
Huddersfield Town	W	L	L	L	D	W	D	W	W	-
W – Home team win, L – Home team loss, D – Draw										

Table 3 - EPL 2017/18 match results

The winner of the tournament is decided by whoever has the most points on the League Table at the end of the tournament. Points are awarded based on performances during the tournament. If a team wins a game, they are awarded 3 points. If the result is a draw, each team receives 1 point.

Entertainment based on Ranking

With the EPL data obtained there is a clear team ranking. Unfortunately, there is no readily available data on the exact constraints on the 2017/18 tournament. For the purpose of analyzing the entertainment heuristic, the only constraint to be added will be the restriction on the number of consecutive home or away games. A maximum of 2 home or away game may occur in a row.

Table 4 represents the schedule produced by the sport scheduling application when the entertainment of all possible games is calculated based on team rankings. The @ symbol in the schedule represents the team is playing away against the other team.

Team/Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Manchester United	@ HUD	@ CHE	NEW	@ WBA	STK	WAT	@ BUR	EVE	LIV	@ WAT	@ STK	BUR	WBA	@ EVE	HUD	@ LIV	@ NEW	CHE
Chelsea	@ LIV	MUN	HUD	@ EVE	NEW	WBA	@ STK	WAT	@ BUR	STK	@ WAT	@ WBA	BUR	@ NEW	EVE	@ HUD	LIV	@ MUN
Liverpool	CHE	EVE	@ BUR	@ HUD	WAT	@ NEW	@ WBA	STK	@ MUN	NEW	WBA	@ STK	HUD	@ WAT	BUR	MUN	@ CHE	@ EVE
Everton	@ BUR	@ LIV	WAT	CHE	@ HUD	@ STK	NEW	@ MUN	@ WBA	WBA	HUD	@ NEW	STK	MUN	@ CHE	@ WAT	BUR	LIV
Burnley	EVE	@ STK	LIV	@ WAT	@ WBA	HUD	MUN	@ NEW	CHE	@ HUD	NEW	@ MUN	@ CHE	WBA	@ LIV	STK	@ EVE	WAT
Watford	STK	WBA	@ EVE	BUR	@ LIV	@ MUN	HUD	@ CHE	@ NEW	MUN	CHE	@ HUD	NEW	LIV	@ WBA	EVE	@ STK	@ BUR
Stoke City	@ WAT	BUR	WBA	@ NEW	@ MUN	EVE	CHE	@ LIV	HUD	@ CHE	MUN	LIV	@ EVE	@ HUD	NEW	@ BUR	WAT	@ WBA
West Bromwich Albion	NEW	@ WAT	@ STK	MAN	BUR	@ CHE	LIV	@ HUD	EVE	@ EVE	@ LIV	CHE	@ MUN	@ BUR	WAT	@ NEW	HUD	STK
Newcastle United	@ WBA	HUD	@ MUN	STK	@ CHE	LIV	@ EVE	BUR	WAT	@ LIV	@ BUR	EVE	@ WAT	CHE	@ STK	WBA	MUN	@ HUD
Huddersfield Town	MUN	@ NEW	@ CHE	LIV	EVE	@ BUR	@ WAT	WBA	@ STK	BUR	@ EVE	WAT	@ LIV	STK	@ MUN	CHE	@ WBA	NEW

Table 4 - Schedule for entertainment by ranking

Using the schedule in Table 4 and the match results in Table 3 it is possible to develop the league table for each round of the tournament. The teams that are currently in the lead are highlighted in green and those in last place are highlighted in red.

Team/Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Manchester United	0	0	3	6	9	12	15	18	21	24	25	26	26	29	32	33	33	**36**
Chelsea	1	4	5	6	9	12	15	18	21	24	24	27	27	27	30	33	36	**36**
Liverpool	1	2	5	8	11	12	13	14	14	17	18	21	24	25	26	27	27	**28**
Everton	0	1	4	5	8	11	14	14	15	16	19	22	25	25	25	25	25	**26**
Burnley	3	4	4	7	10	11	11	12	12	13	16	17	20	20	21	24	27	**30**
Watford	0	3	3	3	3	3	3	3	6	6	9	9	12	13	14	17	18	**18**
Stoke City	3	4	7	7	7	7	7	8	11	11	12	12	12	13	13	13	14	**15**
West Bromwich Albion	1	1	1	1	1	1	2	2	3	4	5	5	8	11	12	15	15	**16**
Newcastle United	1	4	4	7	7	8	8	9	9	9	9	9	9	12	15	15	18	**18**
Huddersfield Town	3	3	4	4	4	5	8	11	11	12	12	15	15	16	16	16	19	**22**

Table 5 - League table standing at each round based on ranking

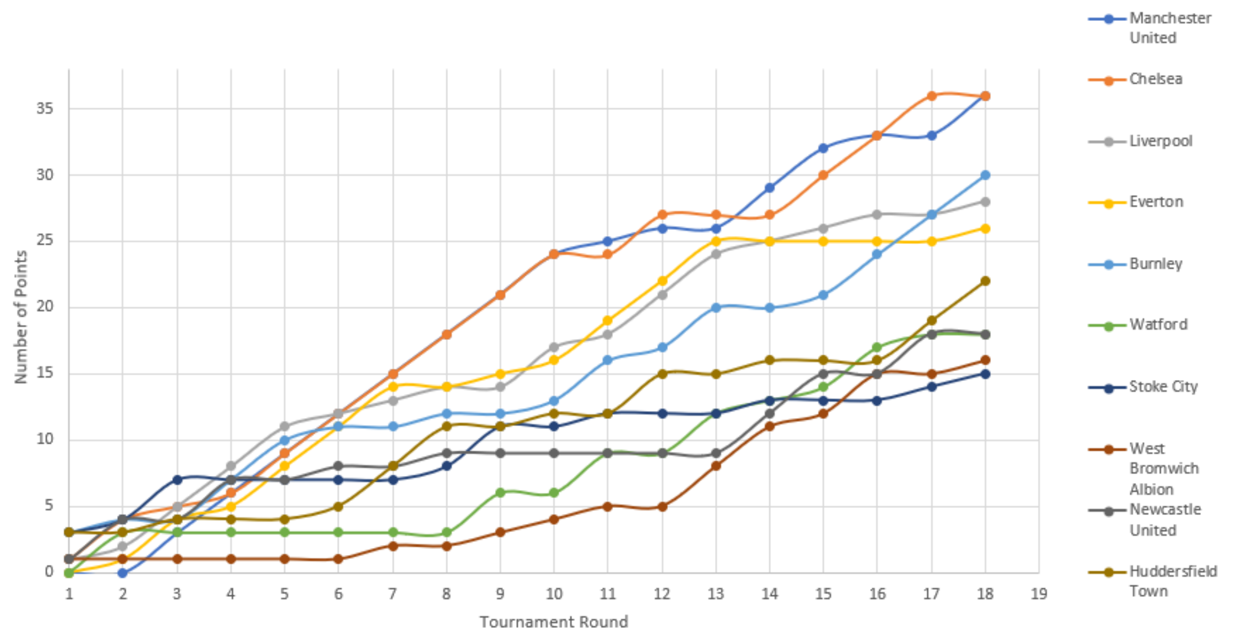

Figure 16 - Graph of league table progression for ranking

The objective for creating entertainment values based on the rank of a team was to force teams ranked closely to play each other near the end of the tournament. Equally, these closely ranked teams should avoid playing each other near the start of the tournament. Looking at the schedule in Table 4 it is evident that games scheduled in the final round all occur with teams ranked next to each other. The rounds 16 and 17 also include several games between teams whose rank differ by one or two places. Understandably, not all the games in the final few rounds are against teams of similar ranks, otherwise it would be impossible to schedule a tournament that meets the DRR and consecutive game constraints.

Table 5 reveals the progression of the league table at each round. It shows that, in the second half of the tournament each team's overall score was within maximum of five points from any other team. This theoretically allows movement within the league table of any team, should one perform well and hence gain points to close that gap. Hence entertainment is maximized since one team has not definitively won or lost overall yet.

While all teams are relatively close to other teams throughout the tournament, it is worth mentioning that the two teams that end up in the top position start jostling for this top position in round 6. In addition to this the two teams at that eventually end up at the bottom of the league, are consistently the last two teams for most of the tournament. Making it clear who is likely win the tournament and who is likely to lose early on.

Entertainment based on Rating and Match Predictions

To be able to compare the different entertainment heuristics, the same dataset and constraints have been used to create the schedule in Table 6. The entertainment values for the games in this schedule were based on the ratings of teams to predict the outcome of games. Then using these predictions, entertainment values are created such that games that are likely to be draw, are more inclined to be scheduled near the end.

Team/Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Manchester United	WBA	STK	@ BUR	@ WAT	EVE	@ NEW	LIV	@ HUD	CHE	WAT	@ STK	@ WBA	NEW	BUR	@ EVE	HUD	@ LIV	@ CHE
Chelsea	BUR	WAT	@ STK	@ HUD	NEW	@ EVE	WBA	@ LIV	@ MUN	STK	@ BUR	HUD	@ WBA	EVE	@ WAT	LIV	@ NEW	MUN
Liverpool	NEW	@ WBA	WAT	@ STK	HUD	BUR	@ MUN	CHE	@ EVE	@ NEW	WBA	@ WAT	@ BUR	STK	@ HUD	@ CHE	MUN	EVE
Everton	@ STK	@ NEW	HUD	WBA	@ MUN	CHE	WAT	@ BUR	LIV	@ WBA	NEW	STK	@ HUD	@ CHE	MUN	BUR	@ WAT	@ LIV
Burnley	@ CHE	@ HUD	MUN	NEW	@ WBA	@ LIV	STK	EVE	@ WAT	HUD	CHE	@ NEW	LIV	@ MUN	WBA	@ EVE	@ STK	WAT
Watford	HUD	@ CHE	@ LIV	MUN	STK	@ WBA	@ EVE	NEW	BUR	@ MUN	@ HUD	LIV	@ STK	WBA	CHE	@ NEW	EVE	@ BUR
Stoke City	EVE	@ MUN	CHE	LIV	@ WAT	HUD	@ BUR	WBA	@ NEW	@ CHE	MUN	@ EVE	WAT	@ LIV	NEW	@ WBA	BUR	@ HUD
West Bromwich Albion	@ MUN	LIV	@ NEW	@ EVE	BUR	WAT	@ CHE	@ STK	HUD	EVE	@ LIV	MUN	CHE	@ WAT	@ BUR	STK	@ HUD	NEW
Newcastle United	@ LIV	EVE	WBA	@ BUR	@ CHE	MUN	@ HUD	@ WAT	STK	LIV	@ EVE	BUR	@ MUN	HUD	@ STK	WAT	CHE	@ WBA
Huddersfield Town	@ WAT	BUR	@ EVE	CHE	@ LIV	@ STK	NEW	MUN	@ WBA	@ BUR	WAT	@ CHE	EVE	@ NEW	LIV	@ MUN	WBA	STK

Table 6 - Schedule for entertainment by rating

Using the schedule in Table 6 and the match results in Table 3 it is possible to produce the league table for each round of the tournament.

Team/Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Manchester United	0	3	6	9	12	12	15	15	18	21	22	25	28	29	32	35	36	**36**
Chelsea	0	3	6	9	12	13	16	17	17	20	23	24	27	30	30	33	33	**36**
Liverpool	3	4	7	10	13	14	14	15	16	17	18	19	22	23	26	26	27	**28**
Everton	3	6	9	10	10	11	14	14	15	16	19	22	25	25	25	25	25	**26**
Burnley	3	4	4	7	10	11	14	17	20	21	21	22	22	23	23	26	27	**30**
Watford	0	0	0	0	0	1	1	4	4	4	4	5	6	9	12	15	18	**18**
Stoke City	0	0	0	0	3	6	6	9	9	9	10	10	11	12	12	13	14	**15**
West Bromwich Albion	3	4	7	8	8	9	9	9	9	10	11	11	11	11	14	15	15	**16**
Newcastle United	0	0	0	0	0	3	3	3	6	7	7	8	8	11	14	14	17	**18**
Huddersfield Town	3	4	7	7	7	7	10	13	13	14	17	18	18	18	18	18	21	**22**

Table 7 - League table standing at each round based on rating

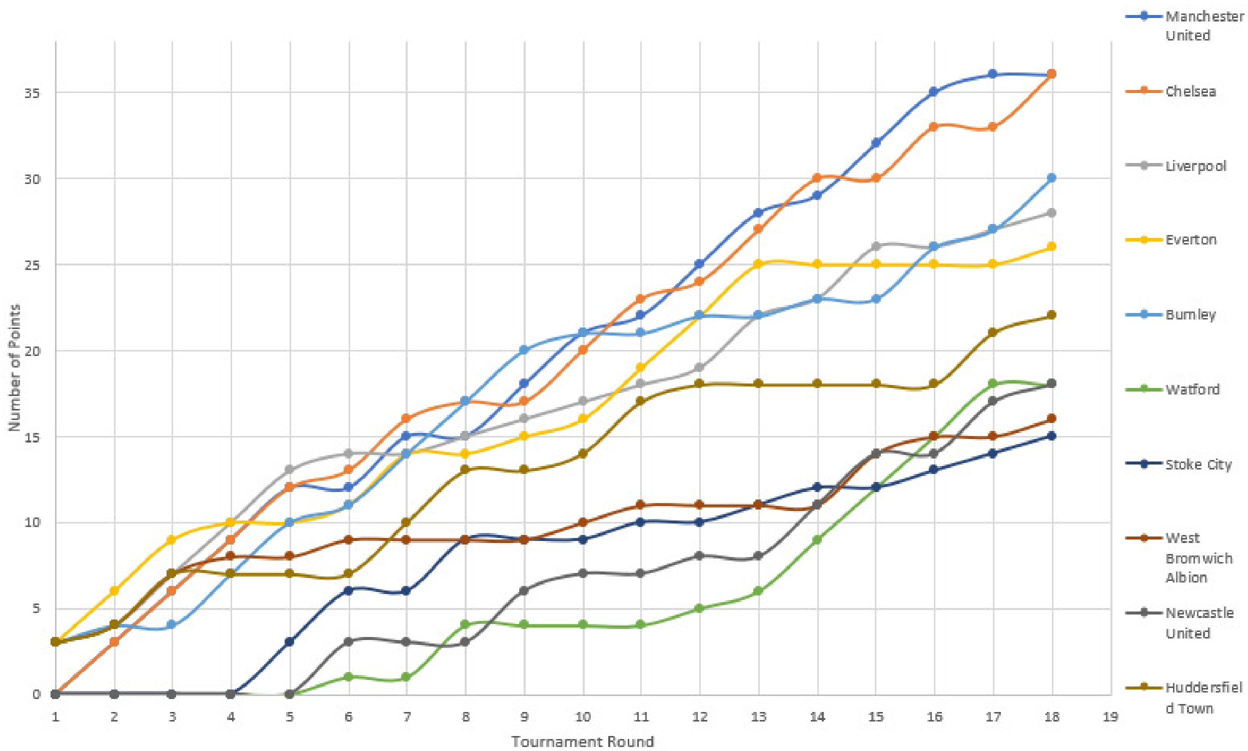

Figure 17 - Graph of league table progression based on rating

It is apparent in Table 7 that there has been a preference to schedule games with more predictable out comes nearer the start and less predictable, i.e. potential draws, near the end of the tournament. This is shown by the steeper increase for some team's overall points at the start of the tournament compared to other teams who struggle to get a single point. Comparing the mean of the top five teams in Table 5 and Table 7 you get 7 and 9.2 respectively. Similarly, the mean of the bottom five teams is 3.8 and 2.8. This makes it evident that some teams will struggle to breakout of the bottom of the league. But this doesn't mean they can give up as there are several teams near the bottom and will still need to perform as well as possible to not come last.

Another difference seen in Table 7 compared to Table 5 is the top and bottom teams at each round. The final top two teams don't start battling for the lead until much later in the tournament. With Liverpool, Everton and Burnley each having a small stint at being the leader. In effect obscuring the final winners for a longer period.

For the majority of the tournament Watford appear as if they are going to come last until the overtake Stoke City in the final few rounds, in contrast to Table 5 where it was West Bromwich Albion at the bottom for a long period of time.

Variation of Entertainment values based on Ratings

Briefly mentioned earlier in the report, a variation producing entertainment values based on ratings would be to swap the preference for where games with predicted outcomes occur in the final schedule. Originally, games that are likely to end in a draw are preferred to occur near the end of the tournament. But swapping this would force them to be near the start.

Although this would result in more predictable games occurring near the end. In theory this would be less entertaining in real-life as more unpredictable games are more entertaining when it could result in the overall winner being decided.

Table 8 is the schedule produced by the application for the variation on the ratings.

Team/Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Manchester United	@ CHE	@ NEW	LIV	HUD	@ EVE	@ WBA	BUR	@ WAT	STK	CHE	@ LIV	@ STK	WAT	EVE	@ BUR	@ HUD	WBA	NEW
Chelsea	MUN	@ LIV	@ HUD	EVE	@ NEW	@ WAT	WBA	@ STK	BUR	@ MUN	HUD	LIV	@ EVE	NEW	@ WBA	STK	WAT	@ BUR
Liverpool	EVE	CHE	@ MUN	STK	@ HUD	@ BUR	WAT	NEW	@ WBA	@ EVE	MUN	@ CHE	@ NEW	HUD	@ WAT	BUR	@ STK	WBA
Everton	@LIV	BUR	@ WAT	@ CHE	MUN	@ HUD	STK	WBA	@ NEW	LIV	@ BUR	WAT	CHE	@ MUN	HUD	@ WBA	NEW	@ STK
Burnley	STK	@ EVE	NEW	@ WAT	@ WBA	LIV	@ MUN	HUD	@ CHE	WAT	EVE	@ NEW	WBA	@ STK	MUN	@ LIV	@ HUD	CHE
Watford	NEW	@ WBA	EVE	BUR	@ STK	CHE	@ LIV	MUN	@ HUD	@ BUR	STK	@ EVE	@ MUN	WBA	LIV	@ NEW	@ CHE	HUD
Stoke City	@ BUR	HUD	@ WBA	@ LIV	WAT	NEW	@ EVE	CHE	@ MUN	WBA	@ WAT	MUN	@ HUD	BUR	@ NEW	@ CHE	LIV	EVE
West Bromwich Albion	@ HUD	WAT	STK	@ NEW	BUR	MUN	@ CHE	@ EVE	LIV	@ STK	NEW	HUD	@ BUR	@ WAT	CHE	EVE	@ MUN	@ LIV
Newcastle United	@ WAT	MUN	@ BUR	WBA	CHE	@ STK	HUD	@ LIV	EVE	@ HUD	@ WBA	BUR	LIV	@ CHE	STK	WAT	@ EVE	@ MUN
Huddersfield Town	WBA	@ STK	CHE	@ MUN	LIV	EVE	@ NEW	@ BUR	WAT	NEW	@ CHE	@ WBA	STK	@ LIV	@ EVE	MUN	BUR	@ WAT

Table 8 - Schedule for a variation of entertainment by rating

Using the schedule in Table 8 and the match results in Table 3 the league table for each round of the tournament can be produced.

Team/Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Manchester United	0	0	3	6	9	12	13	16	19	22	23	24	27	30	33	33	33	**36**
Chelsea	3	4	7	10	10	10	13	16	16	16	17	20	21	24	27	30	33	**36**
Liverpool	1	2	2	3	6	9	12	15	16	17	18	18	19	22	23	24	27	**28**
Everton	1	1	1	1	1	4	7	8	11	12	12	15	16	16	19	20	23	**26**
Burnley	3	6	9	12	15	15	16	17	20	23	26	27	27	28	28	29	30	**30**
Watford	3	4	7	7	8	11	11	11	11	11	11	11	11	14	15	18	18	**18**
Stoke City	0	3	4	5	6	6	6	6	6	9	12	13	14	15	15	15	15	**15**
West Bromwich Albion	0	1	2	5	5	5	5	6	7	7	8	8	11	11	11	12	15	**16**
Newcastle United	0	3	3	3	6	9	12	12	12	12	13	14	15	15	18	18	18	**18**
Huddersfield Town	3	3	3	3	3	3	3	4	7	10	11	14	15	15	15	18	19	**22**

Table 9 - League table standing at each round based on a variation of the rating heuristic

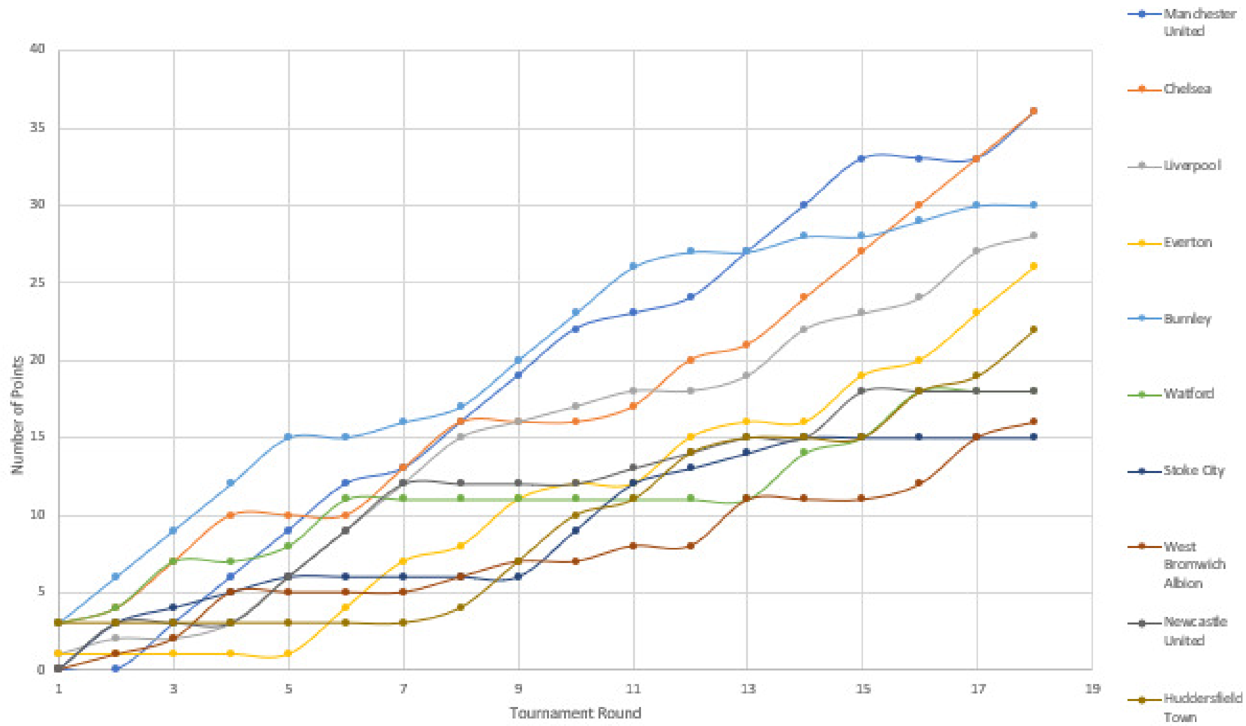

Figure 18 - Graph of league table progression based on a variation of the rating heuristic

As expected, there is not a concentration of games between teams of similar ability in the final few rounds of the tournament. This makes it easier to predict the outcome of each game.

Surprisingly, the development of the league table in Table 9 noticeably differs to both Table 5 and Table 7. While there are few changes in the leader between each round, Manchester United don't obtain the lead until round 13 and Chelsea, when excluding the first round, only becomes joint leader in round 17. At the opposite end of the league table several teams have a period where they are at the bottom, in contrast to just one or two teams. This schedule has effectively hidden the final positions of the teams for more rounds of the tournament than either of the other two schedules.

Travelling Tournament Schedules

The secondary purpose of the is application is to solve the TTP. Using a set of teams with a corresponding distance matrix it is possible to compare the overall distance travelling in a tournament when the ILP is maximizing entertainment, minimizing the distance travelled or a blend of both objectives.

Dataset

The same data from the EPL can be used to evaluate the TTP. Though the number of teams will be reduced to six teams, shown in Table 10, as it is a more computationally intensive task. The application will also be setup to solve a Mirrored DDR tournament with a maximum tour length of 2. These additional constraints reduce the number of feasible solutions, and so speed up the time to find the optimal solution but still provide enough variation in the results between an entertaining schedule and a minimal distance schedule.

Team	FIFA Rating
Chelsea	83
Liverpool	81
Burnley	77
Stoke City	76
West Bromwich Albion	76
Newcastle United	75

Table 10 - Reduced number of teams

Table 11 represents the number of miles between the stadiums for each of the teams as a distance matrix.

Teams	CHE	LIV	BUR	STK	WBA	NEW
CHE	-	224	250	171	134	299
LIV	224	-	52.4	58.6	93.7	174
BUR	250	52.4	-	83.4	118	116
STK	171	58.6	83.4	-	39.5	194
WBA	134	93.7	118	39.5	-	212
NEW	299	174	116	195	212	-

Table 11 - Distance matrix

Control Schedule

To evaluate the distance minimization of TTP a control schedule is required to compare it against. Table 12 is the schedule when using entertainment maximization with the ranking heuristic.

Team/Round	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Chelsea	WBA	@ STK	@ BUR	NEW	LIV	@ WBA	STK	BUR	@ NEW	@ LIV
Liverpool	STK	@ NEW	@ WBA	BUR	@ CHE	@ STK	NEW	WBA	@ BUR	CHE
Burnley	NEW	@ WBA	CHE	@ LIV	STK	@ NEW	WBA	@ CHE	LIV	@ STK
Stoke City	@ LIV	CHE	@ NEW	WBA	@ BUR	LIV	@ CHE	NEW	@ WBA	BUR
West Bromwich Albion	@ CHE	BUR	LIV	@ STK	NEW	CHE	@ BUR	@ LIV	STK	@ NEW
Newcastle United	@ BUR	LIV	STK	@ CHE	@ WBA	BUR	@ LIV	@ STK	CHE	WBA
Total distance travelled – 7178.8										

Table 12 - Control schedule

Distance Minimization

Using the same constraints for the control schedule the application can produce a minimum distance tournament schedule shown in Table 13.

Team/Round	1	2	3	4	5	6	7	8	9	10
Chelsea	@ LIV	@ STK	NEW	BUR	@ WBA	LIV	STK	@ NEW	@ BUR	WBA
Liverpool	CHE	@ BUR	STK	WBA	@ NEW	@ CHE	BUR	@ STK	@ WBA	NEW
Burnley	@ NEW	LIV	WBA	@ CHE	@ STK	NEW	@ LIV	@ WBA	CHE	STK
Stoke City	WBA	CHE	@ LIV	@ NEW	BUR	@ WBA	@ CHE	LIV	NEW	@ BUR
West Bromwich Albion	@ STK	NEW	@ BUR	@ LIV	CHE	STK	@ NEW	BUR	LIV	@ CHE
Newcastle United	BUR	@ WBA	@ CHE	STK	LIV	@ BUR	WBA	CHE	@ STK	@ LIV
Total distance travelled – 6657.3										

Table 13 - Minimal distance schedule

Other than the obvious reduction in the total distance travelled by all teams, there are noticeable differences in the pattern of scheduled games. The control schedule has a total 16 tours that only involve a single game, whilst minimal distance schedule only contains 8 of these such tours. If it was possible to create a schedule where all the teams only went on single game tours, it would result in the maximum total distance that could be travelled. So understandably, the ILP will want to reduce the number of single game tours.

It is apparent that this minimum distance schedule would not be considered an entertaining schedule based on the objective to schedule games of teams ranked closely near the end. Both Liverpool and Chelsea have been scheduled to compete against West Bromwich Albion and Newcastle, respectively, which have a large difference in ranking. Likewise, at the start of the tournament Chelsea and Stoke City are schedule to compete against teams that are ranked next to them.

In a real-world scenario it is unlikely that schedule will only maximize or minimize a single objective. Combining and weighting the two objectives is one solution.

Blended Approach

Blending the two objective functions involves weighting the two objectives individually and then combining them to make one single objective. So that one objective function doesn't overpower, the entertainment values need to be adjusted so that they can be comparable to the distances of the tours. The application has been setup to use a maximum entertainment value of the sum of all distances between teams divided by the number of rounds. Table 14 is the schedule produced in this blended approach.

Team/Round	1	2	3	4	5	6	7	8	9	10
Chelsea	WBA	STK	@ NEW	BUR	LIV	@ WBA	@ STK	NEW	@ BUR	@ LIV
Liverpool	@ NEW	@ WBA	BUR	@ STK	@ CHE	NEW	WBA	@ BUR	STK	CHE
Burnley	@ STK	NEW	@ LIV	@ CHE	WBA	STK	@ NEW	LIV	CHE	@ WBA
Stoke City	BUR	@ CHE	@ WBA	LIV	NEW	@ BUR	CHE	WBA	@ LIV	@ NEW
West Bromwich Albion	@ CHE	LIV	STK	@ NEW	@ BUR	CHE	@ LIV	@ STK	NEW	BUR
Newcastle United	LIV	@ BUR	CHE	WBA	@ STK	@ LIV	BUR	@ CHE	@ WBA	STK
Total distance travelled – 6815.5										

Table 14 - Blended approach schedule

The reduction of single game tours from the control schedule to the blended schedule is then equal to the reduction from the minimal distance schedule. While the blended schedule only contains 8 single games tours, the total distance is still slightly greater than the minimum possible. This is owing to the different combination of two game tours to accommodate an increase in entertainment in this schedule.

As previously stated, the minimal distance schedule doesn't conform to any of the entertainment objectives. The blend approach however does contain some of the desirable features such as two closely ranked teams, Liverpool and Chelsea, being scheduled to compete in the final round. Whilst the other team in the round aren't necessarily against a team ranked directly next to them, the difference in ranking is two, which is adequate. In parallel to this, most teams at the start of the tournament are competing against teams with a reasonable difference in their ranking.

Summary

Overall the application is capable of producing variety of sport schedules for a single set of teams. Each schedule has its own characteristic that makes it better than a simple uninformed schedule. Plus, each one has its own advantages and disadvantages when compared to one another for the dataset used in this evaluation stage.

When the entertainment values are selected based on ranking of teams, it meets the objective to schedule games against teams of similar ranks at the end of the tournament but doesn't obscure the final standing of the teams. Using the team ratings to predict results and assigning entertainment values to encourage games predicted as a draw near the end of the tournament, fairs better at obscuring the final league table result. But this had the negative trait that it effectively split the teams into two half, one which may be able to win the tournament and the other not trying to come last. But it did encourage the games in the final rounds to be between of similar ratings. Unexpectedly, when grouping the games predicted as a draw near the start of the tournament it performed the best at concealing the overall winners and losers of the tournament. But as anticipated, games near the end of the tournament were not against teams of similar rank and games near the start were.

The application also managed to create minimal distance tournament which reduced the overall distance travelled by 7.25% when compared to a similar entertainment schedule of the dataset used. Obviously, this didn't produce an entertaining schedule, but when blending the two approaches a more balanced schedule was created. The reduction in distance was still 5% on the pure entertaining schedule but now includes some of the sought-after features of an entertaining schedule.

Conclusions

The project's original aim was to create a sport schedule application to evaluate the use of Integer Linear Programming for creating entertaining sport schedules. It can be stated that the original aim has been achieved. The application provides a wide range of options to experiment with in a clear and user-friendly manner. It solves sport scheduling problems with the goal of creating an 'entertaining' sport schedule. Furthermore, different 'entertainment' heuristics can be selected. Each one of these heuristics has then been evaluated in this report, highlighting their strengths and weaknesses.

Further topics have also been explored and implemented, such as the Travelling Tournament Problem. These minimal distance schedules have been compared to the entertainment schedules. This has then been extended to create a mixed entertainment and minimal distance schedule as this would attempt to balance the needs of the media companies and sports teams as previously discussed.

Future Works

Although all the aims and requirements of the project have been accomplished, there are still many possibilities to improve and extend the application.

Improvements

A couple of possible improvements that could be made to the application are in the creation of the entertainment values for possible games and the overall performance of the application.

Improved Match Predictions

Currently the calculation of entertainment values based on team ratings is produced by predicting the result of each match by an uninformed difference in the rating of two teams. In the project it has had the intended effect but could be vastly improved.

Rather than basing the predictions of match results on a difference in rating other sources of information could be used. For example, collecting the results of previous years, examining the outcome of each of the games and use this to predict result. An alternative approach could be use betting odds, leaving the predictions to experts. Better predictions of results would then more accurately assign an entertainment value to possible games and increase the overall entertainment of a schedule.

Gurobi Instant Cloud

The main draw-back of the application is the run time of the Gurobi Optimizer to solve ILPs as this is directly affected by the hardware specification machine it is being run on. Whilst not included in the academic licence, one option could be to completely remove the

Gurobi Optimizer from the application. Instead, the application would deal with forming the ILPs and sent a request to Gurobi's Instant Cloud. This would significantly improve current the computational power of the application and allow scheduling a larger number of teams in much less time.

Extensions

Further work to the application and project could be to extend the scope to include 'break' minimization in a tournament.

Break Minimization

A common quality measure for tournament schedule is to look at how many 'breaks' a tournament contains. A break is considered to occur if a team plays two consecutive home or away games[16]. Ideally the fewer breaks in the tournament there are the better the quality of that schedule. Therefore, there is a need to minimize these breaks. As this is another problem that focuses on minimizing an objective, it would be possible to model it using an ILP using a series of constraints.

Reflection of Learning

During this project a large amount of time was used researching the use of Integer Linear Programming in Operations Research. Expanding my knowledge of the use and formulation of Integer Linear Programming, with an emphasis on scheduling problems. To the extent that I was able to create ILP to solve the TTP without examples of decision variables and constraints that could be found for other sport scheduling problems.

Over my time completing this project, I feel that I have improved several areas. My time management has been reasonable as I have been able to meet the initial aim of the project. There has been a significant increase in my ability to research into topic of which I initially only had a basic understanding and alongside an improvement in constructing and writing a report that accurately reflect the aims and results of a project.

Ultimately, I am pleased with how the project panned out and enjoyed researching this area of computer sciences.

Table of Abbreviations

Abbreviation	Definition
ILP	Integer Linear Programming
TTP	Travelling Tournament Problem
OR	Operations Research
LP	Linear Programming
TSP	Travelling Salesman Problem
TS	Tabu Search
WPF	Windows Presentation Foundation
UI	User Interface
EPL	English Premier League
DRR	Double Round Robin
RUI	Ribbon User Interface
MVVM	Model-View-ViewModel
FIFA	Fédération Internationale de Football Association
@	Away at
MUN	Manchester United
CHE	Chelsea
LIV	Liverpool
EVE	Everton
BUR	Burnley
WAT	Watford
STK	Stoke City
WBA	West Bromwich Albion
NEW	Newcastle United
HUD	Huddersfield Town

Table 15 - Table of Abbreviations

List of Figures

List of Tables

List of Equations

References

[1] 'Major League Baseball schedule - Wikipedia'. [Online]. Available: https://en.wikipedia.org/wiki/Major_League_Baseball_schedule. [Accessed: 14-May-2019].
[2] K. Easton, G. Nemhauser, and M. Trick, 'Solving the Traveling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach', p. 10.
[3] H. A. Taha, *Operations Research: An Introduction*, 8th ed. Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©2006, 2006.
[4] 'Knapsack problem', *Wikipedia*. 25-May-2019.
[5] 'Travelling salesman problem', *Simple English Wikipedia, the free encyclopedia*. 22-May-2019.
[6] R. Marti, M. Laguna, and F. Glover, 'Principles of Tabu Search', in *Handbook of Approximation Algorithms and Metaheuristics*, vol. 20073547, T. Gonzalez, Ed. Chapman and Hall/CRC, 2007, pp. 23-1-23–12.
[7] J.-P. Hamiez and J.-K. Hao, 'Solving the Sports League Scheduling Problem with Tabu Search', in *Local Search for Planning and Scheduling*, vol. 2148, A. Nareyek, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 24–36.
[8] B. Mayoh, E. Tyugu, and J. Penjam, *Constraint Programming*. Springer Science & Business Media, 2013.
[9] 'Mathematical Programming Solver | Gurobi'. [Online]. Available: https://www.gurobi.com/products/gurobi-optimizer. [Accessed: 14-May-2019].
[10] gewarren, 'What is WPF? - Visual Studio'. [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf. [Accessed: 14-May-2019].
[11] D. E. Inc, 'WPF MVVM Controls and Windows UI Components for Visual Studio and .NET Developers from DevExpress'. [Online]. Available: https://www.devexpress.com/products/net/controls/wpf/. [Accessed: 05-Jun-2019].
[12] 'How the Premier League fixture list is compiled'. [Online]. Available: http://www.premierleague.com/news/419044. [Accessed: 03-Jun-2019].
[13] J. Nielsen, '10 Heuristics for User Interface Design', *Nielsen Norman Group*, 1994. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed: 03-Jun-2019].
[14] M. Dostál, 'User Acceptance of the Microsoft Ribbon User Interface', *International Conference on Data Networks, Communications, Computers - Proceedings.*, pp. 143–149, 2010.
[15] 'Team Stats Database - FIFA 18 - FIFA Index'. [Online]. Available: https://www.fifaindex.com/teams/fifa18_278/?league=13&order=desc. [Accessed: 04-Jun-2019].
[16] P. V. Hentenryck and Y. Vergados, 'Minimizing Breaks in Sport Scheduling with Local Search', p. 8.
[17] G. Durán *et al.*, 'Scheduling the Chilean Soccer League by Integer Programming', *Interfaces*, vol. 37, no. 6, pp. 539–552, Dec. 2007.